EA4 – Éléments d'algorithmique
TD n° 2 : Notations de Landau et complexité

Les questions facultatives, pour aller plus loin, sont marquées (*).

Exercice 1 : Notation O et Θ

1. Montrer que $3n^2 + 5n + 12 \in O(n^2)$.
2. Montrer que $5n^3 - n^2 \in O(n^4)$.
3. Comparer $n!$ à n^n.
4. Montrer que $n^2 + n \log n \in \Theta(n^2)$.
5. Montrer que $8\sqrt{n} \notin \Theta(n)$.
6. (*) Montrer que $\log(n!) \in \Theta(n \log n)$.
7. (*) Montrer que pour tout $a, b > 1$, $\log_a(n) \in \Theta(\log_b(n))$.

Exercice 2 : Propriétés de Θ

Dans la suite, f et g désignent des fonctions à valeurs dans $\mathbb{N}^*$.

1. Montrer que $\forall f, g,\ f \in \Theta(g) \iff g \in \Theta(f)$.
2. Montrer que $\forall f, g,\ \max(f, g) \in \Theta(f + g)$.
3. Qu'en est-il de $\min(f, g)$?

Exercice 3 : complexité des boucles

Exprimer le plus simplement possible l'ordre de grandeur du nombre d'additions effectuées par les algorithmes suivants :

1.
```
for i in range(n):
  for j in range(n):
    for k in range(n):
      val1 = val1 + 1
```

2.
```
for i in range(n):
  for j in range(i):
    val1 = val1 + 1
```

On considère maintenant des algorithmes prenant un paramètre entier `n`, et deux fonctions de `n` notées f et g.

3. On suppose que l'algorithme `A(n)` se décompose en deux parties consécutives, de complexité (en temps) respective $\Theta(f(n))$ et $\Theta(g(n))$. Quel est l'ordre de grandeur de sa complexité ?
4. On suppose que l'algorithme `A(n)` fait $\Theta(f(n))$ tours de boucle, chacun ayant une complexité $\Theta(g(n))$. Quel est l'ordre de grandeur de sa complexité ?

Exercice 4 : un algorithme récursif

On considère l'algorithme suivant :

```
def F(n) :
  if n < 3 : return 1
  else : return 2 * F(n - 1) + F(n - 3)
```

Soit $A(n)$ le nombre d'additions effectuées lors de l'exécution de F(n).

1. Donner une définition de $A(n)$ par récurrence. En déduire que A est croissante.
2. Montrer que $A(n) \in \Omega(2^{n/3})$ où / est la division euclidienne.
3. Proposer un algorithme qui fait un nombre linéaire d'additions pour calculer les mêmes valeurs que F(n).
4. Quel est la complexité de cet algorithme ?
5. Existe-t-il un algorithme de complexité inférieure ?

Exercice 5 : classement

Classer les fonctions suivantes en fonction de leur ordre de grandeur dans les classes Θ_1 à Θ_7 en respectant les conditions suivantes :

– deux fonctions appartiennent à la même classe Θ_i si et seulement si elles sont du même ordre de grandeur :

$$\forall f, g, \quad f \in \Theta(g) \iff \exists i,\ f, g \in \Theta_i$$

– les classes Θ_i sont rangées en ordre de grandeur croissant :

$$\forall i, \forall f, g, \quad f \in \Theta_i \text{ et } g \in \Theta_{i+1} \implies f \in O(g)$$

Liste des fonctions à traiter :

$n^2 - 5n$, $n + 2n^3 \log n$, $n^2 + n \log n$, $n^2\sqrt{n}$, $\sqrt{n}$, 2^n, e^n, 2^{n+4}, $\log n$, $\log(n^4)$

Θ_1	Θ_2	Θ_3	Θ_4	Θ_5	Θ_6	Θ_7

Notation $\overset{<}{O}$, $\overset{>}{\Omega}$, $\overset{\Leftrightarrow}{\Theta}$ $\Leftrightarrow$ à la fois $\underset{O}{<}$ et $\underset{\Omega}{>}$

$f(n) \underset{=}{\in} O(g(n))$: "$f(n)$ est asymptotiquement plus petite que $g(n)$"

$\Leftrightarrow \exists c, n_0, \forall n \geq n_0, \; f(n) \leq C\, g(n)$ on divise $g(n)$ les 2 côtés

$\Leftrightarrow \exists C$ tq $\lim\limits_{n \to +\infty} \dfrac{f(n)}{g(n)} = C$

$f(n) \in \Omega(g(n)) \Leftrightarrow g(n) \in O(f(n))$

$f(n) \in \Theta(g(n)) \Leftrightarrow f(n) \in O(g(n))$ et $f(n) \in \Omega(g(n))$

$\Leftrightarrow f(n) \in O(g(n))$ et $g(n) \in O(f(n))$

Quand dans O, on trouve Θ, on peut pas tomber

Exemple :

$f(n) = 7n + 3$

$g(n) = n$

Montrer que $f(n) \in O(g(n))$

$7n + 3$

$Cn \geq 7n + 3$

$7 \nearrow C_1 n \geq 7n$

$3 \nearrow C_2 n \geq 3$

dès que $n \geq 1$

$7n + 3n \geq 7n + 3$ pour $n \geq 1$

$10n$

Si $C = 10$ et $n_0 = 1$

$\forall n \geq n_0$

$f(n) = 7n + 3 \leq 10n$

$\hookrightarrow 10g(n)$

$f(n) \in O(g(n))$

2eme meth :

$$\frac{f(n)}{g(n)} = \frac{7n+3}{n} = 7 + \frac{3}{n}$$

$$\lim_{n \to +\infty} \frac{f(n)}{g(n)} = \lim_{n \to +\infty} 7 + \frac{3}{n} = 7 \qquad \frac{g(n)}{f(n)} \longrightarrow 7$$

