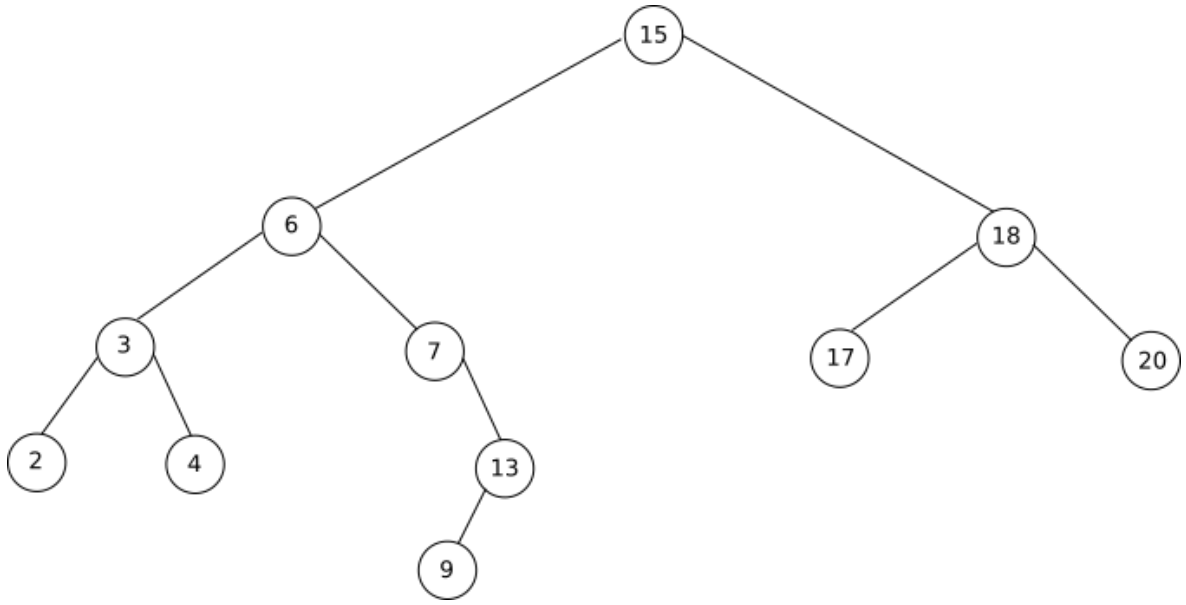
Algorithme sur les arbres binaires de recherche

1. Arbre binaire de recherche

7.Arbre binaire de recherche

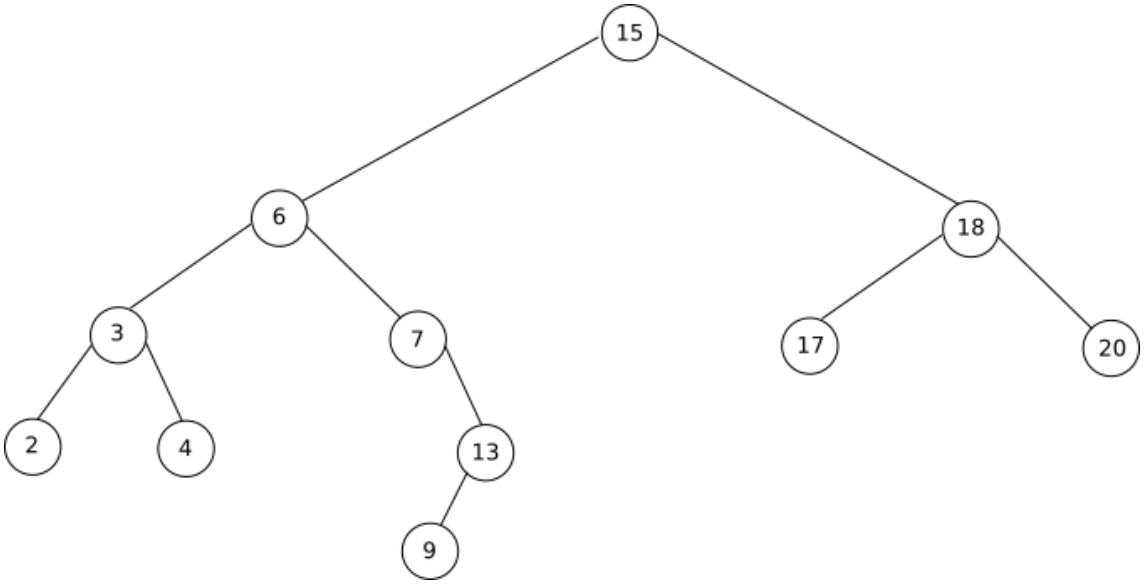
Un arbre binaire de recherche est un cas particulier d'arbre binaire. Pour avoir un arbre binaire de recherche :

