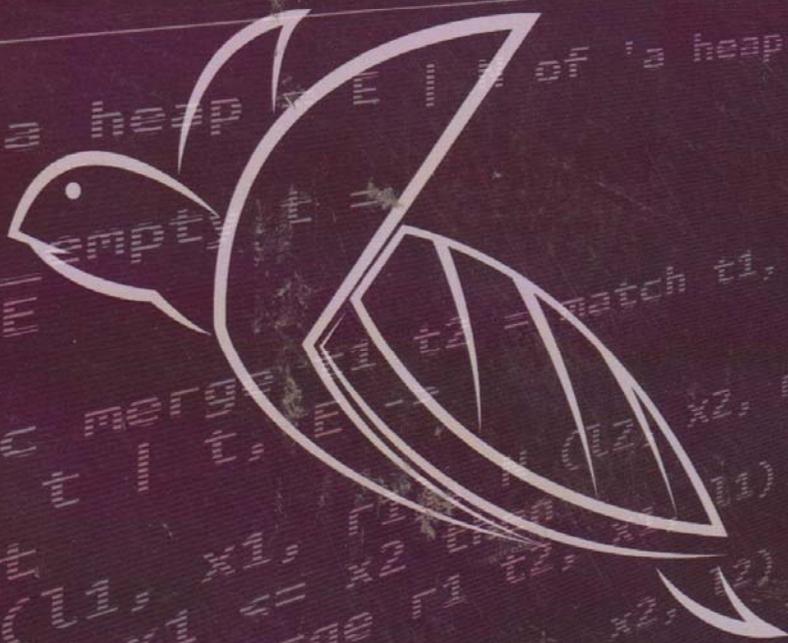


**MP2I
MPI**

INFORMATIQUE

Cours et exercices corrigés



ellipses

Thibaut Balabonski
Sylvain Conchon
Jean-Christophe Filliâtre
Kim Nguyen
Laurent Sartre



2-004-424-1

**MP2
MPI**

INFORMATIQUE

Cours et exercices corrigés

Thibaut Balabonski
Sylvain Conchon
Jean-Christophe Filliâtre
Kim Nguyen
Laurent Sartre

Préface de Laurent Chéno

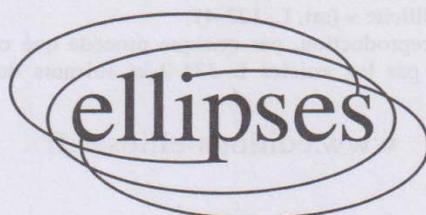


Table des matières

1	Avant-gût	1
2	Notions d'architecture et de système	5
2.1	Arithmétique des ordinateurs	5
2.1.1	Représentation des entiers	6
2.1.2	Représentation approximative des nombres réels	11
2.2	Modèle de von Neumann	16
2.2.1	Composants d'un ordinateur	17
2.2.2	Organisation de la mémoire	20
2.2.3	Langage machine et assembleur	24
2.3	Système d'exploitation	26
2.3.1	L'interface système ou <i>shell</i>	27
2.3.2	Fichiers et redirections	42
2.3.3	Commandes utiles	44
	Exercices	46
3	Programmation fonctionnelle avec OCaml	49
3.1	Premiers pas	50
3.1.1	Déclarations globales et expressions simples	50
3.1.2	Interprétation et compilation d'un programme	51
3.1.3	Inférence de types	53
3.1.4	Expressions simples	54
3.1.5	Fonctions simples	58
3.1.6	Déclarations locales	62
3.1.7	Instructions	63
3.1.8	Commentaires	66
3.1.9	Modules	67
3.2	Données structurées	69
3.2.1	Paires et n -uplets	69
3.2.2	Enregistrements	71

