Algorithmes et structures de données

Domaine	Ingénierie et Architecture
Filière	Informatique et systèmes de communication
Orientation	Réseaux et systèmes (ISCR)
Mode de formation	Plein temps

Informations générales

Nom	:	Algorithmes et structures de données
Identifiant	:	ASD
Années académiques	:	2021-2022, 2022-2023
Responsable	:	Laura Elena Raileanu
Charge de travail	:	180 heures d'études
Périodes encadrées	:	128 (= 96 heures)

Semestre	E1	S1	S2	E2	S3	S4	E3	S5	S6
Cours			64						
Laboratoire			64						

Connaissances préalables recommandées

L'étudiant-e doit connaître et savoir utiliser les notions suivantes du langage C++ :

- constituants d'un programme ;
- types simples et structurés ;
- structures de contrôle ;
- sous-programmes ;
- exceptions ;
- pseudo-code ;
- manipulation de tableaux de capacité fixe ;
- algorithmes de tri simples (bulles, sélection, insertion) ;
- généricité.

L'unité PRG1 permet d'acquérir ces connaissances.

Objectifs

A l'issue de cette unité d'enseignement, l'étudiant-e sera capable de :

- connaître les algorithmes de recherche, de permutations, de tri, de sélection. Savoir les décrire, les appliquer, les classifier selon leurs propriétés et leur complexité, les choisir de façon appropriée pour la résolution des problèmes ;
- connaître les différents types de structures de données (tableaux, listes, tas, arbres, tables de hachage), pouvoir les mettre en œuvre, maîtriser les opérations de manipulation des éléments contenus dans ces structures (insertion, suppression, recherche, parcours, itération, déplacement) ainsi que les algorithmes spécifiques, connaître les complexités temporelles et spatiales associées ;
- concevoir, implémenter et manipuler des nouvelles structures de données adaptées aux problèmes pratiques rencontrés, choisir et adapter les algorithmes pour résoudre de nouveaux problèmes ;
- maîtriser les outils du langage C++ permettant de mettre en œuvre les différents algorithmes et structures de données étudiés. Comprendre et savoir utiliser l'ensemble de conteneurs et algorithmes de la STL.

Contenu et formes d'enseignement

Répartition des périodes indiquée à titre informatif.

Cours: 68 périodes

- Introduction : motivation, pseudo-code, définitions et principes de base pour l'analyse d'un algorithme, la recherche dichotomique, notion de complexité, notion de structure de données et de type de données abstrait	6
- Récursivité : règles de la récursivité, comparaison avec l'itération et récursion terminale, exemples (Factorielle, Fibonacci, Euclide, Hanoï, permutations, MiniMax), complexités	4
- Tris sur tableaux : notions de stabilité et de complexité temporelle et spatiale, algorithmes de tri par fusion, tri rapide, sélection rapide, optimal basé sur des comparaisons	6
- Introduction à l'allocation dynamique en C++	4
- Structures linéaires : tableau (taille fixe, capacité fixe, buffer circulaire, capacité variable), liste (simplement chainée et doublement chainée, notion d'itérateur via itération fonctionelle, tri, splice, tas (insertion, suppression, création depuis un tableau non trié, tri), deque, types de données abstraits (pile, file, file de priorité)	12
- Arbres : quelconques (définitions, représentation, parcours, représentations linéaires, itération), binaires (représentation, parcours symétrique, expressions arithmétiques), binaires de recherche (recherche, insertion, min et max, suppression, itération, rang d'une clé, sélection par le rang, équilibrage statique, équilibrage AVL), types de données abstraits ensemble et tableau associatif, algorithmes sur les ensembles	16
- Tables de hachage : définition, fonction de hachage, résolution de collision, chaînage, adressage ouvert, insertion, suppression, recherche, redimensionnement	6
- Graphes : définition, classification (orientation et pondération), représentation (matrice d'adjacence, listes d'adjacence, implicite), parcours (largeur, profondeur), composantes connexes, tri topologique, composantes fortement connexes (algorithme de Kosaraju), algorithme de Dijkstra, algorithme de Kruskal	10
- Travaux écrits	4

Laboratoire: 60 périodes

- Estimation théorique et expérimentale de la complexité	4
- Utilisation de la récursivité dans un jeu	10
- Implémentation des tris par comparaison et par comptage et étude de leurs complexités	6
- Exercices d'implémentation pour illustrer l'allocation dynamique	4
- Mise en oeuvre d'une deque avec un tableau circulaire : capacité fixe et variable	6
- Mise en oeuvre d'une pile avec une liste chaînée	4
- Tris de listes	6
- Mise en oeuvre d'un arbre binaire de recherche, équilibrage dynamique	10
- Comparaison de l'efficacité de diverses fonctions de hachage	4
- Jeu du taquin 3x3	6

Bibliographie

1. "Algorithms", 4/E, Robert Sedgewick and Kevin Wayne, Addison-Wesley Professional, 2011, ISBN 0132762560, 9780132762564.
2. " Introduction to algorithms", Thomas Cormen, Charles Leiserson, Ronald Rivest, Clifford Stein, Third Edition, MIT Press, 2009, ISBN-10: 0262033844.
3. "Algorithmes: Notions de base", Thomas Cormen, Éditeur Dunod, 2013, ISBN 2100702904, 9782100702909.

Contrôle de connaissances

Cours : l'acquisition des matières de cet enseignement sera contrôlée au fur et à mesure par des tests et des travaux personnels tout au long de son déroulement. Il y aura au moins 2 tests d'une durée totale d'au moins 4 périodes.

Laboratoire : ils seront évalués sur la base des rapports de manipulation, à 3 reprises au minimum.

Examen : L'atteinte de l'ensemble des objectifs de formation sera vérifiée lors d'un contrôle final commun écrit d'une durée de 120 minutes.

Matériel autorisé :

- Information communiquée directement par l'enseignant.

Calcul de la note finale

Note finale = moyenne cours x 0.3 + moyenne laboratoire x 0.2 + moyenne examen x 0.5