Donc $f(n) \in O(g(n))$

Ex 1

1) $f(n) = 3n^2 + 5n + 12$

$g(n) = n^2$

$$\frac{f(n)}{g(n)} = \frac{3n^2 + 5n + 12}{n^2} = 3 + \frac{5}{n} + \frac{12}{n^2}$$

$$\lim_{n \to +\infty} \frac{f(n)}{g(n)} = \lim_{n \to +\infty} 3 + \frac{5}{n} + \frac{12}{n^2}$$

$$= \boxed{3}$$

Donc $f(n) \in O(g(n))$

2e meth

On a

$3n^2 \geq 3n^2$

$+ \; 5n^2 \geq 5n$

$+ \; 12n^2 \geq 12$ pour $n \geq 1$

Donc $20n^2 \geq 3n^2 + 5n + 12$ $\boxed{\text{pour } n \geq 1}$ n_0

Donc pour $C = 20$, $n_0 = 1$ on a $\forall n \geq n_0$

, $f(n) \leq C g(n)$

2) $f(n) = 5n^3 - n^2 \in O(n^4)$

$g(n) = n^4$

$$\frac{f(n)}{g(n)} = \frac{5n^3 - n^2}{n^4} = \frac{5n^3}{n^4} - \frac{n^2}{n^4}$$

$$= \frac{5}{n} - \frac{1}{n^2}$$

$$\lim_{n \to +\infty} \frac{f(n)}{g(n)} = \lim_{n \to +\infty} \frac{5}{n} - \frac{1}{n^2} = 0$$

Donc $f(n) \in O(g(n))$

2ème meth

$f(n) = 5n^3 - n^2$

$g(n) = n^4$

On a :

$$\underbrace{5n^4}_{5g(n)} \geq 5n^3 \geq \underbrace{5n^3 - n^2}_{f(n)}$$

Donc $5n^4 \geq 5n^3 - n^2$

$5g(n) \geq f(n)$

Donc pour $C = 5$, $n_0 = 0$, on a $\forall_n \geq n_0$

$f(n) \leq C g(n)$

3)

$n! = 1 \times 2 \times 3 \times \cdots \times n$

$n^n = n \times n \times n \times \cdots \times n$

Donc $n! \in O(n^n)$

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

$g(n) = n^2$

$f(n) = C(g(n))$

$\lim_{n \to +\infty} \frac{f(n)}{g(n)} = C \neq 0$
$\neq +\infty$

donc $\lim_{n \to +\infty} \frac{g(n)}{f(n)} = \frac{1}{C}$

$\Rightarrow f \in O(g) \quad \Rightarrow \quad g \in O(f)$

$\lim_{n \to +\infty} \frac{n^2 + n\log(n)}{n^2} = \lim_{n \to +\infty} 1 + \frac{\log(n)}{n} = 1$

5) $8\sqrt{n} \notin \Theta(n\log n)$

$8\sqrt{n} \in O(n)$ ✓

~~$8\sqrt{n} \in \Omega(n)$~~

$$\lim_{n \to +\infty} \frac{f(n)}{g(n)} = \lim_{n \to +\infty} \frac{8\sqrt{n}}{n}$$

$$= \lim_{n \to +\infty} \frac{8}{\sqrt{n}}$$

$$= 0$$

Ex3

$f(n)$ nombre d'additions

On veut $g(n)$ tq $f(n) \in \Theta(g(n))$

et $g(n)$ est "simple"

n^2 plus simple que $12n^2 + \frac{3n}{18} + 11$

1) for i in range(n):
  for j in range(n):
    for k in range(n):
      val1 = val1 + 1;

$\Theta(n^3)$

2) for i in range(n)
  for j in range(i):
    v += 1   ) 1 addition) i addition

$$f(n) = \sum_{i=0}^{n-1} i = \frac{n(n-1)}{2} = \Theta(n^2)$$

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

$$f(n) = 1 + 2 + \dots + n-1$$
$$+ \quad + \quad + \quad +$$
$$f(n) = (n-1) + (n-2) + \dots + 1$$

$$2 f(n) = \underbrace{n + n + \dots n + n}_{n-1 \text{ term}}$$

$$\cancel{2} f(n) = \frac{(n-1)n}{2}$$

$f(n) \leq n^2 \Rightarrow f(n) = O(n^2)$

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

$$\geq \frac{n}{2}$$

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

Donc $f(n) = \Omega(n^2)$

3) fonction1() $\Theta(f(n))$ +) Θ somme

fonction2() $\Theta(g(n))$) des 2

$\Theta(f(n) + g(n)) \Leftrightarrow \Theta(\max(f(n), g(n)))$

4 produit [for i in range $\Theta(f(n))$

[fonct(), $\Theta(g(n))$

8)

$$f(n) \leq \max(f(n), g(n))$$
$$g(n) \leq \max(f(n), g(n))$$

$$f(n) + g(n) \leq 2\max(f \quad)$$
$$f(n) + g(n) = O(\max \ldots)$$