NFA006 - Structures de données

Mis à jour le 19-12-2025

Code : NFA006

Unité d'enseignement de type cours

4 crédits

Volume horaire de référence (+/- 10%) : **40 heures**

Responsabilité nationale :
EPN05 - Informatique / Cédric DU MOUZA

Contact national :

EPN05 - Informatique

2 rue Conté
75003 Paris

Sandra Bosse
sandra.bosse@lecnam.net

Présentation

Prérequis

Ce cours s'adresse aussi bien aux élèves en licence qu'à ceux préparant le titre d'analyste programmeur ou le DUT. Il suppose une connaissance minimale en algorithmique et en programmation.

Objectifs pédagogiques

Donner les notions fondamentales de structures de données et de leur utilisation, et montrer comment les implanter à bon escient dans un langage de programmation de haut niveau. Faire comprendre l'importance de la spécification rigoureuse des structures de données, le pourquoi de l'étude de la complexité des algorithmes qui les manipulent, les principes de mise en oeuvre de ces structures.

Compétences

- Savoir évaluer la complexité d'un algorithme simple en fonction de la taille des données.
- Savoir abstraire les principales structures de données, les spécifier et les implanter.

Programme

Contenu

Notions préliminaires
Rappel succinct des propriétés et caractéristiques essentielles des supports de mémorisation, tels que la mémoire centrale, les disques et les bandes. Notion de complexité des algorithmes : mesure d'efficacité en fonction de la taille du problème.
Les structures de données
Les structures séquentielles et les structures arborescentes. Principaux algorithmes liés à ces structures. Différentes techniques d'implantation de ces structures : avantages et inconvénients.
L'utilisation des structures
Principaux algorithmes de tri. Généralités et méthodes simples. Méthodes efficaces. Mesures et comparaisons entre ces algorithmes.
Principes de la recherche d'informations. Recherche séquentielle dans une liste quelconque. Recherche dichotomique dans une liste ordonnée pour laquelle on dispose de l'accès par le rang. Gestion d'un tas : solution efficace pour rechercher le plus petit élément d'un ensemble.
Utilisation de structures arborescentes pour la recherche. Les arbres binaires de recherche : recherche, adjonction et suppression. Évaluation de la complexité logarithmique en moyenne de ces opérations, et comparaison avec les structures séquentielles. Évaluation de la complexité au pire linéaire : amélioration par rééquilibrage donnant les arbres AVL. Analyse des opérations simples de rotation ponctuelle pour conserver l'équilibre.
Généralisation des arbres AVL aux arbres balancés pour prendre en compte une caractéristique des disques : la taille des blocs transférés. Application aux fichiers séquentiels indexés.
Recherche utilisant la notion de hachage : principes et méthodes de résolution des collisions.
Remarque : Implantations proposées au moyen de paquetages Ada génériques disponibles en machine (ou modules Java ou C++), pour que les élèves puissent les utiliser lors de travaux pratiques personnels, et apprennent ainsi les notions fondamentales de réutilisation du logiciel.

Modalités de validation

- Examen final

Description des modalités de validation

Examen

Bibliographie

Titre	Auteur(s)

Structures de données en Java, C++ et Ada 95 (Masson 1997)	C. CARREZ
Types de données et algorithmes (McGraw-Hill, 1990).	M.-C. GAUDEL, M. SORIA, C. FROIDEVAUX
URL:http://deptinfo.cnam.fr/Enseignement/CycleA/SD/	C. CARREZ et al.
Initiation à l'algorithmique et aux structures de données. Volumes 1 et 2 (Dunod).	J. COURTIN, I. KOWARSKI