

Asli Grimaud
Gilles Grimaud

Informatique MP2I

Cours complets, exercices corrigés
et projets guidés pour les classes préparatoires

RESSOURCES
TÉLÉCHARGEABLES



Table des matières

1	Gestion des fichiers	1
1.1	Interface système	2
1.2	Système de fichiers	3
1.3	Droits d'accès	6
1.4	Arborescence	7
1.5	Liens physiques et liens symboliques	8
1.6	Redirections	11
1.7	Commandes indispensables et tube	12
1.8	Makefile	15
1.9	TD/TP - Gestion des fichiers	19
2	Booléens	25
2.1	Algèbre de Boole	26
2.1.1	Tables de vérités	27
2.1.2	Lois	28
2.2	Fonctions booléennes	28
2.3	Un problème de vérité	31
2.3.1	Modélisation	31
2.3.2	Résolution utilisant l'algèbre de Boole	31
2.3.3	Résolution utilisant la programmation fonctionnelle	34
2.3.4	Résolution utilisant les portes logiques	36
2.4	De l'algèbre de Boole au calcul binaire	38
2.5	TD/TP - Booléens	40
3	Programmation	43
3.1	Architecture de von Neumann	44
3.2	Du langage source au langage machine	45
3.2.1	Exemple introductif	46
3.2.2	Langages de programmation	47
3.2.3	Langage machine	47
3.2.4	Interprétation	48
3.2.5	Compilation	50
3.2.6	Machine virtuelle	51
3.2.7	Comparaison des stratégies	52
3.3	Programmation fonctionnelle en OCaml	55
3.3.1	Expressions en OCaml	55
3.3.2	Types en OCaml	57

3.3.3	Fonctions en OCaml	60
3.3.4	Expressions conditionnelles et expressions séquentielles	63
3.3.5	Éléments de programmation impérative en OCaml	66
3.3.6	Déclarations en OCaml et commandes de l'interpréteur OCaml	68
3.4	TD/TP - Programmation	69
4	Codage	77
4.1	Programmation impérative en C	78
4.1.1	Fichier source, fichier objet et fichier exécutable	79
4.1.2	Fonctions en C	81
4.1.3	Utilisation de la fonction printf	83
4.1.4	Variables	84
4.1.5	Conditions en C	85
4.1.6	Instructions conditionnelles	86
4.1.7	Boucles d'itération	88
4.1.8	Boucles conditionnelles	89
4.2	Codage des types	92
4.2.1	Représentation des nombres entiers	92
4.2.2	Représentation des caractères	96
4.2.3	Représentation des nombres flottants	101
4.3	TD/TP - Codage	104
5	Récursivité	109
5.1	Programmation récursive	110
5.1.1	Exemple introductif : Méthode d'Archimède	110
5.1.2	Implémentation récursive de l'approximation d'Archimède	111
5.1.3	Concevoir des fonctions récursives	113
5.1.4	Sous-fonction récursive	114
5.2	Récursivité terminale	115
5.2.1	Appels récursifs	115
5.2.2	Récursivité terminale pour la méthode d'Archimède	118
5.3	Récursivité croisée	119
5.4	TD/TP - Récursivité	120
6	Gestion de la mémoire	123
6.1	Exemple introductif	124
6.2	Pointeurs	126
6.3	Structures de données homogènes	129
6.3.1	Tableaux	129
6.3.2	Chaînes de caractères	131
6.3.3	Arguments de la fonction main	135
6.3.4	Arithmétique des pointeurs	136
6.4	Mémoire	138
6.4.1	Pile d'appel	138
6.4.2	Durée de vie des variables	140
6.4.3	Tas	144
6.4.4	Vue générale	146
6.4.5	Erreurs courantes	147

6.5	Références en OCaml	150
6.5.1	Variables mutables et ramasse-miettes	150
6.5.2	Opérations d'égalités	151
6.6	TD/TP - Gestion de la mémoire	153
7	Structures en C	159
7.1	Définition d'une structure	160
7.2	Pointeurs et structures	164
7.3	Tableaux et structures	166
7.3.1	Structures avec des tableaux	166
7.3.2	Tableaux de structures et tableaux de pointeurs de structures	167
7.4	Tableaux avancés	169
7.4.1	Tableaux de tableaux	169
7.4.2	Tableaux de taille variable	171
7.5	Constantes en C	171
7.6	Accès aux fichiers en C	172
7.7	TD/TP - Structures en C	177
8	Types structurés en OCaml	183
8.1	n-uplets	184
8.2	Enregistrements	185
8.3	Sommes avec constructeurs	188
8.3.1	Constructeurs constants (énumérations)	188
8.3.2	Mot-clé function	189
8.3.3	Filtrage	189
8.3.4	Constructeurs avec arguments	191
8.3.5	Structures de données récursives	193
8.4	Accès aux fichiers en OCaml	194
8.4.1	Lecture d'un fichier	195
8.4.2	Écriture dans un fichier	195
8.5	TD/TP - Types structurés en OCaml	196
9	Fiabilité et tests en programmation	199
9.1	Discipline de programmation	200
9.1.1	Spécification des données	200
9.1.2	Programmation défensive	204
9.2	Tests et validation	207
9.2.1	Graphe de flot de contrôle	209
9.2.2	Critères de couverture	211
9.2.3	Jeu de tests	213
9.2.4	Tests exhaustifs des conditions	213
9.3	TD/TP - Fiabilité et tests en programmation	214
10	Types structurés en OCaml (2)	217
10.1	Listes	218
10.1.1	Fonctions sur les listes	218
10.1.2	Module List	219
10.2	Tableaux	221

