Initiation à la complexité algorithmique

Complexité d'algorithme

- Comment connaître le temps de calcul d'un algorithme?
- Comment comparer deux algorithmes ?

Le temps de calcul d'un algorithme dépend du nombre d'opérations et de l'espace occupé.

Compromis entre le nombre d'opérations et l'espace occupé.

Exemple : la gestion de la mémoire virtuelle.

- Complexité temporelle
- Complexité spatiale

Le nombre d'opérations donne une idée précise sur le temps d'exécution d'un algorithme Exemples : donner le nombre d'opérations pour

- Le calcul du carré d'un nombre
- La somme des n premiers éléments d'un tableau (ou liste)
- Tour de Hanoi

On mesure le nombre d'opérations élémentaires.

Les opérations élémentaires considérées sont :

- Le nombre de comparaison (recherche)
- Le nombre d'affectations (tris)
- Le nombre d'opérations (+, *) (calcul sur les matrices)

Le coût (complexité) d'un algorithme se mesure pour une taille de données, en nombre d'opérations.

Quelle complexité?

- Complexité dans le pire des cas
- Complexité dans le meilleur des cas
- Complexité moyenne

Exemple : calculer la complexité pour l'algorithme de recherche d'un élément dans un tableau

- Le pire des cas : le tableau ne contient pas l'élément (ou bien c'est le dernier élément) -> N
- Le meilleur des cas : l'élément est le premier -> 1
- Cas moyen : difficile à calculer en général.
- Considérer la probabilité d'apparition des nombre de 1 à k. Alors, la complexité moyenne est :

$$N/(k(1-(1-1/k)))$$

Analyse asymptotique

- On calcule le temps de calcul à un facteur multiplicatif près, lorsque la taille des données est grand.
- Notation :

n taille des données et c une constante

$$f(n) < c*g(n)$$

Alors on note f=O(g)

- Le calcul du carré d'un nombre : 2, on dit que la complexité est constante
- La somme des n premiers éléments d'un tableau (ou liste) : n-1 on dit qu'elle est en O(n)

Exercices

Calcul de la complexité pour différents algorithmes à fournir ou bien à faire faire.