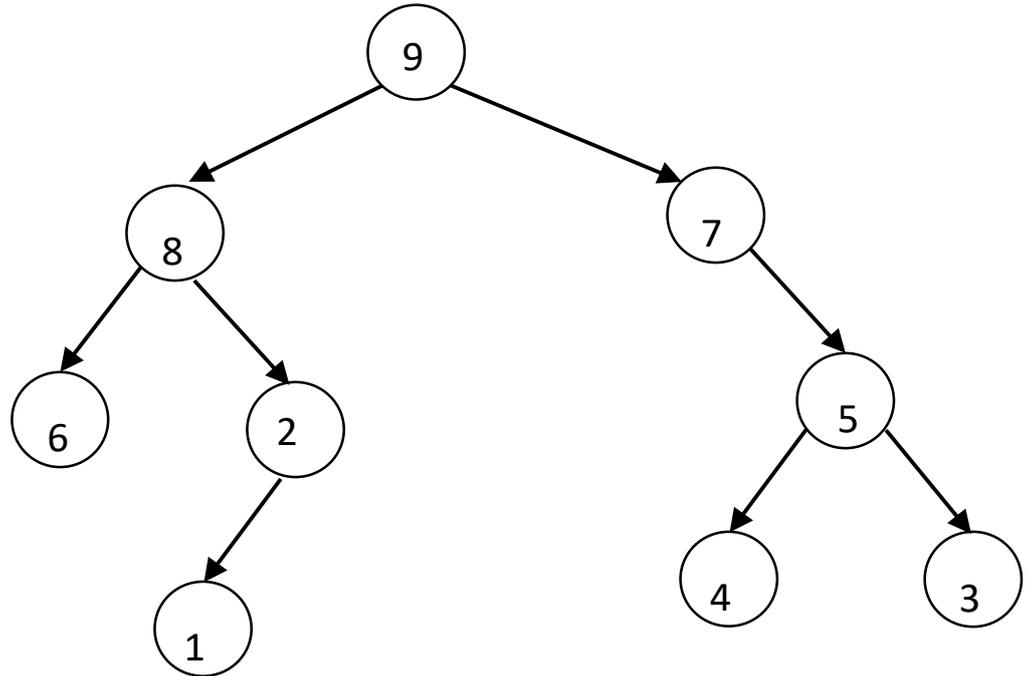


Exercice 1 : Soit l'arbre suivant :



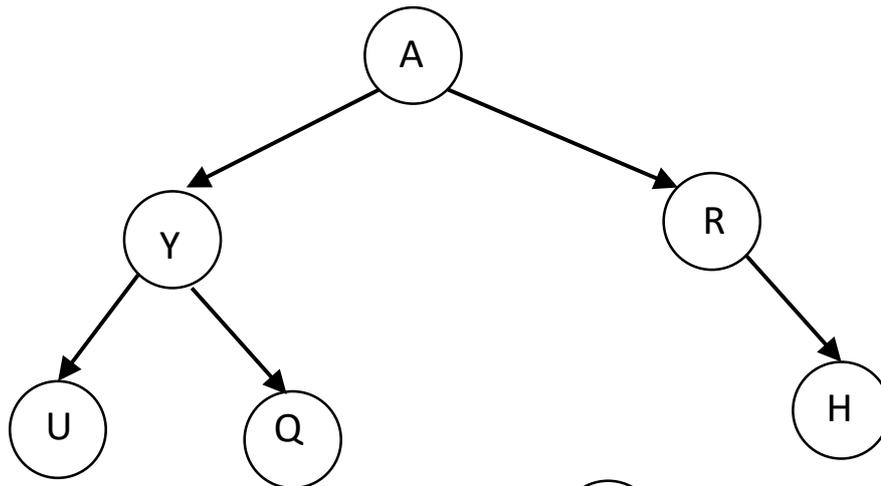
- 1- Donner la hauteur de cet arbre.
- 2- Pour un arbre de cette hauteur, combien de nœuds peut-on avoir au maximum ?
- 3- Donner l'ordre des nœuds dans un parcours en **ordre (infixe)** : fils gauche, nœud, fils droit
- 4- Donner l'ordre des nœuds dans un parcours en **pré-ordre (préfixe)** : nœud, 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