3.2.3	Énumérations	75
3.2.4	Sommes disjointes avec arguments	79
3.3	Récurtivité	81
3.3.1	Fonctions récursives	82
3.3.2	Récursion terminale	87
3.3.3	Types récursifs	94
3.4	Polymorphisme	97
3.5	Ordre supérieur	99
3.6	Traits impératifs	104
3.6.1	Structures de données modifiables	105
3.6.2	Boucles	110
3.6.3	Exceptions	111
3.6.4	Entrées-Sorties	114
	Exercices	116
4	Programmation impérative avec C	123
4.1	Premiers pas	124
4.1.1	Généralités	124
4.1.2	Types et opérations élémentaires	125
4.1.3	Structures de contrôle	127
4.1.4	Modèle d'exécution	131
4.2	Pointeurs, tableaux et structures	134
4.2.1	Pointeurs	134
4.2.2	Tableaux	137
4.2.3	Structures	142
4.3	Entrées-sorties	146
4.3.1	Affichage sur la sortie standard	146
4.3.2	Lecture sur l'entrée standard	146
4.3.3	Ligne de commande	147
4.3.4	Gestion de fichiers	147
4.4	Modularité	148
4.5	Comparaison des langages C et OCaml	154
	Exercices	155
5	Bonnes pratiques de la programmation	159
5.1	Code source	159
5.2	Compilation	162
5.3	Exécution	164
5.4	Validation, test	168
5.5	Quelques conseils	171
	Exercices	172

3.2.3	Énumérations	75
3.2.4	Sommes disjointes avec arguments	79
3.3	Récurtivité	81
3.3.1	Fonctions récursives	82
3.3.2	Récursion terminale	87
3.3.3	Types récursifs	94
3.4	Polymorphisme	97
3.5	Ordre supérieur	99
3.6	Traits impératifs	104
3.6.1	Structures de données modifiables	105
3.6.2	Boucles	110
3.6.3	Exceptions	111
3.6.4	Entrées-Sorties	114
	Exercices	116
4	Programmation impérative avec C	123
4.1	Premiers pas	124
4.1.1	Généralités	124
4.1.2	Types et opérations élémentaires	125
4.1.3	Structures de contrôle	127
4.1.4	Modèle d'exécution	131
4.2	Pointeurs, tableaux et structures	134
4.2.1	Pointeurs	134
4.2.2	Tableaux	137
4.2.3	Structures	142
4.3	Entrées-sorties	146
4.3.1	Affichage sur la sortie standard	146
4.3.2	Lecture sur l'entrée standard	146
4.3.3	Ligne de commande	147
4.3.4	Gestion de fichiers	147
4.4	Modularité	148
4.5	Comparaison des langages C et OCaml	154
	Exercices	155
5	Bonnes pratiques de la programmation	159
5.1	Code source	159
5.2	Compilation	162
5.3	Exécution	164
5.4	Validation, test	168
5.5	Quelques conseils	171
	Exercices	172

6	Raisonner sur les programmes	175
6.1	Correction	180
6.1.1	Spécification d'un problème algorithmique	180
6.1.2	Preuves de correction par récurrence	184
6.1.3	Invariants de boucle	193
6.1.4	Cas d'étude : correction d'algorithmes de tri	199
6.2	Terminaison	212
6.2.1	Technique du variant	213
6.2.2	Relations binaires et ensembles ordonnés	215
6.2.3	Ordres bien fondés	225
6.3	Complexité	233
6.3.1	Cadre pour la complexité temporelle	233
6.3.2	Complexité des boucles	238
6.3.3	Cas d'étude : complexité du tri fusion ascendant	242
6.3.4	Modèles pour la complexité en moyenne et probabilités	245
6.3.5	Complexité des fonctions récursives	251
6.3.6	Cas d'étude : complexité moyenne du tri rapide	255
6.3.7	Complexité amortie	259
6.3.8	Complexité spatiale	264
6.4	Induction structurelle	268
6.4.1	Aperçu : mobiles de Calder	268
6.4.2	Objets inductifs	273
6.4.3	Formalisation des constructions inductives	282
6.4.4	Principe d'induction structurelle	288
6.4.5	Cas d'étude : correction d'un renversement de liste efficace	294
6.5	Cas d'étude : analyse d'un tri de listes	297
	Exercices	306
7	Structures de données	323
7.1	Types et abstraction	323
7.2	Structures de données séquentielles	327
7.2.1	Tableaux	327
7.2.2	Tableaux redimensionnables	328
7.2.3	Listes chaînées	333
7.2.4	Piles	343
7.2.5	Files	351
7.2.6	Tables de hachage	357
7.3	Structures de données hiérarchiques	368
7.3.1	Arbres binaires	368
7.3.2	Arbres binaires de recherche	377
7.3.3	Tas et files de priorité	394

