

Numérique et science informatique
Classe de Terminale

Lycée hoche

année scolaire 2021-2022

Contents

1	Introduction	2
2	Structure d'un ABR	2
	2.1 Attributs des noeuds	2
	2.2 Propriétés des clés	2
3	Affichage et recherche de clés dans un ABR	2
	3.1 Affichage des clés par ordre croissant	2
4	Ajout d'une clé dans un ABR	3
	4.1 Méthode	3
	4.2 Exemple	3

1 Introduction

2 Structure d'un ABR

Cette ressource s'intéresse à une classe particulière d'arbres binaires : les arbres binaires de recherche appelés également les ABR. Ceux-ci définissent des structures de données qui ont pour structure logique un arbre binaire et qui peuvent supporter des opérations courantes sur des ensembles dynamiques ; par exemple : rechercher, minimum, maximum, prédécesseur, successeur, ajouter, supprimer, etc. Ces arbres sont fondamentaux dans beaucoup de domaines : gestion des fichiers sur un disque dur, etc.

2.1 Attributs des noeuds

Un ABR étant un arbre binaire, on utilise pour ses nœuds les mêmes attributs que ceux vus dans la leçon « Généralités sur les arbres binaires » :

- $n.cle$ désigne la « valeur » contenue dans n ;
- $n.père$ désigne le père de n ;
- $n.gauche$ désigne le fils gauche de n ;
- $n.droite$ désigne le fils droit de n . On donne la valeur particulière null à un attribut lorsque celui-ci est vide.

2.2 Propriétés des clés

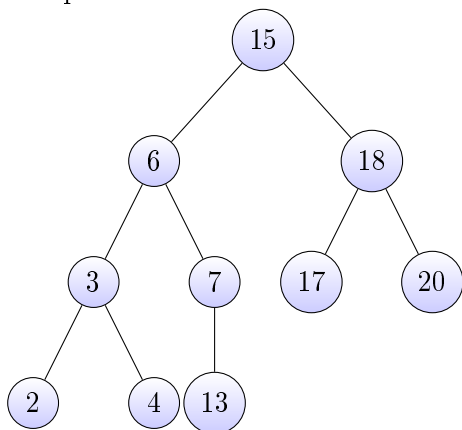
Contrairement à un arbre binaire général, les clés d'un ABR doivent être comparables entre elles. Plus précisément, pour tout nœud x et pour tout descendant y de x , alors :

- si y est un nœud du sous-arbre gauche de x , alors $y.cle \leq x.cle$;
- si y est un nœud du sous-arbre droit de x , alors $y.cle \geq x.cle$

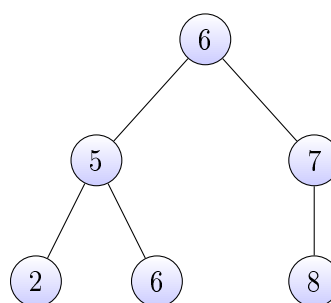
Remarque:

Dans la suite, nous ne considérons que des clés numériques ou chaînes de caractères.

Exemples:



(a) avec des clés numériques distinctes



(b) avec des clés numériques non distinctes

3 Affichage et recherche de clés dans un ABR

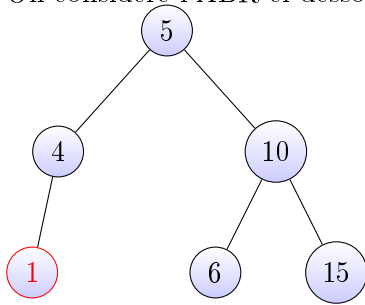
3.1 Affichage des clés par ordre croissant

Exercice:

Construire un ABR dont les clés sont affichées dans l'ordre lors d'un parcours en profondeur avec un traitement préfixé.

Exercice

On considère l'ABR ci-dessous.



- Donner l'affichage des clés que l'on doit obtenir.
- Avec quel type de parcours obtient-on un tel affichage ?

Théorème 1

Un parcours en profondeur d'un ABR avec traitement infixé renvoie les clés de l'ABR par ordre croissant.

4 Ajout d'une clé dans un ABR

L'ajout d'une clé dans un ABR est une opération délicate. En effet, il faut modifier l'ensemble dynamique représenté par l'ABR tout en gardant une structure d'ABR. Il existe principalement deux méthodes d'insertion :

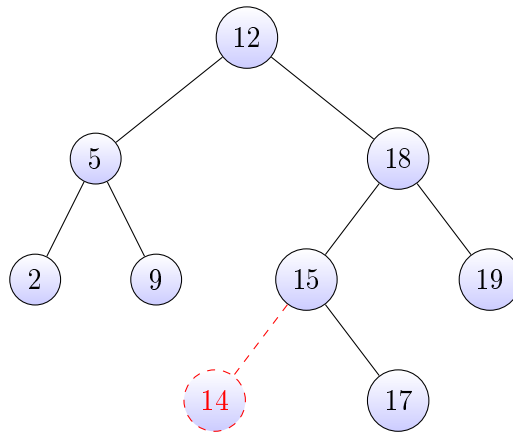
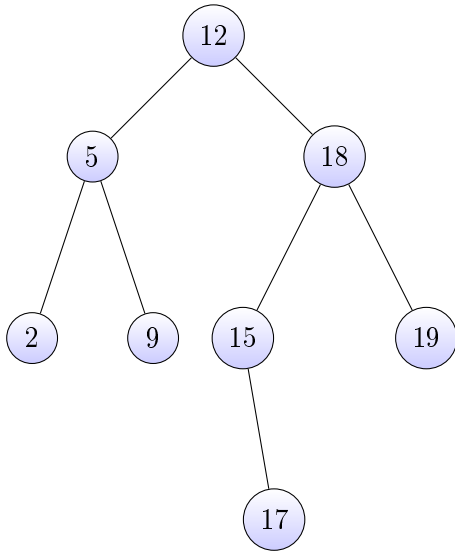
- l'insertion aux feuilles : on modifie dans ce cas la hauteur de l'ABR ;
- l'insertion à la racine : on modifie dans ce cas la structure même de l'ABR. L'insertion aux feuilles est bien sûr la méthode d'insertion la plus simple et c'est uniquement celle-ci que nous traitons ci-dessous.

