

## Feuille de TD/TP 1

---

### Complexité de quelques processus simples

Fixons un entier  $b \geq 2$  qui sera notre base de travail. On suppose que l'on dispose des tables d'addition et de multiplication pour les entiers  $\leq b - 1$ . Le coût d'un calcul est défini comme étant le nombre d'additions et de multiplications d'entiers  $\leq b - 1$  à effectuer au cours de ce calcul.

#### Exercice 1.

1. Evaluer le coût du calcul naïf de  $m + n$  en fonction du nombre de chiffres de  $m$  et  $n$ .
2. Evaluer le coût du calcul naïf de  $mn$  en fonction du nombre de chiffres de  $m$  et  $n$ .

#### Exercice 2.

 Soient  $a, b \geq 2$  deux entiers.

1. Ecrire un algorithme qui transforme l'écriture en base  $a$  d'un entier  $n$  en son écriture en base  $b$ .
2. En admettant que la division d'un nombre à au plus deux chiffres par un chiffre est une opération élémentaire, estimer la complexité de cet algorithme :
  - i) lorsque  $a$  et  $b$  sont quelconques ;
  - ii) lorsque  $a$  est une puissance de  $b$  ;
  - iii) lorsque  $b$  est une puissance de  $a$ .

#### Exercice 3.

 Soient  $A$  un anneau,  $a$  un élément de  $A$  et  $n \geq 1$  un entier.

1. Estimer le coût du calcul (en nombre d'opérations dans  $A$ ) de  $a^n$  par la méthode naïve consistant à calculer les puissances successives de  $a$  jusqu'à aboutir à  $a^n$ .
2. Vérifier que l'algorithme suivant renvoie bien  $a^n$  :  
 $\text{expr}(a, n)$   
Si  $n = 1$  renvoyer  $a$  ;  
Si  $n$  pair renvoyer  $\text{expr}(a, \frac{n}{2})^2$  ;  
Si  $n$  impair renvoyer  $a * \text{expr}(a, \frac{n-1}{2})^2$ .
3. Notons  $C_a(n)$  le coût du calcul de  $a^n$  par l'algorithme  $\text{expr}$ .
  - (a) Donner une expression de  $C_a(2n + 1)$  et de  $C_a(2n)$  en fonction de  $C_a(n)$ .
  - (b) En déduire une majoration de  $C_a(n)$ .
4. Étant donnés des entiers  $a \in \mathbb{Z}$  et  $N \geq 1$ , expliquer comment utiliser Sage pour calculer  $a^n \pmod N$  sans avoir à calculer  $a^n$ . Tester pour de grandes valeurs de  $n$ .

#### Exercice 4.

 On rappelle que la suite de Fibonacci  $(F_n)_{n \geq 0}$  est définie par les relations suivantes :  $F_0 = 0$ ,  $F_1 = 1$  et, pour tout  $n \geq 1$ ,  $F_{n+1} := F_n + F_{n-1}$ .

1. Exprimer  $F_n$  en fonction de  $n$ .
2. En notant  $\phi$  le nombre d'or, montrer que  $F_n$  est équivalent à  $\frac{1}{\sqrt{5}}\phi^n$  lorsque  $n$  tend vers  $+\infty$ .
3. Ecrire un algorithme (récursif) naïf de calcul des nombres de Fibonacci et estimer son coût.
4. Ecrire un algorithme de calcul du  $n$ -ième nombre de Fibonacci ayant un coût en  $\mathcal{O}(n)$  opérations dans  $\mathbb{Z}$ .  
*Indication : Penser à calculer la paire  $(F_{n-1}, F_n)$ .*
5. Ecrire un algorithme de calcul du  $n$ -ième nombre de Fibonacci ayant un coût en  $\mathcal{O}(\log_2(n))$  opérations dans  $\mathbb{Z}$ .  
*Indication : Penser à l'écriture matricielle, comme dans l'algorithme d'Euclide étendu.*
6. Tester ces algorithmes avec le logiciel Sage.

#### Exercice 5.

 Soient  $K$  un corps quelconque et  $m, n$  deux entiers. Calculer le coût (en nombre d'opérations dans  $K$ ) des opérations suivantes :

1. l'addition d'un polynôme de degré  $n$  et d'un polynôme de degré  $m$  ;
2. la multiplication d'un polynôme de degré  $n$  et d'un polynôme de degré  $m$ .

