Fiche du cours

55 h

Titre :

DSA201 - Structures de données & Algorithmes 2

Description :

Suite de DSA-101 : arbres (BST/AVL), tas/priority queue, graphes (BFS/DFS, plus court chemin), tris avancés (merge/quick/heap), greedy et programmation dynamique (DP). Toujours tronc commun multi-langages(Python/C++/Java ...)pour acquérir des réflexes transférables. Prérequis : DSA-101.

Objectifs :

- Implémenter un BST et comprendre AVL (rotations, équilibre).*
- Utiliser un tas binaire (priority queue) et relier à heap sort.*
- Modéliser un graphe (liste/matrice d'adjacence), parcourir avec BFS/DFS.*
- Expliquer/implémenter merge sort, quick sort (pivot), heap sort.*
- Appliquer des greedy classiques (interval scheduling, Huffman) et poser une première DP (memo/Tabulation).

Chapitres :

1. BST : API, insertion/suppression, complexités*
2. AVL : rotations LL/LR/RR/RL, intuition d'équilibrage*
3. Tas / priority queue : push/pop/heapify, applications*
4. Graphes : représentations, BFS (niveaux, composantes)*
5. DFS : détection de cycles, tri topologique (DAG)*
6. Plus courts chemins : Dijkstra (pondérations positives)*
7. Tris avancés : merge/quick/heap — choix & stabilité*
8. Greedy : activités intervalle, Huffman (codage)*
9. DP : Fibonacci, knapsack (0/1) simple, LIS (aperçu)*
10. Mini-projet DSA-201 (au choix : routing simplifié, ordonnanceur par priorité, compresseur Huffman)

À la fin :

Vous pourrez modéliser des problèmes avec arbres/graphes, sélectionner/implémenter un tri avancé, utiliser greedy quand c’est pertinent, et écrire une première solution DP, avec un raisonnement de complexité clair et transposable (Python/C++/Java).