

Sommaire

Partie 1 : Structures de données

1. Listes linéaires chaînées

Notion d'allocation (statique vs dynamique)

Définition et modélisation d'une liste

Algorithmes sur les listes

Listes particulières

Implémentation en Pascal

2. Files d'attente (Queues)

Définition et principes

Implémentation

3. Piles (Stacks)

Définition, principe, applications

Implémentation

4. Récursivité

Introduction par des exemples

Conception d'algorithmes récursifs

Sémantique de la récursivité

Transformation en algorithmes itératifs

Récursivité en Pascal

5. Arbres

Définitions et terminologie

Arbres binaires (modèle, parcours, arbres de recherche)

Arbres de recherche m-aires

Arbres équilibrés : AVL, B-arbres, 2-3, Red-Black

Implémentation des arbres

6. Graphes

Graphes orientés et non orientés

Modèles de graphe

Parcours : DFS (profondeur), BFS (largeur)

Applications des graphes

Implémentation

7. Hachage (Hashing)

Introduction et définitions

Fonctions de hachage

Méthodes de résolution des collisions

Étude comparative des méthodes

Hachage dynamique

Partie 2 : Structures de fichiers

1. Introduction aux fichiers

Besoin de mémoire secondaire

Problèmes liés au stockage externe

Modèles d'adressage et de mesure des performances

2. Structures simples de fichiers

Fichiers comme tableaux

Fichiers comme listes chaînées

Recherche (ex : dichotomique)

3. Méthodes d'indexation

Index à un niveau

Index multiniveaux

Opérations : insertion, suppression, recherche

Index secondaire / multi-clés

4. Méthodes arborescentes

Index avec arbres de recherche m-aires

B-arbres et variantes

5. Méthodes de hachage pour fichiers**

Chaînage externe (listes)

Adressage ouvert

Avantages et inconvénients des méthodes