Université Paris 8 Année 2020–2021

Master Mathématiques et applications

Algorithmes arithmétiques II — Devoir à la maison 2

à rendre pour le samedi 13 février 2021, **dernier délai**

Consigne. Ci-dessous, vous trouverez deux sujets de DM. Vous ne devez en traiter qu'**un seul**. Vous pouvez rendre un travail de groupe de 2 étudiant·e·s maximum.

Documents à fournir. Vous devez rendre par email

1. un rapport contenant vos réponses aux questions, qui sera obligatoirement au format `.pdf`,
2. une archive (format `.zip`) contenant vos programmes.

L'email sera adressé à `julien.lavauzelle@univ-paris8.fr`.

Remarques. Lors de vos implantations, vous allez manier de grands entiers. Certains langages comme `python` les supporte nativement. Pour d'autres, comme `C`, il faut importer une bibliothèque externe (GMP pour `C`, par exemple).

Ressources. On donne à l'adresse

`https://www.math.univ-paris13.fr/~lavauzelle/teaching/2020-21/docs/AA-dm2-aux.zip`

des fichiers auxiliaires contenant des listes de premiers p de taille particulière, ou des listes des nombres premier p tels que $p+1$ est B-superfriable pour des bornes B particulières.

1 Sujet 1 : méthode ρ de Pollard

Consigne. Dans ce sujet, vous devrez répondre aux questions 3 et 4 dans le rapport. Le **soin** et les **justifications** apportées à vos réponses seront évaluées.

On considère la méthode ρ de Pollard comme décrite dans l'Algorithme 1. Cette procédure permet de trouver un facteur d'un entier composé N en entrée de l'algorithme.

Algorithme 1 : Méthode ρ de Pollard avec une fonction d'itération f générique (par exemple on peut poser $f(z) = z^2 + 1$)

Entrée : un entier composé $N \geq 4$
Sortie : un diviseur d de N

Initialiser $d \leftarrow N$
Tant que $d = N$ **faire**
    Tirer a uniformément dans $\{0, \ldots, N-1\}$
    Initialiser $x \leftarrow f(a) \mod N$ et $y \leftarrow f(f(a)) \mod N$
    Calculer $d = \text{pgcd}(y - x, N)$
    Tant que $d = 1$ **faire**
        Calculer $x \leftarrow f(x) \mod N$
        Calculer $y \leftarrow f(f(y)) \mod N$
        Calculer $d = \text{pgcd}(y - x, N)$
Retourner d.

Question 1.– Implanter un algorithme qui calcule le pgcd de deux entiers. Cet algorithme devra supporter des entrées de taille importante (plusieurs centaines de chiffres).

Question 2.– Implanter l'Algorithme 1 avec la fonction $f(z) = z^2 + 1$. Tester votre implantation, par exemple avec certaines valeurs données en annexe.

Question 3.– Calculer **expérimentalement** la complexité de la méthode ρ de Pollard, et comparer avec la valeur **théorique** donnée en cours.

Pour cela, on pourra tracer un graphe du temps de calcul (ou du nombre d'opérations comptées) en fonction de la taille du plus petit facteur premier p de N, et de N. On pourra également s'aider des entiers donnés dans les fichiers auxiliaires, ou implémenter un générateur d'entiers composés ayant la forme désirée pour la méthode « $p + 1$ ». Notons que pour vérifier qu'une fonction $f(x)$ se comporte comme βx^α, on trace $\log(f(x))$ en fonction de $\log(x)$.

Question 4.– Tester votre algorithme avec les fonctions f suivantes (au lieu de $f(z) = z^2 + 1$) :

1. $f(z) = z + 1$,
2. $f(z) = z^2$,
3. $f(z) = z^2 + 2$,
4. $f(z) = z^2 - 1$.

Parmi ces fonctions, lesquelles permettent de trouver efficacement un facteur propre ? Justifier, ou bien par des expérimentations, ou bien par un raisonnement théorique.

