

Informatique

Algorithmes avancés (36 heures)

26_27_4IRC_08_COMSC_UE8_080_MAJ

ACQUIS

- Modéliser et résoudre des problèmes concrets à l'aide de graphes
- Déterminer la complexité temporelle des algorithmes définis par des relations de récurrence, notamment à l'aide du Master Theorem
- Présenter les bases et les techniques avancées de la programmation dynamique pour résoudre des problèmes optimaux, illustrer son efficacité à travers des exemples classiques comme le problème du sac à dos, la plus longue sous-séquence commune ou l'alignement de chaînes ; savoir reconnaître les problèmes où la programmation dynamique est applicable
- Modéliser des problèmes d'optimisation linéaire : Comprendre les fondements de la programmation linéaire, formuler des problèmes réels en programmation linéaire et analyser les limites des solutions proposées par cette méthode.
- Comprendre les limites théoriques et pratiques des modèles de calcul

CONTENU

- Concepts avancés de théorie des graphes (coloration, théorème flot max / coupe min, PageRank, approches de résolution de problèmes difficiles...) ; modélisation à l'aide de graphes
- Complexité des algorithmes récursifs / Master Theorem
- Introduction à la programmation dynamique
- Introduction à la programmation linéaire
- Introduction à la calculabilité (Machines de Turing)

PRÉREQUIS

- Notions de mathématiques discrètes?: combinatoire, raisonnement par récurrence
- Algèbre et analyse?: calcul matriciel, manipulation de fonctions, limites, logarithmes et exponentielles, croissances comparées
- Programmation?: niveau de base en Python (syntaxe, fonctions, classes?)

PÉDAGOGIE

ÉVALUATION

BIBLIOGRAPHIE