

DEVOIR MAISON NSI

Célian BUTRÉ

Janvier 2021

Exercice 3

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

La hauteur d'un nœud correspond au maximum des hauteurs de ses deux fils, plus lui-même. La taille d'un nœud correspond à la somme des tailles de ses deux fils plus lui-même.

Les nœuds C, G, H, I n'ont pas de fils, ils ont donc une taille de 1. Le nœud D a pour unique fils G, qui a une taille de 1, il a donc une taille de 2 ; le nœud F a pour fils H et I, tous deux ayant une taille de 1, il a donc une taille de 3. Le nœud B a pour fils C et D, l'un ayant une taille de 1, l'autre de 2, il a donc une taille de 4 ; le nœud E a pour unique fils F, qui a une taille de 3, il a donc une taille de 4. Finalement, le nœud A a pour fils B et E, tous deux ayant une taille de 4, il a donc une taille de 9. **La taille de l'arbre est de 9.**

Les nœuds C, G, H, I n'ont pas de fils, ils ont donc une hauteur de 1. Le nœud D a pour unique fils G, qui a une hauteur de 1, il a donc une hauteur de 2 ; le nœud F a pour fils H et I, tous deux ayant une hauteur de 1, il a donc une hauteur de 2. Le nœud B a pour fils C et D, l'un ayant une hauteur de 1, l'autre de 2, il a donc une hauteur de 3 ; le nœud E a pour unique fils F, qui a une hauteur de 2, il a donc une hauteur de 3. Finalement, le nœud A a pour fils B et E, tous deux ayant une hauteur de 3, il a donc une hauteur de 4. **La hauteur de l'arbre est de 4**

Question 2 : Étude d'un arbre numéroté en binaire

1. Numéros associés aux nœuds (*en particulier G*)

Point	D	G	H	I
Valeur Binaire	101	1010	1100	1101

2. Nœud numéroté 13 en binaire

On suppose que l'on représente 13 en binaire non signé

$$13 = 2^3 + 2^2 + 2^0 = 1101_{bin}$$

C'est donc le nœud I dont le numéro en binaire vaut 13

3. Bits pour représenter un arbre de hauteur h

Les nœuds les plus en bas seront numérotés sur h bits car chaque hauteur en plus rajoute 1 bit, et qu'un arbre de hauteur 1 est représenté sur 1 bit.

4. Justification de la propriété de cet arbre

$h \leq n$ signifie que la hauteur ne dépasse jamais la taille, or la hauteur correspond à la succession de fils la plus longue, cette suite comporte obligatoirement autant de nœuds que la longueur de la succession. h ne peut pas être plus grande que n d'où $h \leq n$.

On procède à une pseudo-réurrence pour prouver $n \leq 2^h - 1$, en partant du fait qu'un arbre de hauteur h , et de taille maximale est un arbre complet de hauteur h , c'est à dire un arbre qui a récursivement deux fils par nœud. Soit n_{maxh} la taille maximale d'un arbre de hauteur h , pour toute taille d'arbre n de hauteur h on sait que $n_{maxh} \geq n$. On procède donc par prouver que $n_{maxh} = 2^h - 1$ ce qui prouvera que $n \leq 2^h - 1$.

Initialisation : pour un arbre de hauteur 2, un arbre complet comporte 3 sommets, donc $3 = 2^2 - 1$, donc $n_{maxh} = 2^h - 1$ pour le cas de base de $h = 2$.

Hérédité : un arbre complet de hauteur $h+1$ comporte un sommet, et 2 arbres complets de hauteur h . On part de l'hypothèse de récurrence $n_{maxh} = 2^h - 1$, dans le but de prouver $n_{maxh+1} = 2^{h+1} - 1$.

$$\begin{aligned}n_{maxh} &= 2^h - 1 \\2 * n_{maxh} &= 2 * (2^h - 1) \\2 * n_{maxh} + 1 &= 2 * (2^h - 1) + 1 \\n_{maxh+1} &= 2^{h+1} - 1\end{aligned}$$

La propriété ($n_{maxh} = 2^h - 1$ et donc $n \leq 2^h - 1$) est vraie au rang 2 (pour $h = 2$) et héréditaire à partir de celui-ci, d'après le principe de récurrence, la propriété est vraie pour tout $h \geq 2$.

$$h \leq n \leq 2^h - 1$$

Question 3 : Étude d'un arbre binaire complet

1. Tableau représentant un arbre binaire complet

15	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

2. Calcul d'un indice dans le tableau

Le père du nœud d'indice i a pour indice dans le tableau $i//2$, c'est à dire la partie entière de la moitié de l'indice de son fils.

Question 4 : Recherche dans un arbre binaire de recherche

On suppose qu'on possède au préalable une classe arbre avec les attributs `racine`, `fils_droit` et `fils_gauche`. Et que cette classe choisit de représenter ses nœuds sans racines comme ayant 2 fils `None`.

```
def recherche(arbre, élément):
    if arbre == None:
        return(False)
    if arbre.racine == élément:
        return(True)
    if arbre.racine > élément:
        return(recherche(arbre.fils_gauche, élément))
    return(recherche(arbre.fils_droit, élément))
```