Projet d'algorithmique

Contexte

Les nombres premiers sont un sujet de premier plan en mathématiques et trouvent de nombreuses applications pratiques, en particulier en cryptographie. Ce projet d'algorithmique s'intéresse à la factorisation d'un nombre entier en nombres premiers.

On considère les entiers naturels strictement positifs appelés simplement *entiers*. Un entier n est *divisible* par un entier k s'il existe un entier p tel que $n = k \times p$ et on dit que k *divise* n ou k est un *diviseur* de n (on remarque également que p divise n). Un entier n est *premier* s'il n'a pas de diviseur autre que 1 et n. La *factorisation* d'un entier n en nombres premiers est une suite finie unique $(k_1, m_1), \ldots, (k_p, m_p)$ qui vérifie les propriétés suivantes :

- $n = k_1^{m_1} \times k_2^{m_2} \times \cdots \times k_p^{m_p}$;
- k_i est un entier premier appelé *facteur* pour tout i entre 1 et p ;
- m_i est un entier positif non nul appelé *multiplicité* pour tout i entre 1 et p ;
- $k_i < k_j$ pour tout i et j tels que $1 \leq i < j \leq p$ (autrement dit la suite est ordonnée par ordre croissant des facteurs).

Par exemple, la suite $(2, 2), (5, 3), (13, 2), (83, 1)$ est la factorisation de l'entier

$$7013500 = 2^2 \times 5^3 \times 13^2 \times 83.$$

Sujet

L'objectif du projet est de réaliser une structure de données fint pour les factorisations des entiers. Le travail est donc constitué de trois branches :

- La définition d'une SDA munie des opérations suivantes étant donnés deux fint a et b et un entier strictement positif n : création d'un fint à partir de n, tester si a divise b, calcul de $a \times b$, a^n, $a \operatorname{div} b$, $a \bmod b$, $\gcd(a, b)$ et $\operatorname{lcm}(a, b)$, le « gcd » étant le plus grand diviseur commun et le « lcm » étant le plus petit multiple commun ;
- La réalisation d'une SDC (représentation mémoire et algorithmes des opérations) en visant les meilleures complexités possibles ;
- La programmation d'une classe en C++ en respectant les codes de la programmation objet et en exploitant au mieux les capacités du langage (surcharge des opérateurs, réutilisation de la librairie standard, gestion des exceptions, *etc.*).

L'opération de création pose le problème de la recherche des facteurs d'un entier. Un algorithme naïf a été vu en TD. Dans ce projet, on pourra s'intéresser également à l'algorithme « rho » de Pollard.

Pour montrer l'utilité des factorisations, on pourra simplifier une fraction de deux fint en fraction irréductible. Si le temps le permet, on pourra envisager d'utiliser une librarie de nombres entiers en précision infinie comme Gnu MP pour pouvoir factoriser de « grands » entiers.

Rendu et évaluation

Chaque binôme déposera sur madoc, **au plus tard le jeudi 29 avril 2021**, un fichier archive contenant un rapport au format PDF décrivant le couple SDA / SDC et le code source (et seulement le code source) compatible avec le *header* fourni « fint.h ».