MINISTÈRE
DE L'ÉDUCATION
NATIONALE,
DE LA JEUNESSE
ET DES SPORTS

Liberté
Égalité
Fraternité

VOIE GÉNÉRALE

2DE | 1RE | TLE

Numérique et sciences informatiques

ENSEIGNEMENT
SPÉCIALITÉ

ARBRES BINAIRES DE RECHERCHE

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.

Structure d'un ABR

Attributs des nœuds

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

- n.clé 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.

Propriétés des clés

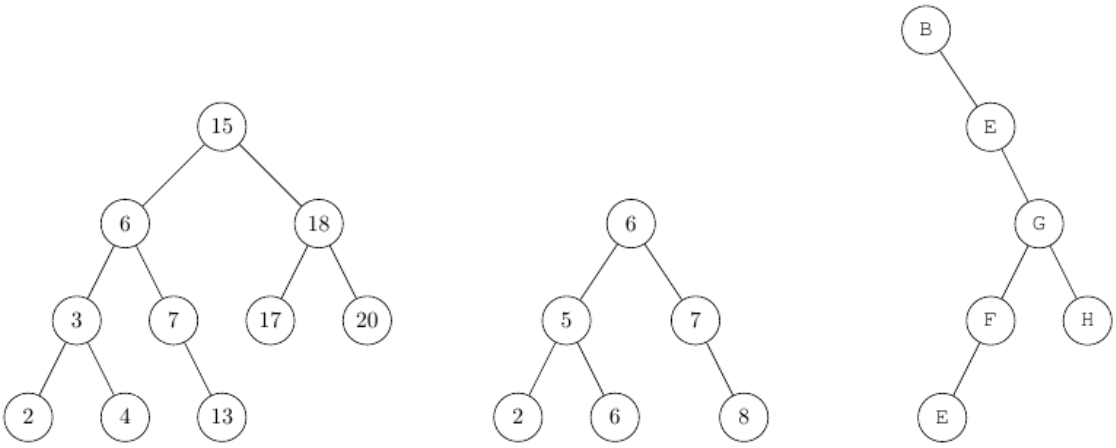
Contrairement à un arbre binaire général, les clés d'un ABR doivent être *comparables entre elles*. Plus précisément, pour tout nœud et pour tout descendant de , alors :

- si y est un nœud du sous-arbre gauche de x, alors y.clé ≤ x.clé ;
- si y est un nœud du sous-arbre droit de x, alors y.clé ≤ x.clé.

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

Retrouvez éduscol sur

Exemples



(a) Avec des clés numériques distinctes

(b) Avec des clés numériques non distinctes

(c) Avec des chaînes de caractères

FIGURE 1 – Quelques exemples d'ABR

Exercice 1

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

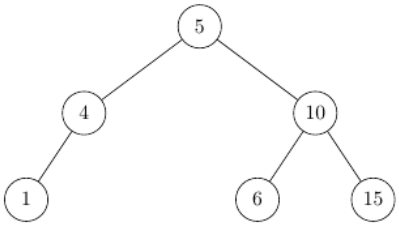
Affichage et recherche de clés dans un ABR

Affichage des clés par ordre croissant

Exercice 2

On considère l'ABR ci-contre.

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



On a plus généralement le théorème suivant :

Théorème 1

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

Exercice 3

En vous aidant du cours sur les arbres binaires, écrire un algorithme renvoyant la liste des clés dans l'ordre croissant. Quelle est la complexité de cet algorithme ?

Recherche de la clé maximale

Exercice 4

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

Exercice 5

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

```
fonction MaximumABR(B:un ABR)
   noeud ← ..............
   tant que .............. ≠ .........
      noeud ← ..............
   fin tant que
   renvoyer ..............
fin fonction
```

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

> **Remarque**
> On a un algorithme similaire pour rechercher la clé minimale d'un ABR. Il peut être proposé en exercice.

Recherche de la clé maximale

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}.

Un algorithme de recherche d'une clé particulière peut être étudié en exercice. On peut en particulier montrer que celui-ci est en , où désigne la hauteur de l'ABR.