7.3.4	Arbres	405
7.3.5	Arbres préfixes	411
7.3.6	Structure unir et trouver (<i>union-find</i>)	416
7.4	Des ensembles	422
	Exercices	425
8	Graphes	437
8.1	Définitions	438
8.1.1	Graphes orientés	438
8.1.2	Graphes non orientés	440
8.1.3	Graphes pondérés	441
8.1.4	Graphes bipartis	442
8.2	Structures de données	443
8.2.1	Matrice d'adjacence	444
8.2.2	Listes d'adjacence	445
8.2.3	Graphes non orientés	445
8.2.4	Graphes pondérés	447
8.3	Algorithmique des graphes	448
8.3.1	Parcours en profondeur	449
8.3.2	Parcours en largeur	457
8.3.3	Plus court chemin	461
8.3.4	Composantes fortement connexes	476
8.3.5	Arbre couvrant de poids minimum	480
8.3.6	Couplage maximum dans un graphe biparti	484
	Exercices	488
9	Algorithmique	493
9.1	Arithmétique	493
9.1.1	Algorithme d'Euclide	493
9.1.2	Arithmétique modulaire	497
9.1.3	Crible d'Ératosthène	499
9.2	Retour sur trace (<i>backtracking</i>)	501
9.3	Algorithme glouton	507
9.4	Décomposition d'un problème en sous-problèmes	512
9.4.1	Diviser pour régner	512
9.4.2	Programmation dynamique	522
9.5	Algorithmique des textes	533
9.5.1	Recherche dans un texte	533
9.5.2	Compression	544
9.6	Algorithmes probabilistes	559
9.6.1	Échantillonnage	560

9.6.2	Problème de N reines	563
9.6.3	Test de primalité	565
9.7	Algorithmique pour l'intelligence artificielle et l'étude des jeux . . .	567
9.7.1	Apprentissage	567
9.7.2	Jeux à deux joueurs	585
	Exercices	596
10	Logique	605
10.1	Logique propositionnelle	608
10.1.1	Variable et formule propositionnelles	608
10.1.2	Sémantique	615
10.1.3	Conséquence logique	621
10.1.4	Équivalence sémantique	622
10.1.5	Substitution	624
10.1.6	Formes normales	625
10.2	SAT	629
10.2.1	Algorithme de Quine	631
10.2.2	Une modélisation SAT	633
10.2.3	2-SAT	636
10.3	Logique du premier ordre	639
10.3.1	Domaine, termes et prédicats	639
10.3.2	Formules du premier ordre	642
10.4	Déduction naturelle	647
10.4.1	Déduire	647
10.4.2	Déduction naturelle propositionnelle	652
10.4.3	Déduction naturelle pour la logique du premier ordre . . .	663
10.5	Prédicats inductifs	667
10.5.1	Systèmes d'inférence et principe d'induction	667
10.5.2	Correction de la déduction naturelle propositionnelle	676
10.5.3	Cas d'étude : caractériser les programmes bien typés	680
	Exercices	685
11	Bases de données	695
11.1	Le modèle entité-association	696
11.2	Le modèle relationnel	700
11.2.1	Relation	700
11.2.2	Clé primaire	704
11.2.3	Clé étrangère	705
11.3	Requêtes SQL	709
11.3.1	Sélection	711
11.3.2	Opérations ensemblistes	718

