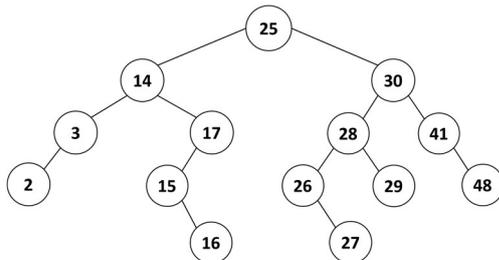


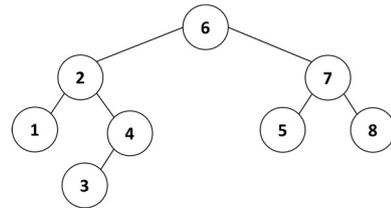
1. Définition

Un arbre binaire de recherche (ABR) est un cas particulier d'arbre binaire. Pour avoir un arbre binaire de recherche :

- Il faut avoir un arbre binaire.
- Il faut que les noms (ou clés) de nœuds composant l'arbre soient ordonnables (on doit pouvoir classer les nœuds, par exemple, de la plus petite clé à la plus grande).
- Les fils gauches d'un nœud doivent avoir une valeur inférieure ou égale à lui.
- Les fils droits d'un nœud doivent avoir une valeur supérieure à lui.



Arbre binaire de recherche



Pas arbre binaire de recherche

✂ Pourquoi le deuxième arbre ci-dessus n'est pas un arbre de recherche ?

2. Construction d'un arbre binaire de recherche (ABR)

Le problème de la construction d'un ABR correspond à celui de l'insertion d'un élément dans un ABR.

2.1. Descriptif de la construction d'un arbre binaire de recherche

Par exemple, on souhaite placer chaque caractère de la liste ['f', 'e', 'w', 'h', 'a', 'c', 'b', 'u', 'x'] dans un ABR.

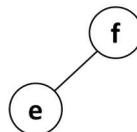
- Dans un premier temps on crée la racine de l'arbre.
- Puis on insère les nœuds en respectant les règles d'un ABR, comme ci-dessous.

① Insérer f



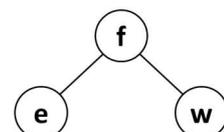
f est la racine de l'arbre

② Insérer e



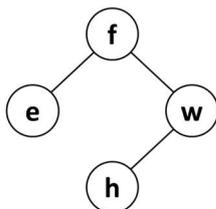
e < f donc fils gauche de f

③ Insérer w



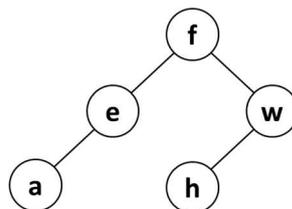
w > f donc fils droit de f

④ Insérer h



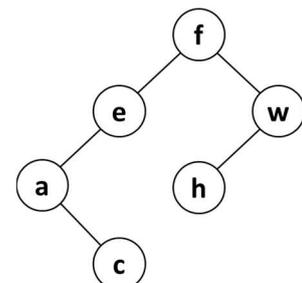
*h > f donc à droite puis
h < w donc fils gauche de w*

⑤ Insérer a

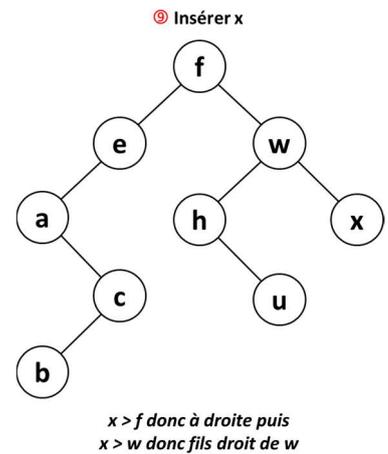
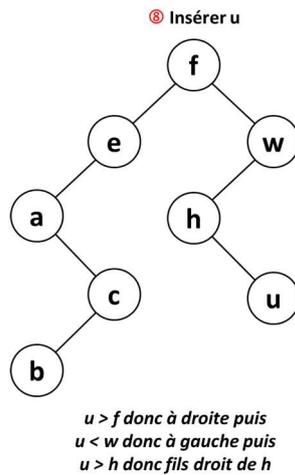
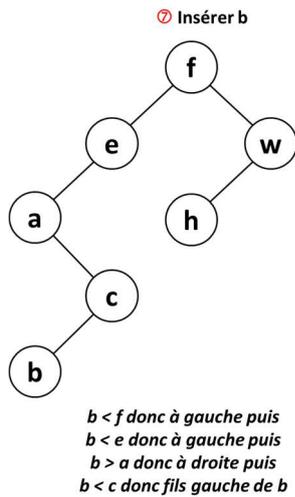


