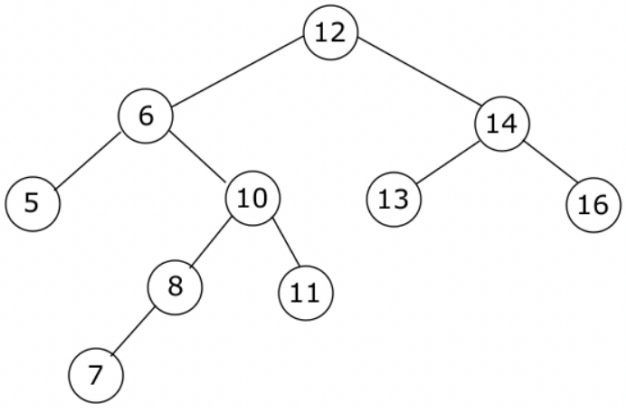
SUJET ZÉRO A: CORRECTION

1. Il s'agit d'une structure FIFO, c'est-à-dire une file
2.
   a. il s'agit de la taille d'un arbre
   b. il s'agit de la racine de l'arbre
   c. il s'agit de la feuille d'un arbre
3.
   a. attributs de la classe Noeud : tache, indice, gauche et droite
   b. La méthode *insere* est dite récursive, car elle s'appelle elle-même. Dans cette méthode récursive, on trouve bien le traitement du cas de base, ce qui permet d'affirmer que cette méthode se termine.
   c. il s'agit du signe > (strictement supérieur)
   d.



4.

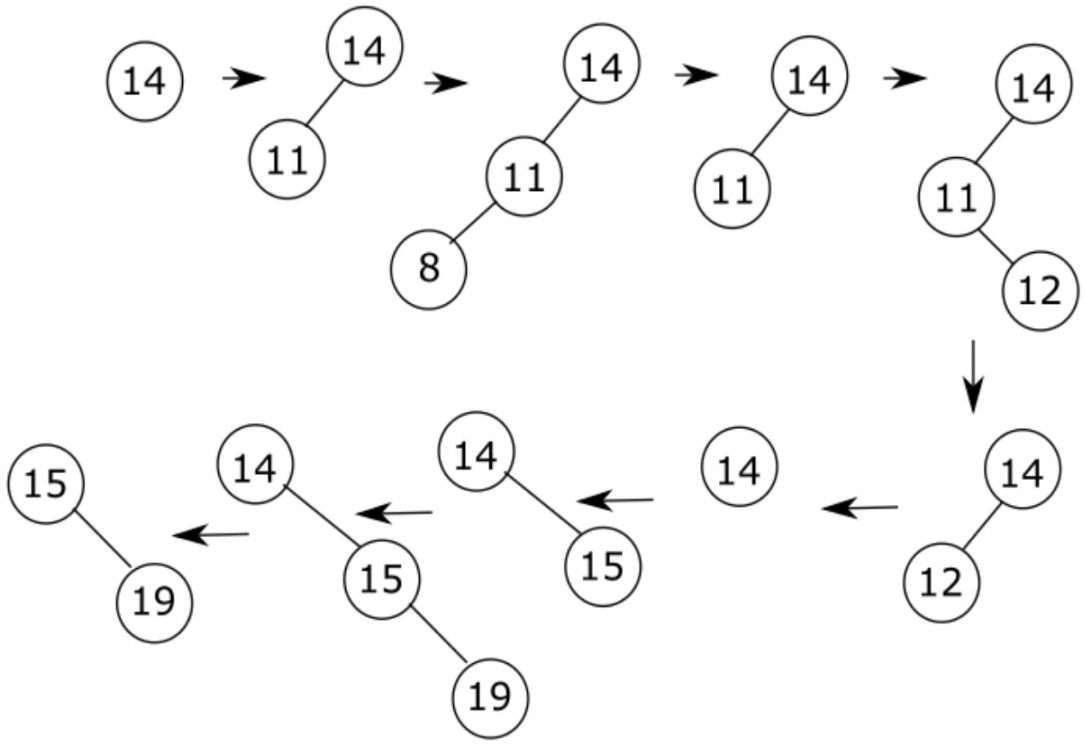
```
def est_present(self, indice_recherche) :
      """renvoie True si l'indice de priorité indice_recherche
      (int) passé en paramètre est déjà l'indice d'un nœud
      de l'arbre, False sinon"""
      if self.est_vide():
          return False
      if self.racine.indice == indice_recherche:
          return True
      if self.racine.indice > indice_recherche :
          return self.racine.gauche.est_present(indice_recherche)
      else :
          return self.racine.droite.est_present(indice_recherche)
```

5.
   a.
   parcours infixe : 6 - 8 - 10 - 12 - 13 - 14
   b.
   Le parcours infixe permet d'obtenir les valeurs des nœuds d'un arbre binaire de recherche dans un ordre croissant. Le parcours infixe va donc permettre d'obtenir les tâches à accomplir dans l'ordre des priorités

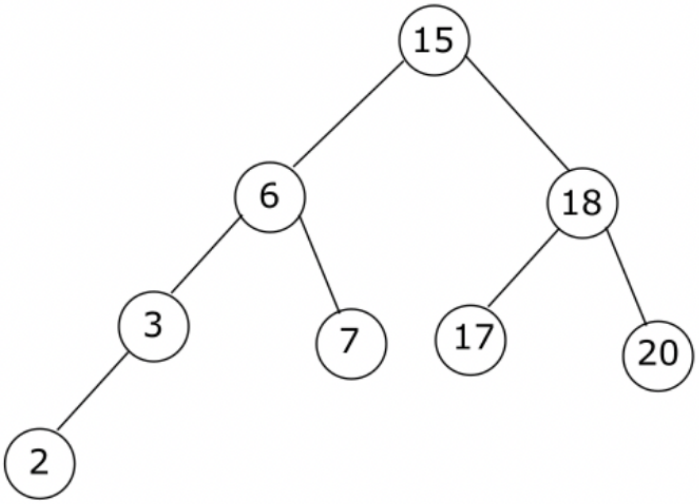
6.

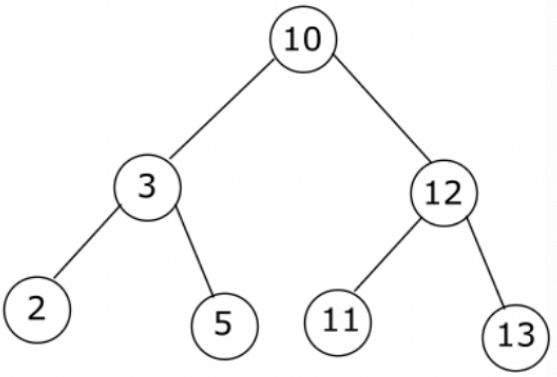
```
def tache_prioritaire(self):
        """renvoie la tache du noeud situé le plus
        à gauche de l'ABR supposé non vide"""
        if self.racine.gauche.est_vide(): #pas de nœud plus à
gauche
            return self.racine.tache
        else:
            return self.racine.gauche.tache_prioritaire()
```

7.



SUJET ZÉRO B: CORRECTION

1.
   a.
   Exemple d'attribut : `enfant_gauche`
   Exemple de méthode : insert_gauche()
   b.
   Nous avons a = 15 et c = 6
2.


3.
Nous avons 11 qui est à droite 12


4.
On obtient le tableau (liste Python) suivant : [1, 6, 10, 15, 16, 18, 25]