

UNIVERSITE MOHAMED KHIDER – BISKRA  
FACULTE DES SCIENCES EXACTES, DES SCIENCES DE LA NATURE ET DE LA VIE  
DEPARTEMENT D'INFORMATIQUE

# Complexité et Optimisation

---

**Support de cours et TD**  
Master d'informatique

Réalisé par : Dr. S. SLATNIA

**Version améliorée**

05/01/2016

## Table des matières détaillées

### Objectifs de cours

#### **CHAPITRE 1. Complexité des algorithmes et mesure de performance**

##### **Partie 1. Théorie de la complexité**

1. Qualités d'un bon algorithme  
Quelques problèmes algorithmiques classiques  
Qualités d'un bon algorithme
2. Analyse et notions de complexité
  - 2.1. Critères d'évaluation de performance de l'algorithme
    - 2.1.1. Calcul de nombre d'opérations
    - 2.1.2. Mesure d'espace mémoire
  - 2.2. Coût d'un algorithme
  - 2.3. Complexité d'un algorithme
  - 2.4. Complexité d'un problème
  - 2.5. Comment calculer la complexité d'un algorithme ?
  - 2.6. Type de complexité
    - 2.6.1. Complexités temporelle et spatiale d'un algorithme
    - 2.6.2. Complexités pratique et théorique
  - 2.7. Notion d'optimalité

##### **Partie 2. Grandeur des fonctions, la Complexité Asymptotique et Calcul de complexité des algorithmes itératifs et récursifs**

1. Optimisation d'exécution de la récursivité
  - 1.1. Définition et types de récursivité
  - 1.2. Critères de terminaison pour un bon algorithme récursif
  - 1.3. Comment gérer la récursivité ?
    - 1.3.1. Gestion de la récursivité sans la pile : Récursivité Terminale (RT)
    - 1.3.2. Gestion de la récursivité avec la pile : Récursivité Enveloppée (RE)
  - 1.4. Optimisation d'exécution de la récursivité
    - 1.4.1. Comparaison d'algorithmes
2. C'est quoi les notions de Landau ?
  - 2.1. Classes de complexité
  - 2.2. Notations asymptotique
    - 2.2.1. Domination asymptotique
    - 2.2.2. Borne Inferieur, Notation  $\Omega$
    - 2.2.3. Equivalence asymptotique
  - 2.3. Calcul de complexité asymptotique
3. Calcul de complexité des algorithmes itératifs et récursifs

- 3.1. Coût uniforme et coût logarithmique
  - 3.1.1. A propos de la notation O
- 3.2. Calcul de complexité des algorithmes itératifs
- 3.3. Calcul de complexité des algorithmes récursifs
- 3.3.1. Master théorème et Équations de récurrences

## **CHAPITRE 2. Classes et complexité d'un problème**

- 1. Complexité d'un problème, Présentation et définitions
- 2. Trois méthodes pour trouver une borne inférieure
- 3. Classes de complexité d'un problème
  - 3.1. Classes de complexité en temps
    - 3.1.1. Classe P
    - 3.1.2. NP
    - 3.1.3. EXPTIME
  - 3.2. Classes de complexité en espace
- 4. NP-complétude
- 5. Réduction d'un problème Q à un problème  $\Pi$

## **CHAPITRE 3. Optimisation de la complexité et la stratégie Diviser pour Régner (Divide&Conquer)**

- 1. Méthode Diviser pour Régner
  - 1.1. Diviser pour Régner : tri fusion
  - 1.2. Diviser pour Régner : produit de deux matrices
- 2. Optimisation de la complexité et la dichotomie
  - 2.1. L'élément majoritaire et la dichotomie
  - 2.2. Optimisation et Algorithmes Avancés, comparaison de complexité
    - 2.2.1. Algorithmes de tri
    - 2.2.2. Algorithmes de recherche

## **CHAPITRE 4. Optimisation combinatoire, méthodes exactes**

- 1. Définitions
  - 1.1. Définition d'optimisation
  - 1.2. Définition de problème d'optimisation
- 2. Programmation dynamique
- 3. Branch and bound
- 4. Recherche arborescente, l'exploration sans information
  - 1.1. Structures de données pour les graphes
  - 1.2. Parcours de graphe
    - 1.2.1. Parcours en largeur
    - 1.2.2. Parcours en profondeur
  - 1.3. Backtracking (retour sur trace)
    - 1.3.1. Application du backtracking dans les arbres de jeux
      - 1.3.1.1. Principe de MIN-MAX

## **CHAPITRE 5. Méthodes approchées, heuristiques spécialisées**

- 1. Algorithme Glouton

# Table des matières

---

- 2. L'exploration avec information
  - 3.1. Définitions
  - 3.2. Algorithme Recherche Locale
  - 3.3. Algorithme A\*
  - 3.4. Algorithme Hill Climbing

## Table des figures

<b>Figure 1.</b> Calcul factorielle, $f(3)$ .....	15
<b>Figure 2.</b> Notations asymptotique [5] .....	20
<b>Figure 3.</b> Classes de complexité .....	21
<b>Figure 4.</b> La fonction Divide pour éclater la donnée .....	35
<b>Figure 5.</b> La stratégie Diviser pour Régner (Divide&Conquer).....	36
<b>Figure 6.</b> Comment Tri-Sélection trie le tableau [7, 1, 15, 8, 2].....	47
<b>Figure 7.</b> Exemple d'arbre .....	52
<b>Figure 8.</b> Exemple de tas .....	53
<b>Figure 9.</b> Implémentation par un tableau .....	54
<b>Figure 10.</b> Tri dans le pire des cas .....	58
<b>Figure 11.</b> Arbre des niveaux de joue de la configuration de la figure 10.....	60
<b>Figure 12.</b> Réduction d'un problème Q à un problème $\Pi$ .....	60
<b>Figure 13.</b> Exemple d'un placement optimal de pièces 2D .....	61
<b>Figure 14.</b> La fonction d'optimisation .....	61
<b>Figure 15.</b> Exemple de fonction d'optimisation .....	65