Université de Bourgogne
U.F.R. Sciences et Technologies
Département Informatique
Année 2016-2017

Master
M1 Informatique

RAPPORT DE PROJET
UE
Algorithme et Complexité

Etudiants : Yann Véron et Marvin Nurit
Sujet : Multiplication rapide de grands entiers

Table des matières

Chapitre 1

Problématique

1.1 Sujet

1.1.1 Description générale

Dans ce sujet, il nous est demandé de programmer en langage OCaml la multiplication de grands entier de deux façons différentes, l'algorithme de Karatsuba [1] [3] [2] et la Transformée de Fourier Rapide.
La multiplication de deux entiers de façon "naïve" de taille n se fait avec n^2 multiplication, ce qui correspond à une complexité de l'ordre de $O(n)$.
Cette complexité, et donc le temps d'exécution de cette multiplication peuvent poser problème sur de grands entiers et polynômes.
C'est pourquoi les deux algorithmes cités précédemment, Fourier et Karatsuba, ont été mis au point.

1.1.2 Multiplication par l'algorithme de Karatsuba

Anatoli Alekseïevitch Karatsuba est un mathématicien Russe ayant obtenus plusieurs récompenses pour ses travaux sur l'algorithmique Multiprécision, plus particulièrement la multiplication.
Il donne ainsi son nom au premier algorithme de multiplication rapide, l'algorithme de Karatsuba, à démontré un théorème d'approximation des séries de Fourier, et amélioré le théorème de Moore sur la machine de Moore.
Dans notre cas, c'est son algorithme de multiplication rapide qui nous intéresse.
Comme dit précédemment, la multiplication "naïve" à une complexité d'ordre $O(n)$, tandis que l'algorithme de Karatsuba à une complexité d'ordre au plus $n^{log_2 3} \approx n^{1.585}$, comme nous le verrons prochainement.
L'algorithme de Karatsuba est de type "Divide and Conquer", ou "Diviser pour régner".
L'objectif est de décomposer les deux entiers en deux plus petit (dont la taille est approximativement égale à la moitié de la taille des deux grands entiers), puis de faire la multiplication des ces deux plus petits entiers.
La méthode de Karatsuba permet de faire cette dernière multiplication en 3 produits au lieu de 4 de façon naïve. Pour de plus grands entiers il suffira d'appliquer récursivement cette méthode.

1.1.3 Multiplication par la transformée de Fourier Rapide

Chapitre 2

Choix d'implémentation

2.1 Algorithmes

2.1.1 Implémentation des polynômes

Nous avons choisi de représenter nos grands entiers en base B en polynômes.
Ces polynômes seront représentés en OCaml sous la forme de listes, grâce au module List implémenté dans ce langage.
Les entiers seront représentés bit de poids faible en premier pour faciliter la manipulation.
Ainsi un entier 1234 en base 10, dont le polynômes vaut $1 \times 10^3 + 2 \times 10^2 + 3 \times 10^1 + 4 \times 10^0$ sera représenté par la liste $L = [4; 3; 2; 1]$.
De même pour une base 16, l'entier 26, codé de manière hexadécimale 1A, et dont le polynôme vaut $1 \times 16^1 + 11 \times 16^0$ sera représenté par la liste $L = [11, 1]$.
Pour cette conversion, et l'utilisation des listes dans les fonctions suivantes il faudra de ce fait passer en paramètre la base sur laquelle le polynôme à été créé, la liste pouvant représenter différents polynômes différents selon la base.
La reconstruction de l'entier depuis la liste fonctionne de la même manière :
En fonction de la base passée en paramètre on additionne les membres du polynôme que l'on multiplie par la base à la puissance adéquate.
Pour plus de facilité d'utilisation, il aurait sans doute été possible de mettre en premier la base dans laquelle la liste à été créé.

2.1.2 Algorithme de Karatsuba

2.1.3 Transformée de Fourier Rapide

Bibliographie

[1] Multiplication rapide. http://algo.inria.fr/bostan/mpri/Cours2.pdf.

[2] Algorithme de karatsuba. https://fr.wikipedia.org/wiki/Algorithme_de_Karatsuba, 2016.

[3] Xavier-Francois Roblot. Arithmétique rapide, multiplication rapide par la méthode de karatsuba. http://math.univ-lyon1.fr/~roblot/resources/ens_partie_2.pdf.