

# Informatique théorique et pratique

## Terminale

Thomas Blanc

Terminale de l'école Alsacienne

20 septembre 2012

- 1 Introduction
  - Présentation du prof
  - Présentation du cours
- 2 L'algorithmique
  - Rappel historique
  - Preuve et complexité
  - Organisation de l'information
- 3 Les réseaux
  - Mise en place des réseaux
  - Les couches de bas niveau
  - Les couches de haut niveau
- 4 Les outils du cours
  - OCaml
  - Les TDs
  - La discussion

# Le prof, c'est qui ?

## Thomas Blanc

- Étudiant à l'ENS Cachan
- Master du MPRI (Master Parisien de Recherche en Informatique)
- En M2 au MVA (Master de Vision et d'Apprentissage)
- [thomas.blanc@crans.org](mailto:thomas.blanc@crans.org)

# Le cours, c'est quoi ?

## Qu'allons nous faire ?

- Une partie algorithmique
  - De la complexité
  - Des maths !
  - De la théorie des ensembles
  - Et de la théorie des graphes
- Une partie réseau
  - La mise en place
  - Le fonctionnement
  - Et l'utilisation !

# Algorithmique

- 1 Introduction
  - Présentation du prof
  - Présentation du cours
- 2 L'algorithmique
  - Rappel historique
  - Preuve et complexité
  - Organisation de l'information
- 3 Les réseaux
  - Mise en place des réseaux
  - Les couches de bas niveau
  - Les couches de haut niveau
- 4 Les outils du cours
  - OCaml
  - Les TDs
  - La discussion

# Un peu d'histoire

## Les premiers algorithmes

- 3000 ans avant JC : les Babyloniens écrivent des méthodes de résolutions d'équation
- 300 avant JC : Euclide écrit et prouve l'algorithme pour trouver le pgcd de deux nombres
- 813-833 : Al Khuwarizmi résout les équations du second degré

# Preuve et complexité d'un algorithme

## Preuve

On veut prouver que l'algorithme donnera le bon résultat, ce qui n'est pas toujours évident. On verra donc les notions d'invariant, d'ordre total et de convergence.

## Complexité

On aimerait aussi savoir en combien de temps un algorithme va s'exécuter, afin d'optimiser ce temps. Parfois, on pourra prouver que la complexité est optimale. On verra la complexité en temps et en espace mémoire.

# Les graphes

## Comment représenter une carte ?

Un graphe est un ensemble de points appelés sommets ou vertex, que l'on relie par des traits nommés arrêtes. Les arrêtes peuvent contenir une information d'orientation ou non. Sommets et arrêtes peuvent être étiquetés par diverses informations.

## Exemple :

On veut représenter une carte, chaque sommet est un croisement, chaque arrête est une route. On prend en compte le fait qu'une voie peut-être à sens unique, on enregistre aussi les limites de vitesse, les règles de priorité, la position des points, la répartition des numéros de rue, l'emplacement des travaux, des ralentisseurs etc. Cette carte peut alors être utilisée par un logiciel GPS pour calculer le meilleur chemin d'un point à un autre.

# Les arbres

## Comment optimiser une recherche ?

Comment trouver un mot dans le dictionnaire en vingt question dont on ne peut répondre que par oui ou non ? C'est possible grâce à notre ami le logarithme ! Plutôt que de ranger nos données dans une liste ou un tableau, on les rangera dans un arbre, qui est une structure facile à modifier (comme une liste) mais qui permet des accès rapide à toutes les données (comme un arbre).

# Connexion !

- 1 Introduction
  - Présentation du prof
  - Présentation du cours
- 2 L'algorithmique
  - Rappel historique
  - Preuve et complexité
  - Organisation de l'information
- 3 **Les réseaux**
  - Mise en place des réseaux
  - Les couches de bas niveau
  - Les couches de haut niveau
- 4 Les outils du cours
  - OCaml
  - Les TDs
  - La discussion

# Les diverses parties du réseau

## Les couches OSI

- 1 Physique (RJ-45)
- 2 Liaison (MAC)
- 3 Réseau (IP)
- 4 Transport (TCP)
- 5 Session
- 6 Présentation (SSL/TLS)
- 7 Application (HTTP, FTP, SSH etc.)

Nous verrons (presque) toutes ces couches !

# Gérer le transport

## Les problèmes physiques :

- Comment assurer qu'un paquet arrive à destination ?
- Comment assurer qu'un paquet n'est pas modifié ?
- Comment réparer les éventuelles modifications ?
- Le chemin d'un point à un autre, comment on le trouve ?

# Utiliser le réseau

## Les problèmes logiciels :

- Comment sécuriser une connexion ?
- Comment gérer de nombreuses connexions à la fois ?
- Et pour écrire un programme qui fait du réseau, comment on fait ?
- À quoi ressemble un protocole réseau ?

# Petite présentation d'OCaml

## OCaml, c'est un langage

- ...fonctionnel et déclaratif
- ...modulaire
- ...compilé
- ...fortement typé

# L'organisation des TDs

Il y aura TD environ toutes les 2-3 semaines. Vous devrez chacun rendre au moins un TD par trimestre.  
Votre meilleur TD comptera dans la note.

# Posez des questions !

Ce cours est le vôtre !  
Des questions ? Des remarques ?