

NSI – Terminale – Sujet « zéro » -

Exercice 3

Question 1 – Déterminer la taille et la hauteur de l'arbre binaire

La taille d'un arbre est le nombre de nœuds.

Ici elle vaut 9.

La hauteur de l'arbre est la longueur du chemin le plus long entre la racine et l'une des feuilles.

Ici 4.

Question 2 -

2.1 - Dans l'exemple précédent, quel est le numéro en binaire associé au nœud G?

Le numéro associé à G est 1010.

2.2 - Quel est le nœud dont le numéro en binaire vaut 13 en décimal ?

Le nombre 13 s'écrit 1101 en binaire. Il correspond donc au nœud I.

2.3. En notant h la hauteur de l'arbre, sur combien de bits seront numérotés les nœuds les plus en bas ?

Les nœuds les plus bas sont numérotés sur h bits (4 dans l'exemple).

2.4. Justifier que pour tout arbre de hauteur h et de taille $n \geq 2$, on a : $h \leq n \leq 2^h - 1$

Un arbre de hauteur h peut avoir au minimum n nœuds (un nœud par niveau). Donc $h \leq n$.

A l'autre extrême, si l'arbre est complet (tous ses niveaux sont remplis) alors la racine a pour code 1, les nœuds du premier niveau ont pour code 10 et 11, ceux du deuxième niveau 100, 101, 110 et 111, etc. . .

Les codes des nœuds du niveau h s'écrivent sur h bits.

L'arbre étant complet, tous les nombres entiers pouvant s'écrire sur h bits correspondent à des codes sauf la valeur 0 car la racine est le 1.

Il y a 2^h valeurs possibles sur h bits. Donc en retirant le 0 on obtient $2^h - 1$.

On a donc bien : $h \leq n \leq 2^h - 1$

Question 3

3.1. Déterminer le tableau qui représente l'arbre binaire complet de l'exemple précédent.

Le tableau est : [15,"A","B","C","D","E","F","G","H","I","J","K","L","M","N","O"]

3.2. On considère le père du nœud d'indice i avec $i \geq 2$. Quel est son indice dans le tableau ?

L'indice du père d'un nœud d'indice $i \geq 2$ est le quotient entier de i par 2 ($i//2$ en python).

Question 4

Écrire une fonction recherche ayant pour paramètres un arbre arbre et un élément element. Cette fonction renvoie True si element est dans l'arbre et False sinon. L'arbre sera représenté par un tableau comme dans la question précédente.

```
def recherche(arbre, element):
    taille = len(arbre)
    i = 1
    while i < taille:
        if arbre[i] == element:
            return True
        elif element < arbre[i]:
            i = 2*i
        else:
            i = 2*i+1
    return False
```