

Éléments d'algorithmique

D. Beauquier, J. Berstel, Ph. Chrétienne

6 février 2005

© D. Beauquier, J. Berstel, Ph. Chrétienne, 2003
Première édition par Masson, 1992.

Version 6 février 2005

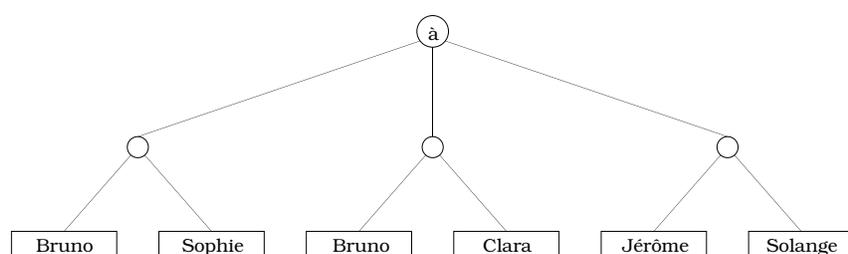


Table des matières

Avant-propos	XIV
1 Préliminaires	1
1.1 Les algorithmes et leur coût	1
1.1.1 Algorithmes	1
1.1.2 Problèmes intraitables	2
1.1.3 Sur la présentation d'algorithmes	3
1.2 Mesures du coût	4
1.2.1 Coût dans le cas le plus défavorable	4
1.2.2 Coût moyen	4
1.2.3 Coût amorti	6
1.3 Une borne inférieure	9
2 Evaluations	13
Introduction	13
2.1 Notations de Landau	14
2.1.1 Notation O	14
2.1.2 Notations Ω et θ	16
2.1.3 Exemples	17
2.2 Récurrences	20
2.2.1 Récurrences linéaires à coefficients constants	20
2.2.2 Récurrences diverses	26
2.2.3 Récurrences de partitions	29
2.2.4 Récurrences complètes	33
Notes	35
Exercices	35
3 Structures de données	37
3.1 Types de données et structures de données	37
3.2 Les structures linéaires	39
3.2.1 Piles	40
3.2.2 Files	42
3.2.3 Listes	44
3.3 Arbres	51
3.3.1 Arbres binaires	51
3.3.2 Dictionnaires et arbres binaires de recherche	52

3.3.3	Arborescences	55
3.4	Files de priorité	56
3.5	Gestion des partitions	61
3.5.1	Le problème « union-find »	61
3.5.2	Union pondérée et compression des chemins	62
3.5.3	Preuve du théorème	65
Notes	69
Exercices	70
4	Graphes	73
Introduction	73
4.1	Définitions et propriétés élémentaires	74
4.1.1	Définitions	74
4.1.2	Implémentations d'un graphe	75
4.1.3	Chemins, chaînes, circuits, cycles	76
4.1.4	Lemme de König	77
4.1.5	Graphes sans circuit	78
4.2	Accessibilité	80
4.2.1	Algorithme de Roy-Warshall	81
4.2.2	Autres problèmes d'accessibilité	82
4.2.3	Semi-anneaux et accessibilité	86
4.2.4	Forte connexité	89
4.3	Arbres et arborescences	90
4.3.1	Arbres	90
4.3.2	Arborescences	93
4.3.3	Arborescences ordonnées	94
4.3.4	Arbres positionnés et arbres binaires	95
4.3.5	Arbre binaire complet	95
4.4	Parcours d'un graphe	97
4.4.1	Parcours d'un graphe non orienté	97
4.4.2	Parcours en profondeur	102
4.4.3	Parcours en largeur	104
4.4.4	Parcours d'un graphe orienté	106
4.4.5	Calcul des composantes fortement connexes	111
4.4.6	Parcours d'une arborescence	115
Notes	117
Exercices	117
5	Tris	121
Introduction	121
5.1	Tri interne	122
5.1.1	Le tri rapide	122
5.1.2	Le tri fusion	128
5.1.3	Le tri par tas	130
5.2	Tri externe	131
5.2.1	Construction des monotopies	133
5.2.2	Répartition des monotopies	134

Notes	140
Exercices	140
6 Arbres et ensembles ordonnés	145
Introduction	145
6.1 Arbres de recherche	147
6.1.1 Définition	147
6.1.2 Rotations	150
6.2 Arbres AVL	152
6.2.1 Définition	152
6.2.2 Insertion	154
6.2.3 Suppression	156
6.2.4 Arbres balisés	159
6.3 Arbres a - b	160
6.3.1 Définition	160
6.3.2 Recherche d'un élément	161
6.3.3 Insertion d'un élément	161
6.3.4 Suppression d'un élément	163
6.3.5 Concaténation et scission	167
6.3.6 Coût amorti des arbres 2-4	169
6.4 Arbres bicolores	173
6.4.1 Présentation	173
6.4.2 Hauteur d'un arbre bicolore	175
6.4.3 Insertion dans un arbre bicolore	177
6.4.4 Suppression d'un élément dans un arbre bicolore	179
6.5 Enrichissement	182
6.6 Arbres persistants	186
6.6.1 Ensembles ordonnés persistants	186
6.6.2 Duplication de chemins	188
6.6.3 Méthode de duplication des sommets pleins	195
Notes	203
Exercices	203
7 Graphes valués	211
Introduction	211
7.1 Arbre couvrant de coût minimum	212
7.1.1 Définition du problème	212
7.1.2 Propriétés des arbres optimaux	212
7.1.3 Algorithme général	214
7.1.4 Algorithmes spécifiques	214
7.2 Chemins de coût minimum	219
7.2.1 Définition du problème	220
7.2.2 Existence d'une solution	221
7.2.3 Itération fondamentale	223
7.2.4 Algorithme de Ford	224
7.2.5 Graphe sans circuit	226
7.2.6 Algorithme de Dijkstra	226

