

Table des matières

Introduction	1
1. Environnement et conventions.....	7
1. Les étapes de développement d'une application	7
1.1. L'analyse	7
1.2. L'algorithme	10
1.3. L'écriture du programme dans un langage de programmation	11
2. Le pseudo-langage.....	18
2.1. Son rôle	18
2.2. Conventions syntaxiques.....	18
3. Le génie logiciel	20
3.1. La modélisation des données	20
3.2. La modularité du programme	20
3.3. La lisibilité du programme.....	20
3.4. Privilégier les solutions simples.....	21
3.5. L'utilisation de conventions d'écriture.....	21
4. La performance algorithmique	21
Résumé.....	22
Exercices	22
Solutions.....	23
2. Les traitements logiques.....	29
1. Les tests ou instructions conditionnelles	29
1.1. Présentation	29
1.2. La logique booléenne	30
1.3. L'instruction SI ALORS SINON	31
1.4. Le choix multiple	33
2. Les boucles ou instructions répétitives	37
2.1. Présentation des boucles	37
2.2. Comprendre le fonctionnement d'une boucle.....	46
2.3. Construire une boucle.....	49
3. Les sous-programmes.....	54
3.1. Les procédures	54
3.2. Les fonctions	54
3.3. Organisation en modules autonomes	55
4. La complexité algorithmique	61
4.1. Principe	61
4.2. L'analyse de la complexité	62
4.3. Les grandes familles de complexité.....	62
4.4. Choisir un algorithme	64
Résumé	64

Exercices	64
Solutions.....	67
3. La gestion des données.....	93
1. Principe de traitement des données	93
1.1. Le rôle des variables	93
1.2. Le rôle des fichiers.....	94
2. Les tableaux	94
2.1. Description	94
2.2. L'allocation mémoire.....	95
2.3. Gestion d'un tableau	96
3. Les enregistrements	121
3.1. Description	122
3.2. Gestion d'un enregistrement	122
4. Les pointeurs.....	126
4.1. Description	126
4.2. Les pointeurs et l'allocation statique de la mémoire	127
4.3. L'allocation dynamique de la mémoire	129
4.4. La particularité de Java.....	130
4.5. La particularité de PHP	130
4.6. La particularité de Python	130
5. Les listes chaînées	130
5.1. Description	130
5.2. L'allocation mémoire.....	131
5.3. Gestion d'une liste chaînée	132
6. Gestion des données complexes.....	159
6.1. Les tableaux d'enregistrements	159
6.2. Les listes d'enregistrements.....	165
7. Variantes sur les tableaux	166
7.1. Les tableaux dynamiques	166
7.2. Les tableaux de pointeurs	170
Résumé	172
Exercices	172
Solutions.....	173
4. La récursivité	195
1. Principe	195
2. Diviser pour résoudre	200
3. Suppression de la récursion	201
4. Récursivité croisée	203
Résumé	206
Exercices	206
Solutions.....	208
5. Les données abstraites	219
1. Les piles	219
1.1. Principe	219
1.2. Les primitives	220

1.3. Utilisation des tableaux.....	220
1.4. Utilisation des listes chaînées	221
1.5. Exemples	223
1.6. Les piles et la récursivité	234
1.7. La classe <i>C++ stack</i>	236
1.8. La classe <i>Java Stack</i>	236
1.9. La classe <i>PHP SplStack</i>	237
1.10. Les listes <i>Python en Piles</i>	238
1.11. La classe <i>Python deque</i>	238
2. Les files	238
2.1. Principe	238
2.2. Utilisation des tableaux.....	239
2.3. Utilisation des listes chaînées	240
2.4. La classe <i>C++ deque</i>	241
2.5. L'interface <i>Java Queue</i> avec la classe <i>LinkedList</i>	242
2.6. La classe <i>PHP SplQueue</i>	243
2.7. Les listes <i>Python en Files</i>	244
2.8. La classe <i>Python deque</i>	244
3. Les arbres	245
3.1. Principe et définitions	245
3.2. Les types d'arbres.....	246
3.3. Création d'un arbre binaire d'expression.....	247
3.4. Parcours d'un arbre binaire.....	249
3.5. Les arbres de recherche	268
Résumé	269
Exercices	270
Solutions.....	271
6. Les tris.....	317
1. Tris élémentaires.....	317
1.1. Le tri par sélection.....	317
1.2. Le tri par insertion.....	325
1.3. Le tri à bulles	329
1.4. Le tri par comptage	333
2. Tris avancés.....	334
2.1. Le tri shell	334
2.2. Le tri indirect	340
2.3. Le tri rapide ou <i>quicksort</i>	351
Résumé	359
Exercices	359
Solutions.....	360
7. Les recherches	371
1. La recherche séquentielle	371
2. La recherche dichotomique	376
2.1. Principe	376
2.2. Version récursive.....	376
2.3. Version itérative	381
3. La recherche par interpolation	382

4. Tables de hachage ou adressage dispersé.....	383
4.1. Principe	383
4.2. La fonction de hachage	384
4.3. Table de hachage chaînée	386
4.4. Table de hachage à adressage ouvert	403
5. Les arbres de recherche équilibrés.....	407
5.1. Principe	407
5.2. Fonctionnement d'un arbre AVL	408
5.3. Gestion du facteur d'équilibre	409
5.4. Rotations des arbres AVL	410
Résumé.....	414
Exercices.....	415
Solutions.....	416
8. Les méthodes numériques.....	463
1. Interpolation polynomiale.....	463
1.1. Principe	463
1.2. Rappel sur les polynômes.....	464
1.3. Le polynôme d'interpolation.....	465
1.4. Le programme	469
1.5. La fonction <i>interpolate.interp1d</i> de la bibliothèque <i>Python scipy</i>	484
2. Méthode des moindres carrés	486
2.1. Principe	486
2.2. Rappel sur le calcul des distances	487
2.3. Le programme	488
2.4. La fonction <i>linalg.lstsq</i> de la bibliothèque <i>Python numpy</i>	498
3. Recherche des solutions d'équations.....	500
3.1. Principe de la méthode de Newton.....	500
3.2. Rappel sur les dérivées.....	501
3.3. Choix du point de départ pour la méthode de Newton	504
3.4. Le programme	506
3.5. La fonction <i>optimize.bisect</i> de la bibliothèque <i>Python scipy</i>	541
Résumé.....	542
9. Les algorithmes classiques.....	543
1. Algorithme du plus court chemin de Dijkstra.....	543
1.1. Rappel sur les graphes	543
1.2. Description de l'algorithme.....	544
1.3. L'algorithme	547
1.4. Application au calcul d'itinéraire.....	573
2. Algorithme de compression de données de Huffman.....	574
2.1. Principe de la compression de données.....	574
2.2. Description de l'algorithme.....	575
2.3. L'algorithme	579
Résumé.....	637
Index.....	639