

L'informaticien qui ne peut pas s'arrêter de raconter des histoires

DONALD E. KNUTH

Pour le pionnier de l'informatique Donald Knuth, une bonne programmation est synonyme d'une belle expression.

Susan D'Agostino, 16 avril 2020

Donald Knuth est un informaticien majeur de sa discipline. Pendant les années du démarrage de l'informatique, dans le milieu du siècle dernier, une compagnie fabriquant du sucre lança un concours qui consacra les talents de Donald Knuth alors qu'il avait 13 ans. Le concours demandait aux enfants de déterminer le nombre de mots qu'on pourrait faire avec les lettres de la société : Ziegler's Giant Bar. C'était un problème bien défini avec différentes parties, juste le genre de problème qu'il adorait.

“J'ai eu une période obsessionnelle-compulsive qui m'a dirigé vers les problèmes digitaux, discrets. Et j'adore me pencher sur de grandes quantités d'information”, dit Knuth.

Knuth feuilleta méthodiquement le dictionnaire intégral Funk & Wagnalls de 2000 pages de sa famille. Il réussit même à convaincre ses parents qu'il était malade, gagnant deux semaines loin de l'école pour travailler sur ce problème. Après avoir étiqueté les index avec des entêtes comme “Aa,” “Ab” et “Ba” basés sur les commencements possibles des mots utilisant les lettres de la société sucrière, il éplucha les colonnes du dictionnaire en notant les mots qui devaient être conservés. Il découvrit qu'il pouvait sauter des sections entières du dictionnaire, comme les pages commençant par la lettre “C,” ou bien les sections des mots commençant par “B” et dont la deuxième lettre était “U.”

Les instances officielles du concours avaient identifié environ 2 000 mots qui pouvaient leur être fournis, mais Knuth en trouva plus de 4 700. Son gain au concours fut un spot publicitaire à la télévision et du chocolat pour toute sa classe. Il continuerait à gagner de nombreuses autres récompenses, incluant la première récompense Grace Murray Hopper de l'ACM Award, la médaille nationale de la Science et la récompense A.M. Turing.

Knuth finit par agréger ses deux amours des problèmes discrets digitaux et des larges collections d'informations dans son œuvre majeure, *L'Art de la programmation des ordinateurs* - une série de livres qu'il commença à écrire pendant ses études en 1962 et qui n'est toujours pas achevée. Il publia le volume 1 en 1968, et la version actuelle en est sa 42^{ème} réimpression. Le volume 2 suivit en 1969 et le volume 3 en 1973. Donald Knuth était alors professeur d'informatique à l'Université Stanford, mais le fait que son travail puisse l'empêcher de terminer ses livres l'inquiétait. Alors il prit un congé sabbatique en 1990 et il prit sa retraite en 1993

traduction (Denise Paule Vella-Chemla, 11.10.2020) d'un article de Quanta magazine, lisible ici <https://www.quantamagazine.org/computer-scientist-donald-knuth-cant-stop-telling-stories-20200416/>.

pour passer le reste de sa vie à terminer l'ensemble des sept volumes. Maintenant âgé de 82 ans, il travaille dur sur la partie B du volume 4, et il anticipe que ce volume aura au moins six parties de A à F.

L'Art de la programmation des ordinateurs est plus qu'un manuel d'utilisation. Comme Isaac Asimov et Eric Temple Bell ont tissé dans leurs romans et autour de leurs personnages des histoires de science et de math, Knuth se régale à raconter des histoires d'informatique.

“Le meilleur moyen de communiquer d'un être humain à un autre consiste à raconter une histoire,” dit-il.

Cette passion pour la communication l'a aidé à jouer un rôle mettant en vedette son œuvre majeure dans l'histoire de l'informatique. Lorsque son éditeur lui envoya les épreuves de la seconde édition du volume 2 dans les années 70, Knuth fut perturbé par l'agencement et l'apparence des nombres, des symboles et des mots sur les pages. Il prit un vol pour Los Angeles pour voir une machine qui imprimait des magazines sur papier brillant, en espérant que cela lui offrirait un gain esthétique, mais c'était trop cher. Toutefois, pendant ce voyage, il commença à développer un langage informatique qui lui permettrait de composer des mathématiques de manière digitale.

