

# Solution série 4

Andrey Martinez Cruz

## Table des matières

<b>1</b>	<b>Exercice 1</b>	<b>3</b>
1.1	Question A . . . . .	3
1.1.1	Méthode itérative . . . . .	3
1.1.2	Méthode en arbre . . . . .	4
1.1.3	Théorème maître . . . . .	5
1.2	Question B . . . . .	5
1.2.1	Méthode itérative . . . . .	5
1.2.2	Méthode en arbre . . . . .	6
1.2.3	Théorème maître . . . . .	7
1.3	Question C . . . . .	7
1.4	Question D . . . . .	7
1.4.1	Méthode itérative . . . . .	8
1.4.2	Méthode en arbre . . . . .	9
1.4.3	Théorème maître . . . . .	9
<b>2</b>	<b>Exercice 2</b>	<b>10</b>
2.1	Question A . . . . .	10
2.2	Question B . . . . .	11
2.3	Question C . . . . .	11
2.4	Question D . . . . .	11
<b>3</b>	<b>Exercice 3</b>	<b>12</b>
3.1	Raccourci . . . . .	12
3.2	Par la définition originale du théorème maître . . . . .	13
<b>4</b>	<b>Exercice 4</b>	<b>13</b>
4.1	Question A . . . . .	13
4.2	Question B . . . . .	13
4.3	Question C . . . . .	13
4.4	Question D . . . . .	13
4.5	Question E . . . . .	13
4.6	Question F . . . . .	14
<b>5</b>	<b>Exercice 5</b>	<b>14</b>

<b>6</b>	<b>Exericse 6</b>	<b>14</b>
<b>7</b>	<b>Exerice 7</b>	<b>14</b>

## 1 Exercice 1

La résolution des équations de récurrence se fera de trois façon : itérative, en arbre et par le théorème maître.

### 1.1 Question A

$$T(n) = 7T(\frac{n}{2}) + n^3$$

#### 1.1.1 Méthode itérative

En posant,  $n = 8$ , on a le pattern suivant :

$$\begin{aligned} T(8) &= 7T(4) + f(8) \\ &= 7(7T(2) + f(4)) + f(8) \\ &= 7(7(T(1) + f(2)) + f(4)) + f(8) \\ &= 7(7(7f(1) + f(2)) + f(4)) + f(8) \\ &= 7(7^2f(1) + 7f(2)) + f(4) + f(8) \\ &= 7^3f(1) + 7^2f(2) + 7f(4) + f(8) \end{aligned}$$

En voyant ce motif, on peut formuler la sommation suivante :

$$T(n) = \sum_{i=0}^{\log_2 n} 7^i f(n^{\log_2 n - i}) \quad (1)$$

en posant  $n = 2^p$  on obtient la substitution suivante :

$$T(2^p) = \sum_{i=0}^p 7^i f(2^{p-i})$$

Maintenant résolvons cette équation :

$$\begin{aligned}
T(2^p) &= \sum_{i=0}^p 7^i f(2^{p-i}) \\
&= \sum_{i=0}^p 7^i (2^{p-i})^3 \\
&= \sum_{i=0}^p 7^i 2^{3p-3i} \\
&= 2^{3p} \sum_{i=0}^p 7^i 2^{-3i} \\
&= 2^{3p} \sum_{i=0}^p \frac{7^i}{2^{3i}} \\
&= 2^{3p} \sum_{i=0}^p \left(\frac{7}{8}\right)^i \\
&= 2^{3p} \left( \frac{1 - \left(\frac{7}{8}\right)^{p+1}}{1 - \frac{7}{8}} \right) \\
&= 2^{3p} \left( \frac{1 - \left(\frac{7}{8}\right)^{p+1}}{\frac{1}{8}} \right) \\
&= n^3 \left( \frac{1 - 2n^{\log_2 \frac{7}{8}}}{\frac{1}{8}} \right) \\
&= n^3 (8 - 16n^{\log_2 \frac{7}{8}}) \\
&= 8n^3 - 16n^{\log_2 7} \in \Theta(n^3)
\end{aligned}$$

### 1.1.2 Méthode en arbre

En dessinant l'arbre d'appel, on remarque qu'à partir du niveau 1 on fait 7 appel de  $T(n)$  dans lequel on divise par 2. Et ce processus est répétée jusqu'à atteindre  $T(1)$  qui atteint la profondeur  $\log_2 n$ . Pour chaque niveau, le travail effectif fait est de  $7^i \frac{n^3}{2^i}$  où  $i$  représente le niveau de profondeur atteint.

