EA4 – Éléments d'algorithmique
TD n° 3 : complexité et récursivité

Exercice 1 : classement

Classer les fonctions suivantes en fonction de leur ordre de grandeur dans les classes Θ_1 à Θ_7 : les fonctions appartiennent à la même classe Θ_i si et seulement si elles sont du même ordre de grandeur, et les classes Θ_i sont rangées en ordre croissant.
Liste des fonctions à traiter (où log désigne le logarithme en base 2) :

$$n^2 + n^4, \quad n^2 + 4^n, \quad n^2 \times 4^n, \quad n^2 + \log n, \quad n^2 + \log(4^n),$$

$$\log(\sqrt{n}), \quad \log(n^4), \quad (\log n)^4, \quad 4^{n-1}, \quad 4^{2n}, \quad 2^{2n}, \quad 4^{\log n}$$

Θ_1	Θ_2	Θ_3	Θ_4	Θ_5	Θ_6	Θ_7

Exercice 2 : algorithme de Karatsuba

Dérouler à la main l'algorithme de Karatsuba vu en cours pour le calcul du produit 3210×1203. Tracer l'arbre des appels récursifs et prendre le produit de chiffres décimaux comme cas de base.

Exercice 3 : complexité d'algorithmes récursifs

Pour chacun des algorithmes `somme_i_` suivants, donner une relation de récurrence satisfaite par le nombre $A_i(n)$ d'additions effectuées pour une entrée de taille n, et en déduire l'ordre de grandeur de $A_i(n)$.

```
def somme_1_(T) :
  return 0 if len(T) == 0 else somme_1_(T[2:]) + 1

def somme_2_(T) :
  m = len(T)//2
  return 0 if len(T) == 0 else somme_2_(T[:m]) + 1

def somme_3_(T) :
  return 0 if len(T) == 0 else somme_1_(T) + somme_3_(T[1:])

def somme_4_(T) :
  m = len(T)//2
  return 0 if len(T) <= 1 else somme_4_(T[:m]) + somme_4_(T[m:])
```

Exercice 4 : complexité de l'algorithme de Karatsuba

Soit respectivement $M(n)$ et $A(n)$ les nombres de multiplications et additions élémentaires effectuées lors de l'exécution de l'algorithme de Karatsuba sur deux polynômes représentés par des tableaux de taille n (donc de polynômes de degré au plus $n-1$). Par souci de simplification, on ne s'intéresse ici qu'au cas où n est une puissance de 2.

1. Étude de $M(n)$

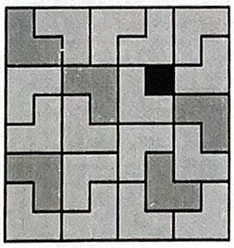
- **a.** Déterminer la relation de récurrence satisfaite par $M(n)$.
- **b.** En déduire une forme close simple pour $M(2^k)$ (en fonction de k).
- **c.** Justifier que, pour tous b et n, $b^{log_2 n} = n^{log_2 b}$, puis en déduire une forme close simple pour $M(n)$ (en fonction de n).

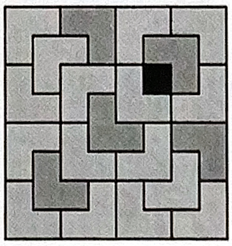
2. Étude de $A(n)$

- **a.** Exprimer le nombre $a(n)$ d'additions élémentaires effectuées par l'algorithme de Karatsuba sur deux entrées de taille n *en dehors des trois appels récursifs.*
- **b.** En considérant successivement chacun des étages de l'arbre de récursion de l'algorithme de Karatsuba sur des entrées de taille 2^k, exprimer $A(2^k)$ en fonction des $a(2^h)$ avec $h \leqslant k$.
- **c.** En déduire que $A(n) \in \Theta(n^{log_2 3})$.

