

Table des matières

Objectifs de cours	1
Chapitre 1. Théorie de la complexité et mesure de performance	3
Partie 1. Théorie de la complexité et mesure de performance	4
1. Généralités	4
1.1. Quelques problèmes algorithmiques classiques	6
1.2. Qualités d'un bon algorithme	7
2. Analyse et notions de complexité	7
2.1. Critères d'évaluation de performance de l'algorithme	8
2.1.1. Calcul de nombre d'opérations	8
2.1.2. Mesure d'espace mémoire	8
2.2. Coût d'un algorithme	9
2.3. Complexité d'un algorithme	9
2.4. Complexité d'un problème	10
2.5. Comment calculer la complexité d'un algorithme?	10
2.6. Type de complexité	11
2.6.1. Complexités temporelle et spatiale d'un algorithme	11
2.6.2. Complexités pratique et théorique	11
2.6. Notion d'optimalité	12
Partie 2. Classes et complexité d'un problème	13
1. Complexité d'un problème, Présentation et définitions	13
2. Trois méthodes pour trouver une borne inférieure	14
2.1. Les méthodes dites d'oracle ou d'adversaire.....	14
2.2. Les méthodes dites d'oracle ou d'adversaire	14
2.3. Les Réductions	14
3. Classes de complexité d'un problème.....	14
3.1. Classes de complexité en temps	14
3.1.1. Classe P	15
3.1.2. NP.....	15
3.1.3. EXPTIME	15
3.2. Classes de complexité en espace.....	16
4. NP-complétude.....	17
5. Réduction d'un problème Q à un problème Π	17
Chapitre 2. Evaluation des algorithmes récursifs, Grandeur des fonctions et la Complexité Asymptotique	19
1. Structures de données.....	19
2. Qu'est ce que la récursivité ?.....	19
3. Types de récursivité	20

4. Critères de terminaison pour un bon algorithme récursif	21
5. Comment gérer la récursivité ?.....	21
5.1. Gestion de la récursivité sans la pile : Récursivité Terminale (RT)	22
5.2. Gestion de la récursivité avec la pile : Récursivité Enveloppée (RE).....	22
6. Optimisation d'exécution de la récursivité	22
6.1. Comparaison d'algorithmes	23
7. C'est quoi les notions de Landau ?	25
7.1. Classes de complexité	25
7.2. Notations asymptotique	26
7.2.1. Domination asymptotique	26
7.2.2. Notation Ω	26
7.2.3. Equivalence asymptotique	27
7.3. Calcul de complexité asymptotique	27
Chapitre 3. Calcul de complexité des algorithmes itératifs et récursifs	30
1. Coût uniforme et coût logarithmique	30
1.1. A propos de la notation O	30
2. Calcul de complexité des algorithmes itératifs	31
3. Calcul de complexité des algorithmes récursifs	32
3.1. Master théorème et Équations de récurrences	33
Chapitre 4. La stratégie Diviser pour Régner (Divide&Conquer)	35
1. Méthode Diviser pour Régner	35
2. Diviser pour Régner : tri fusion	36
3. Diviser pour Régner : produit de deux matrices	39
4. L'élément majoritaire et la dichotomie.....	42
Chapitre 5. Algorithmes de tri et de recherche	46
1. Présentation des différentes méthodes de tri	46
1.1. Algorithme de Tri naïfs (tri par sélection)	47
1.2. Algorithme de Tri Fusion (Merge Sort).....	48
1.3. Le tri rapide (quicksort).....	50
1.4. Le tri par Tas (Heapsort)	52
2. Présentation des différentes méthodes de recherche	55
2.1. Recherche séquentielle : recherche dans un tableau non trié	55
2.2. Recherche séquentielle : recherche dans un tableau trié	56
2.3. Recherche dichotomique	56
Chapitre 6. Exploration des graphes et algorithme des arbres	58
1. Structures de données pour les graphes	58
2. Parcours de graphe	59
2.1. Parcours en largeur	59
2.2. Parcours en profondeur	59
3. Backtracking (retour sur trace).....	60
3.1.1. Application du backtracking dans les arbres de jeux.....	60
3.1.1.1. Principe de MIN-MAX	61

4. Algorithmes des arbres recouvrant optimaux	64
Chapitre 7. Méthodes heuristiques d'optimisation	68
1.1. Définition d'optimisation.....	68
1.2. Problème d'optimisation.....	68
1.3. Optimisation combinatoire	69
1.3.1. Méthodes exactes, Branch and bound.....	69
1.3.2. Méthodes approchées: Algorithme Glouton	70
1.3.2. Quelques Méthodes heuristiques	72
1.3.2.1. Définitions	72
1.3.2.2. Algorithme Recherche Locale	74
1.3.2.2. Algorithme A*	75
1.3.2.2. Algorithme Hill Climbing	76

Table des figures

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