Donc, la sommation du travail total fait à chaque niveau est de

$$T(n) = n^{\log_2 7} + \sum_{i=0}^{\log_2 n - 1} 7^i \left( \frac{n^3}{2^{3i}} \right)$$

La résolution de cette équation de récurrence est la suivante en posant  $n = 2^p$ .

$$\begin{aligned}
T(2^p) &= 2^{p \log_2 7} + \sum_{i=0}^{p-1} 7^i \left( \frac{2^{3p}}{2^{3i}} \right) \\
&= 2^{p \log_2 7} + 2^{3p} \sum_{i=0}^{p-1} \left( \frac{7}{8} \right)^i \\
&= 2^{p \log_2 7} + 2^{3p} \left( \frac{1 - \left( \frac{7}{8} \right)^p}{1 - \frac{7}{8}} \right) \\
&= 2^{p \log_2 7} + 2^{3p} \left( \frac{1 - \left( \frac{7}{8} \right)^p}{\frac{1}{8}} \right) \\
&= n^{\log_2 7} + n^3 \left( \frac{1 - n^{\log_2 \frac{7}{8}}}{\frac{1}{8}} \right) \\
&= n^{\log_2 7} + n^3 (8 - 8n^{\log_2 \frac{7}{8}}) \\
&= n^{\log_2 7} + 8n^3 - 8n^{\log_2 7} \\
&= 8n^3 - 7n^{\log_2 7} \in \Theta(n^3)
\end{aligned}$$

### 1.1.3 Théorème maître

On a  $\alpha = 7$  et  $\beta = 2$ . Cela donne  $c = \log_\alpha \beta = \log_2 7$ . Dans notre cas, on pourrait être dans le cas 3, mais d'abord vérifions cela :

$$\alpha f\left(\frac{n}{\beta}\right) \leq cf(n), 0 < c < 1, n \geq n_0$$

$$7 \frac{n^3}{8} \leq cn^3$$

$$\frac{7}{8} n^3 \leq n^3$$

Cela est vrai pour  $c = \frac{7}{8}$  et  $n_0 = 0$ . Pour l'autre vérification, on a  $n^3 \in \Omega(n^{\log_2 7 + \epsilon})$  qui est vrai pour  $\epsilon = 0, 1$  et donc en ayant vérifié ces deux conditions,  $T(n) \in \Theta(n^3)$ .

## 1.2 Question B

$$T(n) = 4T\left(\frac{n}{3}\right) + n$$

### 1.2.1 Méthode itérative

En posant,  $n = 9$ , on a le pattern suivant :

$$\begin{aligned}
T(9) &= 4T(3) + f(9) \\
&= 4(4T(1) + f(3)) + f(9) \\
&= 4(4f(1) + f(3)) + f(9) \\
&= 16f(1) + 4f(3) + f(9)
\end{aligned}$$

En voyant ce motif, on peut formuler la sommation suivante :

$$T(n) = \sum_{i=0}^{\log_3 n} 4^i f(n^{\log_3 n - i}) \quad (2)$$

en posant  $n = 3^p$  on obtient la substitution suivante :

$$T(3^p) = \sum_{i=0}^p 4^i f(3^{p-i})$$

Maintenant résolvons cette équation :

$$\begin{aligned} T(3^p) &= \sum_{i=0}^p 4^i f(3^{p-i}) \\ &= \sum_{i=0}^p 4^i 3^{p-i} \\ &= 3^p \sum_{i=0}^p \left(\frac{4}{3}\right)^i \\ &= 3^p \left( \frac{1 - \frac{4^{p+1}}{3}}{1 - \frac{4}{3}} \right) \\ &= 3^p \left( \frac{1 - \frac{4^{p+1}}{3}}{-\frac{2}{3}} \right) \\ &= 3^p \left( \frac{1 - \frac{4^{p+1}}{3}}{-\frac{2}{3}} \right) \\ &= n \left( \frac{1 - \frac{4n^{\log_3 \frac{4}{3}}}{3}}{-\frac{2}{3}} \right) \\ &= n(-1, 5 + 2n^{\log_3 \frac{4}{3}}) \\ &= 2n^{\log_3 4} - 1, 5n \in \Theta(n^{\log_3 4}) \end{aligned}$$