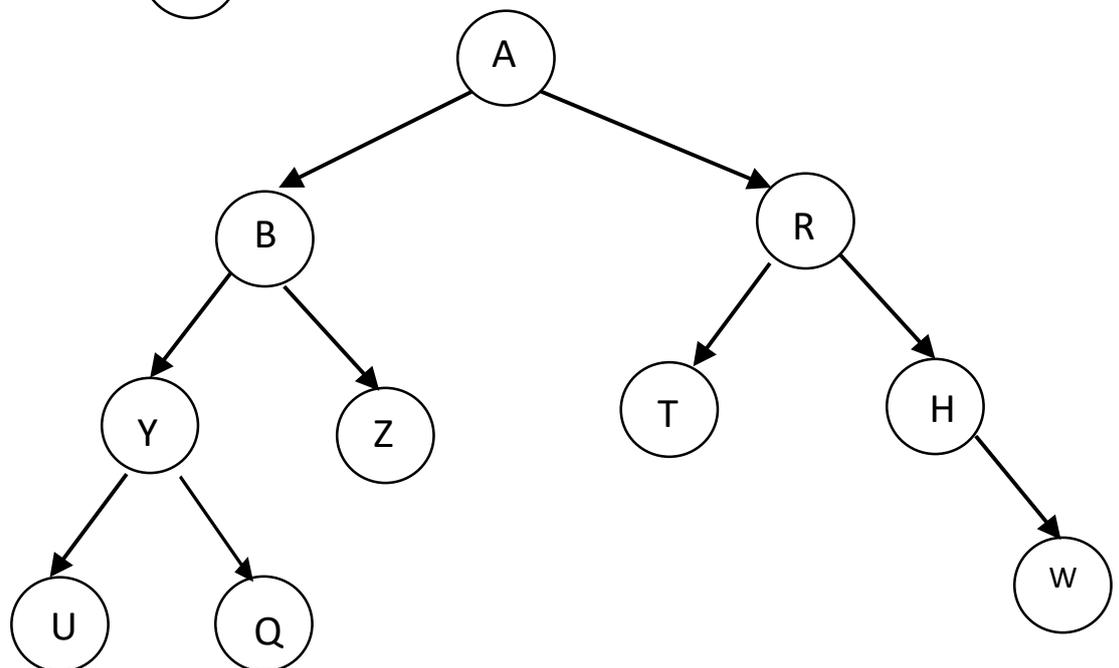
⇒ Télécharger le fichier *tdArbreBinaires.zip* qui contient les fichiers *tdArbres.py* et *trace_arbre.py* à copier dans votre répertoire de travail.

Le fichier *trace_arbre.py* n'est pas à modifier. Il contient les classes *Pile*, *File* et *Graph* qui seront utilisées dans le fichier *tdArbres.py*.

- 1- 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.
- 2- Utiliser les méthodes *insert_gauche()* et *insert_droite()* pour créer l'arbre ci-dessous :



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



- 4- Donner l'ordre des nœuds dans un parcours en **ordre (infixe)** : fils gauche, nœud, fils droit.
- 5- Analyser la méthode fonctionA(). Que fait-elle ?
- 6- 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.
- 7- 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.
- 8- Analyser la méthode fonctionB(). Que fait-elle ?

-
- 9- Analyser la méthode fonctionC() . Que fait-elle ?
 - 10- Quel est la hauteur de l'arbre précédent ? La méthode nommée *hauteur()* renvoie-t-elle la hauteur de l'arbre ?
 - 11- Quel est le nombre de feuilles de l'arbre précédent ? Implémenter une méthode nommée *nb_feuilles()* qui renvoie ce nombre.
 - 12- A l'aide du parcours qu'il vous plait, implémenter une méthode *trouve()* qui renvoie Vrai ou Faux en fonction de la présence d'une valeur ou pas dans l'arbre.