

Objectifs : Dessiner et compléter un arbre binaire grâce à la bibliothèque **graphviz**. Cette bibliothèque dessine en fait des graphes. Donc on va uniquement l'utiliser sans entrer dans le détail de son fonctionnement.

1. Installation de graphviz

Avec internet télécharger le programme d'installation :

<https://graphviz.org/download/>

- [graphviz-7.0.6 \(64-bit\) EXE installer \[sha256\]](#)

Dans le shell : `$ pip install graphviz`

2. Implémentation de la classe Arbre

Afin d'utiliser **graphiv** quatre ajouts devront être mis dans le programme :

- L'appel à la bibliothèque :

```
1 from graphviz import Digraph
2
3 graphe = Digraph(strict=True)
4
```

- Un nœud est nécessaire pour le traçage dans la classe arbre (nb_arbre permet de compter le nombre d'objets Arbre instanciés)

```
class Arbre :
    nb_arbre = 0
    def __init__(self,                 ) :
                        
        self.__class__.nb_arbre += 1
        self.num = str(self.__class__.nb_arbre)
        graphe.node(self.num , self.info) # trace le noeud
```

- La méthode qui trace le Graphe

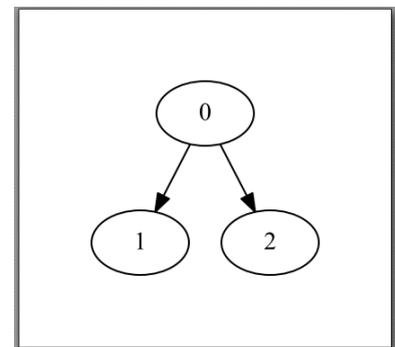
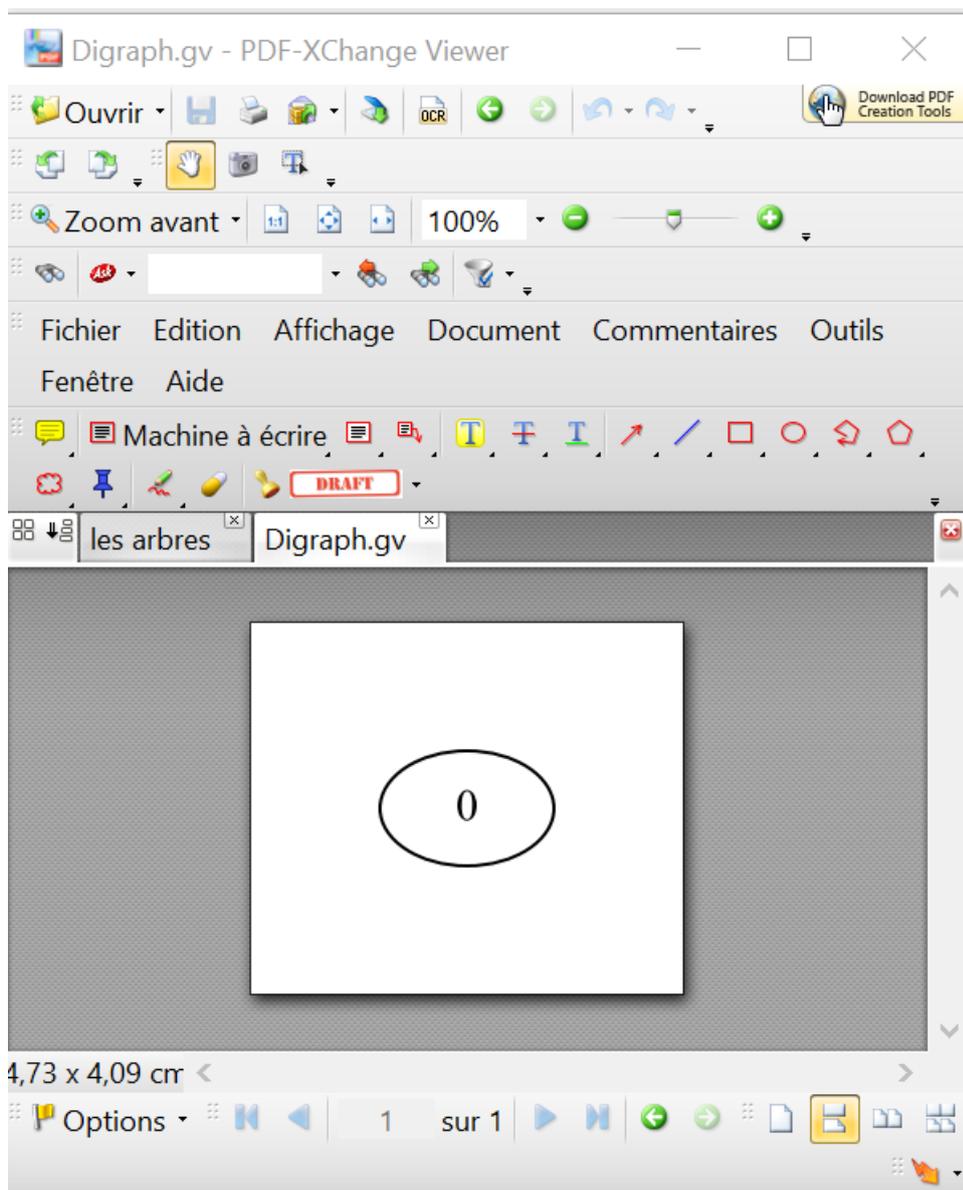
```
def trace_graphique(self) :
    """
    trace les ponts de l'arbre sous le self
    """
    l = [self]
    while l != [] :
        nd = l.pop(0)
        if nd.fg != None :
            graphe.edge(nd.num,nd.fg.num)
            l.append(nd.fg)
        if nd.fd != None :
            graphe.edge(nd.num,nd.fd.num)
            l.append(nd.fd)
```

Enfin l'appel de la méthode et le traçage de l'Arbre A en fin de programme

La valeur du nœud de l'arbre doit être une chaîne de caractères d'où le guillemet.

```
A = Arbre('0')  
  
A.trace_graphique() # permet de tracer les ponts  
graphe.view()
```

Le programme crée alors un fichier pdf qui s'affiche :



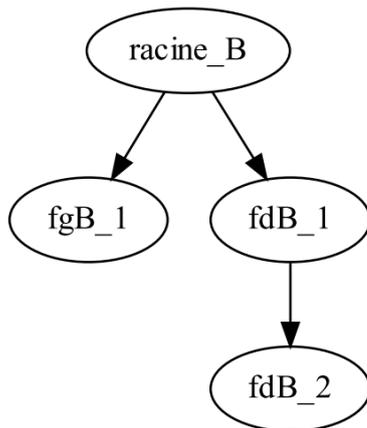
Le nom du fichier Digraph.pdf est toujours le même. Il faut le fermer pour le reconstruire

Question 1

Implémenter l'Objet Arbre du cours ainsi que ses méthodes et fonctions.

Question 2

A la main en ajoutant fils gauche et fils droit construisez l'arbre suivant:



Question 3 tester les différents parcours

Question 4

Ecrire les méthode **insérer_droite_arbre** et **insérer_gauche_arbre** pour permettre d'insérer un arbre entier en fils gauche ou droite et pas seulement une valeur.

Question 5

En vous inspirant de la fonction hauteur qui donne la profondeur de l'arbre, donner le code d'une fonction taille qui donne de manière récursive le nombre de nœuds d'un arbre.