7.2.7	Algorithme A^*	229
7.2.8	Algorithme « PAPS »	232
Notes	234
Exercices	235
8	Flots	239
8.1	Préliminaires	240
8.2	Flots d'un réseau	241
8.2.1	Existence d'un flot compatible de coût minimum	242
8.2.2	Quelques problèmes particuliers	244
8.2.3	Trois propriétés fondamentales	246
8.3	Problème du flot maximum	251
8.3.1	L'algorithme générique de Ford et Fulkerson	255
8.3.2	L'algorithme des distances estimées au puits	257
8.3.3	L'algorithme du préflot	263
8.3.4	L'algorithme de Karzanov	269
8.3.5	L'algorithme des excès échelonnés	270
8.4	Flot de coût minimum	272
8.4.1	Graphe d'écart et conditions d'optimalité	273
8.4.2	Problème dual et conditions d'optimalité	273
8.4.3	Un algorithme primal	275
8.4.4	Flot maximum de coût minimum	280
8.4.5	Plan de transport de coût minimum	285
Notes	289
Exercices	290
9	Automates	293
	Introduction	293
9.1	Mots et langages	294
9.2	Automates finis	295
9.2.1	Définition	295
9.2.2	Exemples	296
9.2.3	Automates déterministes	297
9.3	Opérations	301
9.3.1	Opérations booléennes	301
9.3.2	Automates asynchrones	301
9.3.3	Produit et étoile	303
9.4	Langages rationnels	303
9.4.1	Langages rationnels : définitions	303
9.4.2	Le théorème de Kleene	304
9.4.3	Expressions rationnelles	308
9.5	Automate minimal	311
9.5.1	Quotients	312
9.5.2	Equivalence de Nerode	315
9.6	Calcul de l'automate minimal	318
9.6.1	Construction de Moore	318
9.6.2	Scinder une partition	319

9.6.3	Algorithme de Hopcroft	321
9.6.4	Complexité de l'algorithme	327
Notes	333
Exercices	333
10	Motifs	337
Introduction	337
10.1	Recherche d'un motif	338
10.1.1	Un algorithme naïf	339
10.1.2	L'algorithme de Morris et Pratt	340
10.1.3	Bords	344
10.1.4	L'algorithme de Knuth, Morris et Pratt	346
10.1.5	L'automate des occurrences	350
10.1.6	L'algorithme de Simon	354
10.2	L'algorithme de Boyer et Moore	358
10.2.1	Algorithme de Horspool	358
10.2.2	Algorithme de Boyer et Moore	360
10.2.3	Fonction du bon suffixe	364
10.3	L'algorithme de Aho et Corasick	367
10.4	Recherche d'expressions	369
10.4.1	Calcul efficace d'un automate	370
10.4.2	Recherche d'occurrences	372
Notes	375
Exercices	376
11	Géométrie algorithmique	379
11.1	Notions préliminaires	379
11.1.1	Notations	380
11.1.2	Lignes polygonales, polygones	381
11.1.3	Ordre polaire, circuit polaire	382
11.2	Enveloppe convexe	386
11.2.1	Généralités	386
11.2.2	Marche de Jarvis	389
11.2.3	Algorithme de Graham	390
11.2.4	Algorithme dichotomique	393
11.2.5	Gestion dynamique d'une enveloppe convexe	395
11.3	Localisation de points dans le plan	403
11.3.1	Cas d'un polygone simple	404
11.3.2	Cas d'un polygone simple convexe	405
11.3.3	Cas d'une subdivision planaire généralisée	406
11.4	Diagrammes de Voronoï	408
11.4.1	Diagrammes de Voronoï de points	408
11.4.2	L'algorithme de Fortune	411
11.4.3	Diagramme de Voronoï de segments	422
Notes	426
Exercices	427

12 Planification de trajectoires	431
Introduction	431
12.1 Translation d'un segment	433
12.1.1 Présentation du problème	433
12.1.2 Présentation de l'algorithme	434
12.1.3 Prétraitement des données	440
12.1.4 Résolution du problème de translation	445
12.2 Déplacement d'un disque	446
12.2.1 Introduction	446
12.2.2 Rétraction — Diagramme de Voronoï	447
12.2.3 Exposé de l'algorithme	450
12.2.4 Remarques	455
Notes	455
Index	457
Notes et Compléments	463