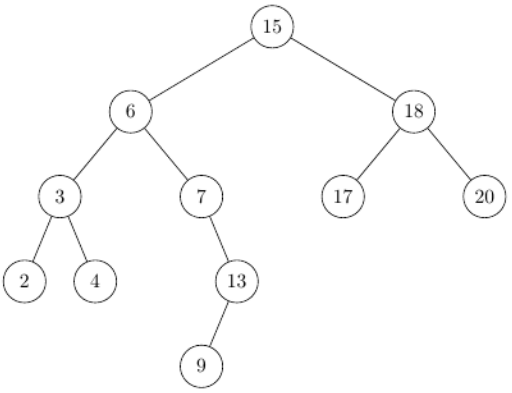
Recherche du successeur d'une clé

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

Exercice 6

On considère l'ABR suivant :

- Quel est le successeur de 15 ? de 17 ? de 13 ? de 20 ?
- Pour la structure d'ABR de l'arbre, que représente chacun de ces successeurs vis-à-vis du nœud initial ?



Méthode 1

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 7

Compléter l'algorithme suivant pour qu'il renvoie le successeur du nœud :

```
fonction Successeur(n:un noeud)
   si n.droite ≠ null alors
      renvoyer .....................................
   fin si
   m ← n.père
   tant que ..... et .....
      n ← .....
      m ← .....
   fin tant que
   renvoyer m
fin fonction
```

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

Remarques

- Le successeur d'un nœud peut avoir la même valeur que le nœud initial dans le cas où les clés ne sont pas distinctes.
- On a une méthode et un algorithme similaire pour rechercher le prédécesseur d'un nœud. Ceci peut être étudié en exercice.

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.

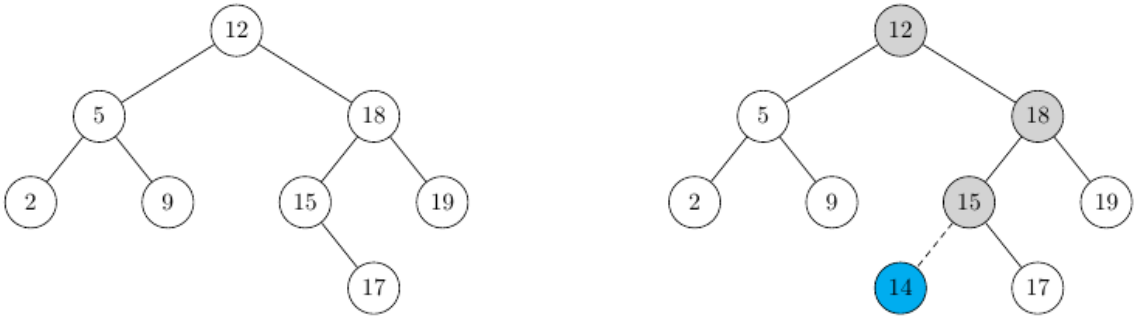
Méthode

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.

Retrouvez éduscol sur

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.



Algorithme

L'algorithme suivant permet d'insérer un nœud aux feuilles dans un ABR :

```
1   fonction InsérerABR(B:un ABR, z:le noeud à insérer)
2     y ← null
3     x ← B.racine
4     tant que x ≠ null
5       y ← x
6       si z.clé < x.clé alors
7         x ← x.gauche
8       sinon
9         x ← x.droite
10      fin si
11    fin tant que
12    z.père ← y
13    si y = null alors
14      B.racine ← z
15    sinon si z.clé < y.clé alors
16      y.gauche ← z
17    sinon
18      y.droite ← z
19    fin si
20  fin fonction
```

- Vis-à-vis de la méthode d'insertion présentée précédemment, à quoi correspondent les lignes 2 à 11 ? les lignes 12 à 19 ?
- Illustrer cet algorithme sur l'exemple ci-dessus.
- Quelle est la complexité de l'algorithme d'insertion aux feuilles dans le pire des cas ?

Implémentation

Une implémentation des ABR à l'aide, par exemple, de la programmation objet et de l'insertion aux feuilles peut être proposée en TP.