- il faut avoir un arbre binaire !
- il faut que les clés des nœuds composant l'arbre soient ordonnables (on doit pouvoir classer les nœuds, par exemple, de la plus petite clé à la plus grande)
- soit x un nœud d'un arbre binaire de recherche. Si y est un nœud du sous-arbre gauche de x, alors il faut que y.clé $\leqslant$ x.clé. Si y est un nœud du sous-arbre droit de x, il faut alors que x.clé $\leqslant$ y.clé



Application-1 : Vérifiez que l'arbre ci-dessus est bien un arbre binaire de recherche.

Application-2 : Appliquez l'algorithme de parcours infixe sur l'arbre ci-dessous :



Dans le parcours infixe, le traitement de la racine est fait entre les appels sur les sous arbres gauche et droit.
Cela revient à lister chaque sommet ayant un fils gauche la seconde fois qu'on le voit et chaque sommet sans fils gauche la première fois qu'on le voit.

Que remarquez-vous ?

2. Recherche d'une clé dans un arbre binaire de recherche

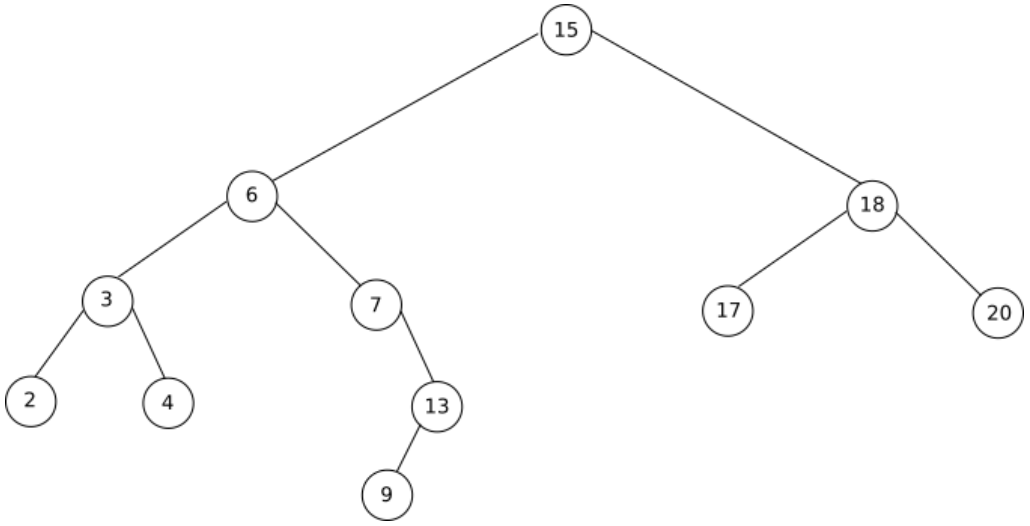
Recherche d'une clé dans un arbre binaire de recherche

Nous allons maintenant étudier un algorithme permettant de rechercher une clé de valeur k dans un arbre binaire de recherche. Si k est bien présent dans l'arbre binaire de recherche, l'algorithme renvoie vrai, dans le cas contraire, il renvoie faux.

