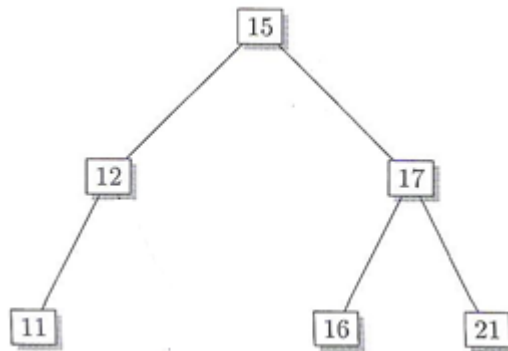


Les arbres binaires de recherche

1. Indiquer la hauteur de l'arbre binaire de recherche *arb0* suivant :

On prendra la convention suivante : la hauteur d'un arbre vide est 0.



2. Insérer la valeur 18 au bon endroit dans cet arbre binaire de recherche. Justifier rapidement.

3. Lister les valeurs de l'arbre *arb0* parcouru de manière postfixe.

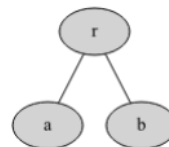
On désigne par SAG le sous-arbre gauche d'un nœud, et par SAD le sous-arbre droit de ce nœud.

En python, un arbre peut être représenté à l'aide d'une liste de la façon suivante :

[valeur du nœud racine, SAG, SAD].

Ainsi, l'arbre *arb1* ci-contre est créé grâce à l'instruction :

```
arb1 = ['r', ['a', [], []], ['b', [], []]]
```



4. Par quelle instruction peut-on accéder à la valeur de la racine de l'arbre *arb1* ? Au SAG de *arb1* ?

Soit la fonction récursive *insérer* fournie ci-dessous, permettant d'insérer la valeur *val* (transmise en paramètre) au bon endroit de l'arbre binaire de recherche *ar* (transmis en paramètre).

5. Compléter cette fonction.

```
def insérer(ar, val):
    if _____ :
        ar = [val, [], []]
    elif val < _____ :
        if ar[1] == []:
            ar[1] = _____
        else :
            insérer(ar[1], val)
    else :
        if _____ == []:
            _____ = [val, [], []]
        else :
            insérer(_____, val)
    return ar
```