Question 5.– Écrire un algorithme qui factorise **complètement** un entier N. L'algorithme utilisera votre méthode ρ comme sous-fonction, et devra retourner une liste

$$L = [(p_1, e_1), \ldots, (p_k, e_k)]$$

telle que l'entier N à factoriser a pour décomposition en facteurs premiers $N = \prod_{i=1}^{k} p_k^{e_k}$.

2 Sujet 2 : méthode $p+1$

Consigne. Dans ce sujet, vous devrez répondre aux questions 2 et 5 dans le rapport. Le **soin** et les **justifications** apportées à vos réponses seront évaluées.

Dans ce sujet, on considère la méthode de factorisation « $p+1$ » de Williams. Cette méthode permet de trouver des facteurs p d'un entier N à factoriser, tels que $p+1$ est superfriable. On rappelle qu'un entier x est B-superfriable si tout entier de la forme p^e qui divise x est plus petit que B.

Question 1.– Implanter une fonction `calcule_indice`(B) qui calcule la valeur de $M = \text{ppcm}(2, \ldots, B)$. On notera que M est aussi égal au produit de tous les entiers de la forme $p_i^{e_i}$, où p_i est premier et e_i est maximal tel que $p_i^{e_i} \leq B$. Pour vous aider, vous pourrez trouver dans le fichier annexe `liste_petits_premiers.txt` la liste de tous les nombres premiers plus petits que 10 000.

La méthode « $p+1$ » repose sur le calcul d'une suite (x_n) définie par les relations de récurrence suivantes :

$$\begin{cases} x_{2n} &= 2x_n^2 - 1 \\ x_{2n+1} &= 2x_n x_{n+1} - x_1 \end{cases} \qquad \text{pour } n \geq 1$$

Ainsi, si z_n désigne le couple (x_n, x_{n+1}), on peut déduire (z_{2n}, z_{2n+1}) de x_1 et z_n. Cela signifie qu'on peut **adapter la méthode d'exponentiation binaire** au calcul de x_n.

Exemple. Si $M = 18$, l'écriture de M en base de M est $(10010)_2$. Les troncations de cette écriture forment les entiers $1 = (1)_2$, $2 = (10)_2$, $4 = (100)_2$, $9 = (1001)_2$ et $18 = (10010)_2$. Donc, on va successivement calculer x_1, x_2, x_4, x_9 et x_{18}.

Il est facile de passer de x_1 à x_2 et de x_2 à x_4, par l'équation $x_{2n} = -1 + 2x_n^2$ donnée plus haut. En revanche, pour calculer x_9, on a besoin à la fois de x_4 et de x_5. Dans l'algorithme de calcul de la suite (x_n), il faudra donc maintenir la connaissance de $z_n = (x_n, x_{n+1})$.

Pour $x_1 = 2$, $f : z \mapsto z^2 + 1$ et $N = 10^{10}$, on doit obtenir les valeurs suivantes

n	x_n	x_{n+1}
1	2	7
2	7	26
4	97	362
9	70226	262087
18	9863382151	–

Question 2.– Écrire dans le rapport le pseudo-code d'une fonction `calcule_x`(x_1, M, N) qui prend en entrée un élément $x_1 \in \{1, \ldots, N-1\}$ et qui calcule la valeur de $(x_M \bmod N)$ par une méthode analogue à l'exponentiation binaire. Puis, implanter la fonction.

On s'intéresse maintenant à l'Algorithme 2, dit méthode de factorisation « $p+1$ » de Williams.