## Algorithmes de Karatsuba

**Exercice 6.** Soit  $n = 2m$  un entier pair non nul et soient  $a, b$  deux entiers s'écrivant avec au plus  $n$  bits. Vérifier que l'algorithme suivant calcule le produit  $ab$ , puis déterminer son coût en fonction de  $n$  :

kara(a,b)  
 Ecrire  $a$  sous la forme  $\alpha 2^m + \tilde{\alpha}$  ;  
 Ecrire  $b$  sous la forme  $\beta 2^m + \tilde{\beta}$  ;  
 Poser  $x := (\alpha + \tilde{\alpha}) * (\beta + \tilde{\beta})$  ;  
 Poser  $y := \alpha\beta$  ;  
 Poser  $z := \tilde{\alpha}\tilde{\beta}$  ;  
 Renvoyer  $y2^n + (x - y - z)2^m + z$ .

**Exercice 7.** Soit  $K$  un corps fini. On notera  $A$  le coût d'une addition dans  $K$ , de sorte que l'on peut calculer la somme de deux polynômes de degrés respectifs  $n, m$  avec un coût égal à  $A(\min(m, n) + 1)$ .

- Supposons que  $P, Q \in K[X]$  soient deux polynômes de degré inférieur ou égal à  $2n$ . En écrivant  $P(X) = P_1(X) + P_2(X)X^n$  et  $Q(X) = Q_1(X) + Q_2(X)X^n$  avec  $P_1, P_2, Q_1, Q_2 \in K[X]$  de degrés inférieur ou égal à  $n$ , exprimer le produit  $PQ$  en fonction des polynômes  $P_1, P_2, Q_1$  et  $Q_2$ .
- En supposant que  $C(n)$  désigne le coût du calcul du produit de deux polynômes de degré au plus  $n$ , montrer que l'on peut calculer le produit  $PQ$  avec un coût d'au plus  $C(2n) = 3C(n) + 8A(n + 1)$ .
- En déduire un algorithme permettant de calculer le produit de deux polynômes de degré au plus  $n$  avec un coût de  $\mathcal{O}(n^{\log_2(3)})$  opérations.
- Montrer que la transformée de Fourier rapide peut améliorer le coût de cet algorithme en  $\mathcal{O}(n \log(n) \log \log(n))$ .

## Division euclidienne et algorithmes d'Euclide

**Exercice 8.** On suppose dans cet exercice que l'on dispose d'une table de division, ce qui nous permet de considérer que la division d'un nombre à au plus deux chiffres par un chiffre est une opération élémentaire. Notre objectif est d'effectuer la division euclidienne de  $a = (a_n \dots a_0)_b$  par  $c = (c_m \dots c_0)_b$ .

- Vérifier que l'algorithme suivant renvoie le quotient  $q$  et le reste  $r$  de la division euclidienne de  $a$  par  $c$  :  
 div(a,c)  
 poser  $x_0 = (a_n \dots a_k)_b$  avec  $k := \max\{i \geq 0 \mid x_0 \geq c\}$  ;  
 pour  $i$  de 1 à  $k + 1$  faire  $q_i := \lfloor \frac{x_{i-1}}{c} \rfloor$ ,  $r_i := x_{i-1} - q_i c$  et  $x_i := r_i b + a_{k-i}$  (sans calculer  $x_i$  pour  $i = k + 1$ ) ;  
 poser  $q := (q_1 \dots q_{k+1})_b$  et  $r := r_{k+1}$ .
- Supposons que  $c$  vérifie  $c_m \geq \lfloor \frac{b}{2} \rfloor$  et que  $a$  vérifie  $n \in \{m, m + 1\}$ . Définissons alors  $q$  comme suit :

$$q := \begin{cases} \min(b - 1, \lfloor \frac{(a_{m+1} a_m)_b}{c_m} \rfloor) & \text{si } n = m + 1 ; \\ \lfloor \frac{a_m}{c_m} \rfloor & \text{si } n = m . \end{cases}$$

Vérifier que  $\lfloor \frac{a}{c} \rfloor$  est alors égal à  $q, q - 1$  ou  $q - 2$ .

- Supposons pour simplifier que  $b$  soit une puissance de 2 (bien que le résultat reste vrai sans cette hypothèse).
  - Montrer qu'il existe  $\lambda \geq 1$  tel que  $\lambda c$  ait le même nombre de chiffres que  $c$  tout en commençant par un chiffre  $\geq \lfloor \frac{b}{2} \rfloor$ .
  - Déterminer le coût du calcul de  $\lambda c$ .
- Montrer que l'algorithme suivant permet d'effectuer la division euclidienne de  $a$  par  $c$  et déterminer son coût :  
 vraidiv(a,c)  
 déterminer le paramètre  $\lambda$  de la question précédente ;  
 calculer  $\lambda a$  et  $\lambda c$  ;  
 calculer  $(q, r) := \text{div}(\lambda a, \lambda c)$  en utilisant la méthode de la question 2 pour calculer les  $q_i$  ;  
 renvoyer  $(q, \frac{r}{\lambda})$ .

