

TD5 Programmation C/C++

Pointeurs & Git

Le travail sera fait par groupe de 3 étudiants : la conception et la mise en place de l'environnement seront collectifs alors que l'implémentation des opérations sera répartie et gérée avec *Git*.

Arbre binaire de recherche

En informatique, un **arbre** est une structure de données qui sert à mémoriser des données. Il est constitué d'éléments que l'on appelle des nœuds. Un nœud peut avoir plusieurs fils et un seul père. Un arbre possède un nœud particulier que l'on appelle sa racine : c'est l'unique nœud qui est père sans être fils d'un autre nœud. Les nœuds ne possédant pas de fils sont appelés les feuilles de l'arbre.

Un **arbre binaire** est un arbre dans lequel chaque nœud peut avoir au maximum deux fils. Chacun des fils est la racine d'un sous-arbre de l'arbre global. Pour différencier ces sous-arbres, on les nomme souvent 'droite' ou 'gauche'.

Un **arbre binaire de recherche** (ABR) est un arbre binaire dans lequel chaque nœud possède une clé (une valeur), telle que :

- chaque nœud du sous-arbre gauche ait une clé inférieure ou égale à celle du nœud considéré, et
- chaque nœud du sous-arbre droit possède une clé supérieure ou égale à celle du nœud considéré.

Selon la mise en œuvre de l'ABR, on pourra interdire ou non des clés de valeur égale. Les nœuds que l'on ajoute deviennent des feuilles de l'arbre.

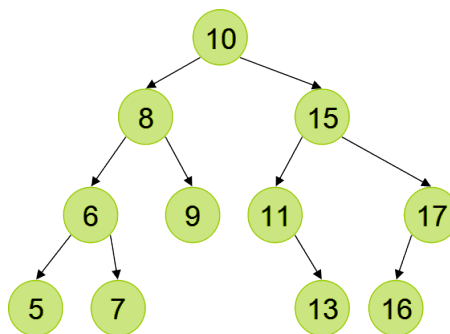


FIGURE 1 – Exemple d'un ABR

Exercice 1 : Initialisation environnement Git (en groupe)

Mettez en place le *dépôt* dans lequel vous allez implémenter la classe ABR (suivre les étapes énoncées dans l'annexe du td ; Ajoutez-moi en tant que collaborateur : mon nom utilisateur *GitHub* est *mcharfi2016*).

Exercice 2 : Conception & Implémentation de la classe ABR (en groupe)

1. Chaque nœud contient 3 pointeurs :
 - (a) fg : qui pointe sur le fils de gauche,

- (b) fd : qui pointe sur le fils de droite, et
 - (c) cle : qui pointe sur la clé qui est de type entier.
2. Quel(s) est(sont) le(s) constructeur(s) nécessaire(s) ?
 3. A-t-on besoin de getteurs, setteurs ?
 4. A-t-on besoin d'un destructeur ?

Exercice 3 : Opérations (séparément)

Répartissez entre vous l'implémentation des différentes opérations présentées ci-dessous comme suit :

- *Étudiant 1* : la création d'un ABR, sa vérification et la suppression,
- *Étudiant 2* : la recherche, le parcours et le min, et
- *Étudiant 3* : l'insertion, le max et la création (main).

N'oubliez pas de mettre à jour votre travail sur le *dépôt* à chaque étape.

Chaque étape doit être commentée (vous allez me montrer le *log* à la fin de la séance).

1. Création d'un ABR

Il s'agit de la fonction qui permet de créer un arbre vide.

2. Vérification d'un ABR

Cette fonction permet de vérifier si un arbre est vide ou non.

3. Recherche

Il s'agit d'une procédure récursive qui cherche un nœud ayant une clé donnée dans l'arbre.

- (a) On commence par examiner la racine. Si sa clé est la clé recherchée, l'algorithme se termine et renvoie la racine.
- (b) Si elle est strictement inférieure, alors elle se trouve dans le sous-arbre gauche, sur lequel on effectue alors récursivement la recherche.
- (c) Si la clé recherchée est strictement supérieure à la clé de la racine, la recherche continue dans le sous-arbre droit.
- (d) Si on atteint une feuille dont la clé n'est pas celle recherchée, on sait alors que la clé recherchée n'appartient à aucun nœud, elle ne figure donc pas dans l'arbre de recherche.

