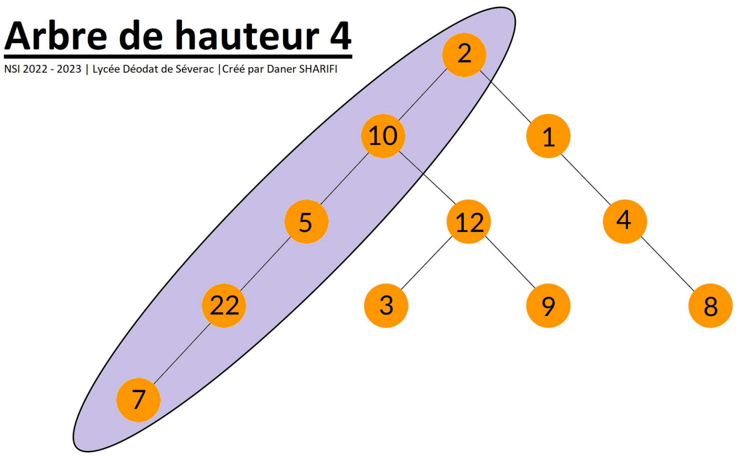
CORRECTION – NSI - 2022 Sujet Polynésie

Exercice 5 : Algorithmique – Les Arbres Binaires

1. a. D'après la définition, la hauteur de cet arbre est 2.

1. b.

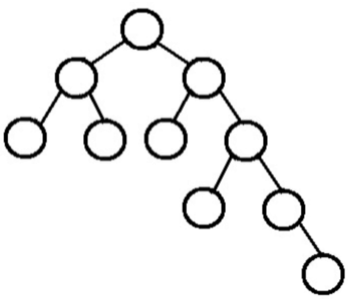


2.

```
Algorithme hauteur(A) :
    test d'assertion : A est supposé non vide
    si sous_arbre_gauche(A) vide et sous_arbre_droit(A) vide :
        renvoyer 0
    sinon si sous_arbre_gauche(A) vide :
        renvoyer 1 + hauteur(sous_arbre_droit(A))
    sinon si sous_arbre_droit(A) vide :
        renvoyer 1 + hauteur(sous_arbre_gauche(A))
    sinon :
        renvoyer 1 + max(hauteur(sous_arbre_gauche(A)),
                                    hauteur(sous_arbre_droit(A)))
```

3. a. On sait que : hauteur(R) = max(hauteur(G), hauteur(D)) = 4
et d'après l'énoncé : hauteur(G) = 2
soit hauteur(R) = max(2, hauteur(D)) = 4
donc il faut que hauteur(D) = 4.

3. b.

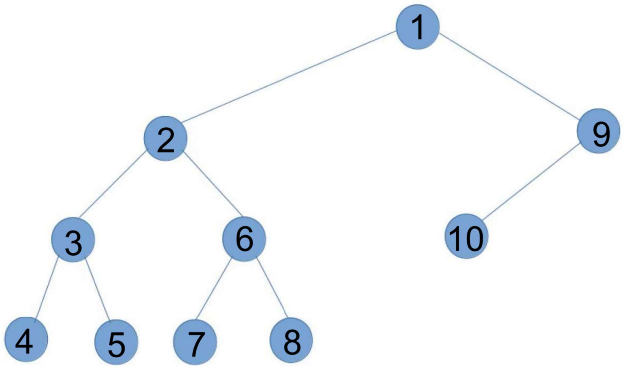


4. a. Pour l'arbre binaire de la question 1. a. de hauteur 2 avec 4 nœuds, on obtient :
$h+1 \leqslant n \leqslant 2^{(h+1)}-1$ soit $3 \leqslant 4 \leqslant 7$
Donc l'inégalité est bien vérifiée.

4. b. Il suffit de construire un arbre binaire où chaque nœud n'a qu'un seul fils. Il s'agit d'une liste.

4. c. Dans ce cas, il faut construire un arbre binaire complet. Tous les nœuds ont deux fils et les feuilles sont toutes à la même hauteur.

5.



6.

```
def fabrique(n, h):
    def annexe(hauteur_max):
        if n == 0:
            return arbre_vide()
        elif hauteur_max == 0:
            n = n - 1
            return arbre(arbre_vide(), arbre_vide())
        else:
            n = n - 1
            gauche = annexe(hauteur_max - 1)
            droite = annexe(hauteur_max - 1)
            return arbre(gauche, droite)
    return annexe(h)
```