Exercice 5 : pavage avec des L

On considère une grille carrée de côté n dans laquelle une case i, j est « interdite » (avec $1 \leqslant i \leqslant n$ et $1 \leqslant j \leqslant n$). On suppose que n est une puissance de 2 ($n = 2^k$ pour un certain $k \geqslant 1$).
On veut paver la grille (excepté la case interdite) avec des tuiles de trois cases en forme de L, orientées dans n'importe quel sens.
Voici deux solutions, pour une grille de côté $8 = 2^3$ dont la case interdite est la case noircie :





Proposer une méthode basée sur la stratégie « diviser pour régner » en ramenant la résolution du problème de départ à la résolution de plusieurs sous-problèmes analogues sur des données de taille inférieure.
Évaluer ensuite le nombre d'appels récursif effectués, d'abord en fonction de k et ensuite en fonction de n.

Ex 1 :

O : $<$ $\quad$ Θ : $\Leftrightarrow$

Ω : $>$

Θ_i	Θ_{i+1}
$f(n), g(n)$	$h(n)$

$f(n) = \Theta(g(n))$

$f(n) = O(h(n))$

$$f(n) = O(h(n)) \Leftrightarrow \exists c, n_0, \forall n \geq n_0,$$

$$f(n) \leq c h(n)$$

$$\Leftrightarrow \exists c < +\infty \quad \lim_{n \to +\infty} \frac{f(n)}{h(n)} = c$$

c : constante

$$a^{bn} = (a^b)^n$$

$$\log(a^b) = b \log(a)$$

$n^2 + n^4$, $\quad n^2 + 4^n$, $\quad n^2 \times 4^n$, $\quad n^2 + \log n$, $\quad n^2 + \log(4^n)$,

$\log(\sqrt{n})$, $\quad \log(n^4)$, $\quad (\log n)^4$, $\quad 4^{n-1}$, $\quad 4^{2n}$, $\quad 2^{2n}$, $\quad 4^{\log n}$

	Θ_1	Θ_2	Θ_3	Θ_4	Θ_5	Θ_6	Θ_7
$\frac{1}{2}\log n$	$= \log(\sqrt{n})$ $= \log(n^{\frac{1}{2}})$ $= \frac{1}{2}\log(n)$ $\log(n^4)$	$(\log n)^4$ n^4 $(\log n)^b = O(n^c)$ $\forall b, c > 0$	$4^{\log n} = n^2$ $n^2 + \log(4^n)$ $n^2 + \log(n)$	$n^2 + n^4$	$n^2 + 4^n$ 4^{n-1} $2^{2n} = 4^n$	$n^2 4^n$	4^{2n}

$\log(n^4)$

$= 4\log(n)$

$4^{\log n}$

$(2^2)^{\log n}$

$= 2^{2\log n}$

$= (2^{\log n})^2$

$= n^2$

$(2^{\log n})^2 = 2$

$(a^b)^c = a^{bc} = (a^c)^b$

$n^2 + \log(4^n)$

$= n^2 + n\log_2 4 = 2$

$= 2^{2n} = (2^2)^n = 4^n$

$4^{n-1} = \frac{4^2}{4}$

$4^{n-1} = \Theta(n^2 + 4^n)$

$\frac{n^2+4^n}{4^{n-1}}$

$= \frac{n^2+4^n}{\frac{4^n}{4}}$

$= \frac{4n^2}{4^n} + 4$

$\xrightarrow[n \to +\infty]{} 4$

$f(n) = n^2 + \log(4^n) = n^2 + 2n$

$g(n) = 4^{\log n} = n^2$

$\lim \frac{f(n)}{g(n)} = \frac{n^2+2n}{n^2} = 1 + \frac{2}{2} \xrightarrow[+\infty]{} 1$

$f(n) = 4^n$

$g(n) = n^2 4^n$

$\frac{f(n)}{g(n)} = \frac{4^n}{n^2 4^n} = \frac{1}{n^2} \xrightarrow[n \to +\infty]{} 0$

