Terence Tao: Hardest Problems in Mathematics, Physics & the Future of AI (Lex Fridman Podcast)

Lex Fridman : The following is a conversation with Terence Tao, widely considered to be one of the greatest mathematicians in history, often referred to as the Mozart of math. He won the Fields Medal and the Breakthrough Prize in mathematics, and has contributed groundbreaking work to a truly astonishing range of fields in mathematics and physics. This was a huge honor for me for many reasons, including the humility and kindness that Terry showed to me throughout all our interactions. It means the world. This is the Lex Fridman podcast. To support it, please check out our sponsors in the description or at lexfreedman.com/sponsors. And now, dear friends, here's Terence Tao. What was the first really difficult research level math problem that you encountered ? One that gives you pause, maybe ?

Terence Tao : Well, I mean, in your undergraduate education you learn about the really hard, impossible problems like the Riemann Hypothesis, the Twin-Primes Conjecture. You can make problems arbitrarily difficult. That's not really a problem. In fact, there's even problems that we know to be unsolvable. What's really interesting are the problems just on the boundary between what we can do relatively easily and what are hopeless. But what are problems where existing techniques can do like 90 % of the job, and then you just need that remaining 10 % ? I think as a PhD student, the Kakeya problem certainly caught my eye and it just got solved, actually. It's a problem I've worked on a lot in my early research. Historically, it came from a little puzzle by the Japanese mathematician Soichi Kakeya in 1918 or so. So the puzzle is that you have a needle on the plane, or think like driving on a road or something, and you wanted to execute a U-turn. You want to turn the needle around, but you want to do it in as little space as possible. So you want to use this little area in order to turn it around, but the needle is infinitely maneuverable. So you can imagine just spinning it around as a unit needle. You can spin it around its center and I think that gives you a disk of area, I think pi over four. Or you can do a three-point U-turn, which is what we teach people in their driving schools to do. And that actually takes area pi over eight. So it's a little bit more efficient than a rotation. And so for a while, people thought that was the most efficient way to turn things around. But Besicovitch showed that in fact you could actually turn the needle around using as little area as you wanted. So 0.001. There was some really fancy multi back and forth U-turn thing that you could do that you could turn a needle around. And in so doing it would pass through every intermediate direction.

Lex Fridman : Is this in the two-dimensional plane ?

Terence Tao : This is in the two-dimensional plane. And yeah, so we understand everything in two dimensions. So the next question is, what happens in three dimensions ? So suppose like the Hubble Space Telescope is a tube in space and you want to observe every single star in the universe, so you want to rotate the telescope to reach every single direction. And here's the unrealistic part. Suppose that space is at a premium, which it totally is not. You want to occupy as little volume as possible in order to rotate your needle around in order to see every single star in the sky. How small a volume do you need to do that ? And so you can modify Besicovitch's construction. And

Reference of the video : https://www.youtube-nocookie.com/embed/HUkBz-cdB-k.
Downsub subtitles Transcription : Denise Vella-Chemla, september 2025.

so if your telescope has zero thickness, then you can use as little volume as you need. That's a simple modification of the two dimensional construction. But the question is that if your telescope is not zero thickness, but just very, very thin, some thickness delta, what is the minimum volume needed to be able to see every single direction as a function of delta ? So as delta gets smaller, as your needle gets thinner, the volume should go down. But how fast does it go down ? And the conjecture was that it goes down very, very slowly, like logarithmically, roughly speaking. And that was proved after a lot of work. So this seems like a puzzle. Why is it interesting ? So it turns out to be surprisingly connected to a lot of problems in partial differential equations, in number theory, in geometry, combinatorics. For example, in wave propagation, you splash some water around, you create water waves and they travel in various directions. But waves exhibit both particle and wave-type behavior. So you can have what's called a wave packet, which is like a very localized wave that is localized in space and moving at a certain direction in time. And so if you plot it in both space and time, it occupies a region which looks like a tube. And so what can happen is that you can have a wave which initially is very dispersed, but it all focuses at a single point later in time. Like you can imagine dropping a pebble into a pond and ripples spread out. But then if you time reverse that scenario and the equations of wave motion are time reversible. You can imagine ripples that are converging to a single point, and then a big splash occurs, maybe even a singularity. And so it's possible to do that. And geometrically what's going on is that there's always sort of light rays. So like if this wave represents light, for example, you can imagine this wave as the superposition of photons all traveling at the speed of light. They all travel on these light rays and they're all focusing at this one point. So you can have a very dispersed wave focus into a very concentrated wave at one point in space and time, but then it defocuses again and it separates. But potentially, if the conjecture had a negative solution, so what that meant is that there's a very efficient way to pack tubes pointing in different directions into a very, very narrow region of very narrow volume, then you would also be able to create waves that start out, there'll be some arrangement of waves that start out very, very dispersed, but they would concentrate not just at a single point, but there'll be a large, there'll be a lot of concentrations in space and time. And you could create what's called a blowup, where these waves, their amplitude becomes so great that the laws of physics that they're governed by are no longer wave equations, but something more complicated and nonlinear. And so in mathematical physics, we care a lot about whether certain equations in wave equations are stable or not, whether they can create these singularities. There's a famous unsolved problem called the Navier-Stokes regularity problem. So the Navier-Stokes equations that govern a fluid flow or incompressible fluids like water. The question asks, if you start with a smooth velocity field of water, can it ever concentrate so much that the velocity becomes infinite at some point ? That's called a singularity. We don't see that in real life. If you splash around water in the bathtub, it won't explode on you, or have water leaving at the speed of light. But potentially it is possible. And in fact, in recent years, the consensus has drifted towards the belief that in fact, for certain very special initial configurations of, say, water, that singularities can form, but people have not yet been able to actually establish this. The Clay Foundation has these seven Millennium Prize Problems, has a million dollar prize for solving one of these problems. This is one of them. Of these seven, only one of them has been solved. The Poincare conjecture by Perelman. So the Kakeya conjecture is not directly, directly related to the Navier-Stokes problem, but understanding it would help us understand some aspects of things like wave concentration, which would indirectly probably help us understand the Navier-Stokes problem better.

Lex Fridman : Can you speak to the Navier-Stokes ? So the existence and smoothness, like you said, Millennium Prize problem. You've made a lot of progress on this one. In 2016, you published a paper, "Finite Time Blowup for an Averaged Three-Dimensional Navier-Stokes Equation."

Terence Tao : Right.

Lex Fridman : So we're trying to figure out if this thing usually doesn't blow up.

Terence Tao : Right.

Lex Fridman : But can we say for sure it never blows up ?

Terence Tao : Right, yeah. So, yeah, that is literally the million-dollar question. So this is what distinguishes mathematicians from pretty much everybody else. Like, if something holds 99.99 % of the time, that's good enough for most things. But mathematicians are one of the few people who really care about whether really 100 % of all situations are covered by. Yeah. So most of the time, water does not blow up. But could you design a very special initial state that does this ?

Lex Fridman : And maybe we should say that this is a set of equations that govern in the field of fluid dynamics. Trying to understand how fluid behaves and it actually turns out to be a really complicated fluid is extremely complicated thing to try to model.

Terence Tao : Yeah, so it has practical importance. So this Clay prize problem concerns what's called the Incompressible Navier-Stokes, which governs things like water. There's something called the Compressible Navier-Stokes, which governs things like air. And that's particularly important for weather prediction. Weather prediction does a lot of computational fluid dynamics. A lot of it is actually just trying to solve the Navier-Stokes equations as best they can. Also gathering a lot of data so that they can initialize the equation. There's a lot of moving parts. So it's a very important problem, practically.

Lex Fridman : Why is it difficult to prove general things about the set of equations like it not blowing up ?

Terence Tao : The short answer is Maxwell's demon. So Maxwell's demon is a concept in thermodynamics, like if you have a box of two gases, oxygen and nitrogen, and maybe you started with all the oxygen on one side and nitrogen on the other side, but there's no barrier between them, then they will mix, and they should stay mixed. There's no reason why they should unmix. But in principle, because of all the collisions between them, there could be some sort of weird conspiracy. Maybe there's a microscopic demon called Maxwell's demon that will, every time an oxygen and nitrogen atom collide, they will bounce off in such a way that the oxygen sort of drifts onto one side and then nitrogen goes to the other, and you could have an extremely improbable configuration emerge, which we never see. And statistically, it's extremely unlikely. But mathematically, it's possible that this can happen and we can't rule that out. And this is a situation that shows up a lot in mathematics. A basic example is the digits of pi. 3.14159 and so forth. The digits look like they have no pattern, and we believe they have no pattern. On the long term, you should see as many ones and twos and threes as fours and fives and sixes. There should be no preference in the digits of pi to favor let's say seven over eight. But maybe there's some demon in the digits of pi that every time you compute more and more digits, it's a biases one digit to another. And this is a conspiracy that should not happen. There's no reason it should happen, but there's no way to prove it with our current technology. Okay, so getting back to Navier-Stokes, a fluid has a certain amount of energy, and because the fluid is in motion, the energy gets transported around. And water is also viscous. So if the energy is spread out over many different locations, the natural viscosity of the fluid will just damp out the energy and it will go to zero. And this is what happens when we actually experiment with water. You splash around, there's some turbulence and waves and so forth, but eventually it settles down. And the lower the amplitude, the smaller the velocity, the more calm it gets. But potentially there is some sort of demon that keeps pushing the energy of the fluid into a smaller and smaller scale. And it will move faster and faster, and at faster speeds, the effective viscosity is relatively less. And so it could happen that it creates some sort of what's called a self-similar blob scenario where the energy of the fluid starts off at some large scale and then it all sort of transfers its energy into a smaller region of the fluid, which then at a much faster rate moves into an even smaller region and so forth. And each time it does this, it takes maybe half as long as the previous one. And then you could actually converge to all the energy concentrating in one point in a finite amount of time. And that scenario is called finite time blow up. So in practice, this doesn't happen. So water is what's called turbulent. So it is true that if you have a big eddy of water, it will tend to break up into smaller eddies, but it won't transfer all the energy from one big eddy into one smaller eddy. It will transfer into maybe three or four, and then those must split up into maybe three or four small eddies of their own. And so the energy gets dispersed to the point where the viscosity can then keep everything under control. But if it can somehow concentrate all the energy, keep it all together, and do it fast enough that the viscous effects don't have enough time to calm everything down, then this blob can occur. So there are papers who had claimed that, oh, you just need to take into account conservation of energy and just carefully use the viscosity and you can keep everything under control for not just the Navier-Stokes, but for many, many types of equations like this. And so in the past, there have been many attempts to try to obtain what's called global regularity for Navier-Stokes, which is opposite of finite-time blowup, that velocity stays smooth. And it all failed. There was always some sign error or some subtle mistake and it couldn't be salvaged. So what I was interested in doing was trying to explain why we were not able to disprove finite-time blowup. I couldn't do it for the actual equations of fluids, which were too complicated. But if I could average the equations of motion of Navier-Stokes, basically, if I could turn off certain types of ways in which water interacts and only keep the ones that I want. So in particular, if there's a fluid and it could transfer its energy from a large eddy into this small eddy or this other small eddy, I would turn off the energy channel that would transfer energy to this one and direct it only into this smaller eddy while still preserving the law of concentration of energy.

Lex Fridman : So you're trying to make it blow up.

Terence Tao : Yeah. So I basically engineer a blowup by changing laws of physics, which is one thing that mathematicians are allowed to do, we can change the equation.

Lex Fridman : How does that help you get closer to the proof of something ?

TERENCE TAO : So it provides what's called an obstruction in mathematics. So what I did was that basically if I turned off certain parts of the equation, which usually when you turn off certain interactions, make it less nonlinear, it makes it more regular and less likely to blow up. But I found that by turning off a very well designed set of interactions, I could force all the energy to blow in finite time. So what that means is that if you wanted to prove global regularity for Navier-Stokes, for the actual equation, you must use some feature of the true equation, which my artificial equation does not satisfy. So it rules out certain approaches. So the thing about math is it's not just about finding, taking a technique that is going to work and applying it, but you need to not take the techniques that don't work. And for the problems that are really hard, often there are dozens of ways that you might think might apply to solve the problem. But it's only after a lot of experience that you realize there's no way that these methods are going to work. So having these counter-examples for nearby problems kind of rules out. It saves you a lot of time, because you're not wasting energy on things that you now know cannot possibly ever work.

LEX FRIDMAN : How deeply connected is it to that specific problem of fluid dynamics ? Or just some more general intuition you build up about mathematics ?

TERENCE TAO : Right, yeah. So the key phenomenon that my technique exploits is what's called supercriticality. So in partial differential equations, often these equations are like a tug of war between different forces. So in Navier-Stokes, there's the dissipation force coming from viscosity, and it's very well understood, it's linear, it calms things down. If viscosity was all there was, then nothing bad would ever happen. But there's also transport that energy from in one location of space can get transported because the fluid is in motion to other locations. And that's a nonlinear effect, and that causes all the problems. So there are these two competing terms in the Navier-Stokes equation, the dissipation term and the transport term. If the dissipation term dominates, if it's large, then basically you get regularity. And if the transport term dominates, then we don't know what's going on. It's a very nonlinear situation. It's unpredictable, it's turbulent. So sometimes these forces are unbalanced at small scales, but not in balance at large scales, or vice versa. So Navier-Stokes is what's called supercritical. So at smaller and smaller scales, the transport terms are much stronger than the viscosity terms. So the viscosity of the things that calm things down. And so this is why the problem is hard. In two dimensions, so the Soviet mathematician Ladyzhenskaya, she in the 60s showed in two dimensions there was no blowup. And in two dimensions, the Navier-Stokes equations is what's called critical. The effect of transport and the effect of viscosity are about the same strength even at very, very small scales. And we have a lot of technology to handle critical and also subcritical equations and prove regularity. But for supercritical equations, it was not clear what was going on. And I did a lot of work. And then there's been a lot of followup showing that for many other types of supercritical equations, you can create all kinds of blow-up examples. Once the nonlinear effects dominate the linear effects at small scales, you can have all kinds of bad things happen. So this is sort of one of the main insights of this line of work, is that supercriticality versus criticality and subcriticality, this makes a big difference. I mean, that's a key qualitative feature that distinguishes some equations for being sort of nice and predictable. And like planetary motion. I mean, there are certain equations that you can predict for millions of years or thousands at least. Again, it's not really a problem. But there's a reason why we can't predict the weather past two weeks into the future. Because it's a super critical equation. Lots of really strange things are going on at very fine scales.

Lex Fridman : So whenever there is some huge source of non-linearity that can create a huge problem for predicting what's going to happen.

Terence Tao : Yeah. And if the nonlinearity is somehow more and more featured and interesting at small scales. I mean, there's many equations that are nonlinear, but in many equations you can approximate things by bulk. So for example, planetary motion. If you want to understand the orbit of the moon or Mars or something, you don't really need the microstructure of the seismology of the moon or exactly how the mass is distributed. You just basically you can almost approximate these planets by point masses. And just the aggregate behavior is important. But if you want to model a fluid like the weather, you can't just say in Los Angeles, the temperature is this, the wind speed is this. For supercritical equations, the fine-scale information is really important.

Lex Fridman : If we can just linger on the Navier-Stokes equations a little bit. So you've suggested maybe you can describe it, that one of the ways to solve it or to negatively resolve it would be to sort of to construct a liquid, a kind of liquid computer.

Terence Tao : Right.

Lex Fridman : And then show that the halting problem from computation theory has consequences for fluid dynamics. So show it in that way. Can you describe this idea ?

Terence Tao : Right, yeah. So this came out of this work of constructing this average equation that blew up. So as part of how I had to do this, so there's sort of this naive way to do it. You just keep pushing. Every time you get energy at one scale, you push it immediately to the next scale as fast as possible. This is sort of the naive way to force blowup. In terms of, in five and higher dimensions this works, but in three dimensions there was this funny phenomenon that I discovered that if you change laws of physics, you just always keep trying to push the energy into smaller scales, what happens is that the energy starts getting spread out into many scales at once. So you have energy at one scale, you're pushing it into the next scale, and then as soon as it enters that scale, you also push it to the next scale. But there's still some energy left over from the previous scale. You're trying to do everything at once, and this spreads out the energy too much. And then it turns out that it makes it vulnerable for viscosity to come in and actually just damp out everything. So it turns out this directive portion doesn't actually work. There was a separate paper by some other authors that actually showed this in three dimensions. So what I needed was to program a delay, so kind of like airlocks. So I needed an equation which would start with a fluid doing something at one scale. It would push its energy into the next scale, but it would stay there until all the energy from the larger scale got transferred. And only after you pushed all the energy in, then you sort of open the next gate, and then you push that in as well. So by doing that, the energy inches forward, scale by scale, in such a way that it's always localized at one scale at a time. And then it can resist the effects of viscosity because it's not dispersed. So in order to make that happen, yeah, I had to construct a rather complicated nonlinearity. And it was basically like it was constructing, like an electronic circuit. So I actually thanked my wife for this because she was trained as an electrical engineer, and she talked about, she had to design circuits and so forth. And if you want a circuit that does a certain thing, like maybe have a light that flashes on and then turns off, and then on and then off, you can build it from more primitive components, capacitors and resistors and so forth. And you have to build a diagram, and these diagrams you can sort of follow with your eyeballs and say, "Oh, yeah, the current will build up here, and then it will stop, and then it will do that." So I knew how to build the analog of basic electronic components like resistors and capacitors and so forth. And I would stack them together in such a way that I would create something that would open one gate, and then there'll be a clock. And then once the clock hits the certain threshold, it would close it. It was kind of a Rube Goldberg type machine, but described mathematically, and this ended up working. So what I realized is that if you could pull the same thing off for the actual equations. So if the equations of water supported computation. So you can imagine kind of a steampunk, but it's really water punk type of thing where so modern computers are electronic. They're powered by electrons passing through very tiny wires and interacting with other electrons and so forth. But instead of electrons, you can imagine these pulses of water moving a certain velocity and maybe there are two different configurations corresponding to a bit being up or down, probably that if you had two of these moving bodies of water collide, it would come out with some new configuration which would be something like an AND gate or OR gate. The output would depend in a very predictable way on the inputs. And like you could chain these together and maybe create a Turing machine. And then you have computers which are made completely out of water. And if you have computers, then maybe you can do robotics, hydraulics and so forth. And so you could create some machine which is basically a fluid analog, what's called a von Neumann machine. So von Neumann proposed if you want to colonize Mars, the sheer cost of transporting people on machines to Mars is just ridiculous. But if you could transport one machine to Mars and this machine had the ability to mine the planet, create some raw materials, smelt them, and build more copies of the same machine, then you could colonize the whole planet over time. So if you could build a fluid machine, which, yeah, so it's a fluid robot. And what it would do, its purpose in life, it's programmed so that it would create a smaller version of itself in some sort of cold state. It wouldn't start just yet. Once it's ready, the big robot configured of water would transfer all its energy into the smaller configuration and then power down and then clean itself up. And then what's left is this newer state which would then turn on and do the same thing, but smaller and faster. And then the equation has a certain scaling symmetry. Once you do that, it can just keep iterating. So this in principle would create a blowup for the actual Navier-Stokes. And this is what I managed to accomplish for this average Navier-Stokes. So it provided this sort of roadmap to solve the problem. Now this is a pipe dream, because there are so many things that are missing for this to actually be a reality. So I can't create these basic logic gates. I don't have these special configurations of water. I mean, there's candidates, things like vortex rings that might possibly work. But also analog computing is really nasty compared to digital computing because there's always errors. You have to do a lot of error correction along the way. I don't know how to completely power down the big machine so that it doesn't interfere with the running of the smaller machine. But everything in principle can happen. It doesn't contradict any of the laws of physics. So it's sort of evidence that this thing is possible. There are other groups who are now pursuing ways to make Navier-Stokes blowup, which are nowhere near as ridiculously complicated as this. They actually are pursuing much closer to the direct self similar model which can, it doesn't quite work as is, but there could be some simpler scheme than what I just described to make this work.

