

# Algorithmique Avancée et Complexité: Présentation du cours AAC

Sophie Tison-USTL-Master1 Informatique

# Objectifs

Avoir des outils pour concevoir un "bon" algorithme pour résoudre un problème.

# Objectifs

Avoir des outils pour concevoir un "bon" -c.à.d. **correct et efficace** - algorithme pour résoudre un problème.

Cela pose de nombreuses questions...et  
demande pas mal de savoir-faire...

Existe-il un algorithme pour résoudre le problème?

Cela pose de nombreuses questions...et  
demande pas mal de savoir-faire...

Existe-il un algorithme pour résoudre le problème?

Connaître quelques Notions de Calculabilité et de décidabilité.

Cela pose de nombreuses questions...et  
demande pas mal de savoir-faire.

Est-ce un problème classique?

Cela pose de nombreuses questions...et  
demande pas mal de savoir-faire.

Est-ce un problème classique?

Connaître et savoir reconnaître des grands classiques

Cela pose de nombreuses questions...et  
demande pas mal de savoir-faire.

Est-ce un problème classique?

Connaître et savoir reconnaître des grands classiques

Tris, méthodes de sélection, recherche



Cela pose de nombreuses questions...et  
demande pas mal de savoir-faire.

Est-ce un problème classique?

Connaître et savoir reconnaître des grands classiques

Tris, méthodes de sélection, recherche  
algorithmique des graphes,

Cela pose de nombreuses questions...et  
demande pas mal de savoir-faire.

Est-ce un problème classique?

Connaître et savoir reconnaître des grands classiques

Tris, méthodes de sélection, recherche  
algorithmique des graphes,  
méthodes de hachages

Cela pose de nombreuses questions...et  
demande pas mal de savoir-faire.

Est-ce un problème classique?

Connaître et savoir reconnaître des grands classiques

Tris, méthodes de sélection, recherche  
algorithmique des graphes,  
méthodes de hachages  
Programmation linéaire...

Cela pose de nombreuses questions...et  
demande pas mal de savoir-faire.

Est-ce un problème classique?

Connaître et savoir reconnaître des grands classiques

Tris, méthodes de sélection, recherche  
algorithmique des graphes,  
méthodes de hachages  
Programmation linéaire...

....

Cela pose de nombreuses questions...et  
demande pas mal de savoir-faire

Comment concevoir un algorithme?

Cela pose de nombreuses questions...et  
demande pas mal de savoir-faire

Comment concevoir un algorithme?

Schémas d'algorithmes, Algorithmic design patterns

Cela pose de nombreuses questions...et  
demande pas mal de savoir-faire

Comment concevoir un algorithme?

Schémas d'algorithmes, Algorithmic design patterns

"Diviser pour régner"

Cela pose de nombreuses questions...et  
demande pas mal de savoir-faire

Comment concevoir un algorithme?

Schémas d'algorithmes, Algorithmic design patterns

"Diviser pour régner"  
Programmation Dynamique



# Cela pose de nombreuses questions...et demande pas mal de savoir-faire

## Comment concevoir un algorithme?

Schémas d'algorithmes, Algorithmic design patterns

"Diviser pour régner"  
Programmation Dynamique  
Algorithmes gloutons

Cela pose de nombreuses questions...et  
demande pas mal de savoir-faire.

L'algorithme est-il correct?

Cela pose de nombreuses questions...et  
demande pas mal de savoir-faire.

L'algorithme est-il correct?

Savoir prouver un algorithme...  
ou tout du moins avoir un minimum de rigueur

Cela pose de nombreuses questions... et  
demande pas mal de savoir-faire.

L'algorithme est-il efficace?

Cela pose de nombreuses questions... et demande pas mal de savoir-faire.

L'algorithme est-il efficace?

Savoir analyser la complexité d'algorithmes

Cela pose de nombreuses questions...et  
demande de multiples compétences.

Peut-on trouver un algorithme plus efficace pour le problème?  
Est-ce un problème dur?

Cela pose de nombreuses questions...et  
demande de multiples compétences.

Peut-on trouver un algorithme plus efficace pour le problème?  
Est-ce un problème dur?