Liste des figures

Préliminaires

3.1	<i>L'arbre de décision d'un algorithme de tri.</i>	10
-----	--	----

Evaluations

2.1	<i>Les arbres binaires de hauteur $-1 \leq h \leq 2$.</i>	28
-----	--	----

Structures de données

2.1	<i>Une liste et une place.</i>	39
2.2	<i>Un pile représentée par une liste chaînée.</i>	41
2.3	<i>La file $(c, a, b, c, c, a, b, d, a, c, c, d)$.</i>	43
2.4	<i>Effet de chaîner (p, q).</i>	48
2.5	<i>Effet de de chaîner (q).</i>	48
3.1	<i>Une arborescence représentée comme arbre binaire.</i>	55
4.1	<i>Les arbres parfaits à 6, 7, et 8 sommets.</i>	56
4.2	<i>Un arbre tournoi.</i>	57
4.3	<i>Un tas (tournoi parfait).</i>	57
4.4	<i>Mise en place de la clé 7 par comparaison aux fils.</i>	59
5.1	<i>Partition en trois classes, de noms 1, 3, et 7.</i>	62
5.2	<i>Compression d'un chemin.</i>	64
5.3	<i>Un arbre fileté.</i>	71
5.4	<i>Un tournoi et sa pagode.</i>	71

Graphes

1.1	<i>Sous-graphes.</i>	74
1.2	<i>Implémentations d'un graphe.</i>	75
1.3	<i>Un graphe sans circuit.</i>	79
1.4	<i>Classement par rang.</i>	80
2.1	<i>Un graphe orienté et sa matrice d'adjacence.</i>	82
2.2	<i>Un graphe étiqueté.</i>	84
2.3	<i>Un graphe orienté et son graphe réduit.</i>	90
3.1	<i>Un arbre à quatorze sommets.</i>	91
3.2	<i>Le graphe G_u.</i>	92
3.3	<i>Structure récursive d'une arborescence.</i>	93
3.4	<i>Une arborescence.</i>	94
3.5	<i>Arbre binaire et arbre binaire complet.</i>	96
3.6	<i>Les premiers arbres binaires complets.</i>	96

3.7	<i>Effeillage d'un arbre binaire complet.</i>	97
3.8	<i>Complétion d'un arbre binaire.</i>	98
4.1	<i>Bordure d'un sous-ensemble de sommets.</i>	98
4.2	<i>Parcours d'un graphe et arbre couvrant.</i>	100
4.3	<i>Parcours en profondeur.</i>	103
4.4	<i>Parcours en largeur.</i>	105
4.5	<i>Un graphe non orienté non connexe.</i>	106
4.6	<i>Bordure pour un graphe orienté.</i>	106
4.7	<i>Un graphe orienté.</i>	107
4.8	<i>Rangs d'attache et points d'attache.</i>	110
4.9	<i>Une exécution de DESC.</i>	113
4.10	<i>Une arborescence.</i>	116
4.11	<i>Un arbre binaire.</i>	117
Tris		
1.1	<i>Un pivotage.</i>	124
1.2	<i>Dernier échange d'un couple inversé.</i>	125
1.3	<i>Appels récursifs du tri rapide.</i>	125
1.4	<i>Un «peigne».</i>	126
1.5	<i>Un tri fusion.</i>	130
1.6	<i>Un tri par tas.</i>	132
2.1	<i>Sélection et remplacement.</i>	134
2.2	<i>Le tri équilibré.</i>	135
2.3	<i>Répartition des monotonies fantômes.</i>	139
Arbres et ensembles ordonnés		
1.1	<i>Un arbre binaire de recherche.</i>	147
1.2	<i>Un arbre sans balises.</i>	148
1.3	<i>L'arbre avec balises.</i>	148
1.4	<i>L'arbre après insertion de la dernière clé.</i>	149
1.5	<i>Insertion de 6 dans l'arbre balisé.</i>	149
1.6	<i>Suppression de 5 dans l'arbre balisé.</i>	150
1.7	<i>L'arbre $A = (x, B, C)$.</i>	150
1.8	<i>Rotations gauche et droite.</i>	151
1.9	<i>Rotation gauche-droite.</i>	151
1.10	<i>Rotation droite-gauche.</i>	151
2.1	<i>Un arbre AVL.</i>	152
2.2	<i>Arbres de Fibonacci Φ_k pour $k = 2, 3, 4, 5$.</i>	153
2.3	<i>Le chemin γ de rééquilibrage.</i>	155
2.4	<i>L'arbre de racine x avant et après insertion.</i>	155
2.5	<i>Cas (1) : une rotation simple rétablit l'équilibre.</i>	156
2.6	<i>Cas (2) : une rotation double rétablit l'équilibre.</i>	156
2.7	<i>Le chemin de rééquilibrage.</i>	157
2.8	<i>L'arbre de racine x avant et après suppression.</i>	157
2.9	<i>Une rotation simple sur le chemin de rééquilibrage.</i>	158
2.10	<i>Une rotation double sur le chemin de rééquilibrage.</i>	158
2.11	<i>Arbre de Fibonacci avant la suppression de la clé 12.</i>	159