Lex Fridman : There is a real leap of genius here to go from Navier-Stokes to this Turing machine. So it goes from what the self-similar blob scenario that you're trying to get the smaller and smaller blob to now having a liquid Turing machine gets smaller and smaller and smaller. And somehow seeing how that could be used to say something about a blowup. I mean, that's a big leap.

Terence Tao : So there's precedent. I mean, so the thing about mathematics is that it's really good at spotting connections between what you think of, what you might think of as completely different problems. But if the mathematical form is the same, you can draw a connection. So there's a lot of work previously on what are called cellular automata, the most famous of which is Conway's Game of Life. There's this infinite discrete grid, and at any given time the grid is either occupied by a cell or it's empty. And there's a very simple rule that tells you how these cells evolve. So sometimes cells live and sometimes they die. And this is when I was a student, it was a very popular screensaver to actually just have these animations going, and they look very chaotic. In fact they look a little bit like turbulent flow sometimes. But at some point people discovered more and more interesting structures within this Game of Life. So for example, they discovered this thing called a glider. So a glider is a very tiny configuration of like four or five cells which evolves and it just moves at a certain direction and that's like these vortex rings. So this is an analogy. The Game of Life is kind of like a discrete equation. And the fluid Navier-Stokes is a continuous equation. But mathematically they have some similar features. And so over time, people discovered more and more interesting things that you could build within the Game of Life. The Game of Life is a very simple system. It only has like three or four rules to do it, but you can design all kinds of interesting configurations inside it. There's something called a glider gun that does nothing of spit out gliders one and one at a time. And then after a lot of effort, people managed to create AND gates and OR gates for gliders. Like there's this massive ridiculous structure which if you have a stream of gliders coming in here and a stream of gliders coming in here, then you may produce a stream of gliders coming out. Maybe if both of the streams have gliders, then there'll be an output stream. But if only one of them does, then nothing comes out. So they could build something like that. And once you could build these basic gates, then just from software engineering, you can build almost anything. You can build a Turing machine. I mean, it's again, enormous steampunk type things, they look ridiculous. But then people also generated self-replicating objects in the Game of Life. A massive machine, a polynomial machine, which over a huge period of time and always look like glider guns inside doing these very steampunk calculations. It would create another version of itself which could replicate.

Lex Fridman : That's so incredible.

Terence Tao : A lot of this was community crowdsourced by amateur mathematicians, actually. So I knew about that work. And so that is part of what inspired me to propose the same thing with Navier-Stokes, which is a much, as I said, analog is much worse than digital. You can't just directly take the constructions in the Game of Life and plunk them in. But again, it just, it shows it's possible.

Lex Fridman : You know, there's a kind of emergence that happens with these cellular automata, local rules. Maybe it's similar to fluids, I don't know. But local rules operating at scale can create these incredibly complex dynamic structures. Do you think any of that is amenable to mathematical analysis ? Do we have the tools to say something profound about that ?

Terence Tao : The thing is, you can get this emergent very complicated structures, but only with very carefully prepared initial conditions. So these glider guns and gates and software machines, if you just plunk on randomly some cells, and you will not see any of these. And that's the analogous situation with Navier-Stokes. Again, with typical initial conditions, you will not have any of this weird computation going on. But basically through engineering, by specially designing things in a very special way, you can pick clever constructions.

Lex Fridman : I wonder if it's possible to prove the sort of the negative of, basically prove that only through engineering can you ever create something interesting.

Terence Tao : This is a recurring challenge in mathematics that I call it the dichotomy between structure and randomness. That most objects that you can generate in mathematics are random. They look like random. The digits of pi, well, we believe is a good example, but there's a very small number of things that have patterns. But now you can prove something as a pattern by just constructing. If something has a simple pattern and you have a proof that it does something like repeat itself every so often, you can do that, and you can prove that, for example, you can prove that most sequences of digits have no pattern. So if you just pick digits randomly, there's something called low large numbers that tells you you're going to get as many ones as twos in the long run. But we have a lot fewer tools, if I give you a specific pattern like the digits of pi, how can I show that this doesn't have some weird pattern to it ? Some other work that I spend a lot of time on is to prove what are called structure theorems or inverse theorems that give tests for when something is very structured. So some functions are what's called additive, like if you have a function that must say natural numbers, the natural numbers. So maybe two maps to four, three maps to six, and so forth. Some functions also additive, which means that if you add two inputs together, the output gets added as well. For example, multiplying by a constant, if you multiply a number by 10, if you multiply A plus B by 10, that's the same as multiplying A by 10 and B by 10 and then adding them together. So some functions are additive. Some functions are kind of additive, but not completely additive. So for example, if I take a number and I multiply by the square root of 2 and I take the integer part of that. So 10 by square root of 2 is like 14 point something. So 10 up to 14. 21 up to 28. So in that case additively is true then. So 10 plus 10 is 20, and 14 plus 14 is 28. But because of this rounding, sometimes there's round off errors, and sometimes when you add A plus B, this function doesn't quite give you the sum of the two individual outputs, but the sum plus minus one. So it's almost additive, but not quite additive. So there's a lot of useful results in mathematics, and I've worked a lot on developing things like this, to the effect that if a function exhibits some structure like this, then it's basically there's a reason for why it's true. And the reason is because there's some other nearby function which is actually completely structured, which is explaining this sort of partial pattern that you have. And so if you have these inverse theorems, it creates this sort of dichotomy that either the objects that you study either have no structure at all, or they are somehow related to something that is structured. And in either case you can make progress. A good example of this is that there's this old theorem in mathematics called Szemeredi's Theorem proven in the 1970s. It concerns trying to find a certain type of pattern in a set of numbers. The pattern is arithmetic progression, things like 3, 5, and 7, or 10, 15, and 20. And Szemeredi, Endre Szemeredi, proved that any set of numbers that is sufficiently big, what's called positive density, has arithmetic progressions in it of any length you wish. So, for example, the odd numbers have a set of density one half, and they contain arithmetic progressions of any length. So in that case, it's obvious because the odd numbers are really, really structured. I can just take 11, 13, 15, 17. I can easily find arithmetic expressions in that set. But Szemeredi also applies to random sets. If I take the set of odd numbers and I flip a coin for each number, and I only keep the numbers for which I got a heads, so I just flip coins. I just randomly take out half the numbers, I keep one half. So that's a set that has no patterns at all, but just from random fluctuations, you will still get a lot of arithmetic progressions in that set.

Lex Fridman : Can you prove that there's arithmetic progressions of arbitrary length within a random ?

Lex Fridman : Yes. Have you heard of the infinite monkey theorem ?

Terence Tao : Usually mathematicians give boring names to theories, but occasionally they give colorful names. The popular version of the infinite monkey theorem is that if you have an infinite number of monkeys in a room with each with a typewriter, they type out text randomly, almost surely one of them is going to generate the entire script of Hamlet or any other finite string of text. It will just take some time, quite a lot of time, actually. But if you have an infinite number, then it happens. So basically, the theorem says that if you take an infinite string of digits or whatever, eventually any finite pattern you wish will emerge. It may take a long time, but it will eventually happen. In particular, arithmetic progressions of any length will eventually happen. But you need an extremely long random sequence for this to happen.

Lex Fridman : I suppose that's intuitive. It's just infinity.

Terence Tao : Yeah, infinity absorbs a lot of sins.

Lex Fridman : Yeah. How are we humans supposed to deal with infinity ?

Terence Tao : Well, you can think of infinity as just an abstraction of a finite number for which you do not have a bound for. I mean, so nothing in real life is truly infinite. But you can ask yourself questions like, what if I had as much money as I wanted ? Or what if I could go as fast as I wanted ? And a way in which mathematicians formalize that is mathematics has found a formalism to idealize. Instead of something being extremely large or extremely small, to actually be exactly infinite or zero. And often the mathematics becomes a lot cleaner when you do that. I mean, in physics we joke about assuming spherical cows. Real world problems have got all kinds of real world effects, but you can idealize, send things to infinity, send something to zero, and the mathematics becomes a lot simpler to work with there.

Lex Fridman : I wonder how often using infinity forces us to deviate from the physics of reality.

Terence Tao : Yeah, so there's a lot of pitfalls. So we spend a lot of time in undergraduate math classes teaching analysis. And analysis is often about how to take limits and whether you, so for example, A plus B is always B plus A. So when you have a finite number of terms and you add them, you can swap them and there's no problem. But when you have an infinite number of terms, there are these sort of shell games you can play where you can have a series which converges to one value, but you rearrange it and it suddenly converges to another value. And so you can make mistakes. You have to know what you're doing when you allow infinity. You have to introduce these epsilons and deltas. And there's a certain type of way of reasoning that helps you avoid mistakes. In more recent years, people have started taking results that are true in infinite limits and what's called finitizing them. So you know that something's true eventually, but you don't know when. Now give me a rate. Okay, so such that if I don't have an infinite number of monkeys, but a large finite number of monkeys, how long do I have to wait for Hamlet to come out ? And that's a more quantitative question. And this is something that you can attack by purely finite methods. And you can use your finite intuition, and in this case it turns out to be exponential in the length of the text that you're trying to generate. And so this is why you never see the monkeys create Hamlet. You can maybe see them create a four-letter word, but nothing that big. And so I personally find once you finitize an infinite statement, it does become much more intuitive and it's no longer so weird.

Lex Fridman : So even if you're working with infinity, it's good to finitize so that you can have some intuition.

Terence Tao : Yeah. The downside is that the finitized proofs are just much, much messier. So the infinite ones are found first, usually, like decades earlier, and then later on people finalize them.

Lex Fridman : So since we mentioned a lot of math and a lot of physics, what to use the difference between mathematics and physics as disciplines, as ways of understanding of seeing the world ? Maybe we can throw in engineering in there. You mentioned your wife is an engineer. Give it new perspective on circuits. So this different way of looking at the world, given that you've done mathematical physics, you've worn all the hats.

Terence Tao : Right, so I think science in general is interaction between three things. There's the real world, there's what we observe of the real world, our observations, and then our mental models as to how we think the world works. So we can't directly access reality. Okay. All we have are the observations, which are incomplete and they have errors. And there are many cases where we want to know, for example, what is the weather like tomorrow ? And we don't yet have the observation and we'd like to predict. And then we have these simplified models, sometimes making unrealistic assumptions, spherical cow type things. Those are the mathematical models. Mathematics is concerned with the models. Science collects the observations and it proposes the models that might explain these observations. What mathematics does, we stay within the model, and we ask, what are the consequences of that model ? What predictions would the model make of future observations or past observations ? Does it fit observed data ? So there's definitely a symbiosis. I guess mathematics is unusual among other disciplines, is that we start from hypotheses, like the axioms of a model, and ask what conclusions come up from that model. In almost any other discipline, you start with the conclusions. I want to do this. I want to build a bridge, I want to make money, I want to do this. And then you find the paths to get there. There's a lot less sort of speculation about, suppose I did this, what would happen ? Planning and modeling. Speculative fiction maybe is one other place, but that's about it, actually. Most of the things we do in life is conclusions driven, including physics and science. I mean, they want to know, where is this asteroid going to go ? What is the weather going to be tomorrow ? But mathematics also has this other direction of going from the axioms.

Lex Fridman : What do you think ? There is this tension in physics between theory and experiment. What do you think is a more powerful way of discovering truly novel ideas about reality ?

Terence Tao : Well, you need both top-down and bottom-up. Yeah, it's a really interaction between all these things. So over time, the observations and the theory and the modeling should both get closer to reality. But initially, this is always the case, they're always far apart to begin with. But you need one to figure out where to push the other. So if your model is predicting anomalies that are not picked up by experiment, that tells experimenters where to look to find more data, to refine the models. So it goes back and forth. Within mathematics itself, there's also a theory and experimental component. It's just that until very recently, theory has dominated almost completely. Like 99 % of mathematics is theoretical mathematics. And there's a very tiny amount of experimental mathematics. I mean, people do do it if they want to study prime numbers or whatever, they can just generate large data sets. So once we had a computers, we began to do it a little bit. Although even before, well, like Gauss, for example, he discovered, he conjectured the most basic theorem in number theory to call the prime number theorem, which predicts how many primes that up to a million, up to a trillion. It's not an obvious question. And basically what he did was that he computed, I mean, mostly by himself, but also hired human computers, people whose professional job it was to do arithmetic, to compute the first hundred thousand primes or something and made tables and made a prediction. That was an early example of experimental mathematics, but until very recently it was not, I mean, theoretical mathematics was just much more successful. I mean, of course, doing complicated mathematical computations was just not feasible until very recently. And even nowadays, even though we have powerful computers, only some mathematical things can be explored numerically. There's something called the combinatorial explosion. If you want to study, for example, Zsigmondy's theorem, you want to study all possible subsets of the numbers 1 to 1000. There's only 1000 numbers. How bad could it be ? It turns out the number of different subsets of 1 to 1000 is 2 to the power 1000, which is way bigger than any computer can currently, in fact, any computer ever will ever enumerate. So there are certain math problems that very quickly become just intractable to attack by direct brute force computation. Chess is another famous example. The number of chess positions, we can't get a computer to fully explore. But now we have AI, we have tools to explore this space, not with 100 % guarantees of success, but with experiment. So we can empirically solve chess now, for example. We have very, very good AIs that can, they don't explore every single position in the game tree, but they have found some very good approximation. And people are using actually these chess engines to do experimental chess that they're revisiting old chess theories about, oh, this type of opening, this is a good type of move, this is not. And they can use these chess engines to actually refine, in some cases overturn, conventional wisdom about chess. And I do hope that mathematics will have a larger experimental component in the future, perhaps powered by AI.

Terence Tao : We'll of course talk about that. But in the case of chess, and there's a similar thing in mathematics, I don't believe it's providing a kind of formal explanation of the different positions. It's just saying which position is better or not. And you can intuit as a human being, and then from that we humans can construct a theory of the matter. You've mentioned the Plato's cave allegory. So in case people don't know, it's where people are observing shadows of reality, not reality itself. And they believe what they're observing to be reality. Is that in some sense what mathematicians and maybe all humans are doing is looking at shadows of reality ? Is it possible for us to truly access reality ?

LEX FRIDMAN : Well, there are these three ontological things. There's actual reality, there's our observations, and our models. And technically they are distinct, and I think they will always be distinct. But they can get closer over time. And the process of getting closer often means that you have to discard your initial intuitions. So astronomy provides great examples. An initial model of the world is flat because it looks flat, and it's big. And the rest of the universe, the sky is not, like the sun, for example, looks really tiny. And so you start off with a model which is actually really far from reality, but it fits kind of the observations that you have. So things look good. But over time, as you make more and more observations bring it closer to reality, the model gets dragged along with it. And so over time, we had to realize that the Earth was round, that it spins, it goes around the solar system, solar system goes around the galaxy, and so on and so forth. And the galaxy universe is expanding. The expansion is itself expanding, accelerating. And in fact, very recently this year or so, even the acceleration of the universe itself, there's evidence that it's non-constant.

LEX FRIDMAN : And the explanation behind why that is.

TERENCE TAO : It's catching up.

LEX FRIDMAN : It's catching up. I mean, it's still the dark matter, dark energy, this kind of thing.

TERENCE TAO : We have a model that sort of explains, that fits the data really well. It just has a few parameters that you have to specify. So people say, oh, that's fudge factors. With enough fudge factors, you can explain anything. But the mathematical point of the model is that you want to have fewer parameters in your model than data points in your observational set. So if you have a model with 10 parameters that explains 10 observations, that is a completely useless model. It's what's called overfitted. But if you have a model with two parameters and it explains a trillion observations, which is basically so, yeah, the dark matter model, I think has like 14 parameters and it explains petabytes of data that the astronomers have. You can think of a theory. Like one way to think about physical mathematical theory is it's a compression of the universe and a data compression. So you have these petabytes of observations. You like to compress it to a model which you can describe in five pages and specify a certain number of parameters. And if it can fit to reasonable accuracy, almost all of your observations. I mean, the more compression that you make, the better your theory.

LEX FRIDMAN : In fact, one of the great surprises of our universe and of everything in it is that it's compressible at all. It's the unreasonable effectiveness of mathematics.

TERENCE TAO : Yeah, Einstein had a quote like that. The most incomprehensible thing about the universe is that it is comprehensible.

LEX FRIDMAN : And not just comprehensible. You can do an equation like E equals MC squared.

Terence Tao : There is actually some mathematical possible explanation for that. So there's this phenomenon in mathematics called universality. So many complex systems at the macro scale are coming out of lots of tiny interactions at the micro scale. And normally, because of the common form of explosion, you would think that the macroscale equations must be infinitely exponentially more complicated than the microscale ones. And they are if you want to solve them completely exactly. Like if you want to model all the atoms in a box of air. That's like Avogadro's number is humongous. There's a huge number of particles. If you actually have to track each one, it'll be ridiculous. But certain laws emerge at the microscopic scale that almost don't depend on what's going on at the macroscale, or only depend on a very small number of parameters. So if you want to model a gas of quintillion particles in a box, you just need to know its temperature and pressure and volume and a few parameters like five or six. And it models almost everything you need to know about these 10 to the 23 or whatever particles. So we don't understand universality anywhere near as we would like mathematically. But there are much simpler toy models where we do have a good understanding of why universality occurs. Most basic one is the central limit theorem that explains why the bell curve shows up everywhere in nature, that so many things are distributed by what's called a Gaussian distribution. Famous bell curve. There's now even a meme but wit this curve.