...Avoir quelques notions de Complexité des problèmes

Cela pose de nombreuses questions...et  
demande pas mal de savoir-faire.

Si le problème est dur, comment l'appréhender!



Cela pose de nombreuses questions...et  
demande pas mal de savoir-faire.

Si le problème est dur, comment l'appréhender!

Connaître quelques techniques d' Algorithmique Avancée:  
Méta-heuristiques, Algorithmes probabilistes,...  
Backtracking, minmax, séparation-évaluation

# Ce que nous ferons

Quelques schémas d'algorithmes (2-3 cours)

Un peu de complexité de problèmes (2-3 cours)

Un peu d'algorithmique avancée (2-3 cours)

Quelques notions de décidabilité et calculabilité (1-2 cours)

# Ce que nous ferons

Quelques schémas d'algorithmes (2-3 cours)

Un peu de complexité de problèmes (2-3 cours)

Un peu d'algorithmique avancée (2-3 cours)

Quelques notions de décidabilité et calculabilité (1-2 cours)

# Ce que nous ferons

Quelques schémas d'algorithmes (2-3 cours)

Un peu de complexité de problèmes (2-3 cours)

Un peu d'algorithmique avancée (2-3 cours)

Quelques notions de décidabilité et calculabilité (1-2 cours)

# Ce que nous ferons

Quelques schémas d'algorithmes (2-3 cours)

Un peu de complexité de problèmes (2-3 cours)

Un peu d'algorithmique avancée (2-3 cours)

Quelques notions de décidabilité et calculabilité (1-2 cours)

# Bibliographie : de nombreuses ressources en ligne

- . Le dictionnaire recensant les algorithmes et problèmes classiques du NIST
- . The Stony Brook Algorithm Repository” qui contient des implémentations d’algorithmes pour des dizaines de problèmes classiques,
- . Algorithms Courses on the WWW, qui contient une collection de cours d’algorithmique,
- . Le site du cours “Algorithms in the Real World”,
- . “A compendium of NP optimization problems”:

# Bibliographie : de nombreuses ressources en ligne

- . Le dictionnaire recensant les algorithmes et problèmes classiques du NIST
- . The Stony Brook Algorithm Repository” qui contient des implémentations d’algorithmes pour des dizaines de problèmes classiques,
- . Algorithms Courses on the WWW, qui contient une collection de cours d’algorithmique,
- . Le site du cours ”Algorithms in the Real World”,
- . ”A compendium of NP optimization problems”:

# Bibliographie : de nombreuses ressources en ligne

- . Le dictionnaire recensant les algorithmes et problèmes classiques du NIST
- . The Stony Brook Algorithm Repository” qui contient des implémentations d’algorithmes pour des dizaines de problèmes classiques,
- . Algorithms Courses on the WWW, qui contient une collection de cours d’algorithmique,
- . Le site du cours “Algorithms in the Real World”,
- . “A compendium of NP optimization problems”:



# Bibliographie : de nombreuses ressources en ligne

- . Le dictionnaire recensant les algorithmes et problèmes classiques du NIST
- . The Stony Brook Algorithm Repository” qui contient des implémentations d’algorithmes pour des dizaines de problèmes classiques,
- . Algorithms Courses on the WWW, qui contient une collection de cours d’algorithmique,
- . Le site du cours ”Algorithms in the Real World”,
- . ”A compendium of NP optimization problems”:

# Bibliographie : de nombreuses ressources en ligne

- . Le dictionnaire recensant les algorithmes et problèmes classiques du NIST
- . The Stony Brook Algorithm Repository” qui contient des implémentations d’algorithmes pour des dizaines de problèmes classiques,
- . Algorithms Courses on the WWW, qui contient une collection de cours d’algorithmique,
- . Le site du cours ”Algorithms in the Real World”,
- . ”A compendium of NP optimization problems”:

# Bibliographie : de nombreux livres

.Cormen, Leiserson, Rivest, "Introduction à l'algorithmique", Dunod (disponible à la BU) vraiment "une" référence essentielle en algorithmique,

. S. Skiena, "Algorithm Design Manual", une "mine"! (Une version on-line proche du livre papier ),

. Sur l'aspect "algorithmic pattern", "Data Structures and Algorithms with object-oriented design patterns in Java". 2000, disponible sur le Web , ...