2.12	<i>Après suppression et rééquilibrage.</i>	159
3.1	<i>Un arbre 2-4.</i>	160
3.2	<i>Insertion d'une feuille, cas $c < c_i$.</i>	162
3.3	<i>Insertion d'une feuille, cas $c > c_i$.</i>	162
3.4	<i>Règle d'éclatement.</i>	163
3.5	<i>L'arbre après insertion de 15.</i>	163
3.6	<i>Règle de fusion.</i>	164
3.7	<i>Règle du partage.</i>	164
3.8	<i>Un arbre 2-4.</i>	165
3.9	<i>Après suppression de 12 et avant rééquilibrage.</i>	166
3.10	<i>Après partage avec le frère droit.</i>	166
3.11	<i>Après fusion avec le frère gauche.</i>	166
3.12	<i>Un arbre 2-4 avant la scission par la clé 36.</i>	168
3.13	<i>Les deux forêts F_1 et F_2 après la séparation.</i>	168
3.14	<i>Reconstitution de l'arbre à partir de la forêt F_1.</i>	169
3.15	<i>Eclatement de x.</i>	171
3.16	<i>Partage entre les frères s et t.</i>	172
3.17	<i>Fusion des frères s et t.</i>	172
4.1	<i>Un arbre bicolore : (a) avec ses feuilles, (b) sans feuilles.</i>	173
4.2	<i>Affectation d'une couleur à un nœud en fonction du rang.</i>	175
4.3	<i>Attribution d'un rang aux sommets de l'arbre de la première figure.</i>	175
4.4	<i>Les trois cas possibles.</i>	176
4.5	<i>Première phase de l'insertion de A dans A.</i>	177
4.6	<i>Rééquilibrage après une insertion : règle γ.</i>	177
4.7	<i>Rééquilibrage après une insertion : règles α.</i>	178
4.8	<i>Rééquilibrage après une insertion : règles β.</i>	178
4.9	<i>Rééquilibrage après insertion de A.</i>	179
4.10	<i>Première phase de suppression d'un élément dans un arbre bicolore .</i>	180
4.11	<i>Rééquilibrage après une suppression : règles $(a_1), (a_2), (a_3)$.</i>	181
4.12	<i>Rééquilibrage après une suppression : règles $(b_1), (b_2), (b_3)$.</i>	182
4.13	<i>Suppression de p.</i>	182
5.1	<i>Un arbre 2-3 à liaisons par niveau.</i>	184
5.2	<i>Une liste doublement chaînée sur les sommets d'un arbre.</i>	184
5.3	<i>Les numéros d'ordre (en blanc) et les tailles (en noir).</i>	185
6.1	<i>Persistance par recopie intégrale.</i>	187
6.2	<i>Duplication de chemin.</i>	189
6.3	<i>Modifications de pointeurs dans une rotation gauche.</i>	189
6.4	<i>Première phase de l'insertion.</i>	190
6.5	<i>Insertion d'un nouvel élément – phase 1.</i>	191
6.6	<i>Rééquilibrage par rotation gauche lors d'une insertion.</i>	191
6.7	<i>Insertion d'un nouvel élément – phase 2.</i>	192
6.8	<i>Suppression d'un élément - phase 1.</i>	192
6.9	<i>Suppression de 0 - phase 1.</i>	193
6.10	<i>Rééquilibrage dans une suppression, règles (a), (b), (c).</i>	194
6.11	<i>Rééquilibrage dans une suppression, règles (d), (e).</i>	194
6.12	<i>Suppression d'un élément-phase 2 (règle (d)).</i>	195
6.13	<i>Recherche de F à l'instant 4.</i>	197

