Fiche de TD4 : diviser pour régner

- Exo1 -

Sur ALGO1 :

* **Equation de récurrence** : on note $t(n)$ le temps mis par l'algo sur une entrée n.

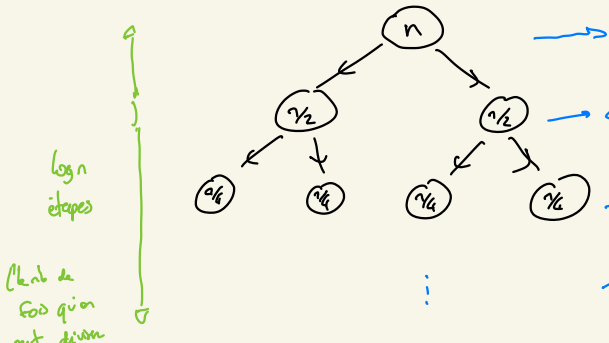
$t(1) = O(1)$

Pour $n > 1$ $\quad t(n) = 2\,t(\lceil \frac{n}{2} \rceil) + O(n^2)$

(la complexité hors des appels récursifs (2 boucles imbriquées))

* **Estimation par arbre des appels récursifs** :

$t(n) \leq 2 \cdot t(\lceil \frac{n}{2} \rceil) + c \cdot n^2$



$\rightsquigarrow c \cdot n^2$ (où c majore la somme des op. constante qui apparaissent dans l'algo).

$\rightsquigarrow 2 \times c\left(\frac{n}{2}\right)^2$ $\quad c\left(\frac{n}{2}\right)^2$: tps lors appel rec lorsque l'algo avec $\frac{n}{2}$ comme paramètre

$\rightsquigarrow 4 \times c\left(\frac{n}{4}\right)^2$

$\rightsquigarrow 8 \times c\left(\frac{n}{8}\right)^2$

Somme : $t(n) \simeq \sum_{i=0}^{\log n} 2^i\, c\left(\frac{n}{2^i}\right)^2$

$$t(n) = cn^2 \sum_{i=0}^{\log n} 2^i \left(\frac{1}{2^i}\right)^2$$

$$= cn^2 \sum_{i=0}^{\log n} \frac{1}{2^i} = cn^2 \, \frac{1 - \left(\frac{1}{2}\right)^{\log n + 1}}{1 - \frac{1}{2}} = cn^2 \, \frac{1 - \left(\frac{1}{2}\right)^{\log n + 1}}{\frac{1}{2}}$$

$$= 2cn^2 \left(1 - \left(\frac{1}{2}\right)^{\log n + 1}\right) \leq 2cn^2.$$

On devrait avoir $t(n) = O(n^2)$.

* **Master Theorem** :

$t(n) \leq 2\,t(\lceil \frac{n}{2} \rceil) + O(n^d)$ $\quad b^d = 2^2 = 4 > a = 2$ donc

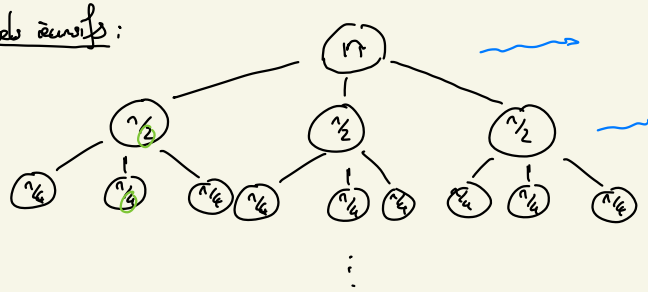
(a = 2, b = 2)

$t(n) = O(n^d) = O(n^2)$.

sur ALGO2 : $t(n)$ le temps mis par Calgo sur l'entrée n.

