Devoir Maison – Test de Miller-Rabin

Master Informatique Université de Lorraine

BAUDON Nicolas, BERNARD Paul-Antoine

février 2022

---

1 Question 1

Nous avons choisi de programmer en python. Nous allons utiliser la bibliothèque **gmpy** (ou **gmpy2**), il s'agit d'un module d'extension de la librairie GMP de C. Elle permet d'utiliser toutes les opérations demandées :

- addition (`add()`)
- division (`div()`)
- multiplication (`mult()`)
- mod, powmod etc

2 Question 2

Selon la définition de wikipédia, un nombre aléatoire est un nombre tel qu'une fois généré il n'existe aucun lien déterministe entre celui-ci et ses prédécesseurs. En bref il s'agit d'un nombre dont le résultat est incertain. Sous python on peut utiliser :

- `os.random`
- `random.SystemRandom([seed])`

Un générateur de nombres aléatoires est un dispositif qui permet de produire une séquence de nombres pour lesquels il n'existe aucun lien déterministes connus.

3 Question 4

Nous avons implémenté ici la fonction d'exponentiation binaire demandée mais nous nous sommes ensuite rendu compte que celle-ci était également fournie dans le module **gmpy2**. Nous avons donc utilisé la fonction `powmod(a, d, n)` fournie par **gmpy2** dans la suite du DM.

4 Question 6

Notre fonction renvoie les résultats suivants :

- n1 : est un nombre premier
- n2 : n'est pas un nombre premier
- n3 : est un nombre premier

Nous avons vérifié ces résultats à l'aide de calculatrice de très grands nombres premiers disponible en ligne (https://bigprimes.org/primality-test).

5 Question 8

Voici les résultats renvoyés par notre programme lorsque l'on cherche le nombre d'itérations nécessaires pour trouver un nombre premier de b bits en moyenne (sur 100 répétitions) :

- b = 128 bits : 165.14 répétitions
- b = 256 bits : 383.91 répétitions

- b = 512 bits : 691.61 répétitions
- b = 1024 bits : 1329.36 répétitions
- b = 2048 bits : 3034.62 répétitions
- b = 4096 bits : 5828.39 répétitions

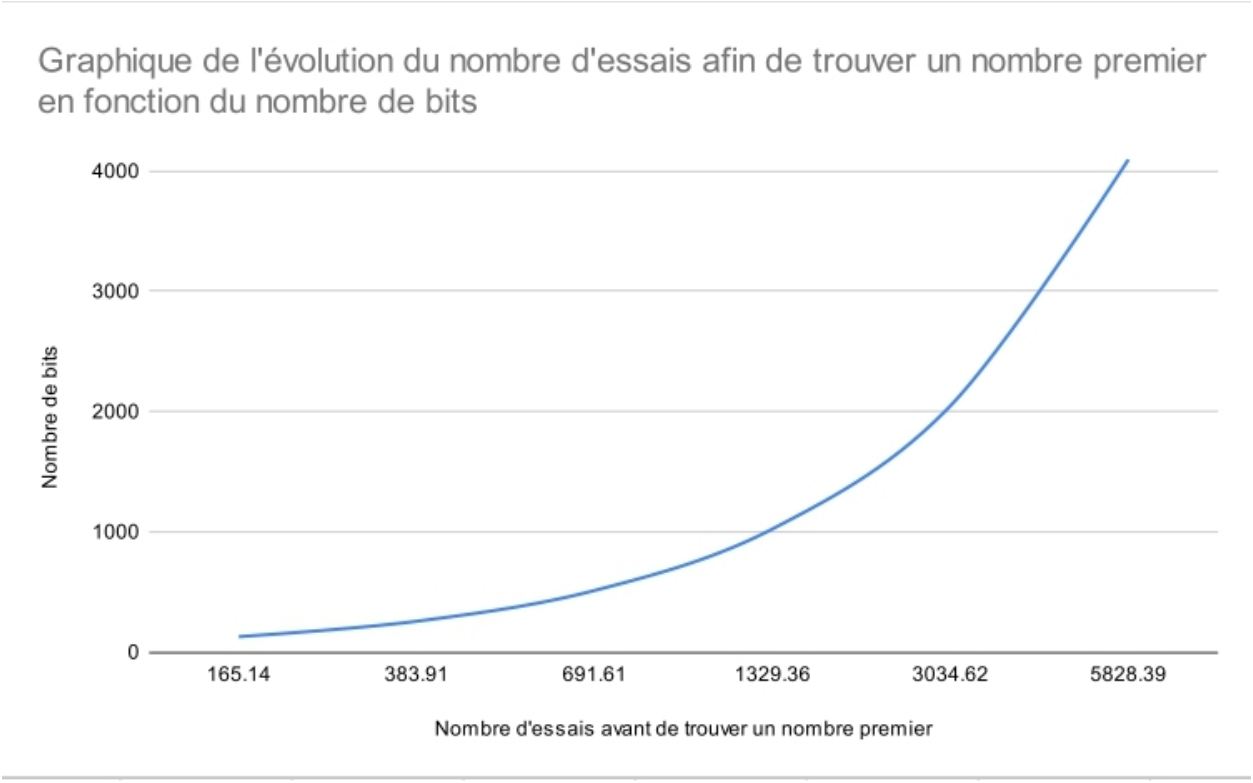

FIGURE 1 – Graphique illustrant l'augmentation du nombre de répétitions de l'algorithme nécessaires pour trouver un nombre premier

6 Question 9

On constate que l'augmentation du nombre d'essais nécessaire pour trouver un nombre premier est croissante. On voit que la courbe augmente significativement lorsque l'on passe de 2048 à 4096 bits. Cela peut-être expliqué par le fait que l'on augmente notre nombre de bits en le multipliant par deux à chaque fois mais il s'agit surtout là d'une preuve visuelle de la raréfaction des nombres premiers. En effet, l'ensemble des nombres premiers possède une densité limite nulle, plus on progresse vers des grands nombres, plus il est difficile d'en trouver des éléments. Ce théorème a été démontré par Legendre en 1808 (théorème de raréfaction des nombres premiers) :

$$lim_{n\to\infty}\frac{\pi(n)}{n} = 0$$

7 Question 10

Il existe un test de primalité deterministe nommé le test cyclotomique ou test de primalité AKS. Sa complexité est polynomiale : $O(n^{clog(log(n))})$.