**TP4 : Les Arbres Binaires de Recherche**

Objectifs du TP

L’objectif de ce TP est de créer un menu permettant à l’utilisateur de manipuler et analyser des arbres binaires de recherche (ABR).

**Fichier 1 : main.c**

Ce fichier contient la fonction main() et constitue l’interface utilisateur permettant d'interagir avec le menu du projet. Il est crucial de prendre en compte des cas extrêmes pour garantir la robustesse du programme :

Suppression propre de l'ABR existant avant la création d'un nouvel ABR vide.

Vérification de la validité de l'élément saisi (doit être un entier).

Vérification de l'existence de l'ABR avant d'ajouter ou de supprimer un élément.

Vérification de l'existence de l'élément à supprimer.

**Fichier 2 : tp4.h**

Ce fichier est divisé en deux parties, l’une pour déclarer les structures nécessaires pour représenter un ABR, et l’autre pour déclarer les prototypes des fonctions nécessaires pour manipuler et analyser cet ABR.

Structures

**T\_Sommet** : représente un sommet dans l'ABR et contient :

borneInf (int) : borne inférieure de l'intervalle représenté par le sommet.

borneSup (int) : borne supérieure de l'intervalle représenté par le sommet.

filsGauche (struct T\_Sommet\*) : pointeur vers le fils gauche.

filsDroit (struct T\_Sommet\*) : pointeur vers le fils droit.

T\_Arbre : représente un ABR de type T\_Sommet\*.

**Prototypes des fonctions**

T\_Sommet \*creerSommet(int element) : créer un sommet.

T\_Arbre insererElement(T\_Arbre abr, int element) : insérer un élément dans l'ABR.

T\_Sommet \*rechercherElement(T\_Arbre abr, int element) : rechercher un élément dans l'ABR.

void afficherSommets(T\_Arbre abr) : afficher tous les sommets de l'ABR.

void afficherElements(T\_Arbre abr) : afficher tous les éléments de l'ABR.

T\_Arbre supprimerElement(T\_Arbre abr, int element) : supprimer un élément de l'ABR.

void tailleMemoire(T\_Arbre abr) : afficher la taille en mémoire de l'ABR.

**Fichier 3 : tp4.c**

Ce fichier contient la définition des fonctions utilisées dans le projet.

**Fonctions :**

T\_Sommet creerSommet(int element)\*

Entrée : un entier element

Sortie : un pointeur vers la structure T\_Sommet

Complexité : O(1)

Justification : Cette fonction alloue de la mémoire pour la structure T\_Sommet et initialise les champs borneInf et borneSup avec la valeur de element.

T\_Arbre insererElement(T\_Arbre abr, int element)

Entrée : un pointeur vers la structure T\_Arbre et un entier element

Sortie : un pointeur vers la racine de l'ABR

Complexité : O(h)

Justification : Cette fonction insère l'élément dans l'ABR en respectant les propriétés de l'ABR.

T\_Sommet rechercherElement(T\_Arbre abr, int element)\*

Entrée : la structure T\_Arbre et un entier element

Sortie : un pointeur vers le sommet contenant l'intervalle avec l'élément, ou NULL si l'élément n'est pas trouvé

Complexité : O(h)

Justification : Cette fonction parcourt l'ABR pour trouver l'élément recherché.

void afficherSommets(T\_Arbre abr)

Entrée : la structure T\_Arbre

Sortie : pas de sortie

Complexité : O(n)

Justification : Cette fonction effectue un parcours en ordre pour afficher les sommets de l'ABR.

void afficherElements(T\_Arbre abr)

Entrée : la structure T\_Arbre

Sortie : pas de sortie

Complexité : O(n)

Justification : Cette fonction effectue un parcours en ordre pour afficher tous les éléments de l'ABR.

T\_Arbre supprimerElement(T\_Arbre abr, int element)

Entrée : un pointeur vers la structure T\_Arbre et un entier element

Sortie : un pointeur vers la racine de l'ABR

Complexité : O(h)

Justification : Cette fonction supprime l'élément de l'ABR et la création de sommets si nécessaire.

void tailleMemoire(T\_Arbre abr)

Entrée : la structure T\_Arbre

Sortie : pas de sortie

Complexité : O(n)

Justification : Cette fonction calcule et affiche la taille en mémoire de l'ABR, la taille qu'occuperait un ABR classique, et la mémoire économisée grâce à la représentation par intervalles.