

Arbre Binaire de Recherche

Programmation Fonctionnelle
Master 2 I2L apprentissage

SÉBASTIEN VEREL

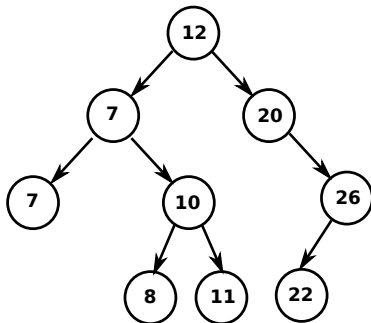
verel@univ-littoral.fr

<http://www-lisic.univ-littoral.fr/~verel>

Université du Littoral Côte d'Opale
Laboratoire LISIC
Equipe OSMOSE

Mai 2021

Définition



Arbre Binaire de Recherche (ABR)

Arbre binaire où pour tous les noeuds, les étiquettes (clés) de l'arbre de gauche sont inférieures (ou égale) à l'étiquette du noeud et les étiquettes de l'arbre droit sont strictement plus supérieures.

Type de données abstrait

Opérations de base sur les ABR

- Insertion d'un élément
- Suppression d'un élément
- Recherche d'un élément

Type de données abstrait

Opérations de base sur les ABR

- Insertion d'un élément
- Suppression d'un élément
- Recherche d'un élément

Exercice

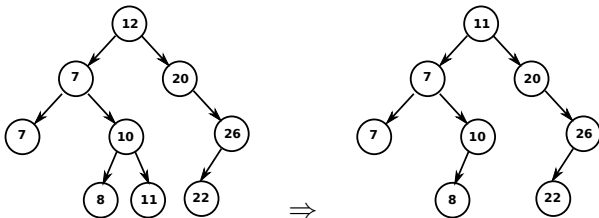
Implémenter le TDA.

Remarque 1 : la suppression est le plus difficile à réaliser.

Remarque 2 : il est bon d'avoir une fonction d'affichage pour les tests.

Suppression simple

La suppression du nombre 12 à la racine peut donner :



Complexité

Complexité des opérations précédentes

Pour un arbre binaire de taille n ,
la complexité dans le pire des cas (*i.e.* le peigne) est :

Complexité

Complexité des opérations précédentes

Pour un arbre binaire de taille n ,
la complexité dans le pire des cas (*i.e.* le peigne) est :

$$\mathcal{O}(n)$$

Pour un arbre binaire **équilibré** de taille n ,
la complexité dans le pire des cas est :

Complexité

Complexité des opérations précédentes

Pour un arbre binaire de taille n ,
la complexité dans le pire des cas (*i.e.* le peigne) est :

$$\mathcal{O}(n)$$

Pour un arbre binaire **équilibré** de taille n ,
la complexité dans le pire des cas est :

$$\mathcal{O}(\log(n))$$

L'insertion et la suppression doivent maintenir la propriété d'arbre équilibré...

Arbre AVL

Arbre presque équilibré

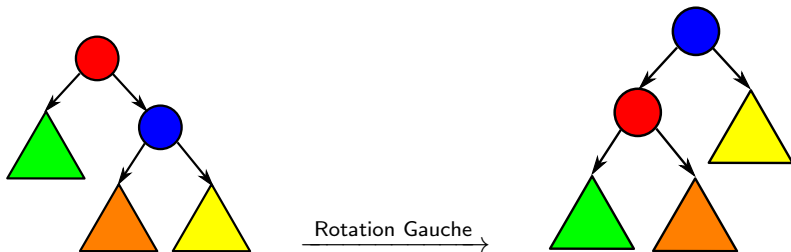
Un arbre est presque équilibré si pour tous sous-arbres la différence de hauteur entre l'arbre de gauche et l'arbre de droite est au plus une unité.

Arbre AVL

- Arbre AVL : arbre automatiquement équilibré, la propriété d'arbre presque équilibré est maintenue lors de l'insertion et la suppression.
- Inventé en 1962 par Georgii Adelson-Velsky et Evgenii Landis (d'où AVL)

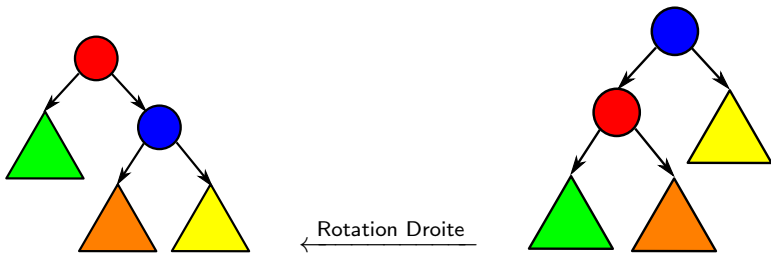
Principes

L'insertion et la suppression utilisent la rotation à gauche et à droite d'un arbre lorsqu'il est trop déséquilibré :



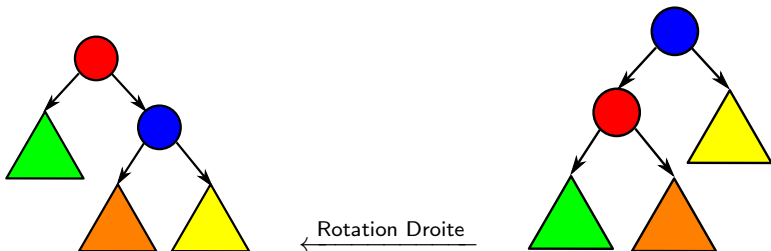
Principes

L'insertion et la suppression utilisent la rotation à gauche et à droite d'un arbre lorsqu'il est trop déséquilibré :



Principes

L'insertion et la suppression utilisent la rotation à gauche et à droite d'un arbre lorsqu'il est trop déséquilibré :



Exercice

Définir les fonctions `rg` et `rd` qui permettent les rotations à gauche et à droite d'un arbre.

Rotation à gauche

Rotation à gauche

```
rg :: Tree a -> Tree a
rg Empty = Empty
rg (Node e ag ad) =
  Node (root ad) (Node e ag (left ad)) (right ad)
```

Rotation à droite

Rotation à droite

```
rd :: Tree a -> Tree a
rd Empty = Empty
rd (Node e ag ad) =
  Node (root ag) (left ag) (Node e (right ag) ad)
```


Insertion

Principe

- Insérer le nombre au bon endroit, le noeud N
- Au fur et à mesure de la remontée, de N à la racine, rééquilibrer l'arbre en effectuant les rotations convenables.

Insertion

A : arbre courant, Ag : arbre gauche de A, Ad : arbre droit de A, Agg : arbre gauche de Ag, Agd : arbre droit de Ag, Adg : arbre gauche de Ad, Add : arbre droit de Ad.

Principe du rééquilibrage en chaque nœud jusqu'à la racine

```
si |hauteur(Ag) - hauteur(Ad)| > 1 alors
  si hauteur(Ag) - hauteur(Ad) = 2 alors
    si hauteur(Agg) ≥ hauteur(Agd) alors
      rd(A)
    sinon
      rg(Ag) puis rd(A)
    fin si
  sinon
    si hauteur(Add) ≥ hauteur(Adg) alors
      rg(A)
    sinon
      rd(Ad) puis rg(A)
    fin si
  fin si
fin si
```

Insertion

Attention

Il faut changer le TDA pour enregistrer la hauteur de chaque noeud, sinon la complexité devient catastrophique...

Suppression

Principe

- Même principe que l'insertion
- Supprimer le nœud au bon endroit, le nœud N
- Au fur et à mesure de la remontée, de N à la racine, rééquilibrer l'arbre en effectuant les rotations convenables.