Lex Fridman : And even the meme applies broadly, there's universality to the meme.

Terence Tao : Yes, you can call it meta if you like, but there are many, many processes. For example, you can take lots and lots of independent random variables and average them together in various ways, you can take a simple average or more complicated average. And we can prove in various cases that these bell curves, these Gaussians emerge. And it is a satisfying explanation. Sometimes they don't. So if you have many different inputs and they're all correlated in some systemic way, then you can get something very far from a bell curve show up. And this is also important to know when a system fails. So universality is not a 100reliable thing to rely on. The global financial crisis was a famous example of this. People thought that mortgage defaults had this sort of Gaussian type behavior that if you ask if you have a population of 100,000 Americans with mortgages, they ask what proportion of them will default on their mortgages. If everything was decorrelated, there would be a nice bell curve, and you can manage risk with options and derivatives and so forth. And it is a very beautiful theory. But if there are systemic shocks in the economy that can push everybody to default at the same time, that's very non-Gaussian behavior. And this wasn't fully accounted for in 2008. Now I think there's some more awareness that this is a systemic risk is actually a much bigger issue. And just because the model is pretty and nice, it may not match reality. The mathematics of working out what models do is really important. But also the science of validating when the models fit reality and when they don't, you need both. But mathematics can help because, for example, these central limit theorems, it tells you that if you have certain axioms like type non-correlation, that if all the inputs were not correlated to each other, then you have these Gaussian behaviors that things are fine. It tells you where to look for weaknesses in the model. So if you have a mathematical understanding of the central limit theorem and someone proposes to use these Gaussian copulas or whatever to model default risk, if you're mathematically trained, you would say, "Okay, but what are the systemic correlation between all your inputs ?" And so then you can ask the economists, "How much of a risk is that ?" And then you can go look for that. So there's always this synergy between science and mathematics.

Lex Fridman : A little bit on the topic of universality, you're known and celebrated for working across an incredible breadth of mathematics. Reminiscent of Hilbert a century ago. In fact, the great Fields Medal winning mathematician Tim Gowers has said that you are the closest thing we get to Hilbert.

Terence Tao : Ha ! (*Lex chuckles*)

Lex Fridman : He's a colleague of yours ?

Terence Tao : Oh yeah, good friend.

Lex Fridman : But anyway, so you are known for this ability to go both deep and broad in mathematics. So you're the perfect person to ask. Do you think there are threads that connect all the disparate areas of mathematics ? Is there a kind of deep underlying structure to all of mathematics ?

Terence Tao : There's certainly a lot of connecting threads, and a lot of the progress of mathematics can be represented by taking by stories of two fields of mathematics that were previously not connected and finding connections. An ancient example is geometry and number theory. So in the times of the ancient Greeks, these were considered different subjects. I mean mathematicians worked on both. Euclid worked both on geometry most famously, but also on numbers. But they were not really considered related. I mean a little bit like you could say that this length was five times this length because you could take five copies of this length and so forth. But it wasn't until Descartes who really realized that develop analytic geometry, you can parameterize the plane, a geometric object, by two real numbers. Every point can be. And so geometric problems can be turned into problems about numbers. And today this feels almost trivial. There's no content to list. Of course a plane is X and Y, because that's what we teach and it's internalized. But it was an important development that these two fields were unified, and this process has just gone on throughout mathematics over and over again. Algebra and geometry were separated and now we have a split algebraic geometry that connects them and over and over again. And that's certainly the type of mathematics that I enjoy the most. So I think there's sort of different styles to being a mathematician. I think hedgehogs and fox. A fox knows many things a little bit, but a hedgehog knows one thing very, very well. And in mathematics there's definitely both hedgehogs and foxes, and then there's people who are kind of, who can play both roles. And I think ideal collaboration between mathematicians involves, you need some diversity, like a fox working with many hedgehogs or vice versa. But I identify mostly as a fox. Certainly I like arbitrage, somehow learning how one field works, learning the tricks of that field and then going to another field which people don't think is related, but I can adapt the tricks.

Lex Fridman : So see the connections between the fields.

Terence Tao : Yeah. So there are other mathematicians who are far deeper than I am, they're really hedgehogs. They know everything about one field, and they're much faster and more effective in that field. But I can give them these extra tools.

Lex Fridman : I mean, you've said that you can be both the hedgehog and the fox, depending on the context, depending on the collaboration. So what can you, if it's at all possible, speak to the difference between those two ways of thinking about a problem ? Say you're encountering a new problem, searching for the connections versus like very singular focus.

Terence Tao : I'm much more comfortable with the fox paradigm. I like looking for analogies, narratives. I spend a lot of time, if there's a result, I see it in one field, and I like the result, it's a cool result. But I don't like the proof. It uses types of mathematics that I'm not super familiar with. I often try to reprove it myself using the tools that I favor. Often my proof is worse, but by the exercise of doing so I can say, "Oh, now I can see what the other proof was trying to do." And from that I can get some understanding of the tools that are used in that field. So it's very exploratory, doing crazy things and crazy builds and reinventing the wheel a lot. Whereas the hedgehog style is, I think, much more scholarly. You're very knowledge-based. You stay up to speed on all the developments in this field. You know all the history. You have a very good understanding of exactly the strengths and weaknesses of each particular technique. Yeah, I think you'd rely a lot more on sort of calculation than sort of trying to find narratives. So, yeah, I mean, I could do that too, but there are other people who are extremely good at that.

Lex Fridman : Let's step back and maybe look at a bit of a romanticized version of mathematics. So I think you've said that early on in your life, math was more like a puzzle solving activity when you were young. When did you first encounter a problem or proof where you realized math can have a kind of elegance and beauty to it ?

Terence Tao : That's a good question. When I came to graduate school in Princeton. So John Conway was there at the time. He passed away a few years ago. But I remember one of the very first research talks I went to was a talk by Conway on what he called extreme proof. So Conway just had this amazing way of thinking about all kinds of things in a way that you would normally think of. So he thought of proofs themselves as occupying some sort of space. So if you want put to prove something, let's say that there's infinitely many primes, okay, you have all different proofs, but you could rank them in different axes. Like some proofs are elegant, some proofs are long, some proofs are elementary and so forth. And so this is cloud. So the space of all proofs itself has some sort of shape. And so he was interested in extreme points of this shape. Like out of all these proofs, what is one that is the shortest at the expense of everything else or the most elementary or whatever. And so he gave some examples of well-known theorems and then he would give what he thought was the extreme proof in these different aspects. I just found that really eye-opening that it's not just getting a proof for a result was interesting, but once you have that proof trying to optimize it in various ways that, but proofing itself had some craftsmanship to it. It certainly informed my writing style that when you do your math assignments and undergraduate your homework and so forth, you're sort of encouraged to just write down any proof that works and hand it in. As long as it gets a tick mark, you move on. But if you want your results to actually be influential and be read by people, it can't just be correct. It should also be a pleasure to read, motivated, be adaptable, to generalize to other things. It's the same in many other disciplines like coding. There's a lot of analogies between math and coding. I like analogies if you haven't noticed. (Lex laughs) But you can code something spaghetti code that works for a certain task, and it's quick and dirty, and it works. But there's lots of good principles for writing code well so that other people can use it, build upon it, and so then has fewer bugs and whatever. And there's similar things with mathematics.

Lex Fridman : Yeah, first of all there's so many beautiful things there. And Conway is one of the great minds in mathematics ever and computer science. Just even considering the space of proofs, and saying, "Okay, what does this space look like and what are the extremes ?" Like you mentioned, coding is an analogy is interesting because there's also this activity called Code Golf.

Terence Tao : Oh, yeah, yeah, yeah.

Lex Fridman : Which I also find beautiful and fun where people use different programming languages to try to write the shortest possible program that accomplishes a particular task.

Terence Tao : Yeah.

Lex Fridman : Then I believe there's even competitions on this. And it's also a nice way to stress test not just the sort of the programs or in this case the proofs, but also the different languages. Maybe that's a different notation or whatever to use to accomplish a different task.

Terence Tao : Yeah, you learn a lot. I mean it may seem like a frivolous exercise, but it can generate all these insights which if you didn't have this artificial objective to pursue, you might not see.

Lex Fridman : What to use the most beautiful or elegant equation in mathematics ? I mean, one of the things that people often look to in beauty is the simplicity. So if you look at E equals MC squared. So when a few concepts come together, that's why the Euler identity is often considered the most beautiful equation in mathematics. Do you find beauty in that one in the Euler identity ?

Terence Tao : Yeah, well, as I said, I mean, what I find most appealing is connections between different things that, so E to the pi I equals minus one. So yeah, people always use all the fundamental constants. Okay, that's cute, but to me. (Lex laughing) So the exponential function was introduced by Euler to measure exponential growth. So compound interest or decay. Anything which is continuously growing, continuously decreasing growth and decay, or dilation or contraction is modeled by the exponential function. Whereas pi comes around from circles and rotation. If you want to rotate a needle, for example, 180 degrees, you need to rotate by pi radians. And i, complex numbers, represents this whole imaginary axes of a 90-degree rotation. So a change in direction. So the X metric function represents growth and decay in the direction that you really are. When you stick an i in the exponential, now, instead of motion in the same direction as your current position, it's motion right angles to your current position, so rotation. And then so E to the pi equals minus one tells you that if you rotate for time pi, you end up at the other direction. So it unifies geometry through dilation and exponential growth or dynamics through this act of complexification, rotation by i. So it connects together all these tools in mathematics, dynamics, geometry, and the complex numbers, they're all considered almost, they're all next door neighbors in mathematics because of this identity.

Lex Fridman : Do you think the thing you mentioned is cute, the collision of notations from these disparate fields is just a frivolous side effect ? Or do you think there is legitimate value in one notation ? All our old friends come together at night.

Terence Tao : Well, it's confirmation that you have the right concepts. So when you first study anything, you have to measure things and give them names. And initially, sometimes, because your model is again too far off from reality, you give the wrong things the best names, and you only find out later what's really important.

Lex Fridman : Physicists can do this sometimes, but it turns out okay.

Terence Tao : So actually with physics, so E equals MC squared. Okay, so one of the big things was the E. So when Aristotle first came up with his laws of motion, and then Galileo and Newton and so forth, they saw the things they could measure. They could measure mass and acceleration and force and so forth. And so Newtonian mechanics, for example, F = ma was the famous Newton's second law of motion. So those were the primary objects. So they gave them the central billing in the theory. It was only later, after people started analyzing these equations that there always seemed to be these quantities that were conserved. So in particular momentum and energy. And it's not obvious that things have an energy. It's not something you can directly measure the same way you can measure mass and velocity and so forth. But over time people realized that this was actually a really fundamental concept. Hamilton eventually in 19th century, reformulated Newton's laws of physics into what's called Hamiltonian mechanics, where the energy, which is now called the Hamiltonian, was the dominant object. Once you know how to measure the Hamiltonian of any system, you can describe completely the dynamics, like what happens to all the states. It really was a central actor which was not obvious initially. And this helped. Actually this change of perspective really helped when quantum mechanics came along, because the early physicists who studied quantum mechanics, they had a lot of trouble trying to adapt their Newtonian thinking because everything was a particle and so forth to quantum mechanics, because now everything was a wave. It just looks really, really weird. You ask what is the quantum version of F = ma ? And it's really, really hard to give an answer to that. But it turns out that the Hamiltonian, which was so secretly behind the scenes in classical mechanics, also is the key object in quantum mechanics, that there's also an object called the Hamiltonian. It's a different type of object. It's what's called an operator rather than a function. But again, once you specify it, you specify the entire dynamics. So there's something called Schrodinger's equation that tells you exactly how quantum systems evolve once you have a Hamiltonian. So side by side, they look completely different objects. One involves particles, one involves waves, and so forth. But with this centrality, you could start actually transferring a lot of intuition and facts from classical mechanics to quantum mechanics. So for example, in classical mechanics there's this thing called Noether's theorem. Every time there's a symmetry in a physical system, there is a conservation law. So the laws of physics are translation invariant. Like if I move 10 steps to the left, I experience the same laws of physics as if I was here. And that corresponds to conservation momentum. If I turn around by some angle, again I experience the same laws of physics. This corresponds to the conservation of angular momentum. If I wait for 10 minutes, I still have the same laws of physics. So this time transition invariance, this corresponds to the law of concentration of energy. So there's this fundamental connection between symmetry and conservation. And that's also true in quantum mechanics, even though the equations are completely different. But because they're both coming from the Hamiltonian, the Hamiltonian controls everything. Every time the Hamiltonian has a symmetry, the equations will have a conservation law. So once you have the right language, it actually makes things a lot cleaner. One of the problems why we can't unify quantum mechanics and general relativity yet, we haven't figured out what the fundamental objects are like. For example, we have to give up the notion of space and time being these almost Euclidean type spaces. And we kind of know that at very tiny scales there's going to be quantum fluctuations, there's spacetime foam, and trying to use Cartesian coordinates, XYZ, it's a non-starter, but we don't know how to what to replace it with. We don't actually have the mathematical concepts. The analog got a Hamiltonian that sort of organized everything.

Lex Fridman : Does your gut say that there is a theory of everything, so this is even possible to unify, to find this language that unifies general relativity and quantum mechanics ?

Terence Tao : I believe so. I mean, the history of physics has been that of unification, much like mathematics. Over the years, electricity and magnetism were separate theories and then Maxwell unified them. Newton unified the motions of heavens with the motions of objects on the earth and so forth. So it should happen. It's just that again, to go back to this model for observations and theory, part of our problem is that physics is a victim of its own success, that our two big theories of physics, general relativity and quantum mechanics, are so good now. Together they cover 99.9 % of sort of all the observations we can make. And you have to either go to extremely insane particle accelerations or the early universe or things that are really hard to measure in order to get any deviation from either of these two theories to the point where you can actually figure out how to combine them together. But I have faith that we've been doing this for centuries and we've made progress before. There's no reason why we should stop.

Lex Fridman : Do you think you'll be a mathematician that develops theory of everything ?

Terence Tao : What often happens is that when the physicists need some theory of mathematics, there's often some precursor that the mathematicians worked out earlier. So when Einstein started realizing that space was curved, he went to some mathematician and asked, "Is there some theory of curved space that mathematicians already came up with that could be useful ?" And he said, "Oh, yeah," I think Riemann came up with something. And so, yeah, Riemann had developed Riemannian geometry, which is precisely a theory of spaces that are curved in various general ways, which turned out to be almost exactly what was needed for Einstein's theory. This is going back to Wigner's unreasonable effectiveness of mathematics. I think the theories that work well to explain the universe tend to also involve the same mathematical objects that work well to solve mathematical problems. Ultimately, they're just sort of both ways of organizing data in useful ways.

Lex Fridman : It just feels like you might need to go some weird land that's very hard to intuit. Like you have like string theory.

Terence Tao : Yeah, that was a leading candidate for many decades. I think it's slowly falling out of fashion, because it's not matching experiment.

Lex Fridman : So one of the big challenges, of course, like you said, is experiment is very tough.

Terence Tao : Yes.

Lex Fridman : Because of how effective both theories are. But the other is like just you're talking about, you're not just deviating from space time, you're going into like some crazy number of dimensions. You're doing all kinds of weird stuff. That to us, we've gone so far from this flat Earth that we started at, like you mentioned.

Terence Tao : Yeah, yeah, yeah.

Lex Fridman : Now we're just, it's very hard to use our limited ape descendants of a cognition to intuit what that reality really is like.

Terence Tao : This is why analogies are so important. I mean, so, yeah, the round Earth is not intuitive, because we're stuck on it. But round objects in general, we have pretty good intuition a little bit. And we have intuition about light works and so forth. And it's actually a good exercise to actually work out how eclipses and phases of the sun and the moon and so forth can be really easily explained by round Earth and round moon models. And you can just take a basketball and a golf ball and a light source and actually do these things yourself. So the intuition is there, but you have to transfer it.

Lex Fridman : That is a big leap intellectually for us to go from flat to round Earth, because our life is mostly lived in flatland, to load that information. And we're all like, take it for granted. We take so many things for granted because science has established a lot of evidence for this kind of thing. But we're around rock flying through space. That's a big leap, and you have to take a chain of those leaps the more and more and more we progress.

Terence Tao : Right, yeah. So modern science is maybe, again, a victim of its own success, is that in order to be more accurate, it has to move further and further away from your initial intuition. And so for someone who hasn't gone through the whole process of science education, it looks more and more suspicious because of that. So we need more grounding, I think. I mean, there are scientists who do excellent outreach, but there's lots of science things that you can do at home. There's lots of YouTube videos. I did a YouTube video recently with Grant Sanderson. We talked about this earlier, how the ancient Greeks were able to measure things like the distance of the moon, distance to the Earth, and using techniques that you could also replicate yourself. It doesn't all have to be fancy space telescopes and really intimidating mathematics.

Lex Fridman : Yeah, I highly recommend that. I believe you gave a lecture and you also did an incredible video with Grant. It's a beautiful experience to try to put yourself in the mind of a person from that time, shrouded in mystery. You're on this planet, you don't know the shape of it, the size of it. You see some stars, you see some things, and you try to localize yourself in this world and try to make some kind of general statements about distance to places.

Terence Tao : Change of perspective is really important. You say travel broadens the mind. This is intellectual travel. You know, put yourself in the mind of the ancient Greeks or some other person, some other time period, make hypotheses, spherical cows, whatever, speculate. And this is what mathematicians do and some artists do, actually.

Lex Fridman : It's just incredible that given the extreme constraints, you could still say very powerful things. That's why it's inspiring, looking back in history, how much can be figured out when you don't have much figure out stuff work.

Terence Tao : If you propose axioms, then the mathematics lets you follow those axioms to their conclusions. And sometimes you can get quite a long way from initial hypotheses.

Lex Fridman : If we can stay in the land of the weird. You mentioned general relativity. You've contributed to the mathematical understanding of Einstein's field equations. Can you explain this work ? And from a sort of mathematical standpoint, what aspects of general relativity are intriguing to you, challenging to you ?

