

**Partiel d'algorithmique - sujet b**

Mardi 9 novembre 2021 14h–16h / Aucun document autorisé

---

**Nom :****Prénom :****Numéro étudiant :****Groupe :**

---

**Exercice 1.** Application du Master theorem - 7 points

On considère l'algorithme **Algo** ci-dessous qui prend en argument un tableau  $T$  et deux indices  $g$  et  $d$  définissant une zone de  $T$ . L'objectif ici est de trouver la complexité de **Algo** en utilisant le Master Theorem (rappelé en annexe). On exprimera cette complexité en fonction de la taille  $n$  de la zone délimitée par  $g$  et  $d$  (inclus), c'est-à-dire de  $n = d - g + 1$ .

L'algorithme **Algo** dépend de la fonction  $F(T, g, d)$  non décrite ici. La complexité de la fonction  $F$  est aussi évaluée en fonction de la taille  $n = d - g + 1$ .

Donner la complexité de l'algorithme **Algo** dans les deux cas ci-dessous :

1. Si  $F(T, g, d)$  est de complexité  $3 \cdot n + 1$ .
2. Si  $F(T, g, d)$  est de complexité  $n^2 + 4$ .

**NB :** Dans les deux cas, on expliquera comment on applique le master Theorem : donner la valeur de  $a$ , celle de  $b$ , celle de  $c$  si besoin, indiquer le cas (parmi les trois possibles) *etc.* Bref, justifiez vos choix !

**Procédure Algo** ( $T, g, d$ )

//  $T$  : un tableau de taille  $k$  avec  $0 \leq g, d \leq k - 1$ .

**begin**

**si**  $g == d + 1$  **alors**

$T[g] = T[g] + T[d]$

**si**  $d > g + 1$  **alors**

**Algo** ( $T, g, g + \frac{d-g+1}{3} - 1$ )

**Algo** ( $T, g + \frac{d-g+1}{3}, g + 2 \cdot \frac{d-g+1}{3} - 1$ )

**Algo** ( $T, g + 2 \cdot \frac{d-g+1}{3}, d$ )

**F** ( $T, g, d$ )

---

---

---

---

---

---

---

---



- [illegible]

- 
- This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

A partir des algorithmes vus en cours (qu'il est inutile de présenter), proposer deux algorithmes efficaces qui, étant donnés un tableau  $T$  de  $n$  valeurs distinctes et un entier  $k$  avec  $1 \leq k \leq n$ , renvoie **l'ensemble** des  $k$  plus grandes valeurs de  $T$ . Donner leurs complexités.

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins or other markings on the paper.

On rappelle que le Master theorem permet d'évaluer les fonctions de la forme  $t(n) = a \cdot t(\frac{n}{b}) + f(n)$  qui apparaissent dans les algorithmes diviser-pour-régner, et où  $a$  et  $b$  sont des rationnels tels que  $a \geq 1$  et  $b > 1$ , et où  $f(n)$  est une fonction positive. On distingue trois cas :

1. Si  $f(n) = O(n^{\log_b(a)-\varepsilon})$  pour  $\varepsilon > 0$ , on a :  $t(n) = \Theta(n^{\log_b a})$ .
2. Si  $f(n) = \Theta(n^{\log_b a})$ , on a :  $t(n) = \Theta(n^{\log_b a} \cdot \log n)$ .
3. Si  $f(n) = \Omega(n^{\log_b(a)+\varepsilon})$  pour  $\varepsilon > 0$  et si il existe  $c < 1$  tel que  $a \cdot f(\frac{n}{b}) \leq c \cdot f(n)$  pour  $n$  assez grand, alors  $t(n) = \Theta(f(n))$ .

4