

Rappel de programmes pour envelopper le log par des droites (Denise Vella-Chemla, août 2023).

On écrit des programmes dans le style de l'article de blog ici : <https://www.cantorsparadise.com/the-twin-prime-conjecture-3671e604818e> qui permettent de voir les courbes de la racine carrée ou du logarithme enveloppées par des droites de pentes de plus en plus faibles.

```
pgm1.py - GNU Emacs at DESKTOP-4PROGGL
File Edit Options Buffers Tools Python Help
import math
from math import sqrt, log
import matplotlib.pyplot as plt
import time

def prime(atester):
    k = 2
    if atester in [0, 1]: return False
    if atester in [2, 3, 5, 7]: return True
    while True:
        if k * k > atester: return True
        else:
            if atester % k == 0: return False
            else: k = k + 1

tic=time.time()
nmax = 1000
for x in range(3,nmax+2,2):
    if prime(x):
        m = 6/x
        p = x-(m*x/6)
        ybas = -m*nmax+p
        yhaut = m*nmax+p
        plt.plot([-nmax,nmax],[ybas,yhaut])
plt.axis([-10,nmax,0,log(nmax)*sqrt(nmax)])
plt.show()
tac=time.time()
print(tac-tic, ' s.')
```

```
pgm2.py - GNU Emacs at DESKTOP-4PROGGL
File Edit Options Buffers Tools Python Isearch Help
← → × ↶ ↷ 📄 🖱️ 🚫

import math
from math import sqrt, log
import matplotlib.pyplot as plt
import time

def prime(atester):
    k = 2
    if atester in [0, 1]: return False
    if atester in [2, 3, 5, 7]: return True
    while True:
        if k * k > atester: return True
        else:
            if atester % k == 0: return False
            else: k = k + 1

tic=time.time()
nmax=100000
for n in range(4,nmax+2,2):
    if prime(n-1) and prime(n+1):
        m = 6/n
        p = n+1-(m*n/6)
        ybas = -m*nmax+p
        yhaut = m*nmax+p
        plt.plot([-nmax,nmax],[ybas,yhaut])
plt.axis([-100,nmax,0,log(nmax)*sqrt(nmax)])
plt.show()
tac=time.time()
print(tac-tic,' s.')
```