**Exercice 9.** Soient  $m, n \geq 1$  deux entiers avec  $n$  impair.

- Vérifier que l'algorithme suivant renvoie le pgcd de  $n$  et  $m$  :  
 pgcd(n,m)  
 si  $m = 1$  ou si  $m = n$  alors renvoyer  $m$  ;  
 si  $n$  est pair alors faire  $(n, m) \mapsto (\frac{n}{2}, m)$  ;  
 si  $n$  est impair et si  $n > m$  alors faire  $(n, m) \mapsto (\frac{n - m}{2}, m)$  ;

si  $n$  est impair et si  $n < m$  alors faire  $(n, m) \mapsto \left(\frac{m-n}{2}, n\right)$ .

- En admettant que la division d'un nombre à au plus deux chiffres par un chiffre est une opération élémentaire, estimer le coût de cet algorithme.

**Exercice 10.**

- Programmer l'algorithme d'Euclide classique.
- Programmer l'algorithme d'Euclide binaire.
- Comparer entre eux les temps d'exécution de ces deux algorithmes, puis les comparer au temps d'exécution de la commande incluse dans Sage. Quelle méthode utilise-t-il ?

**Exercice 11.** Si  $m$  et  $n$  sont deux entiers  $\geq 1$ , on note  $t(m, n)$  le nombre de divisions euclidiennes nécessaires pour calculer le pgcd de  $m$  et  $n$  (on commence par diviser  $m$  par  $n$  et on s'arrête lorsque le reste est nul, ainsi  $t(2, 1) = 1$  et  $t(1, 2) = 2$ ).

- Écrire une procédure calculant  $t(m, n)$ .
- Si  $(F_n)_{n \geq 0}$  est la suite de Fibonacci définie à l'exercice 4, vérifier  $t(F_{n+2}, F_{n+1}) = n$  pour tout  $1 \leq n \leq 50$ .
- Écrire une procédure qui permette de vérifier que le maximum de  $t(m, n)$  pour des entiers  $1 \leq m, n \leq N$  est obtenu pour le couple  $(F_r, F_{r+1})$  où  $r$  est le plus grand entier tel que  $F_{r+1} \leq N$ .
- On considère les fonctions  $T$  et  $\tau$  définies par

$$T(n) = \frac{1}{n} \sum_{1 \leq k \leq n} t(k, n) \quad \tau(n) = \frac{1}{\varphi(n)} \sum_{\substack{1 \leq k \leq n \\ (k, n) = 1}} t(k, n)$$

Tracer les graphes de  $T$  et  $\tau$  sur un même graphique. Que constatez-vous ? Vérifier graphiquement qu'il existe un entier  $c$  pour lequel  $0,843 \ln(n) + 1,47 + c$  est une bonne approximation de  $\tau(n)$ .

On considère la variante de l'algorithme d'Euclide dans laquelle le reste d'une division euclidienne  $m = qn + r$  vérifie  $|r| \leq n/2$ . On note  $t'(m, n)$  le nombre de pas nécessaires pour calculer le pgcd de  $m$  et  $n$  avec cette variante.

- Déterminer  $t'(F_{n+2}, F_{n+1})$ .
- Reprendre les questions 1, 3 et 4 pour la fonction  $t'$  (on notera  $T'$  et  $\tau'$  les fonctions correspondantes).
- Tracer sur un même graphique les fonctions  $\tau$  et  $\tau'$ . Que constatez-vous ?

**Exercice 12.** Soient  $a$  et  $b$  deux entiers. Posons  $a_0 := a$ ,  $a_1 := b$ ,  $u_0 := 1$ ,  $u_1 := 0$ ,  $v_0 := 0$  et  $v_1 := 1$ . Pour tout entier  $i \geq 1$ , on note  $q_i$  le quotient de la division euclidienne de  $a_{i-1}$  par  $a_i$  puis l'on pose  $a_{i+1} := a_{i-1} - q_i a_i$ ,  $u_{i+1} := u_{i-1} - q_i u_i$  et  $v_{i+1} := v_{i-1} - q_i v_i$ . On s'arrête lorsque  $a_{n+1} = 0$ .