6.14	<i>Le sommet contenant L a été dupliqué pour contenir G.</i>	197
6.15	<i>Le fils gauche de L est dupliqué à l'instant t_7.</i>	198
6.16	<i>Cas où z n'est pas plein.</i>	198
6.17	<i>Cas où z est plein.</i>	199
6.18	<i>Insertion d'un nouvel élément-phase 1.</i>	199
6.19	<i>Insertion d'un nouvel élément-phase 2.</i>	200
6.20	<i>Suppression d'un élément-phase 1.</i>	200
6.21	<i>Comparaison des deux méthodes.</i>	201
6.22	<i>Transformation d'un arbre 2-4 en arbre bicolore.</i>	204
6.23	<i>Les quatre double-rotations.</i>	205
6.24	<i>Effet de l'opération $\text{EVASER}(a, A)$.</i>	205
6.25	<i>Arbres binomiaux.</i>	206
6.26	<i>Deux files binomiales,</i>	208
6.27	<i>... et leur union.</i>	208
Graphes valués		
1.1	<i>Un graphe valué.</i>	212
1.2	<i>Cycle et cocycle candidats.</i>	213
1.3	<i>Arbre couvrant de coût minimum.</i>	216
1.4	<i>Arbre couvrant de coût minimum.</i>	218
2.1	<i>Un graphe valué.</i>	220
2.2	<i>Un circuit absorbant.</i>	221
2.3	<i>Construction d'une arborescence des chemins minimaux.</i>	222
2.4	<i>L'itération de Ford.</i>	223
2.5	<i>Le graphe des états d'un sommet.</i>	225
2.6	<i>Graphe initial et première itération.</i>	228
2.7	<i>Les deux itérations suivantes.</i>	228
2.8	<i>Un graphe valué et la fonction h.</i>	230
2.9	<i>Les évaluations de l'algorithme PAPS.</i>	233
2.10	<i>L'algorithme PAPS avec un circuit absorbant.</i>	234
Flots		
1.1	<i>Chaîne et cycle élémentaires.</i>	241
2.1	<i>Amélioration de la compatibilité d'un flot.</i>	243
2.2	<i>Problème de transport et réseau valué équivalent.</i>	245
2.3	<i>Un flot entier positif.</i>	247
2.4	<i>L'application $\psi_{A,B}$.</i>	248
2.5	<i>Matrice d'incidence d'un arbre orienté.</i>	251
3.1	<i>Un problème de flot maximum et l'un de ses flots.</i>	253
3.2	<i>Un graphe d'écart.</i>	254
3.3	<i>Une coupe.</i>	255
3.4	<i>Un problème de flot maximum.</i>	257
3.5	<i>Mise à jour de la distance estimée</i>	259
3.6	<i>Initialisation et première augmentation de distance.</i>	261
3.7	<i>Seconde augmentation de distance.</i>	262
3.8	<i>Premier graphe d'écart d'un flot maximum.</i>	262
3.9	<i>Graphe d'écart du préflot initial.</i>	266

3.10	<i>La valeur maximum est atteinte.</i>	266
3.11	<i>Le dernier graphe d'écart.</i>	267
4.1	<i>Les réseaux R et R'.</i>	281
4.2	<i>Première phase.</i>	287
4.3	<i>Seconde phase.</i>	288
Automates		
2.1	<i>Automate reconnaissant le langage abA^*.</i>	296
2.2	<i>Automate reconnaissant le langage A^*aba.</i>	296
2.3	<i>Automate reconnaissant les mots contenant au moins un b.</i>	296
2.4	<i>Automate reconnaissant les mots contenant un nombre impair de a.</i>	297
2.5	<i>Tous les mots sont reconnus.</i>	297
2.6	<i>Automates reconnaissant (i) le mot vide et (ii) tous les mots sauf le mot vide.</i>	297
2.7	<i>Un automate reconnaissant le langage A^*ab.</i>	299
2.8	<i>L'automate déterminisé.</i>	299
2.9	<i>L'automate déterministe et émondé.</i>	300
2.10	<i>Un automate à $n + 1$ états.</i>	300
3.1	<i>Un automate asynchrone.</i>	302
3.2	<i>L'automate précédent «synchronisé».</i>	302
4.1	<i>Automate reconnaissant exactement le mot $w = a_1a_2 \cdots a_n$.</i>	305
4.2	<i>Quel est le langage reconnu par cet automate?</i>	306
4.3	<i>L'expression rationnelle $a(a + a \cdot b)^*b$.</i>	309
5.1	<i>Un automate reconnaissant $X = b^*a\{a, b\}^*$.</i>	313
5.2	<i>Automate minimal pour $X = b^*a\{a, b\}^*$.</i>	314
5.3	<i>Un automate qui n'est pas minimal.</i>	316
5.4	<i>Un automate quotient.</i>	317
6.1	<i>Un exemple illustrant l'opération \triangleleft.</i>	320
6.2	<i>Exemple.</i>	324
6.3	<i>Automate minimal obtenu à partir de celui de la figure 6.2.</i>	325
6.4	<i>Un exemple.</i>	330
6.5	<i>Deux automates non déterministes reconnaissant le même langage.</i>	334
Motifs		
1.1	<i>Le motif glissant sur le texte.</i>	339
1.2	<i>Echec à la i-ième lettre du motif.</i>	341
1.3	<i>Décalage d'une position.</i>	342
1.4	<i>Décalages successifs du motif.</i>	344
1.5	<i>Lorsque $b \neq a$, le décalage est inutile si $c = a$.</i>	346
1.6	<i>Décalages successifs du motif.</i>	349
1.7	<i>Les graphes des deux fonctions de suppléance.</i>	350
1.8	<i>Automate reconnaissant le langage $A^*abcababcac$.</i>	351
1.9	<i>Automate déterministe $\mathcal{A}(abcababcac)$.</i>	352
1.10	<i>Automate $\mathcal{A}(abcababcac)$, sans ses flèches passives.</i>	354
1.11	<i>Implémentation de l'automate $\mathcal{A}(abcababcac)$.</i>	356
2.1	<i>Algorithme de Horspool.</i>	358
2.2	<i>Algorithme de Boyer-Moore simplifié.</i>	361
2.3	<i>Coïncidence partielle du motif et du texte.</i>	362

