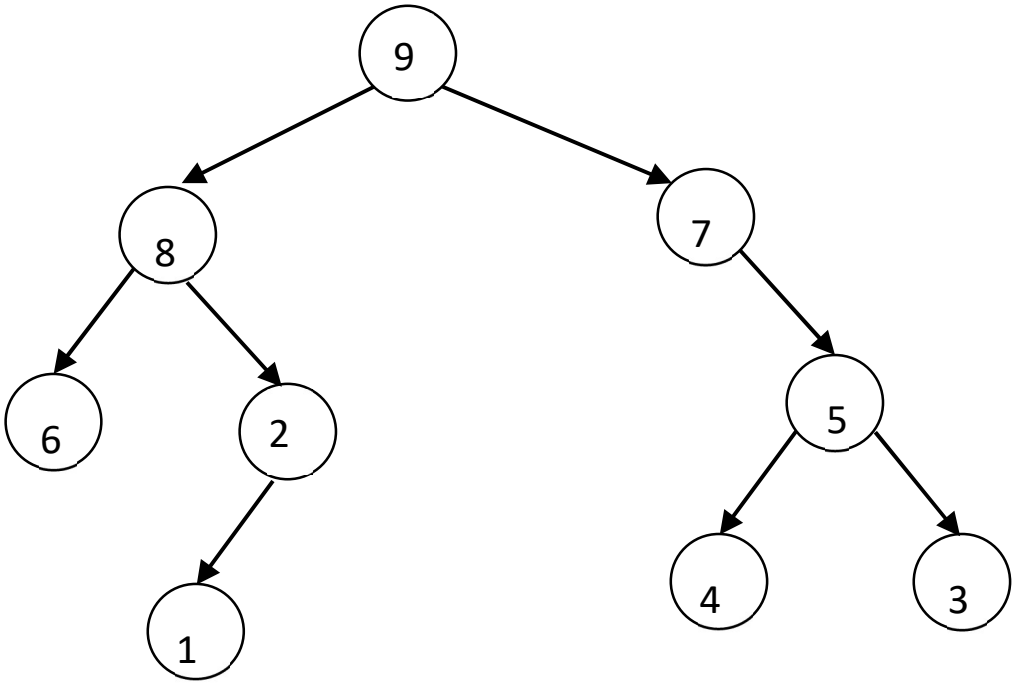
Exercice 1 : Soit l'arbre suivant :



1- Donner la hauteur de cet arbre.

2- Pour un arbre de cette hauteur, combien de nœuds peut-on avoir au maximum ?

3- Donner l'ordre des nœuds dans un parcours en ordre (infixe) : fils gauche, nœud, fils droit

4- Donner l'ordre des nœuds dans un parcours en pré-ordre (préfixe) : nœud, fils gauche, fils droit

5- Donner l'ordre des nœuds dans un parcours en post-ordre (postfixe) : fils gauche, fils droit, nœud

6- Donner l'ordre des nœuds dans un parcours largeur niveau après niveau (i.e. profondeur par profondeur)

Exercice 2 : Implémentation d'une classe arbre en python

1- Implémenter une classe nommée Arbre qui permet d'instancier des objets ayant les attributs d'instance suivants, avec les valeurs par défauts indiquées :

info = *None* ; **fg** = *None* ; **fd** = *None*

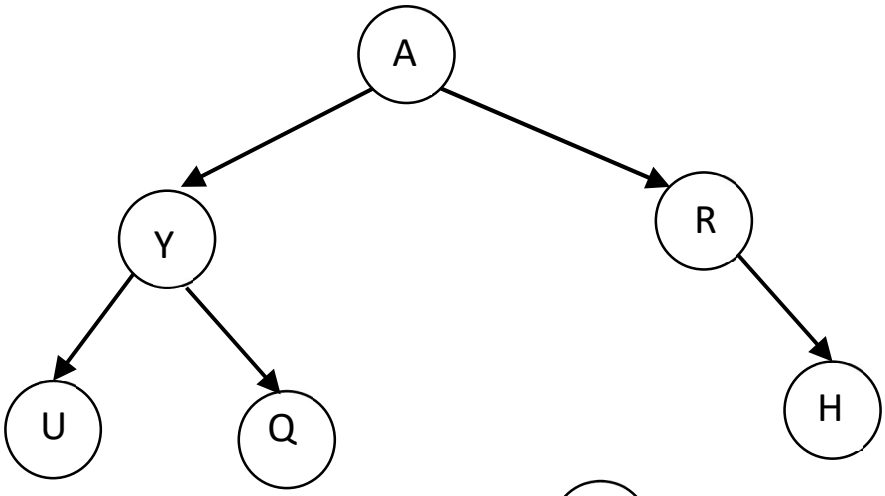
2- Implémenter une méthode nommée *insert_gauche()* ayant comme argument l'attribut *info* du nouvel objet *Arbre()* que l'on veut insérer sur l'emplacement fils_gauche de l'objet.

