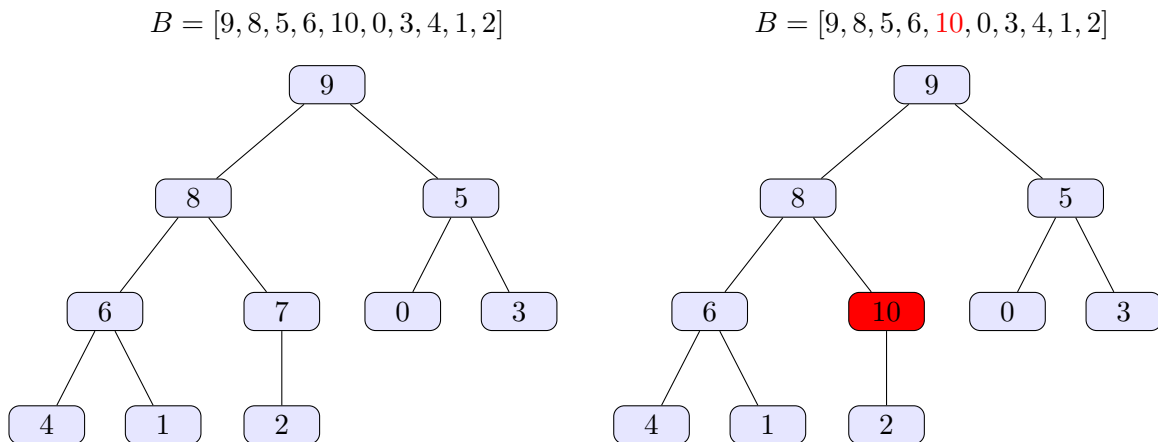


Structure de tas

Rappels de cours : On considère un tableau numérique A où les indices sont numérotés de 0 à $\text{taille}(A) - 1$, comme en Python. Le tableau A définit une structure d'arbre de racine $A[0]$ par la fonction *parent* $p(i) = \lfloor (i-1)/2 \rfloor$. Cet arbre est *binaire* car chaque élément a au plus un enfant *gauche* $g(i) = 2i + 1$ et un enfant *droit* $d(i) = 2i + 2$.

Exercice 1 Vérification d'un tas

Question 1.1 Définissez une fonction $\text{is_heap}(A, i)$ qui à partir d'un tableau A représentant un arbre binaire et un noeud courant i renvoie **True** si A représente un tas et **False** sinon.

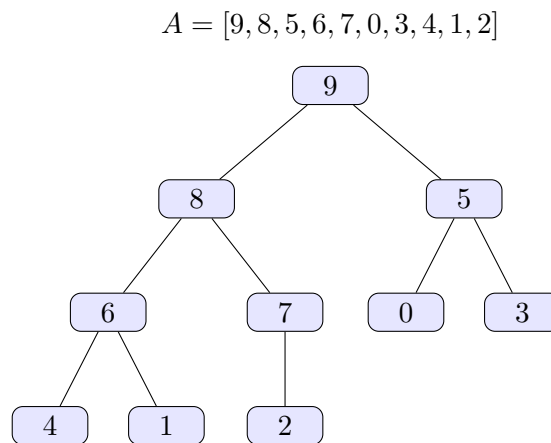


A titre d'exemple l'arbre binaire représenté par A est un tas et l'arbre binaire représenté par B n'est pas un tas car $10 > 8$.

Exercice 2 Affichage d'un tas

Question 2.1 Définissez une fonction $height(A, i)$ qui à partir d'un tableau A représentant un tas et un noeud courant i renvoie la hauteur du tas A c'est-à-dire la hauteur de la racine de A .

Question 2.2 Définissez une fonction $nb_depth(A, p)$ qui à partir d'un tableau A représentant un tas et d'un entier $0 \leq p \leq height(A)$ renvoie le nombre de noeuds du tas A à la profondeur p .



A titre d'exemple pour le tas ci-dessus :

- $height(A) = 3$
- $nb_depth(A, 0) = 1$
- $nb_depth(A, 1) = 2$
- $nb_depth(A, 2) = 4$
- $nb_depth(A, 3) = 3$

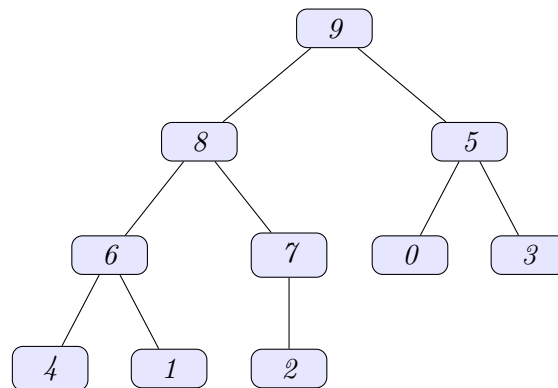
Question 2.3 Définissez une fonction $display_heap(A)$ qui affiche le tas A de la manière suivante (avec l'exemple ci-dessus) :

```
9
8 5
6 7 0 3
4 1 2
```

Indice : Pour un niveau donné l d'un tas A , on doit afficher au pire $nb_level(A, l)$ noeuds.

Question 2.4 Définissez une fonction $nb_height(A, h)$ qui à partir d'un tableau A représentant un tas et d'un entier $0 \leq h \leq height(A)$ renvoie le nombre de noeuds du tas A à la hauteur h .

$A = [9, 8, 5, 6, 7, 0, 3, 4, 1, 2]$



A titre d'exemple pour le tas ci-dessus :

- $height(A) = 3$
- $nb_height(A, 0) = 5$
- $nb_height(A, 1) = 3$
- $nb_height(A, 2) = 1$
- $nb_height(A, 3) = 1$