TD de Complexité, séance 8

Les nombres premiers, et exercices d'arithmétique

Un nombre premier est un nombre entier positif, qui admet exactement deux diviseurs distincts et positifs, qui sont 1 et lui-même[1]. Les 25 nombres premiers inférieurs à 100 sont : 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89 et 97. Un nombre qui n'est pas premier est appelé un nombre composé (par exemple, le nombre 6 a pour diviseurs 1, 2, 3 et 6 et est donc composé).
Les nombres premiers étaient connus dès l'antiquité, et peut-être même à la préhistoire. Euclide (300 *av. JC*) a par exemple démontré qu'il existait un nombre infini de nombres premiers. Plus précisément, on peut montrer que le nombre de nombres premiers inférieurs ou égaux à p a pour ordre de grandeur $\pi(p) = O\left(\frac{p}{\log p}\right)$. Ces nombres jouent un rôle très important dans la cryptographie moderne, et notamment dans les cryptosystèmes à clefs asymétriques comme RSA, inventé en 1978, et qui est encore utilisé aujourd'hui pour chiffrer les communications sur Internet ou les transactions bancaires. Le principe de fonctionnement de cet algorithme est qu'il est relativement aisé de générer un nombre premier arbitrairement grand, mais qu'il est extrêmement complexe de déterminer les facteurs d'un grand nombre composé, égal au produit de deux grands nombres premiers. Le but de ce TD est d'étudier quelques-uns des algorithmes usuels liés aux nombres premiers, et d'étudier leur complexité.

1. Proposez un algorithme ayant pour complexité $O(p)$ permettant de déterminer si le nombre *p* est premier.
2. Quelle est la complexité de cet algorithme, exprimée en fonction de la taille des données d'entrée (*n* = nombre de bits de *p* lorsqu'il est écrit en base 2) ?
3. Montrez qu'une petite astuce permet de réduire la complexité de cet algorithme à $O(\sqrt{p})$.
4. Quelle est la complexité de l'algorithme qui consisterait à utiliser l'algorithme de la question 3 pour déterminer l'ensemble des nombres premiers inférieurs à p ?
5. Le crible d'Eratosthène est un algorithme plus efficace pour résoudre le problème de la question 4 : il construit la liste de tous les entiers inférieurs à p, puis élimine tous les multiples de 2, puis tous les multiples de 3, puis tous les multiples du premier nombre non encore éliminé… et ainsi de suite jusqu'à ce que le premier nombre non éliminé soit supérieur à $\sqrt{p}$. Justifier son fonctionnement, écrire cet algorithme en méta-langage algorithmique (de manière itérative ou de manière récursive), puis évaluer sa complexité.
6. Un nombre parfait est un nombre égal à la somme de ses diviseurs stricts (1 compris, mais lui-même exclus). Proposez un algorithme, pour trouver tous les nombres parfaits inférieurs à *n*, et évaluez sa complexité.
7. Des nombres amis sont deux nombres dont chacun est égal à la somme des diviseurs stricts de l'autre. Proposez un algorithme, pour trouver tous les couples nombres amis inférieurs à *n*, et évaluez sa complexité.

[1] D'après Wikipedia, https://fr.wikipedia.org/wiki/Nombre_premier, consulté le 23/11/2016