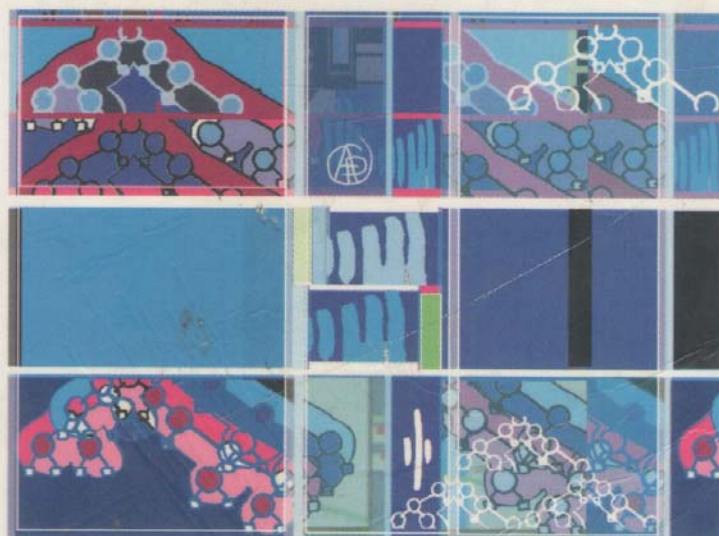


Michel Divay

2^e CYCLE • ÉCOLES D'INGÉNIEURS

Algorithmes et structures de données

Cours et exercices corrigés
en langage C



DUNOD

Table des matières

AVANT-PROPOS

XI

CHAPITRE 1 • RÉCURSIVITÉ, POINTEURS, MODULES

1.1 Récursivité des procédures : définition

1

1.2 Exemples de fonctions récursives

2

1.2.1 Exemple 1 : Factorielle

2

1.2.2 Exemple 2 : Nombres de Fibonacci

5

1.2.3 Exemple 3 : Boucles récursives

7

1.2.4 Exemple 4 : Numération

9

1.2.5 Exemple 5 : Puissance Nième d'un nombre

11

X 1.2.6 Exemple 6 : Tours de Hanoi

12

1.2.7 Exemple 7 : Tracés récursifs de cercles

16

1.2.8 Exemple 8 : Tracé d'un arbre

18

1.2.9 Conclusions sur la récursivité des procédures

20

1.3 Récursivité des objets

20

1.3.1 Rappel sur les structures

20

1.3.2 Exemple de déclaration incorrecte

21

1.3.3 Structures et pointeurs

22

1.3.4 Opérations sur les pointeurs

24

1.4 Modules

26

1.4.1 Notion de module et de type abstrait de données (TAD)

26

1.4.2 Exemple : Module de simulation d'écran graphique

27

1.5 Résumé

36

CHAPITRE 2 • LES LISTES	38
2.1 Listes simples : définition	38
2.2 Représentation en mémoire des listes	39
2.3 Module de gestion des listes	40
2.3.1 Ajout d'un élément	42
2.3.2 Retrait d'un élément	44
2.3.3 Les fonctions de parcours de liste	46
2.3.4 Destruction de listes	46
2.3.5 Recopie de listes	47
2.3.6 Le module de gestion de listes	47
2.4 Vers les applications	49
2.4.1 Redéfinition de l'élément de liste pour une application	49
2.4.2 Modèle de parcours de listes	50
2.4.3 Modèle d'insertion dans une liste ordonnée	52
2.5 Exemples d'applications	53
2.5.1 Liste de personnes	53
2.5.2 Les polynômes	59
2.5.3 Les systèmes experts	65
2.5.4 Les piles	71
2.5.5 Les files d'attente	75
2.6 Avantages - inconvénients des listes	78
2.7 Le Type Abstrait de Données (TAD) liste	79
2.8 Les listes circulaires	79
2.8.1 Le fichier d'en-tête des listes circulaires	79
2.8.2 Insertion en tête de liste circulaire	80
2.8.3 Insertion en fin de liste circulaire	81
2.8.4 Le module des listes circulaires	81
2.8.5 Utilisation du module des listes circulaires	82
2.9 Les listes symétriques	84
2.9.1 Le fichier d'en-tête des listes symétriques	84
2.9.2 Le module des listes symétriques	85
2.9.3 Utilisation du module des listes symétriques	87
2.10 Allocation contiguë	90
2.10.1 Allocation - désallocation en cas d'allocation contiguë	90
2.10.2 Exemple des polynômes en allocation contiguë avec liste libre	92
2.10.3 Exemple de la gestion des commandes en attente	94
2.11 Résumé	106