Terence Tao : I have worked on some equations. There's something called the wave maps equation, or the sigma field model, which is not quite the equation of spacetime gravity itself, but of certain fields that might exist on top of spacetime. So Einstein's equations of relativity just describe space and time itself. But then there's other fields that live on top of that. There's the electromagnetic field, there's things called Yang-Mills fields, and there's this whole hierarchy of different equations of which Einstein is considered one of the most nonlinear and difficult. But relatively low on the hierarchy was this thing called the wave maps equation. So it's a wave which at any given point is fixed to be like on a sphere. So I can think of a bunch of arrows in space and time, so it's pointing in different directions, but they propagate like waves. If you wiggle an arrow, it will propagate and make all the arrows move, kind of like sheaves of wheat in the wheat field. And I was interested in the global regularity problem again for this question. Is it possible for all the energy here to collect at a point ? So the equation I considered was actually what's called a critical equation, where the behavior at all scales is roughly the same. And I was able, barely to show that you couldn't actually force a scenario where all the energy concentrated at one point, that the energy had to disperse a little bit, and the moment it dispersed a little bit, it would stay regular. Yeah, this was back in 2000. That was part of why I got interested in Navier-Stokes afterwards, actually. Yeah, so I developed some techniques to solve that problem. So part of it is this problem is really nonlinear because of the curvature of the sphere, There was a certain nonlinear effect, which was a non-perturbative effect. When you sort of looked at it normally, it looked larger than the linear effects of the wave equation. And so it was hard to keep things under control, even when your energy was small. But I developed what's called a gauge transformation. So the equation is kind of like an evolution of sheaves of wheat and they're all bending back and forth. And so there's a lot of motion. But if you imagine stabilizing the flow by attaching little cameras at different points in space, which are trying to move in a way that captures most of the motion, and under this stabilized flow, the flow becomes a lot more linear. I discovered a way to transform the equation to reduce the amount of nonlinear effects. And then I was able to solve the equation. I found this transformation while visiting my aunt in Australia. And I was trying to understand the dynamics of all these fields. And I couldn't do it with pen and paper. And I had not enough facility of computers to do any computer simulations. So I ended up closing my eyes, being on the floor, and just imagining myself to actually be this vector field and rolling around and to try to see how to change coordinates in such a way that somehow things in all directions would behave in a reasonably linear fashion. And yeah, my aunt walked in on me while I was doing that and she was asking, what am I doing doing this ?

Lex Fridman : It's complicated, is the answer.

Terence Tao : Yeah, yeah. And she goes, "Okay, fine you're a young man. I don't ask questions."

Lex Fridman : I have to ask about the, how do you approach solving difficult problems ? If it's possible to go inside your mind when you're thinking, are you visualizing in your mind the mathematical objects, symbols, maybe ? What are you visualizing in your mind usually when you're thinking ?

Terence Tao : A lot of pen and paper. One thing you pick up as a mathematician is sort of, I call it cheating strategically. So the beauty of mathematics is that you get to change the problem, change the rules as you wish. You don't get to do this for any other field. If you're an engineer and someone says, "Build a bridge over this river," you can't say, "I want to build this bridge over here instead," or, "I want to put it out of paper instead of steel." But a mathematician, you can do whatever you want. It's like trying to solve a computer game where there's unlimited cheat codes available. And so you can set this dimension that's large, I'll set it to one. I'd solve the one dimensional problem first. There's a main term and an error term. I'm going to make a spherical cow assumption. I assume the error term is zero. And so the way you should solve these problems is not in sort of this Iron Man mode where you make things maximally difficult, but actually the way you should approach any reasonable math problem is that if there are 10 things that are making your life difficult, find a version of the problem that turns off nine of the difficulties, but only keeps one of them, and solve that. So you install nine cheats, okay, if you solve 10 cheats, then the game is trivial, but you install nine cheats, you solve one problem, that teaches you how to deal with that particular difficulty, and then you turn that one off, and you turn something else on, and then you solve that one. And after you know how to solve the 10 problems, 10 difficulties separately, then you have to start merging them a few at a time. As a kid, I watched a lot of these Hong Kong action movies from my culture. And one thing is, every time it's a fight scene, some, maybe the hero gets swarmed by 100 bad guy goons or whatever, but it'll always be choreographed so that he'd always be only fighting one person at a time. And then it would defeat that person and move on. And because of that, he could defeat all of them. But whereas if they had fought a bit more intelligently and just swarmed the guy at once, it would make for much worse cinema, but they would win.

Lex Fridman : Are you usually pen and paper ? Are you working with computer and LaTeX ?

Terence Tao : I'm mostly pen and paper, actually. So in my office I have four giant blackboards and sometimes I just have to write everything I know about the problem on the four blackboards and then sit on my couch and just sort of see the whole thing.

Lex Fridman : Is it all symbols like notation, or is there some drawings ?

Terence Tao : Oh, there's a lot of drawing and a lot of bespoke doodles that only make sense to me. And that's the beauty of a blackboard. You erase. And it's a very organic thing. I'm beginning to use more and more computers, partly because AI makes it much easier to do simple coding things. If I wanted to plot a function before, which is moderately complicated as some iteration or something, I'd have to remember how to set up a Python program and how does a for loop work and debug it. And it would take two hours and so forth. And now I can do it in 10, 15 minutes. I'm using more and more computers to do simple explorations.

LEX FRIDMAN : Let's talk about AI a little bit, if we could. So maybe a good entry point is just talking about computer-assisted proofs in general. Can you describe the Lean formal proof programming language and how it can help as a proof assistant, and maybe how you started using it and how it has helped you ?

TERENCE TAO : So Lean is a computer language much like sort of standard languages like Python and C and so forth, except that in most languages the focus is on using executable code. Lines of code do things, they flip bits or they make a robot move, or they deliver you text on the Internet or something. So Lean is a language that can also do that. It can also be run as a standard traditional language, but it can also produce certificates. So a software language like Python might do a computation and give you that the answer is seven. Does the sum of three plus four is equal to seven ? But Lean can produce not just the answer, but a proof that how it got the answer of seven as three plus four, and all the steps involved. It creates these more complicated objects, not just statements, but statements with proofs attached to them. And every line of code is just a way of piecing together previous statements to create new ones. So the idea is not new. These things are called proof assistants, and so they provide languages for which you can create quite complicated, intricate mathematical proofs. And they produce these certificates that give it 100that your arguments are correct, if you trust the compiler, obviously. But they made the compiler really small and there are several different compilers available for the same level.

LEX FRIDMAN : Can you give people some intuition about the difference between writing on pen and paper versus using Lean programming language ? How hard is it to formalize statement ?

TERENCE TAO : So Lean, a lot of mathematicians were involved in the design of Lean, so it's designed so that individual lines of code resemble individual lines in a mathematical argument. You might want to introduce a variable, you might want to prove our contradiction. There are various standard things that you can do, and it's written so ideally it should be like a one-to-one correspondence. In practice it isn't, because Lean is like explaining a proof to an extremely pedantic colleague who will point out, "Okay, did you really mean this ? What happens if this is zero ? Okay, how do you justify this ?" So Lean has a lot of automation in it to try to be less annoying. So, for example, every mathematical object has to come of a type. Like if I talk about X, is X a real number or a natural number or a function or something ? If you write things informally, it's often in terms of context. You say, "Clearly X is equal to," "Let X be the sum of Y and Z," and Y and Z were already real numbers, so X should also be a real number. So Lean can do a lot of that, but every so often it says, "Wait a minute, can you tell me more about what this object is ? What type of object it is ?" You have to think more at a philosophical level, not just sort of computations that you're doing, but sort of what each object actually is in some sense.

Lex Fridman : Is it using something like LLMs to do the type inference ? Or like you mentioned, with the LLMs ?

Terence Tao : It's using much more traditional, what's called good old-fashioned AI. You can represent all these things as trees and there's always algorithms to match one tree to another tree.

Lex Fridman : So it's actually doable to figure out if something is a real number or a natural number ?

Terence Tao : Every object stuff comes with a history of where it came from and you can kind of trace.

Lex Fridman : Oh, I see.

Terence Tao : Yeah, so it's designed for reliability. So modern AIs are not used in, it's a disjoint technology. People are beginning to use AIs on top of Lean. So when a mathematician tries to program proof in Lean, often there's a step. Okay, now I want to use the fundamental calculus to do the next step. So the Lean developers have built this massive project called Mathlib, a collection of tens of thousands of useful facts about mathematical objects. And somewhere in there is the fundamental theme of calculus, but you need to find it. So a lot, the bottle neck now is actually Lemma Search. There's a tool that you know is in there somewhere and you need to find it. And so there are various search engines specialized for Mathlib that you can do, but there's now these large language models that you can say, "I need the fundamental calculus at this point." And it was like, okay, for example, when I code, I have GitHub Copilot installed as a plugin to my IDE and it scans my text, and it sees what I need, says, I might even type, now I need to use the fundamental theorem of calculus. And then it might suggest, "Okay, try this," and maybe 25time it works exactly. And then another 10, 15the time it doesn't quite work, but it's close enough that I can say, "Oh yeah, if I just change it here and here, it'll work." And then like half the time it gives me complete rubbish. But people are beginning to use AIs a little bit on top, mostly on the level of basically fancy autocomplete, that you can type half of one line of a proof and it will find. It will tell you.

Lex Fridman : Yeah, but a fancy, especially fancy with the sort of capital letter F is remove some of the friction mathematician might feel when they move from pen and paper to formalizing.

Terence Tao : Yes, yeah. So right now I estimate that the effort, time and effort taken to formalize a proof is about 10 times the amount taken to write it out. Yeah, so it's doable, but it's annoying.

Lex Fridman : But doesn't it kill the whole vibe of being a mathematician having a pedantic coworker ?

Terence Tao : Right, yeah, if that was the only aspect of it, okay, but, okay, there are some cases where it's actually more pleasant to do things formally. So there was a theorem I formalized and there was a certain constant 12 that came out in the final statement. And so this 12 had to be carried all through the proof and everything had to be checked, that all these other numbers had to be consistent with this final number 12. And then so we wrote a paper through this theorem with this number 12, and then a few weeks later, someone said, "Oh, we can actually improve this 12 to an 11 by reworking some of these steps." And when this happens with pen and paper, every time you change a parameter, you have to check line by line that every single line of your proof still works. And there can be subtle things that you didn't quite realize, some properties on number 12 that you didn't even realize that you were taking advantage of. So a proof can break down at a subtle place. So we had formalized the proof with this constant 12. And then when this new paper came out, we said, "Oh, so that took like three weeks to formalize and like 20 people to formalize this original proof." I said, "Oh, but now let's update the 12 to 11." And what you can do with Lean is that in your headline theorem, you change a 12 to 11. You run the compiler, and of the thousands of lines of code you have, 90 % of them still work. And there's a couple that are lined in red. Now, I can't justify these steps, but it immediately isolates which steps you need to change. But you can skip over everything, which works just fine. And if you program things correctly with good programming practices, most of your lines will not be red. And there'll just be a few places where you, I mean, if you don't hard code your constants, but you sort of use smart tactics and so forth, you can localize the things you need to change to a very small period of time. So it's like within a day or two, we had updated our proof because this is a very quick process. You make a change. There are 10 things now that don't work. For each one, you make a change. And now there's five more things that don't work. But the process converges much more smoothly than with pen and paper.

Lex Fridman : So that's for writing. Are you able to read it ? Like, if somebody else sends a proof, are you able to, what's the, versus paper ?

Terence Tao : Yeah, so the proofs are longer, but each individual piece is easier to read. So if you take a math paper and you jump to page 27 and you look at paragraph six and you have a line of text of math, I often can't read it immediately because it assumes various definitions, which I have to go back, and maybe 10 pages earlier this was defined, and the proof is scattered all over the place, and you basically are forced to read fairly sequentially. It's not like, say, a novel, where in theory you could open up a novel halfway through and start reading. There's a lot of context. But with a proof in Lean, if you put your cursor on a line of code, every single object there, you can hover over it and it will say what it is, where it came from, where the stuff is justified. You can trace things back much easier than of flipping through a math paper. So one thing that Lean really enables is actually collaborating on proofs at a really atomic scale that you really couldn't do in the past. So traditionally with pen and paper, when you want to collaborate with another mathematician, either you do it at a blackboard where you can really interact, but if you're doing it sort of by email or something, basically you have to segment it. I'm going to finish section three, you do section four, but you can't really sort of work on the same thing collaboratively at the same time. But with Lean, you can be trying to formalize some portion of the proof and say, "I got stuck at line 67 here, I need to prove this thing, but it doesn't quite work. Here's the three lines of code I'm having trouble with." But because all the context is there, someone else can say, "Oh, okay, I recognize what you need to do. You need to apply this trick or this tool," and you can do extremely atomic-level conversations. So because of Lean, I can collaborate with dozens of people across the world, most of whom I don't have never met in person, and I may not know actually even whether how reliable they are in the proofs they give me, but Lean gives me a certificate of trust, so I can do trustless mathematics.

Lex Fridman : So there's so many interesting questions. So one, you're known for being a great collaborator. So what is the right way to approach solving a difficult problem in mathematics when you're collaborating ? Are you doing a divide and conquer type of thing, or are you focusing on a particular part and you're brainstorming ?

Terence Tao : There's always a brainstorming process first. Yeah. So math research projects, by their nature, when you start, you don't really know how to do the problem. It's not like an engineering project where somehow the theory has been established for decades and its implementation is the main difficulty body. You have to figure out even what is the right path. So this is what I said about cheating first. It's like, to go back to the bridge building analogy, first, assume you have infinite budget and unlimited amounts of workforce and so forth. Now can you build this bridge ? Okay, now have an infinite budget but only finite workforce. Now can you do that, and so forth. Of course, no engineer can actually do this. They have fixed requirements. Yes, there's this sort of jam sessions always at the beginning, where you try all kinds of crazy things, and you make all these assumptions that are unrealistic but you plan to fix later. And you try to see if there's even some skeleton of an approach that might work, and then hopefully that breaks up the problem into smaller sub-problems, which you don't know how to do, but then you focus on the sub ones, and sometimes different collaborators are better at working on certain things. So one of my theorems I'm known for is a theorem of Ben Green, which is now called the Green-Tao theorem. It's a statement that the primes contain arithmetic progressions of any length. So it's a modification of this theorem of Szemeredi. And the way we collaborated was that Ben had already proven a similar result for progressions of length three. He showed that sets like the primes contain lots and lots of progressions of length three. And even subsets of the primes, certain subsets do. But his techniques only worked for length three progressions. They didn't work for longer progressions. But I had these techniques coming from ergodic theory, which is something that I had been playing with, and I knew better than Ben at the time. And so if I could justify certain randomness properties of some set relating to the primes, like there's a certain technical condition, which if I could have it, if Ben could supply me to this fact, I could conclude the theorem. But what I asked was a really difficult question in number theory, which he said, "There's no way we can prove this." So he said, "Can you prove your part of the theorem using a weaker hypothesis, that I have a chance to prove ?" And he proposed something which he could prove, but it was too weak for me. I can't use this. So there was this conversation going back and forth.

Lex Fridman : Different cheats too.

Terence Tao : Yeah, I want to cheat more, he wants to cheat less. But eventually we found a property which, A, he could prove, and B, I could use, and then we could prove our view. There are all kinds of dynamics. I mean, every collaboration has some story. No two are the same.

Lex Fridman : And then on the flip side of that, like you mentioned with Lean programming, now that's almost like a different story, because you can create, I think you've mentioned a kind of a blueprint for a problem, and then you can really do a divide and conquer with Lean, where you're working on separate parts and they're using the computer system proof checker, essentially to make sure that everything is correct along the way.

Terence Tao : Yeah. So it makes everything compatible and trustable. So currently only a few mathematical projects can be cut up in this way. At the current state of the art, most of the Lean activity is on formalizing proofs that have already been proven by humans. A math paper basically is a blueprint in a sense. It is taking a difficult statement like big theorem and breaking it up into 100 little numbers is, but often not all written with enough detail that each one can be sort of directly formalized. A blueprint is like a really pedantically written version of a paper where every step is explained to as much detail as possible and trying to make each step kind of self-contained or depending on only a very specific number of previous statements that have been proven, so that each node of this blueprint graph that gets generated can be tackled independently of the others, and you don't even need to know how the whole thing works. So it's like a modern supply chain. If you want to create an iPhone or some other complicated object, no one person can build a single object, but you can have specialists who just if they're given some widgets from some other company, they can combine them together to form a slightly bigger widget.

Lex Fridman : I think that's a really exciting possibility because if you can find problems that could be broken down this way, then you could have thousands of contributors, right ? Completely distributed.

Terence Tao : Yes, yes, yes. So I told you before about the split between theoretical and experimental mathematics. And right now most mathematics is theoretical and only a tiny bit is experimental. I think the platform that Lean and other software tools, so GitHub and things like that will allow experimental mathematics to scale up to a much greater degree than we can do now. So right now, if you want to do any mathematical exploration of some mathematical pattern or something, you need some code to write out the pattern. And I mean, sometimes there are some computer algebra packages that help, but often it's just one mathematician coding lots and lots of Python or whatever. And because coding is such an error prone activity, it's not practical to allow other people to collaborate with you on writing modules for your code, because if one of the modules has a bug in it, the whole thing is unreliable. So you get these bespoke spaghetti code written by not professional programmers, but mathematicians, and they're clunky and slow. And so because of that, it's hard to really mass produce experimental results. But I think with Lean, I mean, I'm already starting some projects where we are not just experimenting with data, but experimenting with proofs. So I have this project called the Equational Theories Project. Basically, we generated about 22 million little problems in abstract algebra. Maybe I should back up and tell you what the project is. Okay, so abstract algebra studies operations like multiplication and addition and their abstract properties. So multiplication, for example, is commutative. X times Y is always Y times X, at least for numbers. And it's also associative. X times Y times Z is the same as X times Y times Z. So these operations obey some laws that don't obey others. For example, X times X is not always equal to X, so that law is not always true. So given any operation, it obeys some laws and not others. And so we generated about 4,000 of these possible laws of algebra that certain operations can satisfy. And our question is, which laws imply which other ones ? So, for example, does commutativity imply associativity ? And the answer is no, because it turns out you can describe an operation which obeys the commutative law but doesn't obey the associative of law. So by producing an example, you can show that commutativity does not imply associativity, but some other laws do imply other laws by substitution and so forth. And you can write down some algebraic proof. So we look at all the pairs between these 4,000 laws, and there's about 22 million of these pairs. And for each pair we ask, "Does this law imply this law ? If so, give a proof. If not, give a counterexample." So 22 million problems, each one of which you could give to like an undergraduate algebra student, and they had a decent chance of solving the problem. Although there are a few, at least 22 million, there are like 100 or so that are really quite hard, but a lot are easy. And the project was just to work out to determine the entire graph, like which ones imply which other ones.

Lex Fridman : That's an incredible project, by the way. Such a good idea, such a good test of the very thing we've been talking about at a scale. That's remarkable.