2.4	Décalage : premier cas.	362
2.5	Décalage : deuxième cas.	362
2.6	Algorithme de Boyer-Moore complet.	364
3.1	Automate pour l'ensemble X	368
4.1	Automates normaliss	371
4.2	Automate pour l'union.	371
4.3	Automate pour le produit.	371
4.4	Automate pour l'étoile.	372
4.5	Un automate pour l'expression $(a + b)^*b(1 + a)(1 + a)^*$	372
4.6	L'automate de Boyer et Moore pour aba	377
Géométrie algorithmique		
1.1	Lignes polygonales.	381
1.2	Le contour positif du polygone est $((A, B, C, D, E))$	381
1.3	Ordre polaire.	382
1.4	$p_1 \prec_O p_2, p_2 \prec_O p_3, p_3 \prec_O p_1$	383
1.5	Un secteur angulaire.	384
1.6	Insertion entre p_3 et p_4	386
2.1	Cône enveloppant.	388
2.2	Adjonction d'un point n'appartenant pas à l'enveloppe convexe.	388
2.3	Marche de Jarvis.	389
2.4	Circuit polaire	391
2.5	Algorithme de Graham.	392
2.6	Exemple.	393
2.7	Cas où p est extérieur à P_2	394
2.8	Si p est supprimé, a, b et c apparaissent.	395
2.9	Enveloppes convexes supérieure et inférieure.	396
2.10	Calcul de $\text{Inf}(v)$	397
2.11	Les fonctions p_1, p_2 et G	397
2.12	Les 3 régions I, II, III auxquelles v peut appartenir.	398
2.13	Les 9 cas du lemme.	399
2.14	Insertion d'un nouveau point m	399
2.15	Rotation droite.	401
2.16	Structure de l'arbre avant insertion du point p_{13}	402
2.17	Insertion du point p_{13}	402
3.1	Principe de l'algorithme.	404
3.2	Le décompte des intersections.	405
3.3	Cas d'un polygone convexe.	405
3.4	Une subdivision planaire.	407
3.5	Une subdivision planaire généralisée.	407
4.1	Diagramme de Voronoï : les sites sont pleins.	409
4.2	Tout point suffisamment loin sur D est plus proche de b que de a	410
4.3	La partie grisée est dans $R(a)$	410
4.4	Demi-cône associé au site a	412
4.5	Le plan de balayage.	413
4.6	Le front parabolique au temps t	414
4.7	Tout point du diagramme de Voronoï est point anguleux.	415

4.8	Apparition d'un nouvel arc α .	415
4.9	Apparition de l'arc α : deuxième cas impossible.	416
4.10	Un événement de type site : cas «général».	420
4.11	Disparition de β et création du sommet s .	420
4.12	Événement site avec création d'un sommet.	421
4.13	Un ensemble de 4 segments ouverts et 6 points.	423
4.14	Diagramme de Voronoï formé de 4 demi-droites et d'un arc de parabole.	424
4.15	Le diagramme défini par deux segments.	425
4.16	Un diagramme de Voronoï.	426
4.17	Exemples.	428
4.18	Principe de l'algorithme.	429
Planification de trajectoires		
0.1	La translation $t(x, y)$	432
0.2	p_1 est libre, p_2 et p_3 sont semi-libres, p_4 et p_5 ne le sont pas.	433
1.1	P et θ caractérisent p .	434
1.2	Partitionnement de l'espace libre en cellules.	435
1.3	Le graphe $G_{\pi/2}$ associé à l'exemple de la figure précédente.	435
1.4	Emondage des cellules.	436
1.5	Le graphe $G_{\pi/2}(l)$ obtenu après émondage.	436
1.6	Choix du segment $Sup(p)$ selon la position de p .	437
1.7	Les cellules de $\mathcal{L}_{\pi/2}$.	438
1.8	Une cellule.	438
1.9	Événements (a) simples ou (b) double.	440
1.10	Débuts, milieu et fins.	441
1.11	Segments traités, actifs, non traités.	441
1.12	L'événement en cours est une fin.	443
1.13	L'événement en cours est un milieu.	443
1.14	L'événement en cours est un début.	444
1.15	$g_+(C)$ est calculé au cours de l'insertion de s' .	446
2.1	Exemple d'obstacles.	447
2.2	Une région $R(s)$, et la région $R'(s)$ associée (en gras).	448
2.3	Scission d'un arc de parabole.	451
2.4	$\lambda(e)$ lorsque e est un arc de parabole.	452
2.5	Un exemple de mouvement utilisant cet algorithme.	454
2.6	Composantes connexes des déplacements libres.	455
2.7	Un «mauvais déplacement».	456

Avant-propos

«Encore un livre d’algorithmique!» Les ouvrages consacrés à l’algorithmique et aux structures de données paraissent, depuis quelque temps, à un rythme soutenu et régulier. Aussi devons-nous expliquer pourquoi un livre supplémentaire sur ce sujet nous a paru utile.

Le présent livre se distingue d’autres traités d’algorithmique par deux aspects : d’une part, un accent particulier est mis sur les nouvelles structures d’arbres apparues ces dernières années (arbres bicolores, arbres persistants); d’autre part, nous développons plus en détail trois applications de l’algorithmique : l’optimisation combinatoire, la recherche de motifs dans un texte, et la géométrie algorithmique. Outre leur intérêt propre et leur importance intrinsèque, ces trois applications illustrent de façon exemplaire l’usage que l’on peut faire de structures de données sophistiquées, et les gains en temps et en place qui résultent de leur emploi judicieux.