### 1.2.2 Méthode en arbre

En dessinant l'arbre d'appel, on remarque qu'à partir du niveau 1 on fait 4 appel de  $T(n)$  dans lequel on divise par 3. Et ce processus est répétée jusqu'à atteindre  $T(1)$  qui atteint la profondeur  $\log_3 n$ . Pour chaque niveau, le travail effectif fait est de  $4^i \frac{n}{3^i}$  où  $i$  représente le niveau de profondeur atteint.

Donc, la sommation du travail total fait à chaque niveau est de

$$T(n) = n^{\log_3 4} + \sum_{i=0}^{\log_3 n - 1} \left(\frac{4}{3}\right)^i n$$

La résolution de cette équation de récurrence est la suivante en posant  $n = 3^p$ .

$$\begin{aligned} T(3^p) &= 3^{p \log_3 4} + \sum_{i=0}^{p-1} \left(\frac{4}{3}\right)^i 3^p \\ &= 3^{p \log_3 4} + 3^p \sum_{i=0}^{p-1} \left(\frac{4}{3}\right)^i \\ &= 3^{p \log_3 4} + 3^p \left( \frac{1 - \left(\frac{4}{3}\right)^p}{1 - \frac{4}{3}} \right) \\ &= 3^{p \log_3 4} + 3^p \left( \frac{1 - \left(\frac{4}{3}\right)^p}{-\frac{2}{3}} \right) \\ &= n^{\log_3 4} + n \left( \frac{1 - n^{\log_3 \frac{4}{3}}}{-\frac{2}{3}} \right) \\ &= n^{\log_3 4} + n \left( -\frac{3}{2} + \frac{3}{2} n^{\log_3 \frac{4}{3}} \right) \\ &= n^{\log_3 4} + \frac{3}{2} n^{\log_3 4} - \frac{3}{2} n \\ &= \frac{5}{2} n^{\log_3 4} - \frac{3}{2} n \in \Theta(n^{\log_3 4}) \end{aligned}$$

### 1.2.3 Théorème maître

On a  $\alpha = 4$  et  $\beta = 3$ . Cela donne  $c = \log_\alpha \beta = \log_3 4$ . On a  $n \in \mathcal{O}(n^{\log_3 4 - \epsilon})$  qui est vrai pour  $\epsilon = 0, 1$  et donc,  $T(n) \in \Theta(n^{\log_3 4})$ .

### 1.3 Question C

La solution se trouve dans les diapositives du dépôt. :)

### 1.4 Question D

$$T(n) = 8T\left(\frac{n}{4}\right) + n^3$$

### 1.4.1 Méthode itérative

En posant,  $n = 16$ , on a le pattern suivant :

$$\begin{aligned} T(16) &= 8T(4) + f(16) \\ &= 8(8T(1) + f(4)) + f(16) \\ &= 8(8f(1) + f(4)) + f(16) \\ &= 64f(1) + 8f(4) + f(16) \end{aligned}$$

En voyant ce motif, on peut formuler la sommation suivante :

$$T(n) = \sum_{i=0}^{\log_4 n} 8^i f(n^{\log_4 n - i}) \quad (3)$$

en posant  $n = 4^p$  on obtient la substitution suivante :

$$T(3^p) = \sum_{i=0}^p 8^i f(4^{p-i})$$

Maintenant résolvons cette équation :