De retour à Stanford, Knuth mit en sommeil *L'Art de la programmation des ordinateurs* pendant presque une dizaine d'années pour développer TeX (prononcé “tech”), un langage sophistiqué et qui changea la donne, en pourvoyant les ordinateurs de la typographie digitale. Il le rendit open source, surtout pour le bénéfice des mathématiciens professionnels, des informaticiens, des économistes, des ingénieurs, des linguistes, des statisticiens et de toute autre personne pour qui des symboles techniques manquaient à son clavier mais qui maîtrisait le placement de formules complexes mieux que ses éditeurs. Dans le monde des programmes éphémères d'ordinateurs, TeX a perduré comme le standard en or pour écrire des articles scientifiques plus beaux et plus faciles à lire et à comprendre par des experts.

L'intérêt de Knuth pour la narration d'histoire l'a également amené à développer une philosophie de la programmation littéraire - une méthode pour écrire des programmes d'ordinateurs comme des essais littéraires. Un programme littéraire intercale au sein du source une élégante prose en une langue naturelle, comme l'anglais. Le code source a pour caractéristique d'être fonctionnel et efficace, alors que l'exposé en langage naturel cible le lecteur humain plutôt que le compilateur de l'ordinateur. Quiconque devra un jour mettre à jour ou debugger un programme littéraire évitera le problème chronophage et coûteux consistant à essayer de comprendre les algorithmes qu'a mis en place le programmeur initial, les décisions conceptuelles et les stratégies d'implémentation qu'il a choisis. Knuth est un informaticien qui comprend que les mots comptent.

Quanta Magazine s'est entretenu avec Knuth en Févriererchez lui sur le campus de Stanford. L'interview a été condensée et éditée pour être plus claire.

Avez-vous toujours été intéressé par l'écriture ?

J'ai compris tôt que le monde réel était trop difficile pour moi. Je ne m'attendais pas à découvrir quoi que ce soit de nouveau, mais j'aimais transmettre mon enthousiasme pour les idées en écrivant.

En classe de sixième, deux amis et moi commençâmes un journal de deux pages sur une machine ditto. Nous avions des blagues. Au collège, chaque nuit de lundi en tant qu'éditeur de journal, je faisais une nuit blanche pour que le journal puisse sortir. J'ai vu ma première ligne tapée au lycée, en tant qu'éditeur du journal des étudiants. Quand j'étais junior et senior, nous avons commencé l'ingénierie et les revues scientifiques. Par exemple, j'ai écrit, " $\text{Th}_5\text{E}_4 \text{C}_3\text{EmIC}_2\text{Al}_2 \text{Ca}_3\text{P}_4\text{Er}$." Chaque mot était une formule chimique.

Et cela vous a amené à votre œuvre majeure ? Y pensez-vous comme à une autre histoire ?

L'Art de la programmation des ordinateurs est un manifeste. Il décrit la façon dont j'aime les maths et la manière dont j'aurais aimé qu'on me les apprenne. Dès la page 1, je racontais l'histoire des algorithmes. La plupart des livres à ce moment-là n'exploraient pas le côté humain des découvertes. Ils disaient juste, "Voici comment marche la chimie," ou "Voici comment marche la physique."

Je raconte aussi une histoire technique. Je dis, "Voici quelque chose qui ne marche pas, et voici la manière de résoudre ce problème." Plutôt que de présenter seulement des faits, j'ajoute du drame. La science est beaucoup plus facile à apprendre si vous connaissez l'ordre dans lequel ont eu lieu les découvertes. En outre, je suis incapable de résister à une bonne histoire. Je ne me voyais pas comme un pionnier, mais comme un journaliste.

Au-delà de l'histoire, alors, de quoi parle *L'Art de la programmation des ordinateurs* ?

Après deux années passées à écrire mon livre, j'ai réalisé qu'un programme était quantitativement déterminé comme bon quand il était nouveau. Je ne voulais pas juste dire qu'un programme était meilleur qu'un autre. Je voulais dire celui-ci est de 13.8 % meilleur que celui-là, et expliquer comment les comparer.

Admettons que l'auteur A parle de son algorithme A, et que l'auteur B parle de son algorithme concurrent B. Et l'auteur A n'a jamais écrit à propos de l'algorithme B, et l'auteur B n'a jamais écrit à propos de l'algorithme A. En plus, les auteurs A et B ont utilisé des ordinateurs différents. En tant que journaliste neutre, j'expliquais les deux d'un seul point de vue. Demander "Dans quelle proportion cet algorithme est-il bon ?" est un problème marrant. C'est cela l'analyse des algorithmes.

Est-ce que “l’analyse des algorithmes” est juste une manière différente de dire “l’art de la programmation des ordinateurs” ?

J’assistais à une conférence de la Société de mathématiques industrielles et appliquées en 1967 quand quelqu’un me demanda ce que je faisais. À ce moment-là, l’informatique était découpée en analyse numérique, intelligence artificielle et langages de programmation. C’était ça. J’ai réalisé que j’avais besoin d’un nom pour ce que je faisais.