* éq. de récurrence : $\begin{cases} t(1) = O(1) \\ t(n) = 3\,t(\lceil n/2 \rceil) + O(n). \end{cases}$

* arbre des appels récursifs :



hauteur $\log n$

(log en base 2 : nbre de fois qu'on divise n par 2)

$c.n$

$3(c.\frac{n}{2})$

$9(c.\frac{n}{4})$

$2^t(c.\frac{n}{2^t})$

Somme.

Somme : $t(n) = \sum_{i=0}^{\log n} 3^i \, c\,\frac{n}{2^i} = cn \sum_{i=0}^{\log n} \left(\frac{3}{2}\right)^i$

$= cn\,\frac{1-\left(\frac{3}{2}\right)^{\log n+1}}{1-\frac{3}{2}} = cn\,\frac{\left(\frac{3}{2}\right)^{\log n+1}-1}{\frac{3}{2}-1} = 2cn\left(\left(\frac{3}{2}\right)^{\log n+1}-1\right).$

$\leq 2cn\left(\frac{3}{2}\right)^{\log n+1} = 2.\frac{3}{2}cn\left(\frac{3}{2}\right)^{\log n} = 3cn\left(\frac{3}{2}\right)^{\log n}$

$\leq 3cn\left(2^{\log\frac{3}{2}}\right)^{\log n} = 3c.n.(2^{\log n})^{\log\frac{3}{2}} = 3c.n.n^{\log\frac{3}{2}}$

$\leq 3.c.n^{\log(\frac{3}{2})+1} = \boxed{3.c.n^{\log 3}}$

$\left[\left(\log\tfrac{3}{2}\right)+1 = \log 3 - \log 2 + 1 = \log 3 - 1 + 1 = \log 3\right]$

* Master Theorem : $t(n) \leq \underset{a}{3}\, t(\lceil \tfrac{n}{2} \rceil) + \underset{d}{O(n)}$ $\quad b^d = 2^1 < a = 3$

(b : $t(\lceil \frac{n}{2} \rceil)$)

Par le théo $t(n) = O\left(n^{\frac{\log a}{\log b}}\right) = O(n^{\log 3})$

Exercice 2 -

ALGOA : $t(n)$: temps mis par l'algo sur l'entrée n

$$t(n) \leq 5\, t(\lceil \tfrac{n}{2} \rceil) + O(n)$$

Master Theo : $a = 5$, $b = 2$ et $d = 1$ $\quad b^d = 2^1 < 5 = a$

$$t(n) = O(n^{\log 5 / \log 2}) \quad \text{(cas 3 du Master Theo)}$$

$$\boxed{t(n) = O(n^{\log 5})} \quad (\log 5 \sim 2{,}32)$$

ALGOB : $t(n) \leq 2\, t(n-1) + O(1)$

Master Theo ? $a = 2$ $b = 1$ $d = 0$ $\quad b^d = 1 < 2 = a$

Cas 3 du M.T : $O(n^{\frac{\log 2}{\log 1}}) = O(n^{\frac{1}{0}})$!

On ne peut pas appliquer le Master Theo car b doit être > 1

Du coup, il faut se débrouiller ... A la main :

On dit que $O(1)$ est une constante $c > 0$ et :

$$t(n) \leq 2\, t(n-1) + c \leq 2\,(2\, t(n-2) + c) + c = 4\, t(n-2) + 2c + c$$
$$\leq 4\,(2\, t(n-3) + c) + 2c + c = 8\, t(n-3) + 4c + 2c + c$$
$$\leq 8\,(2\, t(n-4) + c) + 4c + 2c + c = 16\, t(n-4) + 8c + 4c + 2c + c$$

Par récurrence, on montre que $t(n) \leq 2^i\, t(n-i) + (2^{i-1} + 2^{i-2} + \dots + 1)c$ $\quad i \geq 1$

P_1 est vraie : $t(n) \leq 2\, t(n-1) + c$

Si P_i est vraie :

$$t(n) \leq 2^i\, t(n-i) + (2^{i-1} + 2^{i-2} + \dots + 1)c$$
$$\leq 2^i\,(2\, t(n-i-1) + c) + (2^{i-1} + 2^{i-2} + \dots + 1)c \quad (P_i)$$
$$\leq 2^{i+1}\, t(n-i-1) + 2^i c + (2^{i-1} + 2^{i-2} + \dots + 1)c$$
$$\leq 2^{i+1}\, t(n-i-1) + (2^i + 2^{i-1} + 2^{i-2} + \dots + 1)c$$

P_{i+1} est vraie.