$$\begin{aligned} T(3^p) &= \sum_{i=0}^p 8^i f(4^{p-i}) \\ &= \sum_{i=0}^p 8^i (4^{p-i})^3 \\ &= \sum_{i=0}^p 8^i 4^{3p-3i} \\ &= 4^{3p} \sum_{i=0}^p \left(\frac{8}{64}\right)^i \\ &= 4^{3p} \sum_{i=0}^p \left(\frac{1}{8}\right)^i \\ &= 4^{3p} \left( \frac{1 - \left(\frac{1}{8}\right)^{p+1}}{1 - \frac{1}{8}} \right) \\ &= 4^{3p} \left( \frac{1 - \left(\frac{1}{8}\right)^{p+1}}{\frac{7}{8}} \right) \\ &= n^3 \left( \frac{1 - \frac{n^{\log_4 \frac{1}{8}}}{8}}{\frac{7}{8}} \right) \\ &= n^3 \left( \frac{8}{7} - \frac{8n^{\log_4 \frac{1}{8}}}{35} \right) \\ &= \frac{8}{7}n^3 - \frac{8}{35}n^{\log_4 \frac{1}{8} + 3} \in \Theta(n^3) \end{aligned}$$



### 1.4.2 Méthode en arbre

En dessinant l'arbre d'appel, on remarque qu'à partir du niveau 1 on fait 8 appel de  $T(n)$  dans lequel on divise par 4. Et ce processus est répétée jusqu'à atteindre  $T(1)$  qui atteint la profondeur  $\log_4 n$ . Pour chaque niveau, le travail effectif fait est de  $8^i \frac{n^3}{4^{3i}}$  où  $i$  représente le niveau de profondeur atteint.

Donc, la sommation du travail total fait à chaque niveau est de

$$T(n) = n^{\log_4 8} + \sum_{i=0}^{\log_4 n - 1} 8^i \left( \frac{n^3}{4^{3i}} \right)$$

La résolution de cette équation de récurrence est la suivante en posant  $n = 4^p$ .

$$\begin{aligned} T(4^p) &= 4^{p \log_4 8} + \sum_{i=0}^{p-1} 8^i \left( \frac{4^{3p}}{4^{3i}} \right) \\ &= 4^{p \log_4 8} + 4^{3p} \sum_{i=0}^{p-1} \left( \frac{1}{8} \right)^i \\ &= 4^{p \log_4 8} + 4^{3p} \left( \frac{1 - \left( \frac{1}{8} \right)^p}{1 - \frac{1}{8}} \right) \\ &= 4^{p \log_4 8} + 4^{3p} \left( \frac{1 - \left( \frac{1}{8} \right)^p}{\frac{7}{8}} \right) \\ &= n^{\log_4 8} + n^3 \left( \frac{1 - n^{\log_4 \frac{1}{8}}}{\frac{7}{8}} \right) \\ &= n^{\log_4 8} + n^3 \left( \frac{8}{7} - \frac{8}{7} n^{\log_4 \frac{1}{8}} \right) \\ &= \frac{8}{7} n^3 + n^{\log_4 8} - \frac{8}{7} n^{-\log_4 8 + 3} \in \Theta(n^3) \end{aligned}$$

### 1.4.3 Théorème maître

On a  $\alpha = 8$  et  $\beta = 4$ . Cela donne  $c = \log_\alpha \beta = \log_4 8$ . Dans notre cas, on pourrait être dans le cas 3, mais d'abord vérifions cela :

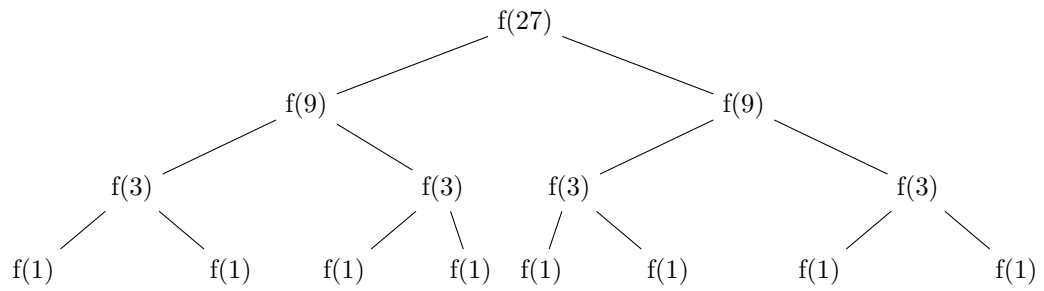
$$\alpha f\left(\frac{n}{\beta}\right) \leq c f(n), 0 < c < 1, n \geq n_0$$

$$\begin{aligned} 8 \frac{n^3}{64} &\leq c n^3 \\ \frac{1}{8} n^3 &\leq n^3 \end{aligned}$$

Cela est vrai pour  $c = \frac{1}{8}$  et  $n_0 = 0$ . Pour l'autre vérification, on a  $n^3 \in \Omega(n^{\log_4 8 + \epsilon})$  qui est vrai pour  $\epsilon = 0,1$  et donc en ayant vérifié ces deux conditions,  $T(n) \in \Theta(n^3)$ .