Terence Tao : Yeah, so it would not have been feasible. I mean, the state of the art in the literature was like 15 equations and sort of how they applied. That's sort of at the limit of what a human repentant paper can do. So you need to scale that up. So you need to crowdsource, but you also need to trust all the, I mean, no one person can check 22 million of these proofs. You needed to be computerized. And so it only became possible with Lean. We Were hoping to use a lot of AI as well. So the project is almost complete. So of these 22 million, all but two have been settled.

Lex Fridman : Wow.

Terence Tao : Actually, and of those two, we have a pen and paper proof for two, and we're formalizing it. In fact, this morning I was working on finishing it. So we're almost done on this.

Lex Fridman : It's incredible.

Terence Tao : Yeah, fantastic.

Lex Fridman : How many people were you able to get ?

Terence Tao : About 50, which in mathematics is considered a huge number.

Lex Fridman : It's a huge number. That's crazy.

Terence Tao : Yeah. So we're going to have a paper of 50 authors and a big appendix of who contributor what.

Lex Fridman : Here's an interesting question. Not to maybe speak even more generally about it. When you have this pool of people, is there a way to organize the contributions by level of expertise of the people of the contributors ? Now, okay, I'm asking a lot of pothead questions here, but I'm imagining a bunch of humans and maybe in the future some AIs, can there be like an ELO rating type of situation where like a gamification of this ?

Terence Tao : The beauty of these Lean projects is that automatically you get all this data. So everything's been uploaded to this GitHub and GitHub tracks who contributed what, so you could generate statistics at any later point in time. You could say, “Oh, this person contributed this many lines of code or whatever.” These are very crude metrics. I would definitely not want this to become part of your tenure review or something. But I think already in enterprise computing, people do use some of these metrics as part of the assessment of performance of an employee. Again, this is a direction which is a bit scary for academics to go down. We don’t like metrics so much.

Lex Fridman : And yet academics use metrics. They just use old ones. Number of papers.

Terence Tao : Yeah, yeah, it’s true. It’s true that, yeah, I mean.

Lex Fridman : It feels like this is a metric, while flawed, is going more in the right direction, right ?

Terence Tao : Yeah.

Lex Fridman : It’s interesting. At least it’s a very interesting metric.

Terence Tao : Yeah, I think it’s interesting to study. I think you can do studies of whether these are better predictors. There’s this problem called Goodhart’s Law. If a statistic is actually used to incentivize performance, it becomes gamed, and then it is no longer a useful measure.

Lex Fridman : Oh, humans always game.

Terence Tao : Yeah, yeah. No, it’s rational. So what we’ve done for this project is self-report. So there are actually standard categories from the sciences of what types of contributions people give. So there’s this concept and validation and resources and coding and so forth. So there’s a standard list of 12 or so categories, and we just ask each contributor to, there’s a big matrix of all the authors and all the categories just to tick the boxes where they think that they contributed, and just give a rough idea. So you did some coding and you provided some compute, but you didn’t do any of the pen and paper verification or whatever. And I think that works out. Traditionally, mathematicians just order alphabetically by certain name. So we don’t have this tradition, as in the sciences, of lead author and second author and so forth, which we’re proud of. We make all the authors equal status, but it doesn’t quite scale to this size. So a decade ago I was involved in these things called Polymath Projects. It was the crowdsourcing mathematics, but without the Lean component. So it was limited by you needed a human moderator to actually check that all the contributions coming in were actually valid. And this was a huge bottleneck, actually. But still we had projects that were 10 authors or so, but we had decided at the time not to try to decide who did what, but to have a single pseudonym. So we created this fictional character called DHJ Polymath. In the spirit of Bourbaki. Bourbaki is the pseudonym for a famous group of mathematicians in the 20th century. And so the paper was authored under pseudonym, so none of us got the author credit. This actually turned out to be not so great for a couple of reasons. So one is that if you actually wanted to be considered for tenure or whatever, you could not use this paper as you submitted on your publications because you didn’t have the formal author credit. But the other thing that we’ve recognized until much later is that when people referred to these projects, they naturally refer to the most famous person who was involved in the project. Oh, so this was Tim

Gowers' Polymath Project, this was Terence Tao's Polymath Project, and not mention the other 19 or whatever people that were involved. So we're trying something different this time around, where everyone's an author, but we will have an appendix with this matrix and we'll see how that works.

Lex Fridman : So both projects are incredible. Just the fact that you're involved in such huge collaborations. But I think I saw a talk from Kevin Buzzard about the Lean programming language just a few years ago, and you're saying that this might be the future of mathematics. And so it's also exciting that you're embracing, one of the greatest mathematicians in world, embracing this what seems like the paving of the future of mathematics. So I have to ask you here about the integration of AI into this whole process. So DeepMind's AlphaProof was trained using reinforcement learning on both failed and successful formal Lean proofs of IMO problems. So this is sort of high-level high school.

Terence Tao : Oh, very high level.

Lex Fridman : Yes, very high-level high school level mathematics problems. What do you think about the system and maybe what is the gap between this system that is able to prove the high school level problems versus graduate level problems ?

Terence Tao : Yeah, the difficulty increases exponentially with the number of steps involved in the proof. It's a combinatorial explosion. So the thing of large language models is that they make mistakes. And so if a proof has got 20 steps and your has a 10at each step of going in the wrong direction, it's extremely unlikely to actually reach the end.

Terence Tao : Actually, just to take a small tangent here, how hard is the problem of mapping from natural language to the formal program ?

Lex Fridman : Oh yeah, it's extremely hard actually. Natural language, it's very fault tolerant. Like you can make a few minor grammatical errors and a speaker in the second language can get some idea of what you're saying. But formal language, if you get one little thing wrong, the whole thing is nonsense. Even formal to formal is very hard. There are different incompatible prefaces in languages. There's Lean, but also Coq and Isabelle and so forth. Even converting from a formal language to formal language is an unsolved, basically unsolved problem.

Terence Tao : That is fascinating. Okay, so but once you have an informal language, they're using their RL-trained model. So something akin to AlphaZero that they used to go to then try to come up with proofs. They also have a model. I believe it's a separate model for geometric problems. So what impresses you about this system, and what do you think is the gap ?

Lex Fridman : Yeah, yeah, we talked earlier about things that are amazing over time become kind of normalized. So now somehow, of course geometry is a solvable problem.

Terence Tao : Right, that's true, that's true. I mean it's still beautiful.

LEX FRIDMAN : Yeah, yeah, no, these are great work that shows what's possible. The approach doesn't scale currently. There are three days of Google server time to solve one high school math problem there. This is not a scalable prospect, especially with the exponential increase as the complexity increases.

TERENCE TAO : We should mention that they got a silver medal performance.

LEX FRIDMAN : The equivalent of.

LEX FRIDMAN : Equivalent of a silver medal performance.

TERENCE TAO : So first of all, they took way more time than was allotted, and they had this assistance where the humans helped by formalizing, but also they're giving us those full marks for the solution, which I guess is formally verified. So I guess that's fair. There are efforts, there will be a proposal at some point to actually have an AI math olympiad where at the same time as the human contestants get the actual olympiad problems, AIs will also be given the same problems with the same time period. And the outputs will have to be graded by the same judges, which means that it will have to be written in natural language rather than formal language.

LEX FRIDMAN : Oh, I hope that happens. I hope that this IMO it happens. I hope the next one.

TERENCE TAO : It won't happen this IMO. The performance is not good enough in the time period. But there are smaller competitions. There are competitions where the answer is a number rather than a long form proof. And AIs are actually a lot better at problems where there's a specific numerical answer, because it's easy to do reinforcement learning on it. You got the right answer, you got the wrong answer, it's a very clear signal. But a long form proof either has to be formal and then the lean can give it a thumbs up, thumbs down, or it's informal. But then you need a human to grade it. And if you're trying to do billions of reinforcement learning runs, you can't hire enough humans to grade those. It's already hard enough for the last language models to do reinforcement learning on just the regular text that people get. But now if you actually hire people, not just give thumbs up, thumbs down, but actually check the output mathematically, yeah, that's too expensive.

LEX FRIDMAN : So if we just explore this possible future, what is the thing that humans do that's most special in mathematics, so that you could see AI not cracking for a while ? So inventing new theories, so coming up with new conjectures versus proving the conjectures, building new abstractions, new representations, maybe an AI Terence style with seeing new connections between disparate fields ?

TERENCE TAO : That's a good question. I think the nature of what mathematicians do over time has changed a lot. So 1,000 years ago, mathematicians had to compute the date of Easter. And there's really complicated calculations, but it's all automated, been automated for centuries. We don't need that anymore. They used to navigate, to do spherical navigation, spherical trigonometry, to navigate how to get from the old world to the new. So I think a very complicated calculation, again, being automated. Even a lot of undergraduate mathematics, even before AI, like Wolfram Alpha, for example, is not a language model, but it can solve a lot of undergraduate level math tasks. So on the computational side, verifying routine things like having a problem, and I say, "Here's a problem in partial differential equations, could you solve it using any of the 20 standard techniques ?" And the AI will say, "Yes, I've tried all 20 and here are the 100 different permutations, and here's my results." And that type of thing, I think it will work very well. Type of scaling to once you solve one problem, to make the AI attack 100 adjacent problems. The things that humans do still. So where the AI really struggles right now is knowing when it's made a wrong turn that it can say, "Oh, I'm going to solve this problem. I'm going to split up this problem into these two cases. I'm going to try this technique." And sometimes if you're lucky and it's a simple problem, it's the right technique and you solve the problem. And sometimes it will have a problem. It would propose an approach which is just complete nonsense, but it looks like a proof. So this is one annoying thing about LLM-generated mathematics. We've had human generated mathematics that's very low-quality. Like submissions from people who don't have the formal training and so forth. But if a human proof is bad, you can tell there's bad proof pretty quickly. It makes really basic mistakes. But the AI-generated proofs, they can look superficially flawless. And that's partly because that's what the reinforcement learning has actually trained them to do, to produce text that looks like what is correct, which for many applications is good enough. So the errors are often really subtle, and then when you spot them, they're really stupid. No human would have actually made that mistake.

Lex Fridman : Yeah, it's actually really frustrating in the programming context because I program a lot. And yeah, when a human makes when a low-quality code, there's something called code smell. You can tell. You can tell immediately, like, okay, there's signs, but with AI generate code.

Terence Tao : Odorless.

Lex Fridman : And then you're right, eventually you find an obvious dumb thing that just looks like good code.

Terence Tao : Yeah.

Lex Fridman : It's very tricky too, and frustrating for some reason to work.

Terence Tao : Yeah. So the sense of smell. There you go. This is one thing that humans have and there's a metaphorical mathematical smell. It's not clear how to get the AIs to duplicate that. Eventually, I mean, so the way AlphaZero and so forth, they make progress on go and chess and so forth, is in some sense they have developed a sense of smell for go and chess positions, that this position is good for white, that's good for black. They can't enunciate why, but just having that sense of smell lets them strategize. So if AIs gain that ability to sort of assess the viability of certain proof strategies. So you can say, I'm going to try to break up this problem into two smaller sub-tasks, and they can say, "Oh, this looks good." Two tasks look like they're simpler tasks than your main task and they still got a good chance of being true. So this is good to try. Or, no, you made the problem worse because each of the two sub-problems is actually harder than your original problem, which is actually what normally happens if you try a random thing to try. Normally it's very easy to transform a problem into an even harder problem. Very rarely do you transform to a simpler problem. So if they can pick up a sense of smell, then they could maybe start competing with human level mathematicians.

Lex Fridman : So this is a hard question, but not competing, but collaborating. Okay, hypothetical. If I gave you an oracle that was able to do some aspect of what you do and you could just collaborate with it.

Terence Tao : Yeah, yeah, yeah.

Lex Fridman : What would that oracle, what would you like that oracle to be able to do ? Would you like it to maybe be a verifier, like, check. Do the code smell, like you're, yes, Professor Tao, this is the correct, this is a promising, fruitful direction.

Terence Tao : Yeah, yeah, yeah.

Lex Fridman : Or would you like it to generate possible proofs and then you see which one is the right one ? Or would you like it to maybe generate different representation, totally different ways of seeing this problem ?

Terence Tao : Yeah, I think all of the above. A lot of it is we don't know how to use these tools because it's a paradigm, that it's not, we have not had in the past assistants that are competent enough to understand complex instructions that can work at massive scale, but are also unreliable. It's interesting, unreliable in subtle ways whilst providing sufficiently good output. It's an interesting combination. You have graduate students that you work with who kind of like this, but not at scale. And we had previous software tools that can work at scale, but very narrow. So we have to figure out how to use them. I mean, so Tim Gowers actually, he actually foresaw in 2000 he was envisioning what mathematics would look like in actually two and a half decades. (Lex and Terence laughing)

Lex Fridman : That's funny.

Terence Tao : Yeah. He wrote in his article, like a hypothetical conversation between a mathematical assistant of the future and himself. He's trying to solve a problem and they would have a conversation. Sometimes the human would propose an idea and the AI would evaluate it, and sometimes the AI would pose an idea and sometimes a computation was required and AI would just go and say, "Okay, I've checked the 100 cases needed here." Or, "You said this is true for all N. I've checked to put N up to 100 and it looks good so far." Or, "Hang on, there's a problem at N equals 46." So just a free form conversation where you don't know in advance where things are going to go. But just based on, I think ideas are good proposed on both sides. Calculations get proposed on both sides. I've had conversations with AI where I say, "Okay, we're going to collaborate to solve this math problem," and it's a problem that I already know the solution to, so I try to prompt it. Okay, so here's the problem. I suggest using this tool and it'll find this lovely argument using a completely different tool which eventually goes into the weeds and say, "No, no, no, try using this," and it might start using this and then it'll go back to the tool that it wanted to do before, and you have to keep railroading it onto the path you want. And I could eventually force it to give the proof I wanted, but it was like herding cats. And the amount of personal effort I had to take to not just sort of prompt it, but also check its output because a lot of what it looked like is going to work. And I know there's a problem on line 17 and basically arguing with was more exhausting than doing it unassisted, but that's the current state the of the art.

Lex Fridman : I wonder if there's a phase shift that happens to where it no longer feels like herding cats, and maybe it'll surprise us how quickly that comes.

Terence Tao : I believe so. In formalization, I mentioned before that it takes 10 times longer to formalize a proof than to write it by hand. With these modern AI tools and also just better tooling, the Lean developers are doing a great job adding more and more features and making it user friendly. It's going from nine to eight to seven. Okay, no big deal. But one day it will drop below one, and that's the phase shift, because suddenly it makes sense when you write a paper to write it in Lean first or through a conversational AI who is generating Lean on the fly with you, and it becomes natural for journals to accept. Maybe they'll offer expedite refereeing. If a paper has already been formalized in Lean, they would just ask the referee to comment on the significance of the results and how it connects to literature and not worry so much about the correctness because that's been certified. Papers are getting longer and longer in mathematics and actually it's harder and harder to get good refereeing for the really long ones unless they're really important. It is actually an issue which the formalization is coming in at just the right time for this to be.

Lex Fridman : The easier and easier it gets because of the tooling and all the other factors, then you're going to see much more, like Mathlib will grow potentially exponentially. It's a virtuous cycle, okay.

Terence Tao : I mean one face shift of this type that happened in the past was the adoption of LaTeX. So LaTeX is this typesetting language that all mathematicians use now. So in the past people used all kinds of word processors and typewriters and whatever, but at some point LaTeX became easier to use than all other competitors and people would switch within a few years. It was just a dramatic phase shift.

Lex Fridman : It's a wild out-there question, but what year ? How far away are we from a AI system being a collaborator on a proof that wins the Fields Medal, so that level ?

Terence Tao : Okay. Well it depends on the level of collaboration.

Lex Fridman : No, like it deserves to be to get the Fields Medal, so half and half.

Terence Tao : Already, I could imagine if it was medal-winning paper having some AI assistance in writing it. The autocomplete alone, I use it. It speeds up my own writing. You can have a theorem, you have a proof, and the proof has three cases, and I write down the proof of the first case, and the autocomplete just suggests now here's how the proof of the second case could work. And it was exactly correct. That was great. Saved me like 5, 10 minutes of typing.

Lex Fridman : But in that case the AI system doesn't get the Fields Medal. (chuckles)

Terence Tao : No.

Lex Fridman : Are we talking 20 years, 50 years, 100 years ? What do you think ?

Terence Tao : Okay, so I gave a prediction in print. So by 2026, which is now next year there will be math collaborations with AI. So not Fields Medal winning, but actual research level papers.

Lex Fridman : Like published ideas that are in part generated by AI.

Terence Tao : Maybe not the ideas, but at least some of the computations, verifications.

Lex Fridman : Has that already happened ?

Terence Tao : That's already happened, yeah. There are problems that were solved by a complicated process conversing with AI to propose things. And the human goes and tries it, and comes back, doesn't work. But it might propose a different idea. It's hard to disentangle exactly. There are certainly math results which could only have been accomplished because there was a human mathematician and an AI involved, but it's hard to sort of disentangle credit. I mean these tools, they do not replicate all the skills needed to do mathematics, but they can replicate sort of some non-trivial percentage of them, 30, 40 % so they can fill in gaps. So coding is a good example. It's annoying for me to code in Python. I'm not a native, I'm not a professional programmer, but with AI the friction cost of doing it is much reduced, so it fills in that gap for me. AI is getting quite good at literature review. I mean there's still a problem with hallucinating references that don't exist, but this I think is a solvable problem. If you train in the right way and so forth, and verify using the Internet, you should in a few years get the point where you have a lemma that you need and say, "Has anyone proven this lemma before ?" And it will do basically a fancy web search AI assistant and say, "Yeah, there are these six papers where something similar has happened." And I mean you can ask it right now and it will give you six papers of which maybe one is legitimate and relevant, one exists but is not relevant and four are hallucinated. It has a non-zero success rate right now, but there's so much garbage, the signal to noise ratio is so poor that it's most helpful when you already somewhat know the literature and you just need to be prompted to be reminded of a paper that was really subconsciously in your memory.

Lex Fridman : Versus helping you discover new you were not even aware of but is the correct citation.

Terence Tao : Yeah, that's, yeah, that it can sometimes do, but when it does, it's buried in a list of options for which the other.

Lex Fridman : They're bad.

Terence Tao : Yeah.

Lex Fridman : I mean being able to automatically generate a related work section that is correct, that's actually a beautiful thing that might be another phase shift, because it assigns credit correctly.

Terence Tao : Yeah.

Lex Fridman : It breaks you out of the silos of thought.

Terence Tao : Yeah, no, there's a big hump to overcome right now. I mean it's like self-driving cars. The safety margin has to be really high for it to be feasible. So yeah, so there's a last mile problem with a lot of AI applications that they can develop tools that work 20 %, 80 % of the time, but it's still not good enough. And in fact even worse than good in some ways.

