

Algorithmique et Programmation avancée - TP n°3

Arbres

Olivier Gérard

27 Novembre 2023

Dans ce TP nous allons étudier et implémenter les structures de données que l'on appelle des arbres. Plusieurs types d'arbres seront évoqués, et vous devrez pour chaque type implémenter les diverses fonctions permettant de les manipuler. Pour cela, n'hésitez à pas utiliser toutes fonctions ou structures que vous aviez déjà développées dans les TPs précédents.

Afin de pouvoir profiter de vos travaux précédents et pour mettre en pratique les notions de compilations séparée vues dans le cours précédent, vous devrez séparer vos structures et fonctions dans des fichiers sources différents, qui implémenteront divers fichiers en-tête dont vous aurez besoin. Ces fichiers seront ensuite inclus dans un fichier contenant votre fonction *main* vous permettant de tester vos fonctions.

Exercice 1

Ce premier exercice se concentrera sur les arbres binaires simples. Par rapport à ce que vous avez vu dans le cours et en vous aidant de vos précédents travaux sur les structures de données, vous devrez implémenter cette structure et les différentes fonctions pour interagir avec.

Implémentez la structure de l'arbre binaire. Développez ces différentes fonctions permettant de réaliser les premières tâches sur cette structure :

- La fonction vérifiant qu'un arbre est vide ou non.
- La fonction pour insérer un élément dans l'arbre.

Implémentez les deux fonctions de recherche vues en cours : la Depth-First Search (DFS) et la Breadth-First Search (BFS). Implémentez les dernières fonctions permettant de manipuler les arbres binaires :

- La fonction renvoyant la hauteur d'un arbre.
- La fonction permettant d'afficher les éléments de l'arbre.
- La fonction pour calculer la profondeur d'un nœud.
- La fonction permettant de supprimer un élément.
- La fonction permettant de supprimer l'arbre.

Question bonus : implémentez les fonctions suivantes :

- Une fonction d'insertion dans l'arbre par rapport à la valeur d'un nœud.
- Une fonction pour supprimer le sous-arbre d'un nœud.
- Une fonction pour ajouter un sous-arbre à un nœud.
- La fonction de recherche Depth-First Search en itératif.

Exercice 2

Implémentez la structure d'arbre binaire de recherche. Cette dernière devra proposer des fonctions équivalentes à celles de votre arbre binaire simple. Pour rappel, dans les arbres binaires de recherche, les nœuds sont ordonnés selon la clé ou la valeur du nœud.

Exercice 3

Les arbres AVL sont des arbres binaires qui ont comme caractéristique que la différence des hauteurs des sous-arbres droit et gauche d'un nœud ne peut excéder 1. À partir de vos implémentations précédentes et du cours, créez la structure d'arbre AVL :

1. Implémentez la fonction pour calculer le facteur d'équilibre d'un nœud.
2. Implémentez les fonctions de rotation gauche et droite.
3. Modifiez vos méthodes d'insertion et de suppression pour les appliquer aux arbres AVL.

Exercice bonus

Les arbres N-aires, à la différence des arbres binaires, permettent à leurs nœud d'avoir plus de deux enfants, souvent utilisés pour représenter des liens hiérarchiques (système de fichiers, par exemple). Implémentez la structure d'arbre N-aire et les fonctions suivantes :

1. Les fonctions pour vérifier qu'un arbre est vide ou non, pour insérer un élément dans l'arbre, pour renvoyer la hauteur de l'arbre et pour afficher ses éléments.
2. La fonction pour calculer le degré d'un nœud, c'est-à-dire son nombre d'enfant.
3. Les fonctions de recherche en profondeur et en largeur.
4. Les fonctions pour supprimer un élément de l'arbre, le sous-arbre d'un nœud et l'arbre lui-même.
5. Les fonctions pour ajouter un sous-arbre à un nœud et pour insérer un élément dans l'arbre par rapport à la valeur d'un nœud.