POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

LOG2810

STRUCTURES DISCRÈTES

TD 6 : ALGORITHMES

A2022

Directives pour la remise :

- Répondez directement sur ce document papier.
- La remise est individuelle, mais le travail en équipe est encouragé.
- La remise se fait à la fin de la séance de TD.
- **Aucun retard ne sera accepté.**
- **Le non-respect des consignes entraînera automatiquement la note 0 pour ce TD.**

Identification

Veuillez inscrire votre section, nom, prénom et matricule ainsi que les noms des collègues avec lesquels vous avez collaboré pour le TD

Section :

Nom :

Prénom :

Matricule :

Collègues :

Exercice 1 :

Le temps de résolution d'un problème dépend de 2 paramètres :

- n : la taille du problème
- f(n) : la complexité du problème, c'est-à-dire le nombre d'opérations élémentaires nécessaire pour trouver la solution

On vous fournit un calculateur sur lequel chaque opération élémentaire s'effectue en 10^{-6} secondes.

Pour chaque complexité suivante, déterminez la taille maximale du problème pouvant être résolu en une seconde. Il n'est pas nécessaire de justifier votre réponse.

a) n

Réponse : :

1 seconde = n x 10^{-6} seconde

Donc n = 10^6

b) n^2

Réponse : :

1 seconde = n^2 x 10^{-6} seconde

Donc $n^2 = 10^6$

Et $n = 10^3$

c) n^3

Réponse :

$N^3 = 10^6$

Donc $n = 10^2$

d) $\sqrt{n}$

Réponse :

$N^{1/2} = 10^6$

Donc $n = 10^{12}$

e) 2^n

Réponse :

$2^n = 10^6$

Donc $n = \log2(10^6)$

N = 19

f) $\log_2(n)$

Réponse :

$Log_2(n) = 10^6$

Donc $n = 2^{10^6} = 2^{1\,000\,000}$

g) n !

Réponse :

Pour cette complexité, un simple calcul n'est pas possible car il n'y a pas d'opération inverse de factorielle. En procédant par essai-erreur, on obtient n = 9.

Exercice 2 : En vous basant sur les définitions des notations de Landau que sont : O, Ω, θ, dites si les affirmations suivantes sont vraies ou fausses. Il n'est pas nécessaire de justifier votre réponse ici.

a) $(3n^4 + 2n^2 – 7) \in O(n^2)$

Réponse : FAUX

En tant que fonction polynôme, $(3n^4 + 2n^2 – 7)$ est en O du monôme de plus haut degré, soit $O(n^4)$. Et $n^4 > n^2$ pour tout n.

b) $(3n^4 + 2n^2 – 7) \in O(n^4)$

Réponse : VRAI

En tant que fonction polynôme, $(3n^4 + 2n^2 – 7)$ est en O du monôme de plus haut degré, soit $O(n^4)$.

c) $(3n^4 + 2n^2 - 7) \in O(n^5)$

Réponse : VRAI

En tant que fonction polynôme, $(3n^4 + 2n^2 - 7)$ est en O du monôme de plus haut degré, soit $O(n^4)$.

Et puisque $n^4 < n^5$, $n^4 \in O(n^5)$, et par conséquent la fonction aussi.

d) $(2n^2 - 7) \in \Omega(n^2)$

Réponse : VRAI

$n^2 < (2n^2 - 7)$ pour $n > k$ avec $k = 3$.

e) $(2n^2 - 7) \in \Omega(n^5)$

Réponse : FAUX

Il n'est pas possible de trouver des constantes k et c telles que

Pour $n > k$ $n^5 < c(2n^2 - 7)$

f) $4n^4 \in \theta(n^5)$

Réponse : FAUX

$4n^4 \in O(n^5)$, mais $4n^4 \notin \Omega(n^5)$

g) $5n^5 \in \theta(n^5)$

Réponse : VRAI

$5n^5 \in O(n^5)$, et $5n^5 \in \Omega(n^5)$

Exercice 3 : Déterminez l'ordre du temps de calcul pour les algorithmes suivants. Justifiez vos réponses tout en montrant toutes les étapes qui ont conduit à votre résultat. Si vous avez fait des hypothèses, énoncez-les.
Une opération s'effectue en une unité de temps arbitraire (UA) .

a) algorithme 1 :

```
1. I = 0
2. Tant que I < N
3.      I = I + 1
```

Réponse :
Comptons le nombre d'opérations.

- Méthode 1

Ligne 1 : 1 opération
Ligne 2 : (N + 1) comparaisons, donc (N + 1) opérations.
Ligne 3 : 1 opération réalisée N fois, donc N opérations. On considère l'incrémentation comme une opération élémentaire.
Au total (1 + (N + 1) + N) opérations, soit (2N + 2) opérations.
L'algorithme 1 est O(N).

- Méthode 2

Ligne 1 : 1 opération
Ligne 2 : 1 comparaison, donc 1 opération
Ligne 3 : 1 opération. On considère l'incrémentation comme une opération élémentaire.
La boucle Tant que s'exécute N fois et une dernière comparaison qui détermine la sortie de la boucle. Le nombre total d'opérations dans la boucle est donc 2N + 1.
Au total (1 + (2N + 1)) opérations, soit (2N + 2) opérations.
L'algorithme 1 est O(N)

b) algorithme 2 :

```
1. I = N
2. Tant que I < N
3.       I = I - 2
```

Réponse :
Comptons le nombre d’opérations.