## 2 Exercise 2

### 2.1 Question A



## 2.2 Question B

Pour savoir le travail totale, il faut faire la sommation du travail de chaque niveau ou le niveau 0 est  $f(27)$  et le dernier niveau est  $f(1)$ .

- Niveau 0 :  $1 \times f(27) = 27$
- Niveau 1 :  $2 \times f(9) = 2 \times 9 = 18$
- Niveau 2 :  $4 \times f(3) = 4 \times 3 = 12$
- Niveau 3 :  $8 \times f(1) = 8$

Donc, la sommation du travail fait à chaque niveau est de  $27 + 18 + 12 + 8 = 65$ .

## 2.3 Question C

Ce qu'on remarque c'est qu'à chaque niveau, le travail effectué à certain niveau est le suivant :  $2^a \frac{n}{3^a}$  où  $a$  correspond au niveau de l'arbre. Donc, en posant  $n = 3^p$ , on a  $2^a \frac{3^p}{3^a} = 2^a \times 3^{p-a}$ .

Par conséquent, le travail effectué à un certain niveau peut être représenté de la manière suivante :

- Niveau 0 :  $3^p$
- Niveau 1 :  $2^1 \times 3^{p-1}$
- Niveau 2 :  $2^2 \times 3^{p-2}$
- Niveau  $i$  :  $2^i \times 3^{p-i}$

## 2.4 Question D

En généralisant cela, la charge de travail total est représentable de la manière suivante :

$$T(3^p) = \sum_{i=0}^p 2^i 3^{p-i}$$

Maintenant on peut faire une analyse classique de celle-ci pour trouver la complexité de  $T(n)$ .

$$\begin{aligned}
T(3^p) &= \sum_{i=0}^p 2^i 3^{p-i} \\
&= \sum_{i=0}^p 2^i 3^{-i} 3^p \\
&= 3^p \sum_{i=0}^p \left(\frac{2}{3}\right)^i \\
&= 3^p \left( \frac{1 - \left(\frac{2}{3}\right)^{p+1}}{1 - \frac{2}{3}} \right) \\
&= 3^p \left( \frac{1 - \left(\frac{2}{3}\right)^{p+1}}{\frac{1}{3}} \right) \\
&= n \left( \frac{1 - \frac{2}{3} n^{\log_3 \frac{2}{3}}}{\frac{1}{3}} \right) \\
&= n(3 - 2n^{\log_3 \frac{2}{3}}) \\
&= 3n - 2n^{\log_3 2} \in \Theta(n)
\end{aligned}$$

Note : le travail totale fait au dernier niveau est égale à  $n^{\log_3 2}$  et le dernier niveau est  $\log_3 n$ . Il était possible d'arriver au même conclusion on utilisant la sommation suivante substitué en  $3^p$  :

$$T(3^p) = 3^{p \log_3 2} + \sum_{i=0}^{p-1} 2^i 3^{p-i}$$

### 3 Exercise 3

Avant de trouver un  $c$  telle que  $T'_a \in o(T_a)$ , il faut d'abord trouver la complexité de  $T_a$ .

En utilisant le théorème maître pour trouver la complexité de  $T_A$ , on  $\alpha = 7$ ,  $\beta = 2$  et  $\lambda = 2$ . On  $c = \log_\alpha \beta = \log_2 7$ .

#### 3.1 Raccourci

Ici  $c > \lambda$ , donc, on est dans le premier cas du théorème maître et donc,  $T_A \in \Theta(n^{\log_2 7})$ .

### 3.2 Par la définition originale du théorème maître

Dans ce cas,  $n^2 \in \mathcal{O}(n^{\log_2 7 - \epsilon})$  est vrai avec  $\epsilon = 0,1$  et donc, on est dans le premier cas du théorème maître ce qui conclut que  $T(n) \in \Theta(n^{\log_2 7})$ .