La nouveauté dans mon livre, c’étaient les études rigoureuses de l’efficacité des algorithmes. Je décidai que la prochaine fois qu’on me poserait la question, je répondrais “l’analyse des algorithmes”. Ma définition était : si ça m’intéresse, c’est de l’analyse des algorithmes. Ça n’était pas une très bonne définition.

Plus tard, je décidai de le justifier. Je décidai que c’était l’étude quantitative, i.e. la mesure de la valeur d’un algorithme, que je divisais en deux parties. Une partie considérait tous les algorithmes possibles pour résoudre un problème donné. L’autre partie considérait un algorithme particulier pour un certain problème.

L’Analyse des algorithmes allait devenir le travail de ma vie. Je dis à mon éditeur de changer le titre de mon livre en *L’analyse des algorithmes*. Mon éditeur dit, “ça ne se vendra jamais”. Ils ont pris la bonne décision. Et j’ai été content quand, 40 ans plus tard, cinq ou six livres sont sortis avec le titre *Analyse des algorithmes*.

Mais pour vous, la programmation est davantage que simplement fonctionnelle. Quand vous avez conçu TeX, par exemple, vous vouliez trouver “la courbe la plus plaisante” reliant certains points. Étiez-vous en train d’essayer de programmer la beauté ?

Mon programme devait relier des points d’une manière qui correspondrait à l’ingénierie-inverse de ce que ferait un expert calligraphe. Il y a un point dans la lettre “S” où la courbure change et de positive devient négative. Alors peut-être peut-elle rester constante un peu de temps. Les concepteurs de lettres suivent une certaine logique pour transformer des lignes en formes de lettres. Je voulais capturer non seulement le résultat de cette conception, mais également l’intelligence qui se cache derrière cette conception. C’est comme écrire un programme d’ordinateur.

J’ai parlé à des designers pour comprendre comment ils parvenaient à procéder de la sorte. Les maths étaient là pour capturer le design de manière quantitative. Avec les mathématiques, j’ai mis des cadrans autour de chaque partie. Je pouvais dire la lettre “A” a ce point, cette épaisseur, ces angles ici, elle diminue là, une bosse en haut et une certaine longueur d’empatement.

Je n’ai jamais essayé de remplacer les concepteurs. Je voulais seulement capturer exactement

pour les générations futures ce que nous étions alors en train de faire. Avec TeX, le design est reproductible.

Aviez-vous anticipé l'acceptation globale de TeX ou sa capacité à perdurer longtemps ?

TeX était seulement conçu pour être utilisé par ma secrétaire et moi-même. Phyllis [Astrid Benson Winkler] était une merveilleuse secrétaire. Elle pouvait lire mes manuscrits et les rendre beaux. La technologie de l'impression était en train de passer de mode parce que les méthodes par essais-erreurs devenaient trop chères. Presque tous les articles publiés dans les années 70 étaient atroces à regarder. Dans les *American Mathematical Monthly*, les indices étaient écrits dans une fonte différente des lignes principales de texte. Je savais que l'informatique rendrait à nouveau les livres beaux.

Je finis de déboguer une version temporaire de TeX en avril 1978. En mai, j'avais 10 utilisateurs. En juin, j'avais 100 utilisateurs. En juillet, j'en avais 1000. Chaque nouveau groupe disait "Est-ce que vous avez cette fonctionnalité dont j'ai besoin?". Cinq ans après, je produisais une mise à jour qui est essentiellement le TeX que nous avons aujourd'hui. Il était destiné à des américains. Alors les européens ont commencé à l'utiliser. De ce fait, dans les années 80, je l'ai fait fonctionner pour les langages du monde entier.

À vous entendre, on a le sentiment que la découverte a toujours fait partie de votre cheminement. Cela reste-t-il vrai aujourd'hui ?

J'écris en moyenne cinq nouveaux programmes par semaine. Les poètes doivent écrire des poèmes. Je dois écrire des programmes d'ordinateur.

Le test ultime pour savoir si je comprends quelque chose est de voir si je peux l'expliquer à un ordinateur. Je peux vous dire quelque chose et vous opinerez de la tête, mais je ne suis pas sûr que je vous l'ai bien expliqué. Mais l'ordinateur n'acquiesce pas de la tête. Il me répète exactement ce que je lui ai appris. Dans la plupart des circonstances dans la vie, vous pouvez bluffer, mais pas avec les ordinateurs.

Vous passez vos journées à écrire, mais vous avez aussi d'autres centres d'intérêt. Comment abordez-vous chaque journée ?

