

TD 1 : Analyse des Programmes

20 septembre 2023

Exercice 1

- 1) Ecrire une fonction qui prend en entrée un entier n et renvoie $n!$ à l'aide d'une boucle.
- 2) Donner un variant de boucle. Montrer que la termine.
- 3) Donner un invariant de boucle. Montrer que la fonction est correcte.

Exercice 2

On remarque que $a^p = (a^2)^{\frac{p}{2}}$ si p est pair, et $a \times (a^2)^{\lfloor \frac{p}{2} \rfloor}$ si p est impair. $\frac{p}{2}$ dans le premier cas et $\lfloor \frac{p}{2} \rfloor$ dans le deuxième cas correspondent au quotient dans la division euclidienne de p par 2.

- 1) Soit a un entier. Comment peut-on faire le calcul de a^p pour $p = 21$ avec la propriété précédente, autrement qu'en faisant le calcul $a^p = a \times a \times \dots \times a$?
- 2) Implémenter la méthode de calcul avec une fonction `exp_rapide` en C prenant deux entiers a et b en entrée et utilisant une boucle `while`.
- 3) Déterminer un variant de boucle dans l'algorithme précédent. Montrer que l'algorithme termine.
- 4) Déterminer un invariant de boucle dans l'algorithme précédent. Montrer que l'algorithme est correct.
- 5) Combien de calculs sont faits quand la fonction est appelée avec un entier p de la forme 2^k ?
- 6) Calculer l'ordre de grandeur de la complexité algorithmique de la fonction. Quelle comparaison peut-on faire avec l'exponentiation naïve ?
- 7) On utilise l'exponentiation rapide dans une fonction qui prend en entrée deux entiers a et n et calcule, à l'aide d'une boucle, la somme des a^p pour p compris entre 0 et n . Quelle est l'ordre de grandeur de la complexité de l'algorithme ?

Exercice 3

Dans un tableau ordonné, la **recherche dichotomique** d'un élément consiste à chercher un élément au milieu d'une portion de tableau restreinte, tant qu'aucun élément n'est égal à celui recherché. Au début, la portion de tableau correspond au tableau complet ; ensuite, elle correspond à chaque étape à la moitié gauche ou droite du tableau précédent. L'algorithme de recherche dichotomique se termine soit lorsqu'un élément est égal à celui recherché, soit

lorsque le dernier tableau parcouru n'a plus qu'un élément.

- 1) Ecrire en pseudo-code l'algorithme de recherche dichotomique (on suppose qu'on a une fonction pour calculer la longueur du tableau en temps constant ; on note $T[i]$ le i ème élément d'un tableau T).
- 2) Montrer que l'algorithme termine.
- 3) Montrer que l'algorithme est correct.
- 4) Calculer la complexité de l'algorithme.

Exercice 4

On considère un programme qui, pour un seuil s donné, calcule le premier terme de la suite de Fibonacci supérieur à s (cf TP 02).

- 1) Montrer que l'algorithme termine.
- 2) Montrer que l'algorithme est correct.
- 3) On admet que le n ème terme de la suite de Fibonacci vaut $F_n = \frac{1}{\sqrt{5}}(\phi^n + \phi'^n)$, où $\phi = \frac{1 + \sqrt{5}}{2}$ et $\phi' = \frac{1 - \sqrt{5}}{2}$. En déduire un majorant de la complexité algorithmique du programme.

Exercice 5

Indiquer dans chacun des cas si la majoration ou l'ordre de grandeur est vrai (en justifiant).

1. $\ln(n) = \Theta(\log_2(n))$
2. $\ln(n) = \mathcal{O}(\sqrt{n})$
3. $2 + (-1)^n = \mathcal{O}(1)$
4. $1 + (-1)^n = \Theta(1)$

Exercice 6

Avec les définitions, démontrer les propriétés suivantes vues en cours :

- \mathcal{O} est réflexive et transitive.
- Ω est réflexive et transitive.
- Θ est réflexive, transitive et symétrique.

Exercice 7

Avec les définitions, démontrer les propriétés suivantes vues en cours :

- Si $C(n) = \Theta(f(n))$ alors $kC(n) = \Theta(f(n))$ pour $k > 0$.
- Si $C(n) = \Theta(f(n))$ et $D(n) = \Theta(f(n))$ alors $C(n) + D(n) = \Theta(f(n))$.
- Si $C(n) = \Theta(f(n))$ et $D(n) = \Theta(g(n))$ alors $C(n)D(n) = \Theta(f(n)g(n))$.

- Si $C(n) = \mathcal{O}(f(n))$ et $D(n) = \mathcal{O}(f(n))$ alors $C(n) + D(n) = \mathcal{O}(f(n))$.
- Si $C(n) = \mathcal{O}(f(n))$ et $D(n) = \mathcal{O}(g(n))$ alors $C(n)D(n) = \mathcal{O}(f(n)g(n))$.
- Si $C(n) = \mathcal{O}(D(n))$ alors $C(n) + D(n) = \Theta(D(n))$.
- Si $C(n) = \Theta(f(n))$ et $D(n) = \mathcal{O}(f(n))$ alors $C(n) + D(n) = \Theta(f(n))$.

Exercice 8

Donner des contre-exemples pour invalider les affirmations suivantes :

- Si $f = \mathcal{O}(h)$ et $g = \Omega(h)$ alors $f + g = \Theta(h)$
- Si $f = \mathcal{O}(h)$ et $g = \mathcal{O}(h)$ alors $fg = \mathcal{O}(h)$

Exercice 9

Soient f et g deux fonctions de \mathbb{N} dans \mathbb{N} telles que f n'est pas majorée à un facteur constant près par g . f est-elle nécessairement minorée par g ?

Exercice 10

Soit f et g deux fonctions de \mathbb{N} dans \mathbb{N} telles que f est majorée par g mais pas minorée.

Est-ce qu'on a nécessairement $\lim_{n \rightarrow +\infty} \frac{g(n)}{f(n)} = +\infty$?