Éléments d'Algorithmique
CMTD1

Christine Tasson
Université de Paris, IRIF

Définition d'un programme

Définition

Un algorithme est

- une description formelle d'un procédé de traitement qui permet, à partir d'un ensemble d'informations initiales, d'obtenir des informations déduites
- une succession finie et non ambigüe d'opérations, et son exécution se termine toujours.

Définition

Un programme

- est une suite d'instructions définies dans un langage donné
- décrit un algorithme.

Efficacité d'un algorithme

Complexité en temps

- compter le nombre d'opérations élémentaires effectuées lors de l'exécution de l'algorithme
- bien choisir les structures de données utilisées.

Complexité en espace

- mesurer la place mémoire maximale occupée durant l'exécution,
- bien choisir les structures de données utilisées.

Terminaison : termine en un temps fini.

Complétude : pour un espace de problèmes donné l'algorithme donne toujours des propositions de solutions.

Nous allons nous concentrer sur la complexité en temps de l'algorithme.

Efficacité d'un algorithme

Sur les machines actuelles, le temps pris par un calcul est très difficile à prévoir, avec une forte composante aléatoire :

- traduction (interprétation, compilation),
- forte dépendance à l'environnement (mémoire, système d'exploitation, multi-threading,...)
- nombreuses optimisations qui dépendent de l'historique (caches, ...).

Nous travaillons avec un modèle de machine simplifiée.

La complexité en temps permet de comprendre si un algorithme peut servir pour de grandes taille de données.

Complexité

Définition

- Complexité temporelle : (ou en temps) temps de calcul,
- Complexité spatiale : (ou en espace) l'espace mémoire requis par le calcul.

Définition

- La complexité pratique est une mesure précise des complexités temporelles et spatiales pour un modèle de machine donné.
- La complexité (théorique) est un ordre de grandeur de ces couts, exprimé de manière la plus indépendante possible des conditions pratiques d'exécution

Analyse de la complexité en temps

Opérations élémentaires :

- assignations: min=tab[0], min=tab[i] ,
- comparaisons: min > tab[i],
- accès à un élément d'un tableau: tab[i].

Ces trois opérations élémentaires se font en temps constant (ne dépend pas de la taille des données).

Pour terminer

- Un algorithme est efficace si sa complexité est au plus un polynôme en n, où n est la taille des données en entrée.
- Il existe des algorithmes pour lesquels nous ne connaissons pas de solutions efficaces. Exemple: le problème du voyageur de commerce.
- Il existe des problèmes qu'on ne sait pas résoudre avec un algorithme. Exemple: Le problème de l'arrêt : peut-on écrire un algorithme A qui pour tout semi-algorithme B et toute instance I détermine si B s'arrête pour la donnée I ? La réponse est non, c'est un problème indécidable.