Lex Fridman : I mean another way of asking the Fields Medal question is what year do you think you'll wake up and be like real surprised you read the headline the news of something happened that AI did, like a real breakthrough, something. It doesn't like Fields Medal, Riemann hypothesis, it could be like really just this AlphaZero moment would go, that kind of thing.

Terence Tao : Yeah. This decade I can see it making a conjecture between two things that people thought was unrelated.

Lex Fridman : Oh, interesting. Generating a conjecture that's a beautiful conjecture.

Terence Tao : Yeah. And actually has a real chance of being correct and meaningful.

Lex Fridman : Because that's actually kind of doable, I suppose. But where the data is. Yeah, no, that would be truly amazing.

Terence Tao : The current models struggle a lot. I mean, so a version of this is, the physicists have a dream of getting the AIs to discover new laws of physics. The dream is you just feed it all this data, and here is a new pattern that we didn't see before. But it actually even struggles, the current state of the art even struggles to discover old laws of physics from the data. Or if it does, there's a big concern of contamination. That it did it only because somewhere in this training did it somehow knew Boyle's Law or whatever law that you're trying to reconstruct. Part of it is that we don't have the right type of training data for this. So for laws of physics, we don't have a million different universes with a million different laws of nature. A lot of what we're missing in math is actually the negative space of, so we have published things of things that people have been able to prove and conjectures that ended up being verified or maybe counterexamples produced. But we don't have data on things that were proposed and they're kind of a good thing to try, but then people quickly realized that it was the wrong conjecture and then they said, "Oh, but we should actually change our claim to modify it in this way to actually make it more plausible." There's a trial and error process which is a real integral part of human mathematical discovery, which we don't record because it's embarrassing. We make mistakes and we only like to publish our wins. And the AI has no access to this data to train on. I sometimes joke that basically AI has to go through grad school and actually go to grad courses, do the assignments, go to office hours, make mistakes, get advice on how to correct the mistakes and learn from that.

Lex Fridman : Let me ask you, if I may, about Grigori Perelman. You mentioned that you try to be careful in your work and not let a problem completely consume you. Just you've really fallen in love with the problem and really cannot rest until you solve it. But you also hasted to add that sometimes this approach actually can be very successful. An example you gave is Grigori Perelman, who proved the Poincare conjecture and did so by working alone for seven years with basically little contact with the outside world. Can you explain this one Millennium Prize Problem that's been solved, Poincare conjecture, and maybe speak to the journey that Grigori Perelman's been on ?

Terence Tao : All right, so it's a question about curved spaces. Earth is a good example. So Earth you can think of as a 2D surface. And just moving around the Earth, it could maybe be a torus with a hole in it, or it could have many holes. And there are many different topologies a priori that a surface could have, even if you assume that it's bounded and smooth, and so forth. So we have figured out how to classify surfaces. As a first approximation, everything's determined by something called the genus, how many holes it has. So a sphere has genus zero, a donut has genus one, and so forth. And one way you can tell these surfaces apart property the sphere has, which is called simply connected. If you take any closed loop on the sphere, like a big closed sort of rope, you can contract it to a point while staying on the surface. And the sphere has this property, but a torus doesn't. If you're on a torus and you take a rope that goes around say the outer diameter torus, there's no way it can't get through the hole. There is no way to contract it to a point. So it turns out that the sphere is the only surface with this property of contractibility up to like continuous deformations of the sphere. So things that are what are called topologically equivalent of the sphere. So Poincare asks the same question in higher dimensions. So this it becomes hard to visualize, because surface you can think of as embedded in three dimensions. But a curved free space, we don't have good intuition of 4D space to live it. And there are also 3D space spaces that can't even fit into four dimensions. You need five or six or higher. But anyway, mathematically you can still pose this question that if you have a bounded three dimensional space now, which also has this simply connective property that every loop can be contracted, can you turn it into a three dimensional version of a sphere ? And so this is the point where conjecture, weirdly in higher dimensions, 4 and 5 was actually easier, so it was solved first in higher dimensions. There's somehow more room to do the deformation, it's easier to move things around a sphere. But three was really hard. So people tried many approaches. There's sort of commentary approaches where you chop up the surface into little triangles or tetrahedron, and you just try to argue based on how the faces interact each other. There were algebraic approaches. There's various algebraic objects, like things called the fundamental group that you can attach to these homology and cohomology and all these very fancy tools. They also didn't quite work. But Richard Hamilton proposed a partial differential equations approach. So the problem is that you have this object which is sort of secretly the sphere, but it's given to you in a weird way. So think of a ball that's being kind of crumpled up and twisted, and it's not obvious that it's a ball, but if you have some sort of surface, which is a deformed sphere, you could, for example, think of it as the surface of a balloon. You could try to inflate it. You blow it up and naturally, as you fill it with air, the wrinkles will sort of smooth out and it will turn into a nice, round sphere. Unless of course, it was a torus or something, in which case it would get stuck at some point. Like if you inflate a torus, there'll be a point in the middle when the inner ring shrinks to zero. You get a singularity, and you can't blow up a any further, you can't flow any further. So he created this flow, which is now called Ricci flow, which is a way of taking an arbitrary surface or space and smoothing it out to make it rounder and rounder, to make it look like a sphere. And he wanted to show that either this process would give you a sphere or it would create a singularity. Very much like how PDEs, either they have global regularity or finite time blowup. Basically, it's almost exactly the same thing. It's all connected. And he showed that for two dimensions, two-dimensional surfaces, if you started something like no singularities ever form, you never ran into trouble, and you could flow, and it will give you a sphere. So he got a new proof of the two dimensional result.

Lex Fridman : But by the way, that's a beautiful explanation of Ricci flow and its application in this context. How difficult is the mathematics here for the 2D case ?

Terence Tao : Yeah, these are quite sophisticated equations, on par with the Einstein equations. Slightly simpler, but yeah, but they were considered hard nonlinear equations to solve. And there's lots of special tricks in 2D that helped. But in 3D, the problem was that this equation was actually supercritical. The same problems as Navier-Stokes, as you blow up, maybe the curvature would get concentrated in finer, smaller regions. And it looked more and more nonlinear and things just looked worse and worse. And there could be all kinds of singularities that showed up. Some singularities, there's these things called neck pinchers, where the surface sort of behaves like a barbell and it pinches at a point. Some singularities are simple enough that you can sort of see what to do next. You just make a snip and then you can turn one surface into two and evolve them separately. But there was the prospect that there's some really nasty knotted singularities showed up that you couldn't see how to resolve in any way that you couldn't do any surgery to. So you need to classify all the singularities, like what are all the possible ways that things can go wrong ? So what Perelman did was, first of all, he turned the problem from a supercritical problem to a critical problem. I said before about how the invention of energy, the Hamiltonian really clarified Newtonian mechanics. So he introduced something which is now called Perelman's reduced volume and Perelman's entropy. And he introduced new quantities, kind of like energy, that looked the same at every single scale and turned the problem into a critical one where the nonlinearities actually suddenly looked a lot less scary than they did before. And then he had to solve. He still had to analyze the singularities of this critical problem. And that itself was a problem similar to this wave maps thing I worked on, actually. So on the level of difficulty of that. So he managed to classify all the singularities of this problem and show how to apply surgery to each of these. And through that was able to resolve the Poincare conjecture. So quite like a lot of really ambitious steps and nothing that a large language model today, for example, could. I mean, at best I could imagine proposing this idea as one of hundreds of different things to try, but the other 99 would be complete dead ends, but you'd only find out after months of work. He must have had some sense that this was the right track to pursue, because it takes years to get from A to B.

Lex Fridman : So you've done, like you said, actually, you see, even strictly mathematically, but more broadly, in terms of the process, you've done similarly difficult things. What can you infer from the process he was going through, because he was doing it alone. What are some low points in a process like that ? When you start to, like you've mentioned hardship, like AI doesn't know when it's failing. What happens to you, you're sitting in your office, when you realize the thing you do did for the last few days, maybe weeks, is a failure ?

Terence Tao : Well, for me, I switch to a different problem. (Lex laughing) So as I said, I'm a fox, I'm not a hedgehog.

Lex Fridman : But you legitimately, that is a break that you can take is to step away and look at a different problem ?

Terence Tao : Yeah, you can modify the problem too. I mean, you can ask some cheat if there's a specific thing that's blocking you that some bad case keeps showing up for which your tool doesn't work. You can just assume by fiat this bad case doesn't occur. So you do some magical thinking, but strategically, okay, to see if the rest of the argument goes through. If there's multiple problems with your approach, then maybe you just give up. But if this is the only problem, then everything else checks out, then it's still worth fighting. So yeah, you have to do some forward reconnaissance sometimes. (*Lex laughs*)

Lex Fridman : And that is sometimes productive. To assume like, okay, we'll figure it out eventually.

Terence Tao : Oh, yeah. Sometimes actually it's even productive to make mistakes. So one of the, I mean there was a project which actually we won some prizes for, with four other people. We worked on this PDE problem again, actually this blowup regularity type problem. And it was considered very hard. Jean Bourgain, who was another Fields Medalist who worked on a special case of this, but he could not solve the general case. And we worked on this problem for two months and we thought we solved it. We had this cute argument that if everything fit and we were excited, we were planning celebration to all get together and have champagne or something, and we started writing it up, and one of us, not me actually, but another co-author said, "Oh, in this lemma here, we have to estimate these 13 terms that show up in this expansion. And we estimated 12 of them. But in our notes I can't find the estimation of the 13th. Can someone apply that ?" And I said, "Sure, I'll look at this." And actually yeah, we didn't cover that. We completely omitted this term. And this term turned out to be worse than the other 12 terms put together. In fact, we could not estimate this term. And we tried for a few more months and all different permutations and there was always this one thing, one term that we could not control. And so this was very frustrating. But because we had already invested months and months of effort in this already, we stuck at this. We tried increasingly desperate things and crazy things. And after two years, we found that approach was actually somewhat different quite a bit from our initial strategy which didn't generate these problematic terms and actually solved the problem. So we solved the problem after two years. But if we hadn't had that initial first dawn of nearly solving the problem, we would have given up by month two or something and worked on an easier problem. If we had known it would take two years, not sure we would have started the project. Sometimes actually having the incorrect, it's like Columbus traveling in the New World. It's a incorrect version of measurement of the size of the Earth. He thought he was going to find a new trade route to India, or at least that was how he sold it in his prospectus. I mean, it could be that he actually secretly knew.

Lex Fridman : Just on the psychological element, do you have emotional or self-doubt that just overwhelms you most like that ? Because this stuff, it feels like math is so engrossing that it can break you. When you invest so much yourself in the problem and then it turns out wrong, you could start to, similar way chess has broken some people.

Terence Tao : Yeah, I think different mathematicians have different levels of emotional investment in what they do. I mean, I think for some people it's just a job. You have a problem and if it doesn't work out, you go on the next one. Yeah, so the fact that you can always move on to another problem, it reduces the emotional connection. I mean there are cases. So there are certain problems that are what are called mathematical diseases where just latch onto that one problem and they spend years and years thinking about nothing but that one problem and maybe their career suffers and so forth. They say, "Okay, but this big win, once I finish this problem, that will make up for all the years of lost opportunity." Occasionally it works, but I really don't recommend it for people without the right fortitude. Yeah, so I've never been super invested in any one problem. One thing that helps is that we don't need to call our problems in advance. Well, when we do grant proposals, we say we will study this set of problems, but even though we don't promise definitely by five years I will supply a proof of all these things. You promise to make some progress or discover some interesting phenomena and maybe you don't solve the problem, but you find some related problem that you can say something new about, and that's a much more feasible task.

Lex Fridman : But I'm sure for you there's problems like this. You have made so much progress towards the hardest problems in the history of mathematics. So is there a problem that just haunts you, it sits there in the dark corners ? Twin prime conjecture, Riemann hypothesis, Goldbach conjecture ?

Terence Tao : Twin prime, that sounds, (Lex laughing) again, so, I mean, the problem is like the Riemann hypothesis, those are so far out of reach.

Lex Fridman : You think so ?

Terence Tao : Yeah. There's no even viable strategy. Like, even if I activate all the cheats that I know of in this problem, like, there's just still no way to get from A to B. I think it needs a breakthrough in another area of mathematics to happen first and for someone to recognize that it would be a useful thing to transport into this problem.

Lex Fridman : So we should maybe step back for a little bit and just talk about prime numbers. So they're often referred to as the atoms of mathematics. Can you just speak to the structure that these atoms provide ?

Terence Tao : So the natural numbers have two basic operations attached to them, addition and multiplication. So if you want to generate the natural numbers, you can do one of two things. You can just start with one and add one to itself over and over again, and that generates you the natural numbers. So additively, they're very easy to generate. 1, 2, 3, 4, 5. Or you can take the prime number if you want to generate multiplicatively, you can take all the prime numbers, 2, 3, 5, 7, and multiply them all together. And together, that gives you all the natural numbers, except maybe for one. So there are these two separate ways of thinking about the natural numbers from an additive point of view and a multiplicative point of view. And separately, they're not so bad. So any question that only involves addition is relatively easy to solve. And any question that only involves multiplication is relatively easy to solve. But what has been frustrating is that you combine the two together and suddenly you get this extremely rich. I mean, we know that there are statements in number theory that are actually as undecidable. There are certain polynomials in some number of variables. Is there a solution in the natural numbers ? And the answer depends on and undecidable statement like whether the axioms of mathematics are consistent or not. But even the simplest problems that combine something multiplicative, such as the primes, with something additive, such as shifting by two, separately, we understand both of them well. But if you ask, when you shift the prime by two, how often can you get another prime ? It's been amazingly hard to relate the two.

Lex Fridman : And we should say that the twin-prime conjecture is just that. It posits that there are infinitely many pairs of prime numbers that differ by two. Now, the interesting thing is that you have been very successful at pushing forward the field in answering these complicated questions of this variety. Like you mentioned, the Green-Tao theorem, it proves that prime numbers contain arithmetic progressions of any length.

Terence Tao : Right.

Lex Fridman : Which is mind-blowing that you could prove something like that.

Terence Tao : Right, yeah. So what we've realized because of this type of research is that different patterns have different levels of indestructibility. So what makes the twin-prime problem hard is that if you take all the primes in the world, 3, 5, 7, 11, so forth, there are some twins in there. 11 and 13 is a twin prime, pair of twin primes and so forth. But you could easily, if you wanted to, redact the primes to get rid of these twins. The twins, they show up and there are infinitely many of them, but they're actually reasonably sparse. Initially there's quite a few, but once you got to the millions, the trillions, they become rarer and rarer. And you could actually just. If someone was given access to the database of primes, you just edit out a few primes here and there. They could make the twin-prime conjecture false by just removing like 0.01of the primes or something. Just well chosen to do this. And so you could present a censored database of the primes which passes all of the statistical tests of the primes. It obeys things like the prime number theorem and other facts about the primes, but doesn't contain any twin primes anymore. And this is a real obstacle for the twin-prime conjecture. It means that any proof structure strategy to actually find twin prime in the actual primes must fail when applied to these slightly edited primes. And so it must be some very subtle, delicate feature of the primes that you can't just get from aggregate statistical analysis.

Lex Fridman : Okay, so that's out. (laughs)

Terence Tao : Yeah, on the other hand, arithmetic progressions has turned out to be much more robust. You can take the primes and you can eliminate 99the prime primes actually, and you can take any 90 participants you want. And it turns out, and another thing we proved is that you still get asthmatic progressions. Asthmatic progressions are much, they're like cockroaches.

Lex Fridman : Of arbitrary length.

Terence Tao : Yes, yes.

Lex Fridman : That's crazy. For people who don't know, arithmetic progressions is a sequence of numbers that differ by some fixed amount.

Terence Tao : Yeah, but it's again like, it's an infinite monkey type phenomenon. For any fixed length of your set, you don't get arbitrary length progressions, you only get quite short progressions.

Lex Fridman : But you're saying twin prime is not an infinite monkey phenomena. I mean, it's a very subtle monkey. It's still an infinite monkey phenomenon.

Terence Tao : Right, yeah, if the primes were really genuinely random, if the primes were generated by monkeys, then yes, in fact, the infinite monkey theorem would.

Lex Fridman : Oh, but you're saying that twin prime is, it doesn't. You can't use the same tools. It doesn't appear random almost.

Terence Tao : Well, we don't know. Yeah, we believe the primes behave like a random set. So the reason why we care about the twin-prime conjecture is it's a test case for whether we can genuinely, confidently, say with 0 % chance of error, that the primes behave like a random set. Random versions of the primes we know contain twins, at least with 100or probably tending to 100as you go out further and further. Yeah. So the primes, we believe that they're random. The reason why arithmetic progressions are indestructible is that regardless of whether it looks random or looks structured like periodic, in both cases, arithmetic regressions appear, but for different reasons. And this is basically all the ways in which the theorem, there are many proofs of these sort of arithmetic progression theorems, and they're all proven by some sort of dichotomy where your set is either structured or random, and in both cases, you can say that something, and then you put the two together. But in twin-primes, if the primes are random, then you're happy, you win. But if the primes are structured, they could be structured in a specific way that eliminates the twins. And we can't rule out that one conspiracy.

Lex Fridman : And yet you were able to make, as I understand, progress on the k-tuple version.

Terence Tao : Right, yeah. So the one funny thing about conspiracies is that any one conspiracy theory is really hard to disprove. That if you believe the world is run by lizards, you say, "Here's some evidence that it's not run by lizards." Well, but that evidence was planted by the lizards. You may have encountered this kind of phenomenon. There's almost no way to definitively rule out a conspiracy. And the same is true in mathematics, that a conspiracy is solely devoted to eliminating twin primes. You would have to also infiltrate other areas of mathematics. But it could be made consistent, at least as far as we know. But there's a weird phenomenon that you can make one conspiracy rule out other conspiracies. So if the world is run by lizards, it can't also be run by aliens.

Lex Fridman : (chuckles) Right.

Terence Tao : So one unreasonable thing is hard to disprove. But more than one, there are tools. So, for example, we know there's infinitely many primes that are, no two of which are, so there are infinitely pairs of primes which differ by at most 246 actually is the current.

Lex Fridman : So there's like a bound.

Terence Tao : Yes, so there's twin primes. There's things called cousin primes that differ by four. There's called sexy primes that differ by six.

Lex Fridman : What are sexy primes ?

Terence Tao : Primes that differ by six. The name is much less, the concept is much less exciting than the name suggests.

Lex Fridman : Got it.