11.3.3	Jointure	722
11.3.4	Fonctions d'agrégation	730
11.3.5	Requêtes de groupe	734
	Exercices	739
12	Langages formels	745
12.1	Langages réguliers	747
12.1.1	Alphabets, mots et langages	747
12.1.2	Langages réguliers, expressions régulières	752
12.2	Automates de mots finis	758
12.2.1	Automates déterministes	759
12.2.2	Automates non déterministes	766
12.2.3	Déterminisation et suppression des transitions spontanées	772
12.2.4	Théorème de Kleene	777
12.2.5	Propriétés des langages réguliers	788
12.2.6	Implémentation des algorithmes	792
12.3	Grammaires non contextuelles	801
12.3.1	Grammaires et langages non contextuels	802
12.3.2	Ambiguïté d'une grammaire	806
12.3.3	Analyse syntaxique	811
	Exercices	816
13	Calculabilité	823
13.1	Décidabilité	826
13.1.1	Règles du jeu	826
13.1.2	Problème de l'arrêt	827
13.1.3	Problèmes de décision et indécidabilité	831
13.1.4	Algorithme universel	836
13.1.5	Réduction calculatoire	837
13.2	Classes de complexité	842
13.2.1	Problèmes de recherche et d'optimisation	842
13.2.2	Modèle de complexité	845
13.2.3	Classe P	846
13.2.4	Réductions polynomiales	850
13.2.5	Classe NP	851
13.3	NP-complétude	855
13.3.1	Problème de référence : SAT	857
13.3.2	Coloriage de graphe	863
13.3.3	La tournée du voyageur de commerce	868
13.4	Algorithmes d'optimisation	874
13.4.1	Algorithmes d'approximation	875

13.4.2	Séparation et évaluation	883
13.5	Modèles historiques et complétude calculatoire	890
13.5.1	Lambda-calcul	891
13.5.2	Machines de Turing	894
13.5.3	Complétude Turing	897
13.5.4	Machines de Turing et complexité	898
	Exercices	901
14	Gestion de la concurrence et synchronisation	905
14.1	Processus	905
14.2	Bibliothèques de <i>Threads POSIX</i>	909
14.3	Atomicité	913
14.4	Mutex	914
14.5	Sémaphores	920
	Exercices	928
	Solutions des exercices	933
	Index	1075

INFORMATIQUE

Cours et exercices corrigés



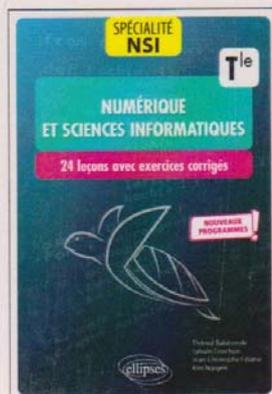
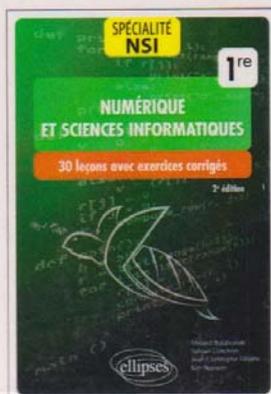
Cet ouvrage propose un cours structuré couvrant l'intégralité du programme de la filière **MP21/MPI** en classes préparatoires aux grandes écoles. Il contient notamment :

- ▶ un **cours complet**, avec une présentation détaillée des concepts et des développements techniques ;
- ▶ de **nombreux exercices corrigés** ;
- ▶ des **milliers de lignes de code** ;
- ▶ des **encarts thématiques et historiques** permettant d'approfondir les notions ;
- ▶ une **initiation aux fondamentaux d'architecture et de système** ;
- ▶ une **initiation aux langages C et OCaml** adaptée aux besoins du programme.

Ce livre ne suppose aucune connaissance préalable en informatique et constitue une introduction complète à ce domaine.

Le site www.informatique-mpi.fr qui accompagne cet ouvrage fournit du matériel librement téléchargeable, dont notamment tout le code source C, OCaml et SQL utilisé dans ce cours.

Thibaut Balabonski, Sylvain Conchon, Jean-Christophe Filliâtre, Kim Nguyen et Laurent Sartre sont enseignants et chercheurs en informatique. Ils enseignent à l'Université Paris-Saclay, à l'École Polytechnique, à l'École Normale Supérieure et en classes préparatoires scientifiques au lycée Montaigne de Bordeaux. À eux cinq, ils totalisent plus de 10000 heures d'enseignement dans de nombreux domaines de l'informatique couvrant notamment tous les aspects du programme de MP21/MPI.



www.editions-ellipses.fr

