

# Algorithmique

## Essai de synthèse

Gilles Aldon, Jérôme Germoni, Jean-Manuel Mény

IREM de Lyon

Mars 2012

# Questions liées à la complexité

- Déroulement de l'algorithme
  - **terminaison**  $\rightsquigarrow$  notion de **convergent**  
(convergent = quantité à valeurs dans un ensemble où toute suite décroissante stationne, qui décroît strictement à chaque passage dans la boucle)

# Questions liées à la complexité

- Déroulement de l'algorithme
  - **terminaison**  $\rightsquigarrow$  notion de **convergent**  
(convergent = quantité à valeurs dans un ensemble où toute suite décroissante stationne, qui décroît strictement à chaque passage dans la boucle)
  - **correction**  $\rightsquigarrow$  notion d'**invariant de boucle**  
(assertion vraie en entrée de boucle, qui reste vraie à chaque passage dans la boucle, donc vraie en sortie de boucle)

# Questions liées à la complexité

- Déroutement de l'algorithme
  - **terminaison**  $\rightsquigarrow$  notion de **convergent**  
(convergent = quantité à valeurs dans un ensemble où toute suite décroissante stationne, qui décroît strictement à chaque passage dans la boucle)
  - **correction**  $\rightsquigarrow$  notion d'**invariant de boucle**  
(assertion vraie en entrée de boucle, qui reste vraie à chaque passage dans la boucle, donc vraie en sortie de boucle)
  - **complexité**  $\rightsquigarrow$  estimation du nombre d'opérations élémentaires en fonction de la taille des données

# Complexité

## Complexité

- en moyenne
- dans le cas le pire

Ex. : tri rapide :  $O(n \log n)$  en moyenne,  $O(n^2)$  dans le cas le pire

# Complexité

## Complexité

- en moyenne
- dans le cas le pire

Ex. : tri rapide :  $O(n \log n)$  en moyenne,  $O(n^2)$  dans le cas le pire

## Algorithmes

- exponentiels  
ex. : Fibonacci naïf, voyageur de commerce...
- polynomiaux  
ex. : tri par tas ( $O(n \log n)$ ), tri sélection ( $O(n^2)$ ), Gauss ( $O(n^3)$ )...

Classes de complexité :  $P$ ,  $NP$ ,  $PSPACE$ ,  $EXPTIME$ ...

Question théorique :  $P = NP$  ?

Question pratique :  $O(n^3)$  est trop grand si  $n = 10^6$ .

# Types d'algorithmes

- **glouton**
  - idée : faire un choix localement optimal en espérant que cela conduise à une solution globalement optimale
  - question du choix du critère non triviale (gymnase, monnaie)
  - ex. : fractions égyptiennes, coloriage d'un graphe, Prim, Dijkstra...

# Types d'algorithmes

- **glouton**

- idée : faire un choix localement optimal en espérant que cela conduise à une solution globalement optimale
- question du choix du critère non triviale (gymnase, monnaie)
- ex. : fractions égyptiennes, coloriage d'un graphe, Prim, Dijkstra...

- **diviser pour régner**

- idée : couper le problème en deux problèmes plus petits, les résoudre et recoller les solutions
- conduit généralement à la récursivité
- ex. : puissances, tri fusion, tri rapide...

# Types d'algorithmes

- **glouton**
  - idée : faire un choix localement optimal en espérant que cela conduise à une solution globalement optimale
  - question du choix du critère non triviale (gymnase, monnaie)
  - ex. : fractions égyptiennes, coloriage d'un graphe, Prim, Dijkstra...
- **diviser pour régner**
  - idée : couper le problème en deux problèmes plus petits, les résoudre et recoller les solutions
  - conduit généralement à la récursivité
  - ex. : puissances, tri fusion, tri rapide...
- **retour sur trace** (*backtracking*)
- **algorithmes probabilistes**
  - invoque un choix au hasard à un certain point
  - ex. : tri rapide, recuit simulé, Min cut...

# Structure des données

- types basiques : nombres, listes
- plus de structure : arbres, graphes
  - arbres : manière de ranger l'information de sorte à pouvoir y accéder rapidement
  - graphes : permet de coder de très nombreux problèmes
  - cachés (reines) ou explicites (Huffman, tri par tas, sudoku)
  -
- codage des structures
  - codage pas nécessairement compliqué  
(ex. : liste comme arbre binaire presque complet dans le tri par tas)
  - crucial pour l'efficacité de l'algorithme