CHAPITRE 3 • LES ARBRES	108
3.1 Les arbres n-aires	108
3.1.1 Définitions	108
3.1.2 Exemples d'applications utilisant les arbres	110
3.1.3 Représentation en mémoire des arbres n-aires,	113
3.2 Les arbres binaires	115
3.2.1 Définition d'un arbre binaire	115
3.2.2 Transformation d'un arbre n-aire en un arbre binaire	115
3.2.3 Mémorisation d'un arbre binaire	116
3.2.4 Parcours d'un arbre binaire	119
3.2.5 Propriétés de l'arbre binaire	128
3.2.6 Création, duplication, destruction d'un arbre binaire	132
3.2.7 Dessin d'un arbre binaire	133
3.2.8 Arbre binaire et questions de l'arbre n-aire	137
3.2.9 Le module des arbres binaires	144
3.2.10 Arbre binaire et tableau	156
3.2.11 Arbre binaire et fichier	160
3.2.12 Arbre binaire complet	161
3.3 Les arbres binaires ordonnés	163
3.3.1 Définitions	163
3.3.2 Arbres ordonnés : recherche, ajout, retrait	165
3.3.3 Le module des arbres ordonnés	170
3.4 Les arbres binaires ordonnés équilibrés	177
3.4.1 Définitions	177
3.4.2 Ajout dans un arbre ordonné équilibré	177
3.4.3 Exemple de test pour les arbres équilibrés	188
3.5 Arbres n-aires ordonnés équilibrés : les B-arbres	190
3.5.1 Définitions et exemples	190
3.5.2 Insertion dans un B-arbre	192
3.5.3 Recherche, Parcours, Destruction	194
3.6 Résumé	204
CHAPITRE 4 • LES TABLES	205
4.1 Cas général	205
4.1.1 Définition	205
4.1.2 Exemples d'utilisation de tables	207
4.1.3 Initialisation de tables	208
4.1.4 Accès séquentiel	209
4.1.5 Accès dichotomique (recherche binaire)	211
4.1.6 Le module des tables	213
4.1.7 Exemples d'applications des tables	218

4.2	Variantes des tables	221
4.2.1	Rangement partitionné ou indexé	221
4.2.2	Adressage calculé	222
4.3	Adressage dispersé, hachage, hash-coding	224
4.3.1	Définition du hachage	224
4.3.2	Exemples de fonction de hachage	225
4.3.3	Résolution des collisions	228
4.3.4	Résolution à l'aide d'une nouvelle fonction	228
4.3.5	Le fichier d'en-tête des fonctions de hachage et de résolution	234
4.3.6	Le type Table (Hachage)	234
4.3.7	Mise en œuvre des fonctions de hachage	239
4.3.8	Résolution par chaînage avec zone de débordement	241
4.3.9	Résolution par chaînage avec une seule table	242
4.3.10	Exemple 1 : Arbre n-aire de la Terre (en mémoire centrale)	251
4.3.11	Exemple 2 : Arbre n-aire du corps humain (fichier)	254
4.4	Résumé	258
CHAPITRE 5 • LES GRAPHERS		259
5.1	Définitions	259
5.1.1	Graphes non orientés (ou symétriques)	259
5.1.2	Graphes orientés	260
5.1.3	Graphes orientés ou non orientés	261
5.2	Exemples de graphes	262
5.3	Mémorisation des graphes	264
5.3.1	Mémorisation sous forme de matrices d'adjacence	264
5.3.2	Mémorisation en table de listes d'adjacence	265
5.3.3	Liste des sommets et listes d'adjacence : allocation dynamique	266
5.4	Parcours d'un graphe	266
5.4.1	Principe du parcours en profondeur d'un graphe	267
5.4.2	Principe du parcours en largeur d'un graphe	268
5.5	Mémorisation (table de listes d'adjacence)	268
5.5.1	Le type Table pour les graphes	269
5.5.2	Le fichier d'en-tête des graphes	269
5.5.3	Initialisation et destruction d'un graphe	270
5.5.4	Insertion d'un sommet ou d'un arc dans un graphe	271
5.5.5	Création d'un graphe à partir d'un fichier	272
5.5.6	Parcours en profondeur (listes d'adjacence)	274
5.5.7	Parcours en largeur (listes d'adjacence)	274
5.5.8	Plus court chemin en partant d'un sommet NSI	276
5.5.9	Menu de test des graphes (listes d'adjacence)	282

5.6	Mémorisation sous forme de matrices	287
5.6.1	Le fichier d'en-tête du module des graphes (matrices)	287
5.6.2	Initialisation et destruction d'un graphe (matrices)	288
5.6.3	Insertion d'un sommet ou d'un arc dans un graphe (matrices)	289
5.6.4	LireGraphe	290
5.6.5	Parcours en profondeur (matrices)	290
5.6.6	Parcours en largeur (matrices)	291
5.6.7	Plus courts chemins entre tous les sommets (Floyd)	292
5.6.8	Algorithme de Floyd	295
5.6.9	Algorithme de calcul de la fermeture transitive	297
5.6.10	Menu de test des graphes (matrices)	300
5.7	Résumé	301
5.8	Conclusion générale	301
CORRIGÉS DES EXERCICES		302
BIBLIOGRAPHIE		321
TABLE DES EXERCICES		323
INDEX		325