Algorithme 2 : Méthode $p+1$ pour une borne de fiabilité B

Entrée : un entier composé $N \geq 4$ possédant un facteur p tel que $p+1$ est B-superfriable
Sortie : un diviseur d de N

Initialiser $d \leftarrow N$.
Tant que $d = N$ **faire**
  Calculer $M \leftarrow \text{ppcm}\{2, \ldots, B\}$.
  Tirer aléatoirement x_1 dans $\{1, \ldots, N-1\}$.
  Calculer $d \leftarrow \text{pgcd}(x_1, N)$.
  Si $d \neq 1$
    Retourner d
  Calculer $x_M \leftarrow$ `calcule_x`(x_1, M, N).
  Calculer $d \leftarrow \text{pgcd}(x_M - 1, N)$.
Retourner d.

Question 3.– Implanter un algorithme qui calcule de pgcd de deux entiers. Cet algorithme devra supporter des entrées de taille importante (plusieurs centaines de chiffres).

Question 4.– Implanter l'Algorithme 2. Tester votre implantation avec certaines valeurs données en annexe.

Question 5.– Calculer **expérimentalement** la complexité de la méthode $p+1$ de Williams, et comparer avec la valeur **théorique** donnée en cours.

Pour cela, on pourra tracer un graphe du temps de calcul (ou du nombre d'opérations comptées) en fonction de N, la borne de friabilité choisie, et de N. On pourra également s'aider des entiers donnés dans les fichiers auxiliaires, ou implémenter un générateur d'entiers composés de taille particulière. Notons que pour vérifier qu'une fonction $f(x)$ se comporte comme βx^{α}, on trace $\log(f(x))$ en fonction de $\log(x)$.

3 Annexe

3.1 Exemples de factorisation par la méthode ρ

$N = pq$	facteurs $(p < q)$
1457	31 , 47
2047	23 , 89
4189	59 , 71
27451	97 , 283
172847	127 , 1361
5506783	1741 , 3163
122839103	107 , 1148029
2373442927	1667 , 1423781
2432194014911	1319 , 1843968169
2386194275560789	1523 , 1566772341143
2543362194387034583	1753 , 1450862632280111
3256575592288000884277	1934351 , 1683549465576827
2149473581756425241209729	1085763209 , 1979689092372281
312560694913751211192104365075234250971 1	1332472387 , 2345719866041405713513704243653

TABLE 1 – Exemples de factorisations « faciles » effectuées par l'algorithme ρ de Pollard.

3.2 Exemples de factorisation par la méthode $p + 1$

$N = pq$	facteurs p, q tels que $p + 1$ est superfriable	
8435923	2243	3761
433214017981	526679	822539
668877085585453	20540519	32563787
29601945037090540097	4302501839	6880170223
2202930212280802191504287	1099511799979	2003553042653
143460616409214794924368837 8019	1125899906956109	1274186235586991
168373945511479629236199196552692 0283	1152921504620379229	1460411180090875927

TABLE 2 – Exemples de factorisations « faciles » effectuées par l'algorithme $p + 1$ de Williams.

Quelques exemples de taille plus conséquente, mais que l'algorithme $p + 1$ règle toujours très rapidement :

- $N = 10562256276314677189367086699051860083179133005474255536770513713410152272327$
  $p = 80694878738093144866358575156611462449$
  $q = 13089128382726742425590053490410316402 3$

- $N = 62168737006042231928510814222178349101156309737965504752177410010962889443724567730 7...$
  $...16203710521739874848738315366444050483000922241076559240303096972312 29$
  $p = 67370446184902616587480990283695167230411304043318606176733774562163367914599$
  $q = 92278945036843081009237393938740885814972681387747472129904570310056540085371$

- $N = 75335665860992821169105470726766540912231610419365378084109059906708119860068536707...$
  $...23673944387751725757220297910047314599198514346142077537459486353899376688526034522045 70...$
  $...46051949701511230965455078708866820795617315314721680376760845 13$
  $p = 25859751346998904721315154541135445772069592027081540466915141922132646156251 8...$
  $...12286986260376922317805736767051853068 79$
  $q = 29132401487586512476354570916689568474194168376146508296871732993914796687724 5...$
  $...60484143488960272064176130757946206028 47$