Programmation Efficace

François Laroussinie *et* ***Arnaud Sangnier***

IRIF - Université de Paris

Rappels algorithmiques

Qu’est-ce-qu’un algorithme ?

Suite d’instructions pour résoudre un problème
ou effectuer un calcul

Deux types d’algorithmes :

1. Résolution d’un problème (Sortie : Oui ou Non) :
   - Trouver si deux sommets dans un graphe sont liés par un chemin
   - Trouver si une formule est satisfiable
   - Vérifier si deux nombres sont premiers entre eux
2. Calcul :
   - Calculer les diviseurs d’un nombre
   - Calculer les composantes fortement connexes d’un graphe

Qu’est-ce-qu’un problème ?

Souvent un problème est défini par une entrée et une sortie attendue.

Exemple : Accessibilité dans un graphe

- Entrées : Un graphe orienté G et deux sommets s et t
- Sortie : Oui si on peut atteindre t depuis s dans G et non sinon

Attention : Ne pas confondre un problème et une instance du problème

- Un algorithme résolvant un problème doit fournir une solution pour toutes les instances du problème
- Une instance étant les données d’entrée

Des problèmes plus difficiles que d'autres

- Tous les problèmes n'ont pas la même complexité
- Il existe des problèmes plus simples à résoudre que d'autres
- Il existe des problèmes pour lesquels il n'existe pas d'algorithme
  $\Rightarrow$ **Problèmes indécidables**
- **Exemple :** L'arrêt des machines à deux compteurs
  - Entrée : Un programme manipulant deux variables entières c0 et c1 qui ne peuvent pas être négatives et initialisées à 0 et avec une suite d'instructions de la forme (c vaut c0 ou c1):
    1. l : c:=c+1;goto l'
    2. l : if (c==0) goto l' else c:c=c−1;goto l'
    3. lH : Halt;
  - Sortie : Est-ce-que l'instruction Halt est atteinte ?
  - Ce problème est indécidable

Classification des problèmes

On peut classer les problèmes **décidable** dans des classes de complexité.

Exemple :

- P (ou PTIME) : problèmes résolvables en temps polynomial
- PSPACE : problèmes résolvables en espace polynomial
- EXPTIME : problèmes résolvables en temps exponentiel
- EXPSPACE : problème résolvable en espace exponentiel
- ...

$$\mathrm{P} \subset \mathrm{PSPACE} \subset \mathrm{EXPTIME} \subset \mathrm{EXPSPACE} \ldots$$

On sait que certains problèmes sont **complets** pour une classe, c'est-à-dire que l'on ne peut pas faire mieux.

- Un problème est PSPACE-complet si on peut réduire tout problème dans PSPACE à ce problème.

Une classe particulière - I

Classe de complexité NP : problèmes résolvables de façon non-déterminisme en temps polynomial

À quoi correspond cette classe ?

- Il s'agit des problèmes pour lesquels on peut donner un certificat de validité qui est vérifiable en temps polynomial
- Un certificat de validité est une preuve sur le résultat
- Par exemple, pour savoir si t est atteignable depuis s dans un graphe, un certificat de validité est un chemin de s vers t
- Il existe des problèmes dans NP que l'on sait pas résoudre en temps polynomial
- On ne sait pas si P $\neq$ NP ou si P $=$ NP
- En pratique pour résoudre un problème dans NP, il faut un temps exponentiel pour énumérer tous les certificats

Une classe particulière - II

La frontière entre P et NP est parfois très subtil

Exemple :

- Problème 2SAT :
  - Entrée : une formule de logique propositionnelle en forme normale conjonctive et chaque clause ne contient au plus que deux variables (par ex: $\phi := (x_1 or \neg x_2) and (x_3 or \neg x_4) and (x_2 or \neg x_3)$)
  - Sortie : Oui si la formule est satisfaisable, non sinon.
  - 2SAT est dans P
- Problème 3SAT : même problème où chaque clause contient trois variables $\Rightarrow$ dans NP

Quelques rappels sur les graphes

- Codage par liste d'adjacence : [[1],[2,3],[0],[4],[2]]
- Avec les distances :
  [[[3,1]],[[4,2],[4,3]],[[2,0]],[[5,4]],[[4,2]]]

Que faut-il savoir faire sur les graphes ?

Maîtriser le cours d'Algorithmique du premier semestre

- Parcours en profondeur
- Parcours en largeur
- Algorithmes de plus courts chemin
- Déterminer les composantes fortement connexes

Encore une différence subtile

- Dire si un graphe est eulérien (existence cycle qui visite toutes les arêtes exactement une fois) $\Rightarrow$ problème dans P
- En effet un graphe est eulérien si et seulement si il est fortement connexe et chaque noeud a le même nombre d'arc entrant que d'arc sortant
- Dire si un graphe est hamiltonien (existence d'un chemin passant par tous les sommets une seule fois) $\Rightarrow$ problème dans NP