Cours Algorithmique Avancée

Master 1 Informatique
RN- Options : RS & SIA

Prof. BOUALLOUCHE Louiza

Chap.1 Introduction à l'Algorithmique Avancée et l'efficacité des Algorithmes

- Un algorithme/programme peut être considéré comme une fonction qui fournit un ensemble de résultats à partir d'un ensemble de données. Ecrire un programme revient généralement à dégager une méthode de résolution faisant intervenir des outils complexes. Le résultat de cette démarche logique de résolution d'un problème donnée est généralement mis sous forme d'un algorithme (à l'aide d'un pseudo-code) avant de le traduire sous forme d'un programme à l'aide d'un langage adéquat.

- L'algorithmique avancée est la construction d'algorithmes efficaces en utilisant des techniques et mécanismes spécifiques.

I.1 Position du problème et recherche d'une méthode de résolution

Il s'agit de poser le problème et de répondre à une double problématique.

1. Il faut tout d'abord trouver une ou plusieurs méthodes de résolution du problème qu'elles soient exactes ou approchées;
2. puis chercher la plus efficace, parmi ces méthodes sélectionnées, en termes de temps d'exécution et d'espace mémoire.

NB. Il est impératif qu'un programme soit efficace. Pour cela, il est essentiel de trouver une méthode de résolution rapide, d'écrire cette dernière sous forme d'un algorithme efficace qu'il faut traduire dans un langage de programmation de manière efficace. Ceci dit, le processus d'optimisation d'un programme commence à partir de la recherche de la méthode de résolution.

Exemple de problèmes à résoudre

- Soient 3 nombres réels a, b et c, on souhaite déterminer les solutions de l'équation

$a*x^2 + b*x + c = 0$

La solution de ce problème est bien connue et est exacte.

- Soient 6 nombres réels a, b, c, d, e et f, on souhaite déterminer les solutions de l'équation

$a*x^5 + b*x^4 + c*x^3 + d*x^2 + e*x + f = 0$

On ne dispose pas de solution générale pour ce problème (cf. Théorie de GALOIS).

Exemple de problèmes à résoudre (suite)

- Soit à calculer la combinaison de P dans N

$$C_N^P = \frac{N!}{P!*(N-P)!}$$

La première solution (naïve) consiste à

1. calculer N! = 1*2*3*…*N (résultat dans X)
2. calculer P! = 1*2*3*…*P (résultat dans Y)
3. calculer (N-p)! = 1*2*3*…*(N-P) (résultat dans Z)
4. Finalement le résultat à retourner est X/(Y*Z)

Cette démarche est correcte mais n'est pas efficace; en effet, il est évident que cette expression peut être simplifiée, par exemple

N!/P! = (P+1)*(P+2)*…*(N)

Voici alors une méthode plus efficace

1. Calculer (P+1)*(P+2)*…(N) (résultat dans X)
2. Calculer (N-p)! = 1*2*3*…*(N-P) (résultat dans Y)
3. Finalement le résultat à retourner est X/Y

- Soit à écrire un algorithme qui délivre la valeur de cette somme

$$S = \log(x1) + \log(x2) + \ldots + \log(xn)$$

Le premier algorithme (naïf) est

```
Fonction S(X: Tab; N: entier) : réel
  Var i: entier;
  Début
    S← 0;
     Pour i allant de 1 à N faire
        S← S + log(X[i]);
     Fin pour
     Retourner S;
  Fin.
```

Cet algorithme fournit un résultat exact mais n'est pas efficace. En effet, la fonction log(.) consomme beaucoup de temps d'exécution; par ailleurs, l'expression S peut être mise sous forme plus simplifiée : $S = \log(x1 * x2 * x3 * \ldots * xn)$.

```
Fonction S(X: Tab; N: entier) : réel
Var  i : entier;
Début
  S← 1;
  Pour i allant de 1 à N faire
      S← S *X[i];
  Fin pour
  Retourner log(S);
Fin.
```