UNIVERSITÉ SAVOIE MONT BLANC

Algorithmique 2

Composante
UFR Sciences et Montagne

En bref

- **Langues d'enseignement:** Français
- **Méthodes d'enseignement:** En présence
- **Ouvert aux étudiants en échange:** Oui

Présentation

Description

Ce module présente de nouveaux éléments d'algorithmique, qui font suite à l'étude des structures de données linéaires et arborescentes, tout en restant assez indépendant des algorithmes sur les graphes. Les notations asymptotiques et la complexité en pire cas sont rappelées. Le module donne alors les outils pour calculer la complexité des algorithmes récursifs, notamment ceux qui ont une stratégie de type divide-and-conquer. Ensuite sont présentés des techniques pour déterminer les temps d'exécution des algorithmes dans les cas où l'analyse en pire cas échoue, notamment l'analyse amortie. Dans la deuxième partie, nous nous intéressons plus à de nouveaux algorithmes pour traiter des problèmes classiques. D'une part, les structures efficaces pour l'union disjointe sont exposées. D'autre part, nous décrivons des algorithmes simples de géométrie algorithmique, ainsi que les structures de localisation spatiale.

Objectifs

Savoir déterminer la complexité en pire cas ou la complexité amortie d'un algorithme donné, qu'il soit itératif ou récursif. Savoir reconnaître quand utiliser des structures de données élaborées (comme les forêts) pour résoudre efficacement des problèmes sur des graphes. Connaître quelques algorithmes pour résoudre des problèmes géométriques. Savoir les coder dans un langage où la complexité attendue est vérifiée.

Heures d'enseignement

Algorithmique II - CM	Cours Magistral	9h
Algorithmique II - TD	Travaux Dirigés	7,5h
Algorithmique II - TP	Travaux Pratiques	12h

Pré-requis obligatoires

Connaître les structures de données simples (liste, pile, arbre) et les principaux algorithmes associés. Savoir utiliser les notations asymptotiques pour décrire la complexité en pire cas d'un algorithme.

Plan du cours

CM : 1 – notations asymptotiques, complexité en pire cas des algorithmes, 2 – complexité des algorithmes récursifs, 3 – complexité amortie, 4 – structures pour ensembles disjoints, 5 – géométrie algorithmique.

TDs : Chacun des TDs permet de mettre en pratique les différents concepts d'algorithmique vus en cours. Les premiers TDs font travailler les notations asymptotiques, puis font calculer les complexités d'algorithmes récursifs. Ensuite, des exercices sur l'analyse amortie sont proposés. Ils terminent par un exemple complet sur les structures de données dynamique par doublement de taille. Les derniers TDs font travailler les étudiants sur de petits problèmes géométriques et leur résolution algorithmique.

TP : Le premier TP montre l'intérêt des structures d'union disjointe lorsque l'on cherche à segmenter une image. Ces structures rendent les approches bottom-up extrêmement efficaces. Le deuxième TP porte sur le calcul de l'enveloppe convexe d'un nuage de point. Le troisième TP montre l'intérêt des structures de localisation spatiale lorsque l'on veut faire un programme de simulation temps réel où de multiples objets se déplacent en même temps et peuvent collisionner avec des obstacles. Les structures k-d-tree sont alors très efficaces pour détecter les collisions lorsque des milliers de particules se déplacent en même temps. Les TP sont réalisés en C, avec la bibliothèque GTK pour la partie graphique/interface.

Infos pratiques

Lieux

› Le Bourget-du-Lac (73)

Campus

› Le Bourget-du-Lac / campus Savoie Technolac