Pour chacun de ces trois thèmes, nous mettons en place les bases nécessaires, tant algorithmiques que théoriques. Puis, nous présentons, à l’aide de problèmes typiques, un échantillon des algorithmes les plus efficaces, employant des structures de données intéressantes. Parmi les algorithmes et les structures de données qui méritent d’être mentionnés plus spécialement, citons :

- le coût amorti des opérations de rééquilibrage dans les arbres 2–4 (chapitre 6);
- les algorithmes de manipulation des arbres bicolores (*ibid.*);
- les arbres persistants (*ibid.*);
- les algorithmes de Goldberg–Tarjan et d’Ahuja–Orlin pour les flots (chapitre 8);
- l’algorithme de Hopcroft de minimisation d’un automate fini (chapitre 9);
- les algorithmes de Simon, et de Boyer et Moore pour la recherche de motifs dans un texte (chapitre 10);
- la gestion dynamique de l’enveloppe convexe d’un ensemble fini de points (chapitre 11);
- l’algorithme de balayage de calcul d’une subdivision plane à l’aide d’arbres persistants (*ibid.*);
- l’algorithme de Fortune de calcul du diagramme de Voronoï (*ibid.*);
- deux algorithmes de planification de trajectoires (chapitre 12).

A qui s'adresse ce livre ?

Ce livre est issu de cours donnés par les auteurs en licence et maîtrise d'informatique et de mathématiques dans les universités Paris VI et Paris VII, en magistère, et pour partie dans divers DEA et DESS.

Dans sa forme présente, il s'adresse principalement aux étudiants en licence et en maîtrise d'informatique et de mathématiques, et aux étudiants des écoles d'ingénieurs. Nous ne supposons chez le lecteur qu'une connaissance rudimentaire de l'informatique. Les premiers chapitres décrivent les bases algorithmiques sur lesquelles s'appuieront les chapitres suivants. Le niveau d'exposition est, à de rares exceptions près, celui des étudiants en licence d'informatique.

Organisation du livre

Ce livre s'organise en deux parties. La première partie, constituée des chapitres 1 à 5, contient un exposé des bases algorithmiques et mathématiques, avec notamment un chapitre sur les techniques d'évaluation de coûts d'algorithmes, un bref chapitre sur les structures de données usuelles, un chapitre plus long sur la théorie des graphes et un chapitre assez succinct sur les algorithmes de tri par comparaison. Les trois premiers chapitres, et en partie les deux chapitres suivants, peuvent être considérés comme des chapitres de référence, auxquels on se reportera si nécessaire.

La deuxième partie, consacrée à des thèmes plus avancés, débute par un long chapitre sur les arbres de recherche. Les deux chapitres suivants sont consacrés à l'optimisation combinatoire. Puis viennent deux chapitres sur la recherche de motifs dans un texte, et enfin deux chapitres sur la géométrie algorithmique et la planification de trajectoires.

Présentation des algorithmes

Les algorithmes sont présentés dans un langage proche de Pascal ; assez rarement, et principalement dans le chapitre 3 (Structures de données), des références explicites à Pascal sont faites, avec des programmes complets, prêts à cuire. Nous nous efforçons de présenter les algorithmes avec suffisamment de détails pour que l'écriture de programmes soit facile ; en particulier, nous décrivons soigneusement les structures de données à employer.

Contenu du livre

Le premier chapitre contient la définition des diverses mesures du coût d'un algorithme : coût dans le cas le plus défavorable, coût moyen, coût amorti ; il se termine par une borne inférieure sur la complexité du tri par comparaison.

La première section du deuxième chapitre contient la définition des notations dites de Landau, ainsi que des exemples de manipulation. Dans la deuxième section, nous abordons la résolution de divers types d'équations de récurrence auxquelles conduit l'analyse d'un algorithme. Nous décrivons notamment la résolution des récurrences linéaires et des récurrences de partition.

Dans le chapitre 3, nous passons en revue les structures de données élémentaires, à savoir les piles, files, listes, ainsi que les arbres binaires et les arbres binaires de recherche. Nous décrivons ensuite les files de priorité et leur implémentation au moyen de tas. Nous terminons par un algorithme de gestion de partitions.

Le chapitre 4 est consacré aux bases de la théorie des graphes. Après les définitions, nous exposons l'algorithme de Roy-Warshall et d'autres problèmes d'accessibilité. Les parcours classiques d'un graphe (parcours en profondeur, en largeur) sont décrits. Puis, nous présentons en détail l'algorithme de Tarjan de calcul des composantes fortement connexes d'un graphe, qui est linéaire en fonction du nombre d'arcs du graphe.

Le chapitre 5 présente trois algorithmes de tri interne éprouvés et efficaces : le tri rapide, le tri fusion et le tri par tas. Nous en donnons l'analyse des coûts en moyenne et dans le pire des cas. Le chapitre se termine par un algorithme de tri externe, le tri polyphasé qui est fondé sur les suites de Fibonacci.

