

# Complexité en temps et en espace

## Introduction

Thomas Blanc

Terminale de l'école Alsacienne

25 octobre 2012

## 1 Introduction

- Problèmes indécidables
- Problèmes longs à résoudre
- Le problème de l'exponentielle
- Le problème de la mémoire
- L'idée d'ordre de grandeur

## 2 Complexité

- Définitions
- Quelques problèmes...

## 3 Classes de complexité

- Classes en temps
- Classes en espace
- Classes non-déterministes
- Hiérarchie des classes

- 1 Introduction
  - Problèmes indécidables
  - Problèmes longs à résoudre
  - Le problème de l'exponentielle
  - Le problème de la mémoire
  - L'idée d'ordre de grandeur
- 2 Complexité
  - Définitions
  - Quelques problèmes...
- 3 Classes de complexité
  - Classes en temps
  - Classes en espace
  - Classes non-déterministes
  - Hiérarchie des classes

# Peut-on tout calculer ?

## Le problème de l'arrêt

On a déjà vu le problème de l'arrêt : peut-on voir à partir du code d'un programme si celui-ci va s'arrêter ?

La réponse est non : c'est impossible. Le problème de l'arrêt et plusieurs autres sont impossibles à résoudre. On ne va pas étudier ces problèmes.

## Certains problèmes sont longs à calculer...

### On voudrait gagner aux échecs à tous les coups...

Supposons qu'on veuille écrire un programme qui joue et gagne à tous les coups aux échecs. Ce programme devrait calculer toutes les possibilités et choisir la suite de coup qui assure sa victoire... En vrai, on n'arrive pas à calculer plus de cinq ou six coups à l'avance. Et encore, il faut un très gros ordinateur. Il en va de même pour le go et plusieurs autres jeux.

# La petite exponentielle...

## $e^x$ et $\log x$

On connaît les fonctions très utiles en maths : exponentielle et logarithme.

Lorsque un programme prend un temps exponentiel par rapport à la taille de l'entrée, le calcul sera très long à moins de prendre de très petites valeurs. En effet, l'exponentielle croît plus vite que tout polynôme (c'est en fait un polynôme de degré infini).

C'est parfois utile : trouver le mot de passe de quelqu'un demande un nombre d'essais exponentiel, c'est pour cela qu'ils sont sûrs !

# Complexité en mémoire

## Retour aux échecs

Supposons qu'on aie beaucoup de temps devant nous (disons plusieurs fois l'âge de l'univers)... Il faudrait quand même enregistrer en mémoire chacune des étapes des échecs.

Les estimations actuelles sont qu'il y a environ  $10^{123}$  parties d'échecs "intéressantes". Il y aurait entre  $4 \times 10^{78}$  et  $6 \times 10^{79}$  atomes dans l'univers, nous avons donc besoin de  $10^{44}$  univers pour stocker nos données. Pour le go, il y a environ  $10^{600}$  parties différentes.

Autant dire que ces problèmes, même s'ils sont théoriquement résoluble, sont dans les faits incalculables.

# Pourquoi oublier les constantes ?

## L'informatique est changeante...

Revenons à des problèmes simples (trier un tableau par exemple). Selon l'ordinateur, les comparaisons, l'accès en mémoire et les autres actions exécutées par l'algorithme n'auront pas toujours le même coût. On veut donc caractériser la complexité d'un algorithme de façon indépendante à la machine qu'on utilisera.

## Rappel sur les opérateurs asymptotiques

On a  $O(n)$  qui signifie "au pire linéaire",  $\Omega(\log n)$  qui signifie "au mieux logarithmique" et  $\Theta(n^2)$  qui veut dire "Équivalent à du quadratique".

- 1 Introduction
  - Problèmes indécidables
  - Problèmes longs à résoudre
  - Le problème de l'exponentielle
  - Le problème de la mémoire
  - L'idée d'ordre de grandeur
- 2 **Complexité**
  - Définitions
  - Quelques problèmes...
- 3 Classes de complexité
  - Classes en temps
  - Classes en espace
  - Classes non-déterministes
  - Hiérarchie des classes

# Complexité d'un algorithme

## Complexité d'un algorithme

La complexité d'un algorithme est la classe asymptotique du temps d'exécution et celle de l'espace mémoire occupé par cet algorithme en fonction de la taille de l'argument donné en entrée. L'argument et la valeur renvoyée en sortie ne sont pas comptés dans la mémoire.

## Exemple

Le tri naïf sur un tableau (chercher le maximum, puis le second plus grand et ainsi de suite) a une complexité en temps  $\Theta(n^2)$  et une complexité en espace  $O(1)$ .

# Complexité d'un problème

## Complexité d'un problème

La complexité d'un problème est la complexité du meilleur algorithme pour ce problème.

## Exemple

La complexité du problème de tri par comparaison est en temps de  $O(n \log n)$  et en espace de  $O(1)$ .

# Quelques problèmes

## Quelques exemples et leur complexité en temps

- Trouver son chemin dans un graphe  $G = (V, E)$  :  
 $O(|E| + |V| \log |V|)$ .
- Trouver l'arbre couvrant dans un graphe :  $O(|E| \log |E|)$ .
- Être sur de gagner au go sur un plateau  $n \times n$  :  $O(2^{n^2})$ .
- Factoriser un nombre :  $O(2^n)$  (pour l'instant...).

## 1 Introduction

- Problèmes indécidables
- Problèmes longs à résoudre
- Le problème de l'exponentielle
- Le problème de la mémoire
- L'idée d'ordre de grandeur

## 2 Complexité

- Définitions
- Quelques problèmes...

## 3 Classes de complexité

- Classes en temps
- Classes en espace
- Classes non-déterministes
- Hiérarchie des classes

# Classes temporelles

## Les classes de complexité temporelles

On définit une classe de complexité temporelle l'ensemble des problèmes dont la complexité en temps est au pire dans une certaine classe asymptotique.

## Exemples

P, EXPTIME, 2EXPTIME,  $k$ -EXPTIME...

# Classes spatiales

## Les classes de complexité spatiales

On définit une classe de complexité spatiale l'ensemble des problèmes dont la complexité en espace est au pire dans une certaine classe asymptotique.

## Exemples

L, PSPACE, EXPSPACE...

# Machines non-déterministes

## Définition

Une machine non-déterministe est un ordinateur ayant droit à une opération en plus : il peut à tout moment “deviner” une valeur. L’ordinateur devra par la suite prouver que la valeur devinée fonctionne.

# Classes non-déterministes

## Les classes de complexité non-déterministes

Chaque classe de complexité a une classe dite non-déterministe associée, on la note  $N\text{Classe}$ .

## Exemples

NL, NP, NPSPACE, NEXPTIME, NEXPSPACE...

# Ordre des classes

Les classes par ordre croissant

# Ordre des classes

## Les classes par ordre croissant

- L

# Ordre des classes

## Les classes par ordre croissant

- L
- NL

# Ordre des classes

## Les classes par ordre croissant

- L
- NL
- P

# Ordre des classes

## Les classes par ordre croissant

- L
- NL
- P
- NP

# Ordre des classes

## Les classes par ordre croissant

- L
- NL
- P
- NP
- PSPACE, NPSPACE

# Ordre des classes

## Les classes par ordre croissant

- L
- NL
- P
- NP
- PSPACE, NPSPACE
- EXPTIME