Complexité - TDs

Nicolas Bourras

September 21, 2020

Contents

0.1 Jeudi 10 Septembre 2020

Planche de TD 1.

0.1.1 Exercice 1

Pour savoir si $g(n) \in O(f(n))$, il suffit de montrer qu'à partir d'un certain rang n_0, la fonction est majorée par $O(Cf(n))$, avec C une constante :

$$\exists C \in R, \exists n_0 \in N, \forall n \geqslant n_0, g(n) \leqslant Cf(n)$$

Question 1

- $g(n) = 6n + 12 \in O(n)$

  On trouve C et n_0, et on vérifie que $g(n) \leqslant Cf(n)$. Dans ce cas, on peut prendre $C = 7$ et $n_0 = 12$.

  Puis-ce que $\forall n \geqslant 12, g(n) < 7n$, alors $g(n) \in O(n)$

- $g(n) = 3n \in O(n^2)$

  $C = 4$, $n_0 = 10$

- $g(n) = 10^{1000000}n \in O(n)$

  $C = 10^{1000000} + 1$, $n_0 = 1$

- $g(n) = 5n^2 + 10n \in O(n^2)$

  $C = 6$, $n_0 = 100$

- $g(n) = n^3 + 1000n^2 + n + 8 \notin O(n^2)$

  Par l'absurde:

$$\forall C \in R, \forall n_0 \in N, \exists n \geqslant n_0, g(n) \geqslant Cf(n)$$

$$\iff \forall C \in R, \forall n_0 \in N, \exists n \geqslant n_0, n^3 + 1000n^2 + n + 8 \geqslant Cn^2$$

$$\iff \forall C \in R, \forall n_0 \in N, \exists n \geqslant n_0, n + 1000 + \frac{1}{n} + \frac{8}{n^2} \geqslant C$$

$\forall C \in N$, il existe $n > C - 1000 - \frac{1}{n} - \frac{8}{n^2}$

Question 2

$g(n) \in \theta(f(n))$ si :

$$\exists C_1, C_2 \in R, \forall n \geqslant n_0, C_1 f(n) \leqslant g(n) \leqslant C_2 f(n)$$

- $g(n) = 5n^2 + 10n \in \theta(n^2)$
  $5n^2 \leqslant 5n^2 + 10n \leqslant 6n^2$ avec $n_0 = 1000$
- $g(n) = n^2 + 1000000n \in \theta(n^2)$
  $n^2 \leqslant n^2 + 1000000n \leqslant 2n^2$ avec $n_0 = 10^{100}$
- $g(n) = 4n^2 + n.log(n) \in \theta(n^2)$
  $4n^2 \leqslant 4n^2 + n.log(n) \leqslant 5n^2$ avec $n_0 = 1000$
- $g(n) = 3n + 8 \notin \theta(n^2)$
  Il suffit de montrer que $\forall C_1 \in R, \forall n_0 \in N, \exists n \geqslant n_0, C_1 n^2 > 3n + 8$
  $\iff C_1 n^2 > 3n + 8 \iff C_1 > \frac{3}{n} + \frac{8}{n^2}$
  $\forall n \in N^*, \frac{3}{n} + \frac{8}{n^2} \leqslant 11$, donc avec $C_1 > 11$ on a $\forall n \in N^*, C_1 n^2 > 3n + 8$

0.1.2 Exercice 2

Question 1

```
puissance(x, p):
  res = 1
  while p > 0:
    res = res * x
    p = p - 1
  return res
```

Question 2

Rappel : $t_x = taille(x) = \lfloor log(x) \rfloor + 1$
On a donc $x = 2^{tx} - 1$ et $p = 2^{tp} - 1$.
La ligne `res = res * x` est au pire $\leqslant tx * p$. La ligne `p = p - 1` est au pire tp opérations.
Ces deux lignes sont donc égales (au pire) à $P(tx*p+tp) \iff 2^{tp}(tx*2^{tp}+tp) \simeq 2^{2tp}tx + tp * 2^{tp} \in \theta(2^{2tp}tx)$.
Note : on aurait du ajouter le test de la boucle.

0.2 17 Septembre 2020

0.2.1 Exercice 3

Question 1

```
def matrice_produit(a, b):
for i in range(1, n):
for j in range(1, n):
out[i][j] = 0 # C1
for k in range(1, n):
out[i][j] += a[i][k] + b[k][j] # C2

return out
```

La taille des données d'entrée = la taille k (bornée par une constante) de codage d'un entier $*n^2$, fois 2 pour car on a 2 matrices.
On pose : $t = 2kn^2 \iff n = \frac{t}{2k}^{\frac{1}{2}}$.
$temps = n^2(c_1 + nc_2) = c_1n^2 + c_2n^3 = \frac{C_1^t}{2K} + C_2(\frac{t}{2k})^{\frac{2}{3}} \in \theta(t^{\frac{3}{2}})$

Question 2

Oui.

Question 3

```
matrice puissance(matriceA, matriceB):
B=A
for i = 1 to P-1
b = produit(B, A)
retourner B
```

Taille de l'entée : $t = kn^2 + k$