- Montrer que  $a_n$  est égal au pgcd  $d$  de  $a$  et  $b$  et que l'on a  $u_n a + v_n b = d$ .
- Montrer que les  $u_i$  sont de signe alterné, de valeurs absolues strictement croissantes pour  $i \geq 1$ , et que  $u_{n+1} = (-1)^{n+1} \frac{b}{d}$ .
- Énoncer et démontrer les propriétés analogues pour les coefficients  $v_i$ .
- En déduire le coût du calcul d'un triplet d'entiers  $(u, v, d)$  vérifiant  $d = \text{pgcd}(a, b)$  et  $ua + vb = d$ .
- En supposant  $a$  et  $b$  premiers entre eux, estimer le coût du calcul de l'inverse de  $a$  dans  $\mathbb{Z}/b\mathbb{Z}$ .

**Exercice 13.** Soient  $a, b \geq 1$  deux entiers premiers entre eux.

- Soit  $(x, y) \in \mathbb{Z}/a\mathbb{Z} \times \mathbb{Z}/b\mathbb{Z}$ . Montrer qu'il existe un unique élément  $z \in \mathbb{Z}/ab\mathbb{Z}$  vérifiant  $z \equiv x \pmod{a}$  et  $z \equiv y \pmod{b}$ , puis estimer le coût du calcul de  $z$  en fonction des tailles de  $a$  et  $b$ .
- Montrer que toute fraction rationnelle de la forme  $\frac{N}{ab}$  s'écrit de manière unique sous la forme  $k + \frac{x}{a} + \frac{y}{b}$  avec  $k \in \mathbb{Z}$  et  $x, y \in \mathbb{N}$  vérifiant  $0 \leq x < a$  et  $0 \leq y < b$ . Estimer le coût du calcul de  $k, x$  et  $y$  en fonction des tailles respectives de  $N, a$  et  $b$ .
- Programmer ces algorithmes.

**Exercice 14.** Soit  $K$  un corps quelconque. Soient  $P, Q \in K[X]$  deux polynômes de degrés respectifs  $n, m$ . Les coûts mentionnés dans cet exercice sont à exprimer en nombre d'opérations dans  $K$ .

- Écrire un algorithme effectuant la division euclidienne de  $P$  par  $Q$  puis déterminer son coût.
- Écrire un algorithme renvoyant le pgcd de  $P$  et  $Q$ , puis déterminer son coût.
- Écrire un algorithme renvoyant le pgcd  $D$  de  $P$  et  $Q$  ainsi qu'une paire de polynômes  $(U, V)$  telle que  $UP + VQ = D$ , puis en estimer le coût.
- En supposant que  $Q$  est premier avec  $P$ , estimer le coût du calcul de l'inverse de  $Q$  dans l'anneau  $K[X]/(P)$ .

## Influence du changement de l'anneau de base

**Exercice 15.** Estimer le coût d'une addition, d'une multiplication et du calcul de l'inverse (en supposant que l'inversibilité de l'élément considéré soit déjà connue par ailleurs) dans :

1. l'anneau  $\mathbb{Z}/n\mathbb{Z}$ , en fonction de  $n$  et du coût des opérations dans  $\mathbb{Z}$  ;
2. le corps fini  $\mathbb{F}_q$ , en fonction du coût des opérations dans  $\mathbb{Z}$  ;
3. l'anneau  $\mathbb{F}_q[X]/(P)$ , avec  $P \in \mathbb{F}_q[X]$ , en fonction du degré de  $P$  et du coût des opérations dans  $\mathbb{Z}$ .

**Exercice 16.** Estimer la complexité du calcul de la somme et du produit de deux polynômes  $P, Q \in \mathbb{Z}[X]$  en fonction de leurs degrés et d'une borne uniforme  $M$  de la valeur absolue de leurs coefficients.

**Exercice 17.** Reprendre l'exercice 3 en travaillant non plus dans un corps mais dans l'anneau  $\mathbb{Z}$ . Vérifier en particulier que l'essentiel du coût du calcul est contenu dans la dernière multiplication, de sorte que l'on ne gagne pas grand chose à utiliser l'exponentiation rapide dans ce cas.

## Quelques applications de la méthode de Newton

**Exercice 18.** Soit  $I$  un intervalle de  $\mathbb{R}$  et  $f : I \rightarrow \mathbb{R}$  une fonction de classe  $\mathcal{C}^\infty$ . Pour tout élément  $x \in I$  n'annulant pas la dérivée  $f'$  de  $f$ , on pose  $g(x) := x - \frac{f(x)}{f'(x)}$ .

