Exercices
Semaine 10

Cours Turing

1 Chercher efficacement des grands nombres premiers

Dans cet exercice, on vous propose de chercher des grands nombres premiers en utilisant tout ce que vous avez vu jusqu'à maintenant, à savoir :

i) un générateur de nombres aléatoires pour tirer au hasard nombre N à 100 chiffres. Pour cela, vous pouvez utiliser celui implémenté en Python, qui fait appel au package random. Voici par exemple un moyen simple de générer un nombre aléatoire avec (à peu près) 100 chiffres :

```
N = random.randint(10**99,10**100-1)
```

ii) le test de Fermat qui teste si, pour un nombre donné N, et un nombre tiré au hasard A, $A^{N-1} \pmod N = 1$ (à répéter avec 30 valeurs différentes de A pour minimiser la probabilité de faire une erreur).

NB : En Python, $A^B \pmod N$ s'écrit pow(A, B, N).

iii) et si vous voulez encore optimiser votre code, vous pouvez d'abord tester si le nombre N n'est multiple ni de 2, ni de 3, ni de 5, avec la solution de l'exercice de la semaine dernière.

2 Exponentiation rapide

Ecrire un programme qui calcule efficacement $A^B \pmod N$, en vous basant sur ce que vous avez vu en cours. Il s'agit donc :

- d'utiliser la décomposition binaire du nombre B ;

- de calculer successivement $A^2 \pmod N$, $(A^2)^2 \pmod N$, etc. (en stockant éventuellement ces calculs préliminaires dans un vecteur), puis d'effectuer les multiplications appropriées (toujours $\pmod N$) pour trouver le résultat final.

Bonus :* Vous pouvez aussi écrire une version récursive de ce programme !

3 Factorisation

a*) (*Note :* Cette première partie est plus difficile : si vous voulez, vous pouvez aller chercher le code solution sur Moodle, et l'utiliser comme une "boîte noire" pour la partie b)

Ecrire un programme qui prenne en entrée un nombre entier positif N et dont la sortie soit :

- la liste P de tous les nombres premiers plus petits que N

- la liste L de taille N, où $L[i]$ soit le plus petit facteur premier de i. Par exemple, si $N = 10$, alors la liste $L = [0, 1, 2, 3, 2, 5, 2, 7, 2, 3]$ (où on a posé ici $L[0] = 0$ et $L[1] = 1$ par convention).

b) Ecrire un programme qui prenne en entrée un nombre entier positif k, ainsi que k nombres entiers positifs $N_1, \ldots, N_k$ et imprime ensuite k lignes, avec sur chaque ligne i la liste des facteurs premiers de N_i (par exemple, si $N_i = 60$, alors la sortie sur la ligne i doit être 2 2 3 5).

Note : Il va sans dire que cet programme ne fonctionnera que très lentement pour des grands nombres entiers $N_1, \ldots, N_k$!