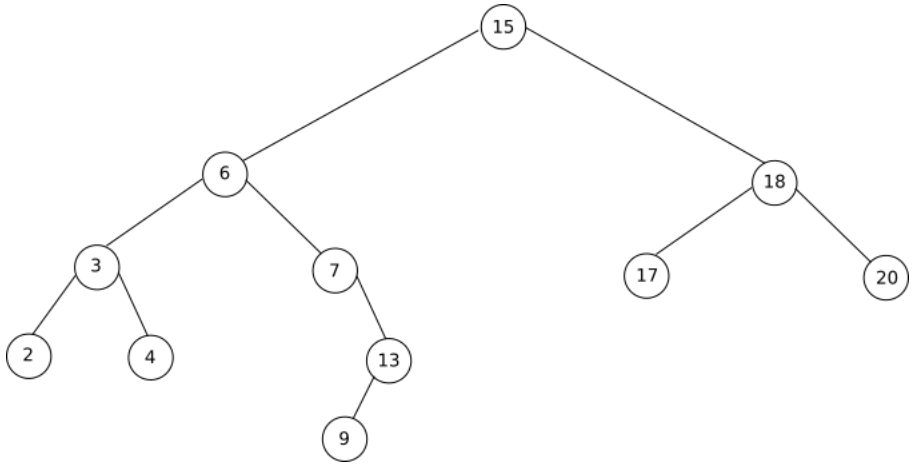
Étudiez l'algorithme suivant :

```
VARIABLE
T : arbre
x : nœud
k : entier
DEBUT
ARBRE-RECHERCHE(T,k) :
        Si T==NIL :
                Renvoyer Faux
        Fin Si
        x ⟵ T.racine
        Si k==x.cle :
                Renvoyer Vrai
        Fin Si
        Si k< x.cle :
                ARBRE-RECHERCHE(x.gauche,k)
        Sinon :
                ARBRE-RECHERCHE(x.droit,k)
        Fin si
FIN
```

Application-3 : Appliquez l'algorithme de recherche d'une clé dans un arbre binaire de recherche sur l'arbre ci-dessous. On prendra k = 13.



Application-4 : Appliquez l'algorithme de recherche d'une clé dans un arbre binaire de recherche sur l'arbre ci-dessous. On prendra k = 16.



Cet algorithme de recherche d'une clé dans un arbre binaire de recherche ressemble beaucoup à la recherche dichotomique vue en première dans le cas où l'arbre binaire de recherche traité est équilibré. À noter qu'il existe une version dite "itérative" (qui n'est pas récursive) de cet algorithme de recherche :

Étudiez l'algorithme suivant :

```
VARIABLE
T : arbre
x : nœud
k : entier
DEBUT
ARBRE-REHERCHE_ITE(T,k) :
        X ⟵ T.racine
        Tant que T ≠ NIL et k ≠ x.cle :
                X ⟵ T.racine
                Si k < x.cle :
                        T ⟵ x.gauche
                Sinon :
                        T ⟵ x.droit
                Fin si
        Fin tant que
        Si k == x.cle :
                Renvoyer Vrai
        Sinon :
                Renvoyer Faux
        Fin Si
FIN
```

3. Insertion d'une clé dans un arbre binaire de recherche

Il est tout à fait possible d'insérer un nœud « y » dans un arbre binaire de recherche (non vide) :

Étudiez l'algorithme suivant :

```
VARIABLE
T : arbre
x : nœud
y : noeud
DEBUT
ARBRE-INSERTION(T,y) :
        x ⟵ T.racine
        Tant que T ≠ NIL :
                x ⟵ T.racine
                Si y.cle < x.clé
                        T ⟵ x.gauche
                Sinon :
                        T ⟵ x.droit
                Fin si
        Fin tant que
        Si y.cle < x.clé :
                Inserer y à gauche de x
        Sinon :
                Inserer y à droite de x
        Fin Si
FIN
```

Application-5 : Appliquez l'algorithme d'insertion d'un nœud « y » dans un arbre binaire de recherche sur l'arbre ci-dessous. On prendra y.clé = 16.