Ligne 1 : 1 opération
Ligne 2 : 1 comparaison, donc 1 opération
Ligne 3 : 0 opération. La ligne ne s’exécutera pas car on ne rentrera pas dans la boucle.
Au total (1 + 1)) opérations, soit 2 opérations.
L’algorithme 2 est $\theta(1)$.

c) algorithme 3 :

```
1. I=1
2. P = 0
3. Tant que I <= N
4.      J = 1
5.      Tant que J <= N
6.              P = P + J
7.              J = J + 1
8.      I = I + 1
```

Réponse :

Comptons le nombre d'opérations.

Ligne 1 : 1 opération

Ligne 2 : 1 opération

Ligne 3 : (N + 1) comparaisons, donc (N + 1) opérations.

Ligne 4 : 1 opération réalisée N fois, donc N opérations.

Ligne 5 : 1 comparaison réalisée N fois, donc N opérations.

Ligne 6 : 1 opération réalisée N fois, donc N opérations.

Ligne 7 : 1 opération réalisée N fois, donc N opérations.

Ligne 8 : 1 opération réalisée N fois, donc N opérations.

La boucle de la ligne 5 à 7 s'exécute N fois en tant que boucle imbriquée. Cette boucle fait donc N(N+N+N) soit $3N^2$.

Au total $(1 + 1 + (N + 1) + N + 3N^2 + N)$ opérations, soit $(3N^2 + 3N + 3)$ opérations.

L'algorithme 3 est $O(N^2)$.

Exercice 4 : Démontrez que (n3 + 2n)/(2n + 1) est O(n2).

Réponse : :

Par définition, si $(n^3 + 2n)/(2n + 1)$ est $O(n^2)$ alors, il existe c et k tel $n \geq k$ et $(n^3 + 2n)/(2n + 1) \leq c.n^2$

Supposons que l'inégalité est vérifiée et trouvons les valeurs c et k.
$(n^3 + 2n)/(2n + 1) \leq c.n2$

On sait que pour $n \geq 1$, $2n + 1 > 0$. On peut donc écrire successivement :
$(n^3 + 2n) \leq c.n^2.(2n + 1)$
$(n^3 + 2n) \leq c.(2n^3 + n^2)$
$0 \leq (2c - 1).n^3 + c.n^2 – 2n$

En posant c = 1. On a :
$0 \leq (2 - 1).n^3 + n^2 – 2n$
$0 \leq n^3 + n^2 – 2n$
$0 \leq n(n^2 + n – 2)$
$0 \leq n(n + 2)(n – 1)$
- Lorsque $n \geq 1$, on peut déduire que $n \geq 0$.
- Lorsque $n \geq 1$, on a $(n + 2) \geq 3$ et on déduit que $(n + 2) \geq 0$.
- Lorsque $n \geq 1$, on a $(n – 1) \geq 0$

Ainsi, lorsque $n \geq 1$, $0 \leq n(n + 2)(n – 1)$ est vérifiée.

On peut donc considérer que lorsque k = 1 et c = 1 on a : pour $n \geq k$, $(n^3 + 2n)/(2n + 1) \leq c.n2$
D'où $(n^3 + 2n)/2n + 1)$ est $O(n^2)$

Exercice 5 : Donnez une évaluation du comportement asymptotique de chacune des fonctions suivantes, en utilisant le O.

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

Réponse :

$\log(n^2) = 2.\log(n)$

ainsi 4 $\log(n^2)$ = 8 $\log(n)$ qui est $O(\log(n))$

donc 4 $\log(n^2)$ est $O(\log(n))$

b) $12n^7 + n!$

Réponse :

Évaluons d'abord le comportement asymptotique de $12n^7$.

On a $n^7 < n!$ pour $n>14$

D'où $12n^7 + n! < 13(n!)$ pour n grand

ainsi $13(n!)$est $O(\log(n))$

D'où $12n^7 + n!$ est $O(n!)$

c) $3n^3(\sqrt{n} + \log(n))$

Réponse :

$\sqrt{n} + \log(n)$ est $O(\sqrt{n})$ *d'après le cours*

$3n^3$ est $O(n^3)$

$3n^3(\sqrt{n} + \log(n))$ est O $(n^3. \sqrt{n})$

Donc $3n^3(\sqrt{n} + \log(n))$ est O $(n^{3,5})$

c) $(3^n + n^6)(7^n + \sqrt{n}\ )$

Réponse :

$(3^n + n^6)$ est $O(3^n)$

$(7^n + \sqrt{n}\ )$ est $O(7^n)$

Donc $(3^n + n^6)(7^n + \sqrt{n}\ )$ est $O(3^n. 7^n)$

c'est-à-dire $O(21^n)$

Exercice 6 : Ordonnez les fonctions suivantes de sorte que chaque fonction soit un grand-O de la suivante.

1.5^n, n^{100}, $(log(n))^3$, $\sqrt{n}.log(n)$, 10^n , $(n!)^2$ et $n^{99} + n^{98}$

Exemple: n, n^3 , 2^n

Réponse : :

L'ordre est simple lorsque nous nous souvenons que les fonctions factorielles croissent plus vite que les fonctions exponentielles, que les fonctions exponentielles croissent plus vite que les fonctions polynomiales, et que les fonctions logarithmiques croissent très lentement. L'ordre est :

$(log(n))^3$, $\sqrt{n}.log(n)$, $n^{99} + n^{98}$, n^{100}, 1.5^n, 10^n , $(n!)^2$