Jack London écrivait 1000 mots chaque jour avant de s'adresser à tout le monde. Il était assez totalitaire par rapport à ça, "Laissez-moi seul jusqu'à ce que j'aie écrit mon millier de mots!" Alors il pouvait boire ou relire le reste de la journée. Non, mon principe en terme d'emploi du temps est de faire la chose que je déteste le plus qui est sur ma liste de choses à faire. Quand approche le week-end, je suis très heureux.

Vraiment ? Comment faire des choses que vous détestez peut-il vous rendre heureux ?

Ce serait très facile pour moi de dire “Oh, laissez-moi être un génie et ne jamais nettoyer les toilettes.” Mais même laver les toilettes est une tâche faisable. [Mon épouse] Jill et moi avons des uniformes dotés d’une fente dans laquelle on peut glisser un flacon de nettoyant 409. Vous y allez, faites gicler le produit et vous vous sentez bien d’avoir nettoyé les toilettes !

Le succès d’une personne dans la vie est déterminé par le fait qu’elle ait un potentiel minimum élevé, pas un potentiel maximum élevé. Si vous pouvez vraiment bien faire quelque chose mais qu’il y a d’autres choses que vous ne savez pas bien faire, ces dernières choses vous retarderont. Mais si pour presque tout ce que vous faites, vous êtes au top, alors vous aurez une bonne vie. Et c’est pour ça que j’essaie d’apprendre comment passer à travers les choses que les autres trouvent déplaisantes.

Vous avez aussi de nombreux projets qui n’ont rien à voir avec l’informatique, comme la composition musicale, *Fantasia Apocalyptica*. Vous avez même construit votre maison autour d’un orgue à tuyaux à deux étages. Est-ce que cette variété fait aussi partie de ce qui vous rend heureux ?

J’ai écrit deux livres, incluant *Les choses dont un informaticien parle rarement*, qui parlent de théologie - des choses que vous ne pouvez pas prouver - plutôt que de mathématiques ou d’informatique. Ma vie ne serait pas complète si elle était entièrement consacrée à des choses déconnectées et arides. Les choses mystiques que je ne comprends pas me donne de l’humilité. Il y a des choses au-delà de ma compréhension.

En mathématiques, je sais quand un théorème est correct. J’aime ça. Mais la vie ne vaudrait pas le coup si tout était faisable. Cette connaissance ne m’affaiblit pas. Plutôt, elle m’assure de ne pas rester coincé dans une ornière.

Est-ce que cela importe que vous terminiez *L’Art de la programmation des ordinateurs* ?

Oh, je réalise que l’informatique continuera de vivre et de se développer. L’un des scénari est que tout le monde cessera de travailler sur les ordinateurs que nous avons actuellement. On ira tous vers l’apprentissage machine et les ordinateurs quantiques. Alors je pourrai assister à la fin de l’histoire des ordinateurs non quantiques. Je suis toujours plus content quand je peux dire “Voici la fin de l’histoire”. C’est la façon la plus facile d’imaginer que moi aussi, je finirai. Mais je ne réponds pas à votre question.

Légendes des photos

Donald Knuth a travaillé sa vie entière pour raconter des histoires avec et à propos des programmes d'ordinateur. (Vivian Cromwell)

Knuth travaille en ce moment sur la partie B du volume 4 de L'Art de la programmation des ordinateurs. Il envisage que ce volume ait quatre parties de plus, et que l'ensemble de la série ait trois volumes de plus. (Jill Knuth)

Knuth a toujours raconté des histoires, en leur donnant une tournure scientifique. Au lycée, il écrivit une courte histoire dans laquelle chaque mot était une formule chimique. (Jill Knuth)

Knuth dans son bureau à Stanford avec le concepteur de fontes Herman Zapf en 1980, juste quelques années après la version du programme de typographie de Knuth, TeX. (Chuck Painter / Stanford News Service)

“Le meilleur moyen de communiquer d'un être humain à un autre consiste à raconter une histoire,” dit Knuth. Cette approche lui a permis d'analyser les algorithmes plus rigoureusement et de façon neutre. (Vivian Cromwell)

Knuth chez lui en 2020. Il a pris sa retraite en 1993 pour terminer son œuvre majeure, L'Art de la programmation des ordinateurs. (Jill Knuth)

Textes des “accroches” ou “coiffes”¹

Je suis incapable de résister à une bonne histoire. Je ne me voyais pas comme un pionnier, mais comme un journaliste.

Je savais que la programmation des ordinateurs rendraient à nouveau les livres beaux à regarder.

1. terme journalistique exact ?