Du coup pour $i = n$: $\quad t(n) \leq 2^n\, t(0) + (2^{n-1} + \dots + 1)c$

$$\leq 2^n\, t(0) + (2^n - 1)c$$
$$\leq 2^n\,(t(0) + c).$$

$$\boxed{t(n) = O(2^n).}$$

ALGOC : $t(n) \leq 9\, t(\lceil \tfrac{n}{3} \rceil) + O(n^2)$ $\quad$ MT : $a = 9$, $b = 3$, $d = 2$ $\quad b^d = 9 = a$

$$\boxed{t(n) = O(n^2 \log n)}$$

Comme $2 < \log 5$ $\quad \frac{n^2 \log n}{n^{\log 5}} = \frac{\log n}{n^{\log 5 - 2}} \xrightarrow[n \to +\infty]{} 0$ donc asymptotiquement

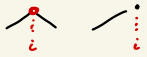
$O(n^2 \log n)$ est plus petit que $O(n^{\log 5})$. **le meilleur est ALGO C**.

Exercice 4.

Q.1 : $[1,2,3,0]$ ou $[1,3,2,0]$ ou $[4,3,2,1]$ ou $[1,2,3,4]$

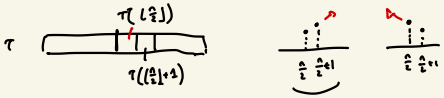
Q.2 : Une solution :

```
i ← 0 ;
Tant que ( i < n-1 et T[i] < T[i+1])
    i++ ;
Retourner i
```



Algo en $O(n)$

Q.3 :




```
PICRECURSIF (T).
n ← taille de T ;
si (n=1)
    retourner T[0] ;
m ← ⌈n/2⌉ - 1 ;          n-⌈n/2⌉ = ⌊n/2⌋
si T[m+1] > T[m]
    retourner PIC-RECURSIF ( T[m+1, ..., n-1]) ;
sinon
    retourner PIC-RECURSIF ( T[0, ..., m]) ;
                            ⌈n/2⌉
```

(si $n > 1$ $\lceil \frac{n}{2} \rceil \geq 1$ et $m \geq 0$)

Complexité : $t(n)$ sur un tableau de taille n :

$t(n) \leq t(\lceil \frac{n}{2} \rceil) + O(1)$.

Master Theo : $a = 1$, $b = 2$, $d = 0$

$b^d = 2^0 = 1 = a$

$t(n) = O(n^d \log n) = \boxed{O(\log n)}$

Exercice 5 -

① : trier T : $O(n \log n)$

récupérer $T[n-p], T[n-p+1] \dots T[n-1] \rightsquigarrow O(p)$.

En tout $O(n \log n) + O(p) = \boxed{O(n \log n)}$ (car $p \leq n$)

② Construction du tas : $O(n \log n)$ [en fait, on peut même faire $O(n)$]

Extraire max : $O(\log n)$

En tout : $O(n \log n) + p\, O(\log n) = \boxed{O(n \log n)}$ (ou $O(n + p \log n)$)

③ On fait p calculs de rang le $n-p+1^{ième}$, $n-p+2^{ième}$ ... $n^{ième}$

En tout $pO(n) = \boxed{O(pn)}$

④ Calcul du $n-p^{ième}$ rang $\rightarrow O(n)$ On note x l'élément correspondant.

On récupère dans T les p elts $\geq x$: $O(n)$ On le stocke dans T'

Trier T'. $O(p \log p)$ $\boxed{O(n) + O(p \log p)}$ ← la meilleure stratégie

---

Ex 3 et 6...