*a < f donc à gauche puis
a < e donc fils gauche de e*

⑥ Insérer c



*c < f donc à gauche puis
c < e donc à gauche puis
c > a donc fils droit de a*



2.2. Algorithme de construction d'un arbre binaire de recherche

✍ D'après le descriptif de la construction d'un arbre binaire de recherche, compléter l'algorithme `insert(arbre, valeur)` qui permet d'insérer un élément dans un arbre binaire de recherche.

```

fonction insert(arbre, valeur)
    si valeur ≤ racine alors
        si l'arbre n'a pas de fils gauche alors
            fils gauche ← Noeud(valeur, None, None)
        sinon
            insert(. . . . ., valeur)
    sinon
        si . . . . .
            . . . . .
        sinon
            insert(. . . . ., valeur)
    
```

2.3. Implémentation de la construction d'un arbre binaire de recherche

✍ Télécharger et décompresser le fichier `abr`. Il contient le fichier `abr.py` à compléter et le module `graph_tree` permettant de dessiner l'arbre.

✍ Ouvrir le fichier principal (`abr.py`).

✍ Compléter le fichier afin d'importer le module `graph_tree` (voir début du fichier). Le module contient une fonction `graphic_arbre(nom_arbre)` permettant de dessiner un arbre binaire.

✍ Le fichier contient aussi la classe `Noeud` qu'il faut compléter avec la méthode `insert(self, value)`. Elle permet d'insérer au bon endroit le nœud d'étiquette (de valeur) `value` dans l'arbre préexistant `self`.

```

# ===== création de l'arbre =====
#
#
# construction de l'arbre
arbre = Noeud('f', None, None)
arbre.insert('e')
arbre.insert('w')
arbre.insert('h')
arbre.insert('a')
arbre.insert('c')
arbre.insert('b')
arbre.insert('u')
arbre.insert('x')
# dessin de l'arbre
graphic_arbre(arbre)
    
```

3. Parcours d'un arbre binaire de recherche

☞ Parmi les parcours préfixe, infixe et suffixe, lequel permet de récupérer la liste triée d'un arbre binaire de recherche ? Le vérifier en appliquant les fonctions ou les méthodes des parcours vu précédemment.

4. Valeur maximum et minimum

☞ Compléter les méthodes `ABRmax(self)` et `ABRmin(self)` qui renvoie respectivement la valeur maximum et minimum de l'arbre.

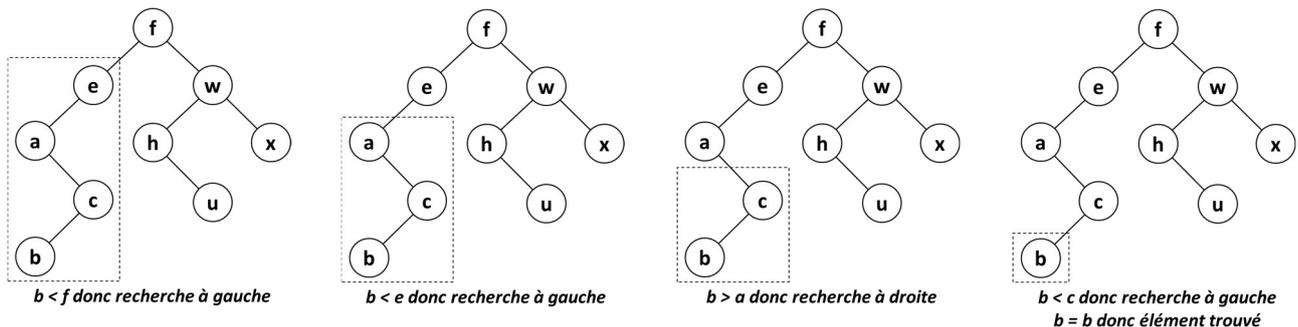
5. Recherche d'un élément dans un arbre binaire de recherche

Les arbres binaires de recherche permettent de rechercher un élément dans un arbre de n nœuds en temps $O(h)$ où h est la hauteur de l'arbre binaire de recherche.

- Si l'arbre est équilibré, on a $h \cong \log_2(n)$ et la recherche a une complexité en temps logarithmique.
- Au maximum, si l'arbre n'est pas équilibré on peut avoir alors une complexité au plus linéaire.

5.1. La méthode pour rechercher un élément

Exemple de la recherche (qui est récursive) de la valeur **b** un arbre binaire de recherche.



5.2. Algorithme de la recherche d'un élément

Recherche récursive

- Si l'élément recherché est égal à la racine alors on retourne vrai.
- Sinon si l'élément recherché est inférieur et que le fils gauche est non vide, on continue la recherche vers la gauche.
- Sinon si l'élément recherché est supérieur et que le fils droit est non vide, on continue la recherche vers la droite.
- Sinon on retourne faux.

5.3. Implémentation de la recherche d'un élément dans un arbre binaire de recherche

☞ Implémenter la méthode `chercher(self, k)` qui recherche un élément k dans un ABR.