10.2.1	Accès à un élément	222
10.2.2	Module Array	222
10.2.3	Matrices	224
10.3	Chaînes de caractères	225
10.4	Structures de données abstraites	226
10.5	TD/TP - Types structurés en OCaml (2)	229
11	Analyse des algorithmes	233
11.1	Terminaison	234
11.1.1	Variants	235
11.1.2	Exemple illustratif : Cas de la couverture des points	236
11.1.3	Analyse de terminaison d'un algorithme itératif	236
11.1.4	Analyse de terminaison d'un algorithme récursif	238
11.2	Correction	239
11.2.1	Invariants	240
11.2.2	Analyse de correction - version impérative	241
11.2.3	Analyse de correction - version récursive	242
11.3	Complexité	244
11.3.1	Exemple illustratif : Exponentiation	245
11.3.2	Complexité en temps	246
11.3.3	Classes de complexité	247
11.3.4	Types de complexité	250
11.3.5	Complexité des boucles	253
11.3.6	Complexité amortie	256
11.3.7	Complexité spatiale	259
11.4	TD/TP - Analyse des algorithmes	260
12	Structures de données séquentielles	263
12.1	Principe général des files et des piles	264
12.2	Tableaux redimensionnables	264
12.2.1	Principe	265
12.2.2	Complexité	266
12.3	Listes chaînées	267
12.4	Structure de pile	268
12.4.1	Principe des piles	268
12.4.2	Création d'une librairie en C	269
12.4.3	Implémentation avec les tableaux	272
12.4.4	Implémentation avec les listes chaînées	274
12.5	Structure de file	274
12.5.1	Principe des files	274
12.5.2	Implémentation avec les tableaux	276
12.5.3	Implémentation avec les listes chaînés	278
12.6	Implémentation des piles et des files en OCaml	279
12.6.1	Types structurés	279
12.6.2	Module Stack	281
12.6.3	Module Queue	282
12.7	TD/TP - Structures de données séquentielles	284

13	Réversivité et induction	291
13.1	Réversivité avec des appels multiples	292
13.1.1	Exemple introductif : Tours de Hanoï	292
13.1.2	Algorithme et complexité	292
13.1.3	Implémentation et activations	293
13.2	Équations de récurrence	294
13.3	Induction	296
13.3.1	Réurrence faible et récurrence forte	296
13.3.2	Principe d'induction	297
13.3.3	Preuve par induction structurelle	300
13.3.4	Preuve par induction bien fondée	301
13.4	Analyse d'algorithmes récursifs	306
13.4.1	Exemple illustratif : Codage de formules logiques	306
13.4.2	Preuve de terminaison d'un algorithme récursif	308
13.4.3	Preuve de correction d'un algorithme récursif	309
13.5	TD/TP - Réversivité et induction	310
14	Structures de données séquentielles (2)	313
14.1	Tableaux associatifs	314
14.1.1	Table de hachage	314
14.1.2	Implémentation en C	316
14.1.3	Module <code>Hashtbl</code> en OCaml	319
14.2	Sérialisation	322
14.2.1	Format binaire	325
14.2.2	Format textuel	330
14.3	TD/TP - Structures de données séquentielles (2)	333
15	Structures de données hiérarchiques	339
15.1	Arbres	340
15.2	Arbres binaires	341
15.2.1	Vocabulaire	342
15.2.2	Conversion d'un arbre quelconque en arbre binaire	343
15.2.3	Représentation d'un arbre binaire complet dans un tableau	344
15.2.4	Implémentation	346
15.3	Parcours d'arbre	347
15.3.1	Parcours en largeur	347
15.3.2	Parcours en profondeur	348
15.4	TD/TP - Structures de données hiérarchiques	351
16	Bases de données	353
16.1	Introduction	354
16.2	Notion de table	355
16.2.1	Attributs	355
16.2.2	Entités	357
16.3	Clés primaires	358
16.3.1	Déclaration	358
16.3.2	Complexité des recherches	359
16.3.3	Clés étrangères	360