4. Insertion

L'insertion d'un nœud commence par une recherche : on cherche la clé du nœud à insérer ; lorsqu'on arrive à une feuille, on ajoute le nœud comme fils de la feuille en comparant sa clé à celle de la feuille : si elle est inférieure, le nouveau nœud sera à gauche ; sinon il sera à droite. En fait, le nouveau nœud sera insérer à l'endroit où la recherche s'est arrêtée.

5. Suppression

La suppression d'un nœud commence aussi par une recherche : on cherche le nœud à supprimer en fonction de sa clé. Ensuite, selon la position du nœud en question dans l'arbre, il faut identifier l'un des trois cas suivants :

- (a) *suppression d'une feuille* : Il suffit de l'enlever de l'arbre vu qu'elle n'a pas de fils.
- (b) *suppression d'un nœud avec un seul enfant* : Il faut l'enlever de l'arbre en le remplaçant par son fils.
- (c) *suppression d'un nœud avec deux enfants* : Supposons que le nœud à supprimer soit appelé N.
 - i. On échange le nœud N avec son successeur le plus proche (i.e., le nœud le plus à gauche du sous-arbre droit) ou son plus proche prédécesseur (i.e., le nœud le plus à droite du sous-arbre gauche). Cela permet de garder une structure d'arbre binaire de recherche.
 - ii. Puis on applique à nouveau la procédure de suppression à N, qui est maintenant une feuille ou un nœud avec un seul fils.

6. Parcours

Il s'agit de récupérer les clés d'un arbre binaire de recherche dans l'ordre croissant en réalisant un parcours en profondeur. On parcourt récursivement le sous-arbre gauche, puis en ajoutant la racine, puis en parcourant récursivement le sous-arbre droit.

7. **Min** (resp. **Max**)

Il s'agit de retourner la valeur la plus petite (resp. grande) contenue dans l'arbre.

8. **Création (main)**

Il s'agit d'implémenter le programme qui se charge de créer une instance d'un objet de la classe ABR et tester les différentes opérations.

Les figures ci-dessous contiennent des exemples des opérations d'insertion, recherche et suppression.

Insertion de la valeur 12

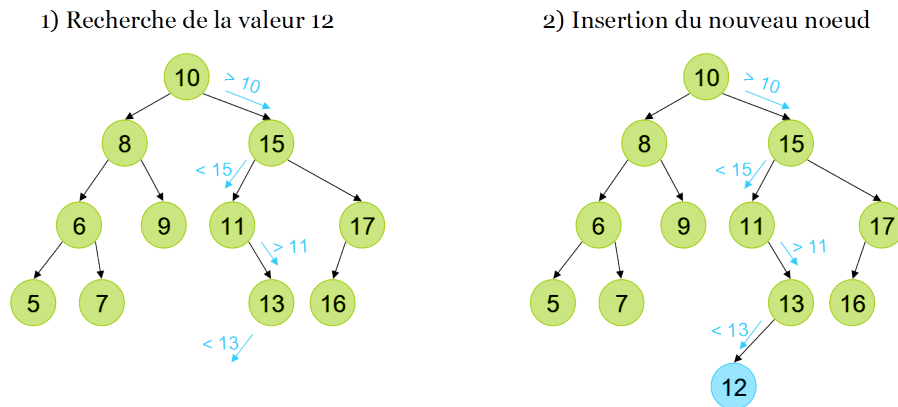


FIGURE 2 – Exemple d'insertion dans un ABR

Recherche de la valeur 7

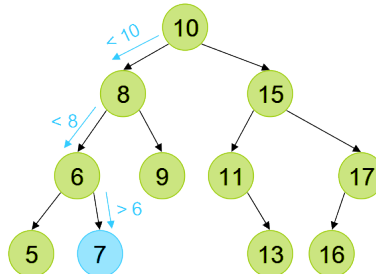
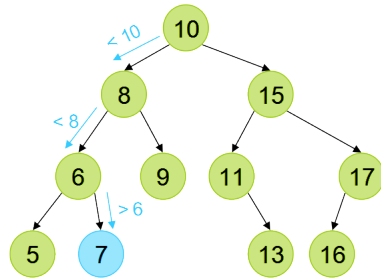


