Algorithmes et structures de données : TD 5

Temps d'un algorithme T(n) - Notation Grand-O

Exercice 5.1 *Temps d'un algorithme T(n)*

Pour chacun des fonctions $T_i(n)$ suivant, déterminer sa complexité asymptotique dans la notation Grand-O. Exemple : $T_0(n) = 3n \in O(n)$.

1. $T_1(n) = 6n^3 + 10n^2 + 5n + 2$
2. $T_2(n) = 3\log_2 n + 4$
3. $T_3(n) = 2^n + 6n^2 + 7n$
4. $T_4(n) = 7k + 2$
5. $T_4(n) = 4\log_2 n + n$
6. $T_5(n) = 2\log_{10} k + kn^2$

Exercice 5.2 *Temps d'un algorithme T(n)*

Considérer les deux algorithmes A1 et A2 avec leurs temps d'exécution $T_1(n) = 9n^2$ et $T_2(n) = 100n + 96$ respectives.

1. Déterminer la complexité asymptotique des deux algorithmes dans la notation Grand-O. Quel algorithme a la meilleure complexité asymptotique?
2. Montrer que vos solutions sont correctes en spécifiant un c et un n_0 par algorithme afin que la relation suivante soit satisfaite :
   $O(f) = \{g | \exists c > 0 : \exists n_0 > 0 : \forall n \geq n_0 : g(n) \leq cf(n)\}$
3. Calculer les temps maximales d'exécution des deux algorithmes $T_i(n)$ pour $n = 1, n = 3, n = 5, n = 10, n = 14$.
4. Ebaucher les graphes des deux fonctions T_i dans un même système de coordonné (abscisse n, ordonné $T_i(n)$).
5. Pour quelles longueur de donnée n quel algorithme est le plus efficace ? (Calculer l'intersection ...)
6. Quelle est la complexité asymptotique de l'algorithme suivant ? Quelle règle avez vous appliqué ?

```
début
appeler A1      {Ici l'algorithme 1 est exécuté. }
appeler A2      {Ici l'algorithme 2 est exécuté. }
fin
```

La feuille continue au verso ..

Exercice 5.3 *Addition de matrices*

Considérer les deux matrices carrées A et B de taille n :

$$A = \begin{pmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ ... & ... & ... & ... \\ a_{n1} & a_{n2} & ... & a_{nn} \end{pmatrix}, B = \begin{pmatrix} b_{11} & b_{12} & ... & b_{1n} \\ b_{21} & b_{22} & ... & b_{2n} \\ ... & ... & ... & ... \\ b_{n1} & b_{n2} & ... & b_{nn} \end{pmatrix},$$

L'addition de ces deux matrices donne la matrice C quadratique de taille n:

$$C = \begin{pmatrix} c_{11} & c_{12} & ... & c_{1n} \\ c_{21} & c_{22} & ... & c_{2n} \\ ... & ... & ... & ... \\ c_{n1} & c_{n2} & ... & c_{nn} \end{pmatrix}$$

avec

$$c_{ij} = a_{ij} + b_{ij} \quad \forall i, \forall j$$

1. Définir le type des matrices carrées et déclarer les variables A,B, et C.
2. Ecrire un algorithme qui effectue l'addition des deux matrices A et B et stocke les résultats en C.
3. Déterminer la fonction de temps maximale ("worst case") T(n) pour des matrices de taille n.
4. Déterminer la complexité Grand-O pour des matrices de taille n.

Exercice 5.4 *Multiplication de matrices*

La multiplication des deux matrices carrées de taille n donne la matrice C quadratique de taille n:

$$C = \begin{pmatrix} c_{11} & c_{12} & ... & c_{1n} \\ c_{21} & c_{22} & ... & c_{2n} \\ ... & ... & ... & ... \\ c_{n1} & c_{n2} & ... & c_{nn} \end{pmatrix}$$

avec

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj} \quad \forall i, \forall j$$

1. Ecrire un algorithme qui effectue la multiplication des deux matrices A et B et stocke les résultats en C.
2. Déterminer la fonction de temps maximale ("worst case") T(n) pour des matrices de taille n.
3. Déterminer la complexité O(n) pour des matrices de taille n.