Rapport TP4 NF16 : Les Arbres Binaires de Recherche

Lola Trupin

Alexandre Eberhardt

Introduction

L'objectif de ce TP est de développer un système pour gérer les inscriptions des étudiants aux UVs en utilisant des structures de données telles que les arbres binaires et les listes chaînées.

1. Structure

Les structures de données utilisées dans ce TP sont les suivantes :

* Structure T\_Element : Représente une inscription individuelle à une UV. Elle est composée d'un code d'UV et d'un pointeur vers l'inscription suivante, formant ainsi une liste chaînée.
* Structure T\_Noeud : Représente un étudiant dans l'arbre binaire. Elle contient le nom et le prénom de l'étudiant, un pointeur vers la liste chaînée de ses inscriptions (T\_Element), et des pointeurs vers les fils gauche et droit, représentant d'autres étudiants dans l'arbre.
* Structure T\_Arbre : Représente l'arbre binaire dans son ensemble. Il s'agit d'un pointeur vers un T\_Noeud qui agit comme la racine de l'arbre.

2. Fonctions implémentées et complexité

* creerInscription : Crée une nouvelle inscription. Complexité en O(1).
* rechercherInscription : Recherche une inscription spécifique dans la liste. Complexité en O(n), où n est le nombre d'inscriptions.
* ajouterInscription : Ajoute une inscription à la liste d'un étudiant. Complexité en O(n).
* rechercherNoeud : Recherche un étudiant dans l'arbre binaire. Complexité en O(h), où h est la hauteur de l'arbre.
* creerNoeud : Crée un nouveau nœud pour un étudiant. Complexité en O(1).
* inscrire : Inscrit un étudiant à une UV. Complexité en O(h).
* afficherInscriptions : Affiche toutes les inscriptions d'un étudiant. Complexité en O(n), où n est le nombre total d'inscriptions.
* Autres fonctions : Incluent des fonctions auxiliaires pour la gestion des chaînes de caractères et le nettoyage du buffer d'entrée.

3. Conclusion

Grâce à ce TP, nous avons pu expérimenter l’implémentation de listes chaînées et d’arbres binaires de recherche en C, ainsi que leur gestion. Bien que le code soit fonctionnel, il existe des opportunités d'optimisation et d'amélioration, notamment en matière de gestion de la mémoire et de performance.