FIGURE 3 – Exemple de recherche dans un ABR

1^{er} cas de Suppression : Suppression du noeud avec la valeur 7

1) Recherche de la valeur 7



2) Suppression de la valeur 7

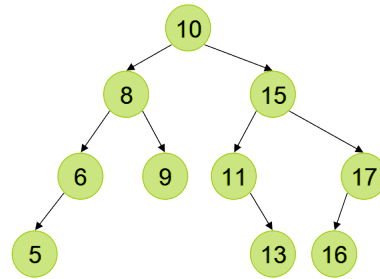
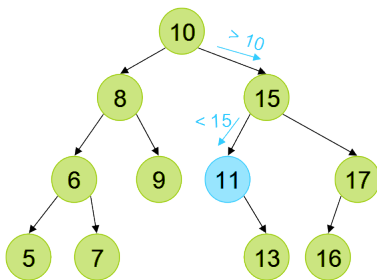


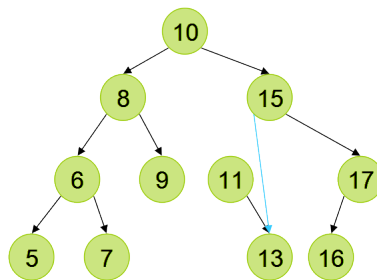
FIGURE 4 – Exemple de suppression d'un nœud dans un ABR : nœud feuille

2^{em} cas de Suppression : Suppression du noeud avec la valeur 11

1) Recherche de la valeur 11



2) Remplacer le noeud par son fils



3) Suppression du noeud

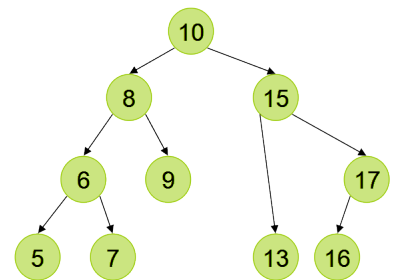
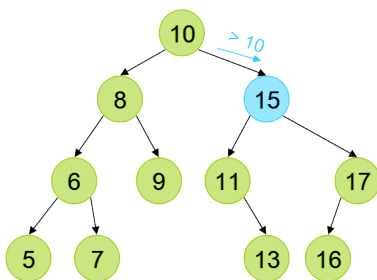


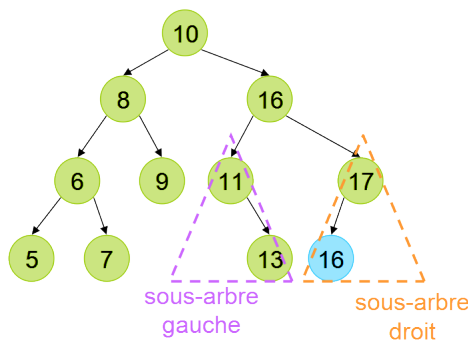
FIGURE 5 – Exemple de suppression d'un nœud dans un ABR : nœud avec un seul fils

3^{em} cas de Suppression : Suppression du noeud avec la valeur 15

1) Recherche de la valeur 15



2) Remplacer le noeud par son successeur le plus proche



3) Suppression du noeud

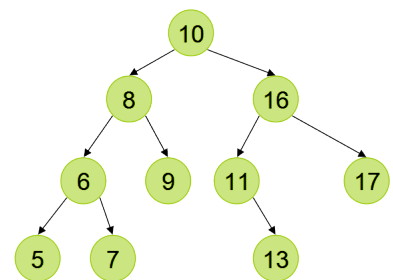


FIGURE 6 – Exemple de suppression d'un nœud dans un ABR : nœud avec deux fils