1. Etudier la régularité de  $g$  sur son ensemble de définition.
2. Comparer l'ensemble des zéros de la fonction  $f$  et l'ensemble des points fixes de la fonction  $g$ .
3. Supposons que  $a$  soit un zéro de  $f$  en lequel  $g$  est bien définie.
  - (a) Calculer  $g'(a)$ .
  - (b) Démontrer l'existence d'un voisinage  $U$  de  $a$  tel que pour tout élément  $x_0 \in U$ , la suite  $(x_{n+1} := g(x_n))_{n \geq 0}$  est bien définie et converge vers  $a$ .
  - (c) Supposons qu'il existe deux réels  $M$  et  $r$  strictement positifs tels que l'on ait  $2Mr < 1$  et  $|g''(x)| \leq M$  pour tout  $x \in ]a-r, a+r[$ . Montrer que la suite  $(x_n)_{n \geq 0}$  définie ci-avant vérifie  $|x_n - a| \leq (\frac{1}{2})^{2^n - 1} |x_0 - a|$  pour tout entier  $n \geq 0$ .
4. Que se passe-t-il lorsque  $a$  est un zéro commun à  $f$  et  $f'$  ?
5. Supposons que  $f$  est convexe et strictement croissante et considérons  $x_0 \in I$  vérifiant  $f(x_0) > 0$ . Montrer que si  $f$  s'annule sur  $I$ , alors la suite  $(x_n)_{n \geq 0}$  définie comme ci-avant est bien définie et converge vers le point d'annulation de  $f$  dans  $I$ .

**Exercice 19.** Soit  $P \in \mathbb{R}[X]$  un polynôme non constant de coefficient dominant strictement positif. On suppose que  $P$  est scindé dans  $\mathbb{R}$ .

1. Montrer qu'il existe un réel  $A > 0$  tel que pour tout  $x_0 > A$ , la suite de Newton  $(x_n)_{n \geq 0}$  associée à  $x_0$  et à  $P$  est bien définie et converge vers la plus grande racine réelle de  $P$ .
2. Expliciter une valeur possible de  $A$  à l'aide des coefficients de  $P$ .

**Exercice 20.** Soit  $a$  un réel strictement positif dont on souhaite approcher l'inverse.

1. A quelle fonction  $f$  peut-on appliquer la méthode de Newton pour approcher  $a^{-1}$  ?
2. Choisissons  $x_0 > 0$ . Montrer que la suite de Newton associée à  $f$  et à  $x_0$  est donnée par la relation de récurrence suivante : pour tout  $n \geq 0$ ,  $x_{n+1} = x_n(2 - ax_n)$ .
3. Pour tout  $n \geq 0$ , posons  $\varepsilon_n := ax_n - 1$ .
  - (a) Exprimer  $\varepsilon_{n+1}$  en fonction de  $\varepsilon_n$ .
  - (b) En déduire que si  $|\varepsilon_0| < 1$ , alors la suite  $(x_n)_{n \geq 0}$  converge vers  $a^{-1}$ .
4. Posons  $I := [\frac{1}{2}, 1]$ . Déterminer une paire de réels  $(\lambda, \mu)$  permettant de minimiser la quantité  $\sup_{a \in I} |\lambda - \mu a - a^{-1}|$ .

*Remarque : L'intérêt de la manœuvre est d'accélérer la convergence de la suite  $(x_n)_{n \geq 0}$  lorsque  $a$  appartient à  $I$  en choisissant  $x_0 := \lambda - \mu a$ .*

**Exercice 21.** Soient  $a$  et  $x_0$  deux réels strictement positifs.

1. Pour tout  $n \geq 0$ , on pose  $x_{n+1} := \frac{1}{2}(x_n + \frac{a}{x_n})$ . Montrer que la suite  $(x_n)_{n \geq 0}$  est bien définie et converge vers  $\sqrt{a}$ .
2. Pour tout  $n \geq 0$ , posons  $y_{n+1} := \frac{y_n}{2}(3 - ay_n^2)$  avec  $y_0 := x_0$ . Montrer que la suite  $(y_n)_{n \geq 0}$  est bien définie et converge vers  $\sqrt{a}^{-1}$ .
3. Comparer les deux méthodes ainsi obtenues pour approcher  $\sqrt{a}$  : laquelle semble la plus efficace en temps de calcul ?