16.3.4	Schémas des tables	361
16.4	Modèle entité-association	361
16.4.1	Associations 1-1	362
16.4.2	Associations 1-*	363
16.4.3	Associations *-*	364
16.5	Consultation des données	366
16.5.1	Langage SQL	366
16.5.2	Extraction des données	367
16.5.3	Formatage des données extraites	367
16.5.4	Filtrage des sélections de données	368
16.5.5	Jointure de données	370
16.5.6	Opérateurs ensemblistes	372
16.5.7	Agrégation de données	373
16.5.8	Imbrication de requêtes SQL	375
16.6	TD/TP - Bases de données	377
17	Structures de données hiérarchiques (2)	381
17.1	Arbres binaires de recherche	382
17.1.1	Arbres binaires de recherche équilibrés et dégénérés	383
17.1.2	Implémentation en C	385
17.2	Arbres bicolores	388
17.2.1	Rotations	391
17.2.2	Insertion	393
17.2.3	Suppression	394
17.3	Propriété de tas	396
17.3.1	File de priorité	397
17.3.2	Tri par tas	399
17.4	TD/TP - Structures de données hiérarchiques (2)	400
18	Structures de données relationnelles	403
18.1	Graphes	404
18.1.1	Graphes non orientés et orientés	404
18.1.2	Chemins	406
18.1.3	Graphes bipartis	410
18.1.4	Graphes pondérés	410
18.1.5	Arbres	411
18.2	Structures de données	412
18.2.1	Enregistrement graph	413
18.2.2	Matrice d'adjacence	414
18.2.3	Listes d'adjacence	415
18.2.4	Création des graphes	415
18.3	Parcours	416
18.3.1	Parcours en largeur	417
18.3.2	Parcours en profondeur	420
18.3.3	Tri topologique d'un graphe orienté acyclique	423
18.3.4	Recherche des composantes connexes d'un graphe non orienté	424
18.4	TD/TP - Structures de données relationnelles	425

19 Logique	429
19.1 Logique propositionnelle	430
19.1.1 Syntaxe des formules logiques	431
19.1.2 Substitution	433
19.1.3 Sémantique de vérité du calcul propositionnel	434
19.1.4 Formes normales	438
19.2 SAT	442
19.2.1 Algorithme de recherche exhaustive	443
19.2.2 Algorithme de Quine	443
19.2.3 Modélisation du problème de la coloration de graphe	446
19.3 Logique du premier ordre	447
19.4 TD/TP - Logique	450
20 Structures de données relationnelles (2)	453
20.1 Algorithmes de plus courts chemins	454
20.2 Algorithme de Dijkstra	455
20.2.1 Algorithmes gloutons	456
20.2.2 Principe de l'algorithme	456
20.2.3 Implémentation en C	460
20.2.4 Analyse	462
20.3 Algorithme de Floyd-Warshall	465
20.3.1 Programmation dynamique	465
20.3.2 Principe	465
20.3.3 Implémentation en C	467
20.3.4 Analyse	469
20.4 TD/TP - Structures de données relationnelles (2)	470
21 Algorithmique des textes	477
21.1 Recherche dans un texte	478
21.1.1 Algorithme de recherche exhaustive	478
21.1.2 Algorithme de Boyer-Moore	480
21.1.3 Algorithme de Rabin-Karp	484
21.2 Compression	488
21.2.1 Algorithme de Huffman	489
21.2.2 Algorithme de Lempel-Ziv-Welch	496
21.3 TD/TP - Algorithmique des textes	501
22 Projets et problèmes	505
22.1 Afficheur 7 segments	505
22.2 Coefficients binomiaux	506
22.3 Flocon de Koch	509
22.4 Ellipse	513
22.5 <i>Look and Say</i>	516
22.6 Méthode de Newton	521
22.7 Cartes	524
22.8 Algorithme de PageRank	532
22.9 Mandelbrot	540
22.10 Arbres binaires	546

22.11 Compression	554
23 Solutions des exercices	561
23.1 TD/TP - Gestion des fichiers	561
23.2 TD/TP - Booléens	565
23.3 TD/TP - Programmation	569
23.4 TD/TP - Codage	573
23.5 TD/TP - Récursivité	579
23.6 TD/TP - Gestion de la mémoire	580
23.7 TD/TP - Structures en C	585
23.8 TD/TP - Types structurés en OCaml	590
23.9 TD/TP - Fiabilité et tests en programmation	594
23.10 TD/TP - Types structurés en OCaml (2)	596
23.11 TD/TP - Analyse des algorithmes	598
23.12 TD/TP - Structures de données séquentielles	602
23.13 TD/TP - Récursivité et induction	607
23.14 TD/TP - Structures de données séquentielles (2)	611
23.15 TD/TP - Structures de données hiérarchiques	616
23.16 TD/TP - Bases de données	617
23.17 TD/TP - Structures de données hiérarchiques (2)	619
23.18 TD/TP - Structures de données relationnelles	622
23.19 TD/TP - Logique	625
23.20 TD/TP - Structures de données relationnelles (2)	630
23.21 TD/TP - Algorithmique des textes	635
24 Solutions des projets et problèmes	641
24.1 Afficheur 7 segments	641
24.2 Coefficients binomiaux	644
24.3 Flocon de Koch	645
24.4 Ellipse	647
24.5 <i>Look and Say</i>	649
24.6 Méthode de Newton	655
24.7 Cartes	657
24.8 Algorithme de PageRank	664
24.9 Mandelbrot	670
24.10 Arbres binaires	674
24.11 Compression	679
Bibliographie	685
Liste des figures	689
Liste des tableaux	693
Liste des codes	695
Index	699