3- Implémenter une méthode nommée *insert_droite()* ayant comme argument l'attribut *info* du nouvel objet *Arbre()* que l'on veut insérer sur l'emplacement fils_droit de l'objet.

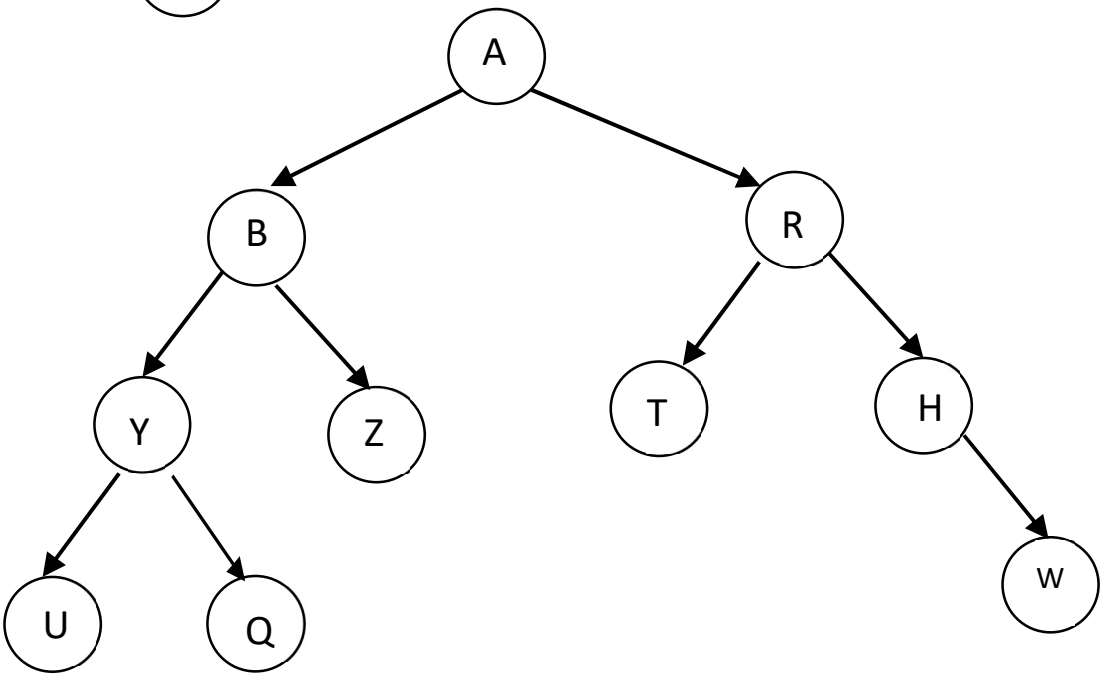
4- Dans le programme principal, créer un objet *Arbre* nommé A et ayant comme attribut info la chaine de caractère 'A' :

```
# Main
A = Arbre("A")
```

Utiliser les méthodes *insert_gauche()* et *insert_droite()* pour créer l'arbre ci-dessous :



5- Utiliser à nouveau les méthodes *insert_gauche()* et *insert_droite()* pour modifier l'arbre précédent et obtenir :



6- Quel est la hauteur de l'arbre précédent ? Implémenter une méthode nommée *hauteur()* qui renvoie la hauteur de l'arbre.

7- Donner l'ordre des nœuds dans un parcours en ordre (infixe) : fils gauche, nœud, fils droit.
Implémenter une méthode nommée *parcours_infixe()* qui renvoie la liste des nœuds de l'arbre analysé en utilisant ce type de parcours.

8- Donner l'ordre des nœuds dans un parcours en pré-ordre (préfixe) : nœud, fils gauche, fils droit
Implémenter une méthode nommée *parcours_preordre()* qui renvoie la liste des nœuds de l'arbre analysé en utilisant ce type de parcours.

9- Donner l'ordre des nœuds dans un parcours en post-ordre (postfixe) : fils gauche, fils droit, nœud.
Implémenter une méthode nommée *parcours_postfixe()* qui renvoie la liste des nœuds de l'arbre analysé en utilisant ce type de parcours.