Terence Tao : So you can make a conspiracy rule out one of them these, but once you have 50 of them, it turns out that you can't rule out all of them at once. It requires too much energy somehow in this conspiracy space.

Lex Fridman : How do you do the bound part ? How do you develop a bound for the difference between the primes that there's an infinite number of ?

Terence Tao : So it's ultimately based on what's called the pigeonhole principle. So the pigeonhole principle is the statement that if you have a number of pigeons and they all have to go into pigeonholes and you have more pigeons than pigeonholes, then one of the pigeonholes has to have at least two pigeons there. So there has to be two pigeons that are close together. So, for instance, if you have 100 numbers and they all range from one to a thousand, two of them have to be at most 10 apart, because you can divide up the numbers from 1 to 100 into 100 pigeon holes. Let's say if you have 101 numbers, 101 numbers, then two of them have to be distance less than 10 part because two of them have to belong to the same pigeonhole. So it's a basic feature of a basic principle in mathematics. So it doesn't quite work with the primes directly because the primes get sparser and sparser as you go out, that fewer and fewer numbers are prime. But it turns out that there's a way to assign weights to numbers. So there are numbers that are kind of almost prime, but they don't have no factors at all other than themselves in one, but they have very few factors. And it turns out that we understand almost primes a lot better than understand primes. And so, for example, it was known for a long time that there were twin almost-primes. This has been worked out. So almost primes are something we kind of understand. So you can actually restrict attention to a suitable set of almost primes. And whereas the primes are very sparse overall relative to the almost primes, they actually are much less sparse. You can set up a set of almost primes where the primes have density like say one percent. And that gives you a shot at proving by applying some sort of original principle that there's pairs of primes that are just only 100 about. But in order to prove the twin-prime conjecture, you need to get the density of primes inside the almost primes up to a threshold of 50Once you get up to 50you will get twin primes. But unfortunately there are barriers. We know that no matter what kind of good set of almost primes you pick, the density primes can never get above 50It's called the parity barrier. And I would love to find. Yeah, so one of my long-term dreams is to find a way to breach that barrier, because it would open up not only to twin-prime conjecture, but the Goldbach conjecture and many other problems in number theory are currently blocked because our current techniques would require going beyond this theoretical parity barrier. It's like going faster than the speed of light.

LEX FRIDMAN : Yeah. So we should say a twin-prime conjecture, one of the biggest problems in the history of mathematics. Goldbach conjecture also. They feel like next door neighbors. Has there been days when you felt you saw the path ?

TERENCE TAO : Oh yeah. Sometimes you try something and it works super well. You again, the sense of mathematical smell we talked about earlier. You learn from experience when things are going too well because there are certain difficulties that you sort of have to encounter. I think the way a colleague of mine put it is that if you are on the streets of New York and you put on a blindfold, and you're put in a car, and after some hours the blindfold is off and you're in Beijing. That was too easy. Somehow there was no ocean being crossed. Even if you don't know exactly what was done, you're suspecting that something wasn't right.

LEX FRIDMAN : But is that still in the back of your head to do you return to these. Do you return to the prime numbers every once in a while to see ?

TERENCE TAO : Yeah, when I have nothing better to do, which is less and less now. I get busy with so many things these days. But yeah, when I have free time, and I'm too frustrated to work on my sort of real research projects, and I also don't want to do my administrative stuff, or I don't want to do some errand for my family. I can play these things for fun. And usually you get nowhere. Yeah, you have to learn to just say, "Okay, fine, once again, nothing happened. I will move on." Very occasionally, one of these problems I actually solved, or sometimes as you say, you think you solved it, and then you forward for maybe 15 minutes, and then you think I should check this, because this is too easy, too good to be true. And it usually is.

LEX FRIDMAN : What's your gut say about when these problems would be solved, twin-prime and Goldbach ?

TERENCE TAO : Prime I think we will keep getting more partial results. It does need at least one. This parity barrier is the biggest remaining obstacle. There are simpler versions of the conjecture where we are getting really close. So I think we will, in 10 years we will have many more much closer results. We may not have the whole thing, so twin-primes is somewhat close. Riemann hypothesis, I have no clue. I mean, it has happened by accident, I think.

LEX FRIDMAN : So the Riemann hypothesis is a kind of more general conjecture about the distribution of prime numbers.

TERENCE TAO : Right, yeah. It states that are sort of viewed multiplicatively for questions only involving multiplication, no addition, the primes really do behave as randomly as you could hope. So there's a phenomenon in probability called square root cancellation that if you want to poll, say America upon some issue and you ask one of the two voters, you may have sampled a bad sample and then you get a really imprecise measurement of the full average. But if you sample more and more people, the accuracy gets better and better, and it actually improves like the square root of the number of people you sampled. So yeah, if you sample 1,000 people, you can get like a 2 or 3So in the same sense, if you measure the primes in a certain multiplicative sense, there's a certain type of statistic you can measure. It's called the Riemann zeta function, and it fluctuates up and down. But in some sense, as you keep averaging more and more, if you sample more and more, the fluctuation should go down as if they were random. And there's a very precise way to quantify that. And the Riemann hypothesis is a very elegant way that captures this. But as with many other ways in mathematics, we have very few tools to show that something really genuinely behaves really random. And this is actually not just a little bit random, but it's asking that it behaves as random as an actually random set, this square root cancellation. And we know because of things related to the parity problem, actually that most of the usual techniques cannot hope to settle this question. The proof has to come out of left field. Yeah, but what that is, no one has any serious proposal. And there's various ways to sort of, as I said, you can modify the primes a little bit and you can destroy the Riemann hypothesis. So it has to be very delicate. You can't apply something that has huge margins of error. It has to be would just barely work. And there's all these pitfalls that you have to dodge very adeptly to it.

Lex Fridman : The prime numbers are just fascinating.
Terence Tao : Yeah.

Lex Fridman : What to you is most mysterious about the prime numbers ?

Terence Tao : That's a good question. So conjecturally we have a good model of them. As I said, they have certain patterns like the primes are usually odd for instance. But apart from these obvious patterns, they behave very randomly. And just assuming that they behave. So there's something called the Cramer random model of the primes, that after a certain point primes just behave like a random set. And there's various slight modifications to this model, but this has been a very good model. It matches the numerics. It tells us what to predict. Like I can tell you with complete certainty the twin-prime conjecture is true. The random model gives overwhelming odds that it's true. I just can't prove it. Most of our mathematics is optimized for solving things with patterns in them. And the primes have this anti-pattern, as do almost everything really. But we can't prove that. I guess it's not mysterious that the primes, it's kind of random, because there's no reason for them to have any kind of secret pattern. But what is mysterious is what is the mechanism that really forces the randomness to happen. And this is just absent.

Lex Fridman : Another incredibly surprisingly difficult problem is the Collatz conjecture.

Terence Tao : Oh, yes.

Lex Fridman : Simple to state, beautiful to visualize in its simplicity, and yet extremely difficult to solve. And yet you have been able to make progress. Paul Erdos said about the Collatz conjecture that mathematics may not be ready for such problems. Others have stated that it is an extraordinarily difficult problem, completely out of reach. This is in 2010, out of reach of present-day mathematics. And yet you have made some progress. Why is it so difficult to make ? Can you actually even explain what it is if it's easy to ?

Terence Tao : Yeah, so it's a problem that you can explain. It helps with some visual aids, but yeah. So you take any natural number, like say 13, and you apply the following procedure to it. So if it's even you divide it by two, and if it's odd, you multiply it by three and add one. So even numbers get smaller, odd numbers get bigger. So 13 would become 40, because 13 times three is 39. Add one, you get 40. So it's a simple process for odd numbers and even numbers, they're both very easy operations. And then you put them together, it's still reasonably simple. But then you ask what happens when you iterate it ? You take the output that you just got and feed it back in. So 13 becomes 40. 40 is now even, divided by 2, 20. 20 still even, divided by 2, 10, 5. And then 5 times 3 plus 1 is 16, and then 8, 4, 2, 1. And then from 1 it goes 1, 4, 2, 1, 4, 2. It cycles forever. So this sequence I just described, 13, 40, 20, 10. So these are also called hailstone sequences, because there's an oversimplified model of hailstone formation which is not actually quite correct, but it's still somehow taught to high school students, as the first approximation, is that a little nugget of ice gets, an ice crystal forms in a cloud, and it goes up and down because of the wind. And sometimes when it's cold, it acquires a bit more mass, and maybe it melts a little bit. And this process of going up and down creates this partially melted ice, which eventually creates this hailstone, and eventually it falls out the earth. So the conjecture is that no matter how high you start up, you take a number which is in the millions or billions, this process that goes up if you're odd, and down if you're even, it eventually goes down to earth all the time.

Lex Fridman : No matter where you start, with this very simple algorithm, you end up at one.
Terence Tao : Right.

Lex Fridman : And you might climb for a while.
Terence Tao : Right, yeah.
Lex Fridman : Up and down.

Lex Fridman : Yeah, if you plot it, these sequences, they look like Brownian motion, they look like the stock market. They just go up and down in a seemingly random pattern. And in fact, usually that's what happens, that if you plug in a random number, you can actually prove, at least initially, that it would look like random walk. And that's actually a random walk with a downward drift. It's like if you're always gambling on a roulette at the casino with odds slightly weighted against you. So sometimes you win, sometimes you lose, but over in the long run, you lose a bit more than you win. And so normally your wallet will go to zero if you just keep playing over and over again.

Lex Fridman : So statistically it makes sense.

Terence Tao : Yes. So the result that I proved, roughly speaking, asserts that statistically, like 99 % of all inputs would drift down to maybe not all the way to one, but to be much, much smaller than what you started. So it's like if I told you that if you go to a casino, most of the time you end up, if you keep playing for long enough, you end up with a smaller amount in your wallet than when you started. That's kind of like the result that I proved.

Lex Fridman : So why is that result like can you continue down that thread to prove the full conjecture ?

Terence Tao : Well, the problem is that I used arguments from probability theory, and there's always this exceptional event. So in probability we have this law of large numbers which tells you things like if you play a casino with a game, at a casino with a losing expectation, over time, you are guaranteed almost surely with probability as close to 100you're guaranteed to lose money. But there's always this exceptional outlier. It is mathematically possible that even when the game is, the odds are not in your favor, you could just keep winning slightly more often than you lose. Very much like how in Navier-Stokes there could be most of the time your waves can disperse. There could be just one outlier choice of initial conditions that would lead you to blow up. And there could be one outlier choice of special number that you stick in that shoots off to infinity or other numbers crash to earth, crash to one. In fact, there's some mathematicians, Alex Kontorovich for instance, who've proposed that actually these Collatz iterations are like these cellular automata. Actually if you look at what they happen in binary, they do actually look a little bit like these Game of Life type patterns. And in an analogy to how the Game of Life can create these massive self-replicating objects and so forth, possibly you could create some sort of heavier than air flying machine, a number which is actually encoding this machine, whose job it is to encode is to create a version of itself which is larger.

Terence Tao : Heavier than air machine encoded in a number that flies forever.

Lex Fridman : Yeah, so Conway in fact worked on this problem as well.

Lex Fridman : Oh, wow.

Terence Tao : Conway, similar, in fact, that was more inspirations for the Navier-Stokes project that Conway studied generalizations of the Collatz problem, where instead of multiplying by three and adding one or dividing by two, you have more complicated branching rules. But instead of having two cases, maybe you have 17 cases and you go up and down. And he showed that once your iteration gets complicated enough, you can actually encode Turing machines, and you can actually make these problems undecidable and do things like this. In fact, he invented a programming language for these kind of fractional linear transformations. He called it Fractran as a play on Fortran, and he showed that you can program, it was Turing complete. You could make a program that if the number you inserted in was encoded as a prime it would sync to zero, it would go down, otherwise it would go up, and things like that. So the general class of problems is really as complicated as all the mathematics.

Lex Fridman : Some of the mystery of the cellular automata that we talked about having a mathematical framework to say anything about cellular automata, maybe the same kind of framework is required for the Collatz conjecture.

Terence Tao : Yeah, if you want to do it, not statistically, but you really want 100all inputs for the earth. So what might be feasible is statistically 99But like everything, that looks hard.

Lex Fridman : What would you say is out of these within reach famous problems is the hardest problem we have today ? Is it the Riemann Hypothesis ?

Terence Tao : Riemann is up there. P = NP is a good one because that's a meta problem. If you solve that in the positive sense that you can find a P = NP algorithm, then potentially this solves a lot of other problems as well.

Lex Fridman : And we should mention some of the conjectures we've been talking about. A lot of stuff is built on top of them now. There's ripple effects. P = NP has more ripple effects than basically any other.

Terence Tao : Right, if the Riemann hypothesis is disproven, that'd be a big mental shock to the number theorists, but it would have follow-on effects for cryptography. Because a lot of cryptography uses number theory, uses number theory constructions involving primes and so forth. And it relies very much on the intuition that number theory has built over many, many years of what operations involving primes behave randomly and what ones don't. And in particular our encryption methods are designed to turn text written information on it into text which is indistinguishable from random noise. And hence we believe to be almost impossible to crack, at least mathematically. But if something as core to our beliefs as the Riemann hypothesis is wrong, it means that there are actual patterns of the primes that we're not aware of. And if there's one, there's probably going to be more. And suddenly a lot of our crypto systems are in doubt.

Lex Fridman : Yeah, but then how do you then say stuff about the primes ?
Terence Tao : Yeah.

Lex Fridman : Like you're going towards the Collatz conjecture again, because you want it to be random, right ? You want it to be random.

Terence Tao : Yeah, so more broadly, I'm just looking for more tools, more ways to show that things are random. How do you prove a conspiracy doesn't happen ?

Lex Fridman : Is there any chance to you that P = NP ? Can you imagine a possible universe ?

Terence Tao : It is possible. I mean, there's various scenarios. I mean, there's one where it is technically possible, but in fact it's never actually implementable. The evidence is sort of slightly pushing in favor of no, that probably P is not equal to NP.

Lex Fridman : I mean, it seems like it's one of those cases similar to Riemann hypothesis. I think the evidence is leaning pretty heavily on the no.

Terence Tao : Certainly more on the no than on the yes. The funny thing about P = NP is that we have also a lot more obstructions than we do for almost any other problem. So while there's evidence, we also have a lot of results ruling out many, many types of approaches to the problem. This is the one thing that the computer science has actually been very good at. It's actually saying that certain approaches cannot work, no-go theorems. It could be undecidable. We don't know.

Lex Fridman : There's a funny story I read that when you won the Fields Medal, somebody from the Internet wrote you and asked, what are you going to do now that you've won this prestigious award ? And then you just quickly, very humbly said that this shiny metal is not going to solve any of the problems I'm currently working on. I'm going to keep working on them. It's just, first of all, it's funny to me that you would answer an email in that context. And second of all, it just shows your humility. But anyway, maybe you could speak to the Fields Medal. But it's another way for me to ask about Grigori Perelman. What do you think about him famously declining the Fields Medal and the Millennium Prize, which came with a one million dollar of prize money ? He stated that I'm not interested in money or fame. The prize is completely irrelevant for me. If the proof is correct, then no other recognition is needed.

Terence Tao : Yeah, no. He's somewhat of an outlier, even among mathematicians who tend to have somewhat idealistic views. I've never met him. I think I'd be interested to meet him one day, but I never had the chance. I know people who've met him, but he's always had strong views about certain things. I mean, it's not like he was completely isolated from the math community. I mean, he would give talks and write papers and so forth, but at some point he just decided not to engage with the rest of the community. He was disillusioned or something, I don't know. And he decided to peace out and collect mushrooms in St. Petersburg or something. And that's fine, you can do that. I mean that's another sort of flip side to, I mean, a lot of our problems that we solve, some of them do have practical application and that's great. But if you stop thinking about a problem, so he hasn't published since in this field, but that's fine. There's many, many other people who've done so as well. Yeah, so I guess one thing I didn't realize initially with the Fields Medal is that it sort of makes you part of the establishment. So most mathematicians, just career mathematicians, you just focus on publishing the next paper, maybe getting one promotion, one rank, and starting a few projects, maybe taking some students or something. But then suddenly people want your opinion on things and you have to think a little bit about things that you might just have foolishly say because you know no one's going to listen to you. It's more important now.

Lex Fridman : Is it constraining to you ? Are you able to still have fun and be a rebel and try crazy stuff and play with ideas ?

Terence Tao : I have a lot less free time than I had previously. I mean mostly by choice. I mean obviously I have the option to sort of decline. So I decline a lot of things. I could decline even more, or I could acquire a reputation being so unreliable that people don't even ask anymore.

Lex Fridman : I love the different algorithms here. This is great.

Terence Tao : It's always an option. But you know, there are things that are like, I mean I don't spend as much time as I do as a postdoc just working on one problem at a time or fooling around. I still do that a little bit. But yeah, as you advance in your career, some of the more soft skills, so math somehow front loads all the technical skills to the early stages of a career. So as a post office publish or perish, your incentive to advised to basically focus on proving very technical theorems, to sort of prove yourself as well as prove the theorems. But then as you get more senior, you have to start mentoring and giving interviews and trying to shape direction of the field, both research-wise and sometimes you have to do various administrative things, and it's kind of the right social contract because you need to work in the trenches to see what can help mathematicians.

Lex Fridman : The other side of the establishment sort of the really positive thing is that you get to be a light that's an inspiration to a lot of young mathematicians or young people that are just interested in mathematics. It's like, it's just how the human mind works. This is where I would probably say that I like the Fields Medal, that it does inspire a lot of young people somehow. This is just how human brains work.
Terence Tao : Yeah.

Lex Fridman : At the same time, I also want to give sort of respect to somebody like Grigori Perelman, who is critical of awards in his mind. Those are his principles. And any human that's able for their principles to do the thing that most humans would not be able to do. It's beautiful to see.

Terence Tao : Some recognition is necessary and important. But yeah, it's also important to not let these things take over your life and only be concerned about getting the next big award or whatever. I mean, yeah, so again, you see, these people try to only solve really big math problems and not work on things that are less sexy, if you wish, but actually still interesting and instructive. As you say, the way the human mind works, we understand things better when they're attached to humans, and also if they're attached to a small number of humans. The way our human mind is wired, we can comprehend and the relationships between 10 or 20 people. But once you get beyond 100 people, there's a limit. I think there's a name for it, beyond which it just becomes the other. And so you have to simplify the whole mass of 99.9humanity becomes the other. And often these models are incorrect and this causes all kinds of problems. So to humanize a subject, like if you identify a small number of people and say, "These are representative people of a subject," role models, for example, that has some role, but it can also be, too much of it can be harmful. Because I'll be the first to say that my own career path is not that of a typical mathematician. I had a very accelerated education. I skipped a lot of classes. I think I had very fortunate mentoring opportunities, and I think I was at the right place at the right time. Just because someone doesn't have my trajectory doesn't mean that they can't be good mathematicians. I mean, they may be good mathematicians in a very different style. And we need people with a different style. And sometimes too much focus is given on the person who does the last step to complete a project in mathematics or elsewhere that's really taken centuries or decades with lots and lots of building and lots of previous work. But that's a story that's difficult to tell if you're not an expert, because it's easier to just say, "One person did this one thing." It makes for a much simpler history.