Maintenant, dans le cas de  $T_{A'}$ , on aura la valeur de  $\log_\alpha \beta$  suivante :  $\log_4 c$ . En revanche,  $\lambda$  est la même valeur que dans  $T_A$ . Ici, ce qu'on peut remarquer c'est que  $\log_2 7 \approx 2,8074$ . Et donc, pour trouver le plus gros  $c$  possible qui fait que  $T'_a \in o(T_a)$  est vraie, il faut que  $T_{a'}$  se retrouve dans le premier cas du théorème maître. On remarque que si  $c = 16$ , on tomberais dans le deuxième cas du théorème ( $\log_4 16 = 2$ ) et grossir un peu plus  $c$  nous rapprocherai de la valeur  $\log_2 7$ . Donc,  $c$  est au moins supérieur à 16. Pour trouver la plus grande valeur de  $c$  possible qui rend la complexité citée précédemment vraie, il faut trouver le point d'intersection entre  $\log_4 c$  et  $\log_2 7$  de la manière suivante :

$$\begin{aligned}\log_4 c &= \log_2 7 \\ c &= 4^{\log_2 7} \\ c &\approx 4^{2,8074} \\ c &\approx 49\end{aligned}$$

Or,  $\log_4 49 \approx 2,8074$ . Donc, pour prendre la plus grande valeur possible qui fait que la complexité précisée précédemment soit vrai, il faut prendre le prédécesseur de 49 qui est 48.  $\log_4 48 \approx 2,7925 < 2,8074$  respecte notre contrainte. Donc,  $c = 48$ .

## 4 Exercice 4

### 4.1 Question A

$$T(n) \in \Theta(\log n)$$

### 4.2 Question B

$$T(n) \in \Theta(n)$$

### 4.3 Question C

$$T(n) \in \Theta(n!)$$

### 4.4 Question D

$$T(n) \in \Theta(n^2 \log n)$$

### 4.5 Question E

$$T(n) \in \Theta(n)$$

#### 4.6 Question F

$$T(n) \in \Theta(n^2 \log^3 n)$$

### 5 Exercice 5

En posant  $p = 2$  en à la le développement :

$$\begin{aligned} T(2^{2^2}) &= T(16) = T(\sqrt{16}) + f(16) \\ &= T(\sqrt{4}) + f(4) + f(16) \\ &= f(1) + f(4) + f(16) \end{aligned}$$

En essayant de développer plus en détail, on remarque le motif suivant :

$$T(2^{2^p}) = T(2^{2^{p-1}}) + 1 = T(2^{2^{p-2}}) + 2 = T(2^{2^{p-i}}) + i = \dots = T(2^{2^0}) + p = T(2) + p$$

Or,  $p = \log \log n$  et donc  $T(n) \in \Theta(\log \log n)$

### 6 Exercice 6

1. **Procédure** TRANSFORMER( $x, b, P$ )
2.     Si  $x \geq b$  alors
3.         empiler( $P, x \bmod b$ )
4.      $x \leftarrow x \div b$
5.     transformer( $x, b, P$ )
6.     Fin Si alors
7.     empiler( $P, x$ )
8. Fin Procédure

Complexité temporelle :  $T(n) \in \Theta(\log_b n)$

### 7 Exercice 7

Les fonctions utilisées pour les liste chaînées sont les suivantes :

- estVide : Vérifie si la liste chaînée est vide  $\Theta(1)$
- obtenirTete : Obtient l'élément en tête de la liste chaînée  $\Theta(1)$
- inserer : Inserer l'élément au début de la liste  $\Theta(1)$
- retirerDebut : Retire l'élément en tête de la liste chaînée  $\Theta(1)$

1. **Fonction** RENVERSER( $L_1, L_2$ )
2.     Si estVide( $L_1$ ) alors
3.         Renvoyer  $L_2$
4.     element  $\leftarrow$  obtenirTete( $L_1$ )
5.     insérer( $L_2$ , element)
6.     retirerDebut( $L_1$ )
7.     Renvoyer Renverser( $L_1, L_2$ )
8. Fin Fonction

La complexité de cette fonction peut être représentée ainsi :

$$T(n) = T(n - 1) + 1 \in \Theta(n)$$