$\frac{n^2 4^n}{4^{2n}} = \frac{n^2 4^n}{4^n 4^n} = \frac{n^2}{4^n} \xrightarrow[n \to +\infty]{} 0$

$$\frac{f(n)}{g(n)} \xrightarrow[n+\infty]{} \begin{cases} +\infty \Rightarrow g(n) = O(f(n)) \\ c \neq 0,\ c \neq \infty \Rightarrow f(n) = \Theta(g(n)) \\ 0 \Rightarrow f(n) = O(g(n)) \end{cases}$$

Algorithme de Karatsube

$$a = \underbrace{13578}_{a_1}\underbrace{43152}_{a_2}$$

$$= a_{1n} \times 10^5 + a_2$$

$$b = \underbrace{98243}_{b_1}\underbrace{15798}_{b_2} = b_1 \times 10^5 + b_2$$

$$a \times b = (a_1 \times 10^5 + a_2)(b_1 \times 10^5 + b_2)$$

10 chiffres

$$= a_1 b_1 \times 10^{10} + (a_1 b_2 + b_1 a_2)10^5 + a_2 b_2$$

5 chiffres

$$= a_1b_1 \times 10^{10} + (z_1 - a_1b_1 - a_2b_2)\,10^5 + a_2b_2$$

$$z_1 = (a_1+a_2)(b_1+b_2)$$

$$= a_1b_1 + a_1b_2 + a_2b_1 + a_2b_2$$

Ex 2

$$a = \overbrace{32}^{a_1}\overbrace{10}^{a_2}$$

$$b = \underbrace{12}_{b_1}\underbrace{03}_{b_2}$$

a_1, b_1 $\qquad$ a_2, b_2 $\qquad$ a_1+a_2, b_1+b_1

```
def mult(a,b):
    a1, a2 = sep(a)        -> sep : séparer
    b1, b2 = sep(b)
    P1 = mult(a1, b1)
    P2 = mult(a2, b2)
    Z = mult(a1 + a2, b1 + b2)
```

et mul (7,6)

$32 + 10 = 42$

$12 + 03 = 15$

puis $4 + 2 = 7$

$1 + 5 = 6$

return $P_1 * 10^n + (Z - P_1 - P_2) \times 10^{\frac{n}{2}} + P_2$

$$a = \overline{32}|\overline{10} = a_1^{3200} \times 10^2 + a_2^{10}$$

$$b = 1203 = b_1 \times 10^2 + b_2$$

$$\begin{aligned}
a \times b &= (a_1 \times 10^2 + a_2)(b_1 \times 10^2 + b_2) \\
&= a_1 b_1 \times 10^4 + (a_1 b_2 + b_1 a_2) 10^2 + a_2 b_2 \\
&= a_1 b_1 \times 10^4 + (z - a_1 b_1 - a_2 b_2) 10^2 + a_2 b_2
\end{aligned}$$

$$\begin{aligned}
z_1 &= (a_1 + a_2)(b_1 + b_2) \\
&= a_1 b_1 + a_1 b_2 + a_2 b_1 + a_2 b_2
\end{aligned}$$

3210×1203

P_1: $a_1 b_1$

$32 \times 12 = 384$

- $3 \times 2 \to 6$
- $1 \times 2 \to 2$
- $(3+2)(1+2) \to 15$

P_2: $a_2 b_2$

$10 \times 03 = 30$

- $1 \times 0 \to 0$
- $0 \times 3 \to 0$
- $(1+0)(0+3) \to 3$

Z: $(a_1 + a_2)(b_1 + b_2)$

$(32 + 10)(12 + 03) = 42 \times 15$

- $4 \times 1 \to 4$
- $2 \times 5 \to 10$
- $(4+2)(1+5) \to 36$

3210×1203

$$
\begin{array}{r}
3840000 \\
+ \quad 21600 \\
+ \quad 30 \\
\hline
3861630
\end{array}
$$

Karatsuba : $O(n^{\log_2 3})$

Basique : $O(n^2)$

FFT : $O(n \log n)$

Ex3 : $n = len(T)$

```
def somme_1_(T) :
  return 0 if len(T) == 0 else somme_1_(T[2:]) + 1
```