Le chapitre 6 contient la description de plusieurs familles d'arbres binaires de recherche équilibrés. Nous décrivons les arbres AVL, puis les arbres a - b , avec notamment des opérations plus élaborées comme la concaténation et la scission, et évaluons le coût amorti d'une suite d'opérations. Ensuite, nous présentons les arbres bicolores, et enfin une réalisation de structures persistantes sur ces arbres.

Le chapitre 7 traite deux problèmes fondamentaux d'optimisation combinatoire : la recherche d'un arbre couvrant de coût minimum et la recherche des chemins de coût minimum issus d'un sommet dans un graphe orienté valué. Pour le premier problème, nous présentons une implémentation efficace des algorithmes de Kruskal et de Prim. Quant à la recherche des chemins de coût minimum, nous exposons l'itération fondamentale de Ford, l'algorithme de Dijkstra pour des coûts positifs et sa variante A^* , l'algorithme de Bellman pour un graphe sans circuits et enfin l'algorithme PAPS.

Le chapitre 8 traite des problèmes de flots optimaux dans les graphes. Nous prouvons d'abord le théorème d'Hoffman sur l'existence d'un flot compatible, puis étudions les décompositions d'un flot. Après avoir établi le théorème de Ford et Fulkerson, nous analysons les algorithmes les plus performants pour le calcul du flot maximum, à savoir l'algorithme primal des distances estimées au puits (algorithme d'Ahuja et Orlin) et deux variantes efficaces de l'algorithme dual du préflot, l'algorithme de Karzanov et l'algorithme des excès échelonnés. Nous considérons ensuite le problème du flot de coût minimum et présentons un algorithme dû à Goldberg et Tarjan. Nous terminons par l'algorithme d'Edmonds et Karp pour la recherche d'un plan de transport de coût minimum.

Dans le chapitre 9, préliminaire au chapitre suivant, nous présentons les bases de la théorie des automates finis. Après avoir démontré l'équivalence entre les automates finis et les automates finis déterministes, nous établissons le théorème de Kleene qui montre que les langages reconnaissables et les langages rationnels sont une seule et même famille de langages. Nous prouvons l'existence et l'unicité

d'un automate déterministe minimal reconnaissant un langage donné. Pour la construction de cet automate, nous présentons l'algorithme de Hopcroft de minimisation.

Dans le chapitre 10, nous considérons d'abord le problème de la recherche d'une ou de toutes les occurrences d'un mot dans un texte. Nous présentons l'algorithme naïf, l'algorithme de Morris et Pratt, et sa variante due à Knuth, Morris et Pratt, l'implémentation par automate fini et l'algorithme de Simon, et pour finir l'algorithme de Boyer et Moore, dans sa version de Horspool et dans la version complète. Ensuite, nous considérons la recherche d'une occurrence de plusieurs motifs, et décrivons l'algorithme de Aho et Corasick. Dans la dernière section, nous étudions la recherche d'occurrences de mots décrits par une expression rationnelle.

La première section du chapitre 11 contient un rappel de quelques notations classiques en géométrie euclidienne, ainsi que la mise en place d'outils qui seront utiles dans les sections suivantes. La deuxième section est consacrée aux algorithmes de calcul de l'enveloppe convexe d'un ensemble fini de points du plan. Nous terminons par un algorithme plus sophistiqué de gestion dynamique d'enveloppes convexes. Quelques problèmes de localisation d'un point dans le plan divisé en régions sont étudiés dans la troisième section; on y utilise des structures de données assez complexes, en particulier les ensembles ordonnés persistants. La dernière section est consacrée aux diagrammes de Voronoï de points et de segments.

Dans le chapitre 12, nous étudions un problème de planification de trajectoires, aussi appelé le problème du «déménageur de piano». Nous donnons des algorithmes performants dans deux cas particuliers typiques : l'algorithme de translation d'un segment dans un environnement polygonal dû à Schwartz et Sharir, et celui du déplacement d'un disque dû à Ó'Dúnlaing et Yap. Ce dernier utilise les diagrammes de Voronoï de segments.

Exercices et notes

Les chapitres sont en général suivis de notes et d'exercices. Les notes bibliographiques sont volontairement succinctes. Leur but n'est pas de retracer la paternité des résultats ou des algorithmes présentés, mais de conseiller des lectures complémentaires. On y trouvera notamment des renvois nombreux au *Handbook of Theoretical Computer Science* qui contient, lui, d'abondantes listes de références bibliographiques.

Remerciements

Nous avons bénéficié, durant la préparation de ce livre, de commentaires, discussions et remarques de nombreux collègues. Nous remercions en particulier Paul Blanchard, Luc Boasson, Maxime Crochemore, Clara Daquin, Christiane Frougny, Marie-Paule Gascuel, Irène Guessarian, Michelle Morcrette, Dominique Perrin, Michel Pocchiola, Andreas Podelski, Imre Simon, Michèle Soria, Volker Strehl, Anne Verroust, Mariette Yvinec.