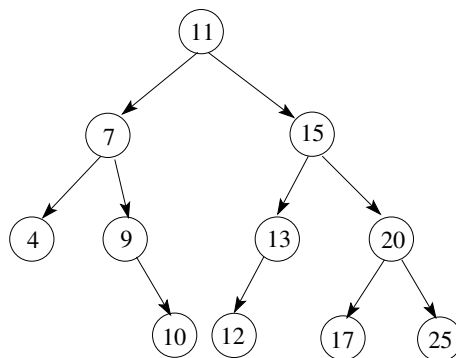


TP1-2 : Arbre binaire de recherche

Un arbre binaire de recherche (ABR) est un arbre binaire ordonné tel que pour tout nœud N , toutes les valeurs du sous-arbre gauche de N sont inférieures ou égales à la valeur de N , et toutes les valeurs du sous-arbre droit de N sont supérieures (strictement) à la valeur de N .

Un avantage de ces arbres est qu'ils permettent de rechercher un élément plus rapidement qu'avec un tableau ou une liste chaînée.



Exemple d'arbre binaire de recherche

Bonnes pratiques :

1. tester les fonctions au fur et à mesure en faisant dans un fichier de test autant de fonctions de test que nécessaire.
2. usage de Valgrind (cf. à la fin du sujet)

Exercice 1 : Écrire le package des ABR en définissant :

- les structures de données nécessaires (Noeud, Abr, Boolean);
- `creerArbrevide` : fonction qui crée un ABR vide;
- `estArbreVide` : fonction qui indique si l'arbre est vide ou pas;
- `insérerEnFeuille` : fonction qui insère une valeur donnée comme feuille dans l'ABR donné;
- `viderArbre` : fonction qui vide/détruit l'ABR donné;
- `afficherCroissant` : procédure qui affiche dans l'ordre croissant les valeurs de l'ABR;
- `afficherDecroissant` : procédure qui affiche dans l'ordre décroissant les valeurs de l'ABR;
- `afficherCouche` : procédure qui dessine l'ABR couché (la racine à gauche et les feuilles à droite)
- `rechercherVal` : fonction qui recherche une valeur donnée dans un ABR donné. Elle retourne un booléen indiquant si la valeur a été trouvée ou pas;
- `supprimerVal` : fonction qui enlève la valeur donnée de l'arbre. la fonction `supprimerVal` appellera une fonction locale `oterMax` qui permet de supprimer la valeur la plus grande d'un arbre donné et de retourner sa valeur.

Exercice 2 : On va écrire le package des AVL en faisant quelques "copier/coller intelligents" du package précédent pour gagner un peu de temps.

Attention, ici :

- un Noeud connaît sa hauteur (h) ou plus précisément la hauteur de l'arbre dont il est racine ;
- et l'AVL doit rester équilibré même après une insertion ou une suppression.

Vous ajouterez les fonctions locales :

- `mettreAJourHauteur` qui met à jour la hauteur de l'arbre (le champ h de la racine de l'arbre) passé en paramètre en fonction de celles des sous-arbres.
- `rotationGauche` et `rotationDroite` qui effectuent une rotation à gauche ou à droite de l'ABR pris en paramètre.
- `reequilibrer` qui met en oeuvre la ou les bonnes rotations pour rétablir l'équilibre de l'arbre passé en paramètre. Si l'arbre est bien équilibré, elle met à jour la hauteur de la racine de l'arbre.

Exercice 3 : Comparaison dans les deux types d'ABR : équilibré ou pas.

Faire des tests qui construisent des ABR et des AVL contenant les valeurs de 1 à 10000 (resp. 50000, resp. 100000) font des recherches et les détruisent. Et mesurer pour chacun le temps mis aux différentes étapes (construction, recherche et destruction de l'arbre).

Vous pouvez utiliser les fonctions `time()` ou `clock()` de `time.h`.

ABR déséquilibré avec des nœuds de 1 à n	tps construct.	tps rech. de -1	tps rech. de $n/2$	tps rech. de $n+1$	tps de destruc.
nœuds 1 à 10000					
nœuds 1 à 50000					
nœuds 1 à 100000					

AVL avec des nœuds de 1 à n	tps construct.	tps rech. de -1	tps rech. de $n/2$	tps rech. de $n+1$	tps de destruc.
nœuds 1 à 10000					
nœuds 1 à 50000					
nœuds 1 à 100000					

Valgrind : Valgrind est un outils permettant de déboguer, d'effectuer du profilage de code et mettre en évidence des fuites mémoires. Il est composé de différents modules dont `memcheck` qui piste des erreurs comme :

- l'utilisation de variables ou de pointeurs non initialisés ;
- l'accès à des zones mémoire libérées ou non allouées ;
- l'oubli de libérer la mémoire allouée ;
- la libération d'une zone mémoire déjà désallouée ;
- l'accès à une case qui n'existe pas dans un tableau dynamique

Pour pouvoir l'utiliser on compile avec l'option `-g` (mode debug) :

```
gcc -Wall -g abr.c progPrinc.c -o exo
```

puis on exécute avec `valgrind` : `valgrind ./exo` par exemple, ou en mettant des options pour avoir plus de précisions dans les messages d'erreur.

lien utile : <https://valgrind.org/docs/>