Lex Fridman : I think on the whole it is a hugely positive thing to talk about Steve Jobs as a representative of Apple when I personally know, and of course everybody knows, the incredible design, the incredible engineering teams, just the individual humans on those teams. They're not a team. They're individual humans on a team. And there's a lot of brilliance there, but it's just a nice shorthand, like pi.
Terence Tao : Yeah.

Lex Fridman : Steve Jobs, pi.

Terence Tao : Yeah, yeah. As a starting point, as a first approximation.

Lex Fridman : And then read some biographies and then look into much deeper first approximation.
Terence Tao : Yeah.

Lex Fridman : That's right. So you mentioned you were at Princeton to Andrew Wiles at that time.
Terence Tao : Oh, yeah.

Terence Tao : He was a professor there.

Lex Fridman : It's a funny moment how history is just all interconnected. And at that time he announced that he proved the Fermat's Last Theorem. What did you think, maybe looking back now with more context about that moment in math history ?

Terence Tao : Yeah, so I was a graduate student at the time. I vaguely remember there was press attention, and we all had the same. We had pigeonholes in the same mail room. So we all pitched our mail, and suddenly Andrew Wiles' mailbox exploded to be overflowing.

Lex Fridman : (laughs) That's a good metric.

Terence Tao : Yeah. So we all talked about it at tea and so forth. I mean, we didn't understand, most of us sort of didn't understand the proof. We understand sort of high-level details. In fact, there's an ongoing project to formalize it in Lean. Kevin Buzzard is actually.

Lex Fridman : Yeah. Can we take that small tangent ? How difficult does that, because as I understand, the proof for Fermat's Last Theorem has like super complicated objects.

Terence Tao : Yeah.

Lex Fridman : Really difficult to formalize, no ?

Terence Tao : Yeah, I guess, yeah, you're right. The objects that they use, you can define them. So they've been defined in Lean. So just defining what they are can be done. That's really not trivial, but it's been done. But there's a lot of really basic facts about these objects that have taken decades to prove and that they're in all these different math papers. And so lots of these have to be formalized as well. Kevin Buzzard's goal, actually, he has a five-year grant to formalize Fermat's Last Theorem. And his aim is that he doesn't think he will be able to get all the way down to the basic axioms. But he wants to formalize it to the point where the only things that he needs to rely on as black boxes are things that were known by 1980 to number theorists at the time. And then some other person, some other work would have to be done, to get from there. So it's a different area of mathematics than the type of mathematics I'm used to. In analysis, which is kind of my area, the objects we study are kind of much closer to the ground. I study things like prime numbers and functions and things that are within scope of a high school math education to at least define. Yeah, but then there's this very advanced algebraic side of number theory where people have been building structures upon structures for quite a while. And it's a very sturdy structure at the base at least, it's extremely well developed in the textbooks and so forth. But it does get to the point where if you haven't taken these years of study and you want to ask about what is going on at level six of this tower, you have to spend quite a bit of time before they can even get to the point where you see something you recognize.

Lex Fridman : What inspires you about his journey, that was similar as we talked about seven years, mostly working in secret ?

Terence Tao : Yeah. That is a romantic, so it kind of fits with sort of the romantic image I think people have of mathematicians to the extent that they think of them at all, as these kind of eccentric wizards or something. So that certainly kind of accentuated that perspective. I mean, it is a great achievement. His style of solving problems is so different from my own, which is great. I mean, we need people like that.

Lex Fridman : Can you speak to it, in terms of you like the collaborative ?

Terence Tao : I like moving on from a problem if it's giving too much difficulty. But you need the people who have the tenacity and the fearlessness. I've collaborated with people like that where I want to give up, because the first approach that we tried didn't work, and the second one didn't work, but they're convinced and they have third, fourth, and the fifth approach works. And I have to eat my words. Okay, I didn't think this was going to work, but yes, you were right.

Lex Fridman : And we should say for people who don't know, not only are you known for the brilliance of your work, but the incredible productivity, just the number of papers which are all of very high quality. So there's something to be said about being able to jump from topic to topic.

Terence Tao : Yeah, it works for me. Yeah. I mean, there are also people who are very productive and they focus very deeply on, yeah. I think everyone has to find their own workflow. One thing which is a shame in mathematics is that we have, mathematics, there's sort of a one size fits all approach to teaching mathematics. So we have a certain curriculum and so forth. I mean, maybe if you do math competitions or something, you get a slightly different experience. But I think many people, they don't find their native math language until very late or usually too late. So they stop doing mathematics, and they have a bad experience with a teacher who's trying to teach them one way to do mathematics. They don't like it. My theory is that humans don't come, evolution has not given us a math center of a brain directly. We have a vision center and a language center and some other centers which evolution has honed, but we don't have innate sense of mathematics. But our other centers are sophisticated enough that different people, we can repurpose our other areas of our brain to do mathematics. So some people have figured out how to use the visual center to do mathematics, and so they think very visually when they do mathematics. Some people have repurposed their language center and they think very symbolically. Some people, if they are very competitive and they like gaming, there's a part of your brain that's very good at solving puzzles and games, and that can be repurposed. But when I talk to other mathematicians, they don't quite think, I can tell that they're using somehow different styles of thinking than I am. I mean, not disjoint, but they may prefer visual. I don't actually prefer visual so much. I need lots of visual aids myself. Mathematics provides a common language, so we can still talk to each other even if we are thinking in different ways.

Lex Fridman : But you can tell there's a difference set of subsystems being used in the thinking process.

Terence Tao : Yes, they take different paths, they're very quick at things that I struggle with and vice versa, and yet they still get to the same goal.

Lex Fridman : That's beautiful.

Terence Tao : But I mean, the way we educate, unless you have a personalized tutor or something, I mean, education sort of just by nature of scale has to be mass produced. You have to teach the 30 kids. If they have 30 different styles, you can't teach 30 different ways.

Lex Fridman : On that topic, what advice would you give to students, young students who are struggling with math but are interested in it and would like to get better ? Is there something in this complicated educational context, what would you ?

Terence Tao : Yeah, it's a tricky problem. One nice thing is that there are now lots of sources for mathematical enrichment outside the classroom. So in my day already there are math competitions and there are also popular math books in the library. But now you have YouTube. There are forums just devoted to solving math puzzles. And math shows up in other places. For example, there are hobbyists who play poker for fun, and they, for very specific reasons, are interested in very specific probability questions. And there's a community of amateur probabilists in poker, in chess, in baseball. I mean, there's math all over the place. And I'm hoping actually with these new tools for Lean and so forth, that actually we can incorporate the broader public into math research projects. This almost doesn't happen at all currently. So in the sciences there is some scope for citizen science, like astronomers looking at the amateurs who would discover comets, and there's biologists that people who could identify butterflies and so forth. There are a small number of activities where amateur mathematicians can discover new primes and so forth. But previously, because we had to verify every single contribution, most mathematical research projects, it would not help to have input from the general public. In fact, it would just be time-consuming, because just error checking and everything. But one thing about these formalization projects is that they are bringing in more people. So I'm sure there are high school students who've already contributed to some of these formalizing projects, who contributed to Mathlib. You don't need to be a PhD holder to just work on one atomic thing.

Lex Fridman : There's something about the formalization here that also, as a very first step, opens it up to the programming community too, the people who are already comfortable with programming. It seems like programming is somehow, maybe just the feeling, but it feels more accessible to folks than math. Math is seen as this, like extreme, especially modern mathematics seen as this extremely difficult to enter area. And programming is not. So that could be just an entry point.

Terence Tao : You can execute code and you can get results, you can print out "Hello world" pretty quickly. If programming was taught as an almost entirely theoretical subject, where you just taught the computer science, the theory of functions and routines and so forth, and outside of some very specialized homework assignments, you would not actually program on the weekend for fun. They would be considered as hard as math. So as I said, there are communities of non-mathematicians where they're deploying math for some very specific purpose, like optimizing their poker game, and for them, and then math becomes fun for them.

Lex Fridman : What advice would you give in general to young people how to pick a career, how to find themselves ?

Terence Tao : That's a tough, tough question. Yeah, so there's a lot of certainty now in the world. I mean, there was this period after the war where at least in the West, if you came from a good demographic, there was a very stable path to a good career. You go to college, you get an education, you pick one profession, and you stick to it. It's becoming much more a thing of the past. So I think you just have to be adaptable and flexible. I think people have to get skills that are transferable, like learning one specific programming language or one specific subject of mathematics or something, that itself is not a super transferable skill, but knowing how to reason with abstract concepts or how to problem solve when things go wrong, these are things which I think we will still need even as our tools get better, you'll be working with AI, sports, so forth.

Lex Fridman : But actually you're an interesting case study. I mean you're like a, one of the great living mathematicians, right ? And then you had a way of doing things, and then all of a sudden you start learning, first of all, you kept learning new fields, but you learn Lean. That's a non-trivial thing to learn.
Terence Tao : Yeah.

Lex Fridman : For a lot of people, that's an extremely uncomfortable leap to take, right ? A lot of mathematicians.

Terence Tao : First of all, I've always been interested in new ways to do mathematics. I feel like a lot of the ways we do things right now are inefficient. Me and my colleagues who spend a lot of time doing very routine computations or doing things that other mathematicians would instantly know how to do and we don't know how to do them, and why can't we search and get a quick response and so forth. So that's why I've always been interested in exploring new workflows. About four or five years ago, I was on a committee where we had to ask for ideas for interesting workshops to run at a math institute. And at the time Peter Scholze had just formalized one of his new theorems, and there were some other developments in computer-assisted proof that looked quite interesting. And I said, "Oh, we should run a workshop on this. This would be a good idea." And then I was a bit too enthusiastic about this idea. So I got voluntold to actually run it. So I did with a bunch of other people, Kevin Buzzard and Jordan Ellenberg and a bunch of other people, and it was a nice success. We brought together a bunch of mathematicians and computer scientists and other people, and we got up to speed on the state of the art, and it was really interesting developments that most mathematicians didn't know was going on that lots of nice proofs of concept just saw hints of what was going to happen. This was just before ChatGPT, but there was, even then there was one talk about language models and the potential capability of those in the future. So that got me excited about the subject. So I started giving talks about, this is something more of us should start looking at, now that I run this conference. And then ChatGPT came out, and suddenly AI was everywhere. And so I got interviewed a lot about this topic, and in particular the interaction between AI and formal proof assistants. And I said, "Yeah, they should be combined. This is perfect synergy to happen here." And at some point I realized that I have to actually not just talk the talk, but walk the walk. I don't work in machine learning, and I don't work in proof formalization, and there's a limit to how much I can just rely on authority and saying, "I'm a well-known mathematician. Just trust me when I say that this is going to change mathematics," and I'm not doing it any, when I don't do any of it myself. So I felt like I had to actually justify it. A lot of what I get into actually I don't quite see in advance just how much time I'm going to spend on it. And it's only after I'm sort of waist deep in a project that I realized by that point I'm committed.

Lex Fridman : Well, that's deeply admirable that you're willing to go into the fray, be in some small way a beginner. Or have some of the sort of challenges that a beginner would. So new concepts, new ways of thinking, also you know, sucking at a thing that others. I think in that talk, you could be a Fields Medal winning mathematician and an undergrad knows something better than you.

Terence Tao : Yeah. I think mathematics inherently, I mean mathematics is so huge these days that nobody knows all of modern mathematics, and inevitably we make mistakes, and you can't cover up your mistakes with just sort of bravado, because people will ask for your proofs and if you don't have the proofs, you don't have the proofs.
Lex Fridman : I love math.

Terence Tao : Yeah. (Lex laughs) So it does keep us honest. It's not a perfect panacea, but I think we do have more of a culture of admitting error because we're forced to all the time.

Lex Fridman : Big ridiculous question. I'm sorry for it once again. Who is the greatest mathematician of all time ? (Terence laughing) Maybe one who's no longer with us. Who are the candidates ? Euler, Gauss, Newton, Ramanujan, Hilbert.

Terence Tao : So first of all, as mentioned before, there's some time dependence.

Lex Fridman : On the day.

Terence Tao : Yeah. If you cumulatively over time, for example, Euclid, sort of is one of the leading contenders. And then maybe some unnamed anonymous mathematicians before that, whoever came up with the concept of numbers.

Lex Fridman : Do mathematicians today still feel the impact of Hilbert ?

Terence Tao : Oh yeah.

LEX FRIDMAN : Directly of everything that's happened in the 20th century ?

TERENCE TAO : Yeah, Hilbert spaces. We have lots of things that are named after him, of course. Just the arrangement of mathematics and just the introduction of certain concepts. I mean, 23 problems have been extremely influential.

LEX FRIDMAN : There's some strange power to the declaring which problems are hard to solve. The statement of the open problems.

TERENCE TAO : Yeah, I mean this is bystander effect everywhere. If no one says, "You should do X," everyone just will move around waiting for somebody else to do something and nothing gets done. One thing that actually you have to teach undergraduates in mathematics is that you should always try something. So you see a lot of paralysis in an undergraduate trying a math problem. If they recognize that there's a certain technique that can be applied, they will try it. But there are problems for which they see none of their standard techniques obviously applies. And the common reaction is then just paralysis. I don't know what to do. I think there's a quote from the Simpsons. "I've tried nothing, and I'm all out of ideas." (Lex laughing) So the next step then is to try anything, no matter how stupid. And in fact, almost the stupider the better. A technique which is almost guaranteed to fail. But the way it fails is going to be instructive. It fails because you're not at all taking into account this hypothesis. Oh, this hypothesis must be useful. That's a clue.

LEX FRIDMAN : I think you also suggested somewhere this fascinating approach which really stuck with me as they're using it and it really works. I think you said it's called structured procrastination.
TERENCE TAO : No, yes.

LEX FRIDMAN : It's when you really don't want to do a thing, then you imagine a thing you don't want to do more.

TERENCE TAO : Yes.

LEX FRIDMAN : That's worse than that. And then in that way you procrastinate by not doing the thing that's worse.
TERENCE TAO : Yeah, yeah.

LEX FRIDMAN : It's a nice hack. It actually works. (chuckles)

TERENCE TAO : Yeah, yeah, it is. I mean, with anything, I mean, like psychology is really important. Like you talk to athletes like marathon runners and so forth, and they talk about what's the most important thing, is it their training regimen or their diet and so forth ? So much of it is actually psychology, just tricking yourself to think that the form is feasible so that you're motivated to do it.

LEX FRIDMAN : Is there something our human mind will never be able to comprehend ?

Terence Tao : Well, as a mathematician, I mean, by induction, there must be some sufficient large number that you can't understand. (Lex laughing) That was the first thing that came to mind.

Lex Fridman : So that, but even broadly, is there, is there something about our mind that we're going to be limited even with the help of mathematics ?

Terence Tao : Well, okay, I mean, how much augmentation are you willing, for example, if I didn't even have pen and paper, if I had no technology whatsoever, so I'm not allowed blackboard, pen and paper.

Lex Fridman : You're already much more limited than you would be.

Terence Tao : Incredibly limited. Even language, the English language is a technology. It's one that's been very internalized.

Lex Fridman : So you're right, the formulation of the problem is incorrect because there really is no longer just a solo human. We're already augmented in extremely complicated, intricate ways, right.

Terence Tao : Yeah.

Lex Fridman : So we're already like a collective intelligence.

Terence Tao : Yes, yeah, yes. So humanity, plural, has much more intelligence in principle on his good days, than the individual humans put together. It can all have less. Okay, but yeah, so yeah, the mathematical community, plural, is an incredibly super intelligent entity that no single human mathematician can come close to replicating. You see it a little bit on these question analysis sites. So there's Math Overflow, which is the math version of Stack Overflow. And sometimes you get these very quick responses to very difficult questions from the community and it's a pleasure to watch, actually as an expert.

Lex Fridman : I'm a fan spectator of that site. Just seeing the brilliance of the different people, the depth of knowledge some people have, and the willingness to engage in the rigor and the nuance of the particular question. It's pretty cool to watch. It's almost like just fun to watch. What gives you hope about this whole thing we have going on, human civilization ?

Terence Tao : I think the younger generation is always really creative and enthusiastic and inventive. It's a pleasure working with young students. The progress of science tells us that the problems that used to be really difficult can become trivial to solve. I mean it was like navigation. Just knowing where you were on the planet was this horrendous problem. People died or lost fortunes because they couldn't navigate. And we have devices in our pockets that do this automatically for us. It's a completely solved problem. So things that seem unfeasible for us now could be maybe just homework exercise sort of thing. (laughs)

Lex Fridman : Yeah, one of the things I find really sad about the finiteness of life is that I won't get to see all the cool things we're to going create as a civilization, you know ? That because in the next 100 years, 200 years, just imagine showing up in 200 years.

Terence Tao : Yeah, well, already plenty has happened. You know, like if you could go back in time and talk to your teenage self or something. You know what I mean ? Just the Internet and now AI, I mean, again, they're beginning to be internalized and say, "Yeah, of course an AI can understand our voice and give reasonable, slightly incorrect answers to any question." But yeah, this was mind-blowing even two years ago.

Lex Fridman : And in the moment, it's hilarious to watch on the Internet and so on, the drama, people take everything for granted very quickly. And then we humans seem to entertain ourselves with drama. Out of anything that's created, somebody needs to take one opinion, another person needs to take an opposite opinion, argue with each other about it. But when you look at the arc of things, I mean, just even in progress of robotics. Just to take a step back and be like, wow, this is beautiful the way humans are able to create this.

Terence Tao : Yeah, when the infrastructure and the culture is healthy, the community of humans can be so much more intelligent and mature and rational than the individuals within it.

Lex Fridman : Well, one place I can always count on rationality is the comments section of your blog, which I'm a big fan of. There's a lot of really smart people there. And thank you, of course, for putting those ideas out on the blog. And I can't tell you how honored I am that you would spend your time with me today. I was looking forward to this for a long time. Terry, I'm a huge fan. You inspire me. You inspire millions of people. Thank you so much for talking.

Terence Tao : Thank you, it was a pleasure.

Lex Fridman : Thanks for listening to this conversation with Terence Tao.