$T[i] \rightarrow i^e$ valeur

$T[a:] \rightarrow [T[a], T[a+1], \dots, T[b-1]]$

$$\begin{array}{cccccc} & 0 & 1 & 2 & & \\ T = [& 9, & 5, & 8, & 9, & 3] \end{array}$$

$T[2:] = [8, 9, 3]$

$$A_1(n) = \begin{cases} 0 \text{ si } n = 0 \\ 1 + A_1(n-2) \end{cases}$$

$$A_2(n) = \begin{cases} 0 \text{ si } n = 0 \\ 1 + A_2(n//2) \end{cases}$$

$$A_3(n) = \begin{cases} 0 \text{ si } n = 0 \\ 1 + A_1(n) + A_3(n-1) \text{ sinon} \end{cases}$$

$$A_4(n) \begin{cases} 0 \text{ si } n \leq 0 \\ 1 + A_4(n//2) + A_4(\lceil \frac{n}{2} \rceil) \end{cases}$$

09/02

Ex 3 :

```
def somme_1_(T) :
    return 0 if len(T) == 0 else somme_1_(T[2:])
```

+ 8 ou peu importe

$$A_1(n) = \begin{cases} 0 & \text{si } n \leq 0 \\ 1 + A_1(n-2) \end{cases}$$

```
def somme_2_(T) :
    m = len(T) // 2
    return 0 if len(T) == 0 else somme_2_(T[:m]) + 1
```

$$A_2(n) = \begin{cases} 0 & \text{si } n \leq 0 \\ 1 + A_2(n // 2) \end{cases} \rightarrow m = \mathrm{Len}(T) // 2$$

```
def somme_3_(T) :
    return 0 if len(T) == 0 else somme_1_(T) + somme_3_(T[1:])
```

$$A_3(n) = \begin{cases} 0 & \text{si } n \leq 0 \\ 1 + A_1(n) + A_3(n-1) \end{cases}$$

```
def somme_4_(T):
  m = len(T)//2
  return 0 if len(T) <= 1 else somme_4_(T[:m])
                               + somme_4_(T[m:])
```

:m : les éléments jusqu'à m inclus

m: : les éléments sauf ceux d'avant m et m exclus

$$A_4(n) \begin{cases} 0 \text{ si } n \leq 0 \\ 1 + A_4\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + A_4\left(\left\lceil \frac{n}{2} \right\rceil\right) \end{cases}$$

→ il additionne les somme_4_(T[:m]) somme_4_(T[m:])

$n = 3$

$$m = \left\lfloor \frac{n}{2} \right\rfloor + \left\lceil \frac{n}{2} \right\rceil = n$$

$\left\lfloor \frac{n}{2} \right\rfloor$: inférieur ou égale à $\frac{n}{2}$

$\left\lceil \frac{n}{2} \right\rceil$: supérieur ou égale

$$n/2 = \left\lfloor \frac{n}{2} \right\rfloor$$

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

Exemple

$n = 10$

$A_1(10) = 1 + \underbrace{A_1(8)}_{1 + \underbrace{A_1(6)}_{1 + \underbrace{A_1(4)}_{1 + \underbrace{A_1(2)}_{1 + A_1(0)}}}}$

10
9
8
7
6
5
4
3
2
1
0

$$A_1(n) \in \Theta(n)$$

$$P(n) : A_1(n) \leq C_n$$
$$\text{et } A_1(n) \geq d_n$$

$A_1(n) = \frac{1}{2} n$ Si n est pair ;

$\frac{n-1}{2}$ sinon

$$T(n) = \begin{cases} 0 & \text{si } n = 0 \\ a\, T\left(\frac{n}{b}\right) + O(n^c) & \end{cases}$$

$a = 1$, $b = 2$, $c = 0$

$$A_2(2^k) = 1 + A_2(2^{k-1})$$
$$= 1 + 1 + A_2(2^{k-2})$$
$$\vdots$$
$$= k + A_2(2^0)$$
$$= k + 1$$