4.1 Méthode

Pour insérer un nœud aux feuilles d'un ABR, il suffit de partir de la racine de l'ABR, de parcourir un chemin descendant jusqu'à un attribut null en testant la valeur de la clé du nœud à insérer avec la valeur de la clé de chaque nœud rencontré sur le chemin, puis de remplacer l'attribut null par le nœud voulu. Ce nœud devient alors une feuille de l'ABR.

4.2 Exemple

Dans l'exemple ci-dessous, la figure de gauche montre l'ABR initial et la figure de droite le nouvel ABR une fois que le nœud 14 a été inséré. Les nœuds en gris clair indiquent le chemin descendant de la racine vers la position où l'élément doit être inséré et la ligne en pointillés indique le lien qui a été ajouté à l'ABR pour insérer l'élément 14.



4.3 Recherche de la clé maximale

Exercice 3

1. Quelle est la clé maximale de l'ABR de l'exercice 2 ? Où est-elle située ?
2. De façon générale, où est située la clé maximale dans un ABR et comment doit-on la chercher ?

Exercice 4

Compléter l'algorithme suivant pour qu'il renvoie la clé maximale de l'ABR B :

Algorithme 1 : recherche de la clé maximale

```

1 fonction MaximumABR(B)
2   noeud ← .....
3   tant que ..... ≠ ..... faire
4     |   noeud ← .....
5     |   afficher x.clé
6   fin
7   Renvoyer .....
8 fin fonction
  
```

Quelle est la complexité de cet algorithme dans le pire des cas?

4.4 Recherche d'une clé particulière

Pour rechercher une clé particulière dans un ABR, la méthode est extrêmement simple : il suffit de parcourir l'arbre depuis la racine jusqu'à la clé (ou une feuille si la clé n'existe pas) en branchant si nécessaire à chaque nœud en fonction de la valeur de la clé :

- dans le sous-arbre gauche si $clé < noeud.clé$;
- dans le sous-arbre droit si $clé > noeud.clé$.

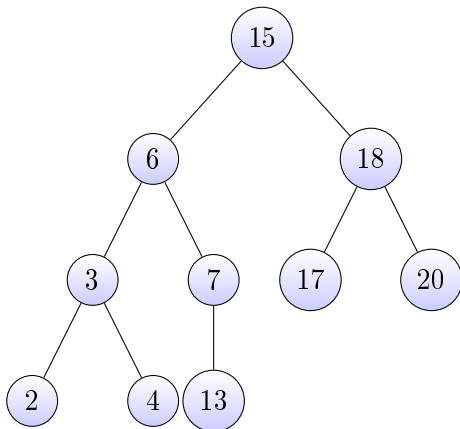
On peut en particulier montrer que celui-ci est en $O(h)$, où h désigne la hauteur de l'ABR.

4.5 Recherche du successeur d'un nœud

On souhaite déterminer le successeur d'un nœud dans l'ordre déterminé par un parcours infixe de l'arbre. Pour rendre l'algorithme très efficace, il faut réaliser cette recherche sans effectuer de comparaison entre les clés.

Exercice 5

- Quel est le successeur de 15 ? de 6 ? de 18 ? de 2 ?
- Pour la structure d'ABR de l'arbre, que représente chacun de ces successeurs vis-à-vis du nœud initial ?



Méthode:

En utilisant la structure d'un ABR, on déduit la méthode suivante pour déterminer le successeur d'un nœud :

- si le sous-arbre droit de est non vide, alors le successeur de est située dans ce sous-arbre droit ;
- si le sous-arbre droit de est vide et que a un successeur , alors est de dont est aussi

Attention, chaque nœud est un ancêtre et un descendant de lui-même.

Exercice 6

Compléter l'algorithme suivant:

Algorithme 2 : recherche du successeur d'une clé

```

1 fonction Successeur (n)
2 si n.droite ≠ null alors
3   renvoyer.....
4 fin
5 m ← n.père
6 tant que .....et ..... faire
7   n ←.....
8   m ←.....
9 fin
10 Renvoyer m
11 fin fonction
  
```

Algorithme

Algorithme 3 : Insertion d'un noeud dans un ABR

```

1  tant que  $x \neq \text{null}$  faire
2     $y \leftarrow x$ 
3    si  $z.cle < x.cle$  alors
4       $x \leftarrow x.gauche$ 
5    sinon
6       $x \leftarrow x.droite$ 
7    fin
8  fin
9   $z.pere \leftarrow y$ 
10 si  $y == \text{null}$  alors
11    $B.racine \leftarrow z$ 
12 sinon si  $z.cle < y.cle$  alors
13    $y.gauche \leftarrow z$ 
14 sinon
15    $y.droite \leftarrow z$ 
16 fin

```