# Bibliographie : de nombreux livres

. Cormen, Leiserson, Rivest, "Introduction à l'algorithmique", Dunod (disponible à la BU) vraiment "une" référence essentielle en algorithmique,

. S. Skiena, "Algorithm Design Manual", une "mine"! (Une version on-line proche du livre papier ),

. Sur l'aspect "algorithmic pattern", "Data Structures and Algorithms with object-oriented design patterns in Java". 2000, disponible sur le Web , ...

# Bibliographie : de nombreux livres

. Cormen, Leiserson, Rivest, "Introduction à l'algorithmique", Dunod (disponible à la BU) vraiment "une" référence essentielle en algorithmique,

. S. Skiena, "Algorithm Design Manual", une "mine"! (Une version on-line proche du livre papier ),

. Sur l'aspect "algorithmic pattern", "Data Structures and Algorithms with object-oriented design patterns in Java". 2000, disponible sur le Web , ...

# Organisation: les TPs

Il y aura 6 séances de TP (langage : Java) encadrées pour la mise en oeuvre directe des méthodes étudiées en cours:

- . Programmation dynamique, Algorithmes gloutons(2 séances)
- . Propriétés NP, réductions polynômiales (2 séances)
- . Heuristiques, Métaheuristiques (2 séances)

# Organisation: les TPs

Il y aura 6 séances de TP (langage : Java) encadrées pour la mise en oeuvre directe des méthodes étudiées en cours:

- . Programmation dynamique, Algorithmes gloutons(2 séances)
- . Propriétés NP, réductions polynômiales (2 séances)
- . Heuristiques, Métaheuristiques (2 séances)

# Organisation: les TPs

Il y aura 6 séances de TP (langage : Java) encadrées pour la mise en oeuvre directe des méthodes étudiées en cours:

- . Programmation dynamique, Algorithmes gloutons(2 séances)
- . Propriétés NP, réductions polynômiales (2 séances)
- . Heuristiques, Métaheuristiques (2 séances)



# Organisation: les TPs

Il y aura 6 séances de TP (langage : Java) encadrées pour la mise en oeuvre directe des méthodes étudiées en cours:

- . Programmation dynamique, Algorithmes gloutons(2 séances)
- . Propriétés NP, réductions polynômiales (2 séances)
- . Heuristiques, Métaheuristiques (2 séances)

# Organisation.... :l'évaluation

Le contrôle continu sera basé sur les TPs et un DS en milieu de semestre. La note de contrôle continu sera  $1/3 * (\textit{note DS}) + 2/3 * (\textit{note TP})$  avec éventuellement un bonus donné par des "devoirs maison".

# Quelques exemples

.Problème 1: Trouver le plus court chemin  
entre deux sommets d'un graphe

.Problème 2: Trouver le chemin le plus long  
-sans cycle- dans un graphe

# Quelques exemples

.Problème 1: Trouver le plus court chemin  
entre deux sommets d'un graphe

.Problème 2: Trouver le chemin le plus long  
-sans cycle- dans un graphe

# Quelques exemples

.Problème 1: Trouver un chemin qui passe une et une seule fois par tous les sommets d'un graphe

.Problème 2: Trouver un chemin qui passe une et une seule fois par tous les arcs d'un graphe

# Quelques exemples

.Problème 1: Trouver un chemin qui passe une et une seule fois par tous les sommets d'un graphe

.Problème 2: Trouver un chemin qui passe une et une seule fois par tous les arcs d'un graphe

# Quelques exemples

.Problème 1: Planifier  $n$  cours dont les horaires sont donnés dans  $k$  salles

. Problème 2: Planifier  $n$  cours dans  $k$  créneaux, en connaissant les incompatibilités entre deux cours

# Quelques exemples

.Problème 1: Planifier  $n$  cours dont les horaires sont donnés dans  $k$  salles

. Problème 2: Planifier  $n$  cours dans  $k$  créneaux, en connaissant les incompatibilités entre deux cours