$\uparrow$ n

$\log(n)$

$$A_2(1) = 1$$

$$n = 2^k$$
$$\Rightarrow k = \lceil \log_2(n) \rceil$$

$$A_3(n) = \underbrace{A_1(n)} + \underbrace{A_3(n-1) + 1}$$

$$= n + 1 + \overbrace{A_1(n-1) + A_3(n-2) + 1}$$
$$= n + n-1 + A_1(n-2) + A_3(n-3) + 1 + 1 + 1$$
$$\vdots$$
$$= n + n-1 + n-2 + \dots + 1 + A_3(0) + \overbrace{1+1+\dots+1}^{n \text{ fois}}$$
$$= \frac{n(n+1)}{2} + n$$
$$= \frac{n^2+n}{2} + n$$

$$S = n + n-1 + (n-2) + \dots + 1$$

$$+$$

$$S = 1 + 2 + 3 + \dots + n$$

$$2S = \underbrace{(n+1) + (n+2) + (n+1) \dots + n+1}_{n \text{ termes}}$$

$$2S = n(n+1)$$

$$\Rightarrow S = \frac{n(n+2)}{2}$$

Pour $A_4(n)$

$$A_4(n) = n-1$$

$$A_4(n) = A_4\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + A_4\left(\left\lceil \frac{n}{2} \right\rceil\right) + 1$$

$$= \left\lfloor \frac{n}{2} \right\rfloor - \cancel{1} + \left\lceil \frac{n}{2} \right\rceil - 1 + \cancel{1}$$

$$= n - 1$$

Algorithme de Karatsuba

$$a = a_1 + 2^m + a_2$$

$$b = b_1 \times 2^m + b_2$$

Récursion

$$\left(\begin{array}{l} z = (a_1 + a_2)(b_1 + b_2) \\ P_1 = a_1 b_1 \\ P_2 = a_2 b_2 \end{array}\right.$$

res : $P_1 2^n + (z - P_1 - P_2) 2^m + P_2$

Ex 4

2) $A(n) = \begin{cases} 0 & \text{si } n \leq 1 \\ 4 \times \frac{n}{2} + 3A\left(\frac{n}{2}\right) \end{cases}$

$$A(n) = 6 + 3A\left(\frac{n}{2}\right)$$

$$a = \underbrace{1 + 2x}_{a_2} + \underbrace{4x^2 + 8x^3}_{x^2 a_1}$$

$\hookrightarrow 4 + 8x$

$a_1 + a_2 = 5 + 10X$

$a_1 + a_2 X^t$

$n = 2^k$

$A(2^n) = 2^{k+1} + 3A(2^{k-1})$

$= 2^{k+1} + 3(2^k + 3A(2^{k-2}))$

$0a$

$= 2n + 3A(2^{k-1})$

$= 2n + 3(2\frac{n}{2} + 3A(2^{k-2}))$

$= 2n + 3 \cdot \frac{2n}{2} + 9(2\frac{n}{4} + A(2^{k-3}))$

$= 2n(1 + \frac{3}{2} + \frac{9}{4} + \frac{3^3}{2^3} + \cdots + (\frac{3}{2})^k)$

$= 2n(\sum_{i=0}^{k} (\frac{3}{2})^i)$

q

$= 2n\left(\frac{1-q^{k-1}}{1-q}\right)$

$= 2n\left(\frac{q^{k+1}-1}{\frac{3}{2}-1}\right)$

$= 4n\left(\frac{3}{2}^{\log_2 n + 1} - 1\right)$

$$A(n) = \begin{cases} 0 & \text{si } n \leq 1 \\ 4 \times \frac{n}{2} + 3A\left(\frac{n}{2}\right) \end{cases}$$

$$\in \Theta\left(4n\left(n^{\log_2 \frac{3}{2}}\right)\right)$$

$$= 4n^{\log_2\left(\frac{3}{2}\right)+1}$$

$$= 4n^{\log_2 3}$$