

Algorithmique

Correction Partiel n° 2 (P2)

INFO-SUP S2# – EPITA

8 janvier 2018 - 11 : 00

Solution 1 (Combien? – 3,5 points)

Spécifications :

La fonction `nb_inter(B, a, b)` calcule le nombre de valeurs de l'arbre binaire de recherche B dans l'intervalle $[a, b[$.

```
1 def nb_inter(B, a, b):
2     if B == None:
3         return 0
4     elif B.key < a:
5         return nb_inter(B.right, a, b)
6     elif B.key > b:
7         return nb_inter(B.left, a, b)
8     else:
9         return 1 + nb_inter(B.left, a, b) + nb_inter(B.right, a, b)
```

Solution 2 (ABR → AVL – 4,5 points)

Spécifications :

La fonction `makeAVL(B)` construit une copie de l'arbre binaire B avec les déséquilibres renseignés en chaque nœud.

```
1 def __makeAVL(B):
2     if B == None:
3         return(None, -1)
4
5     else:
6         A = AVL(B.key)
7
8         (A.left, hl) = __makeAVL(B.left)
9         (A.right, hr) = __makeAVL(B.right)
10
11        A.bal = hl - hr
12        return (A, 1 + max(hl, hr))
13
14
15 def makeAVL(B):
16     (A, h) = __makeAVL(B)
17     return A
```

Solution 3 (AVL - Ajout 0 – 5 points)

Spécifications :

La fonction `add0(A)` ajoute la valeur 0 dans l'AVL A (ne contenant que des entiers naturels non nuls) et retourne un couple : l'arbre résultat et un booléen indiquant si la hauteur de A a changé.

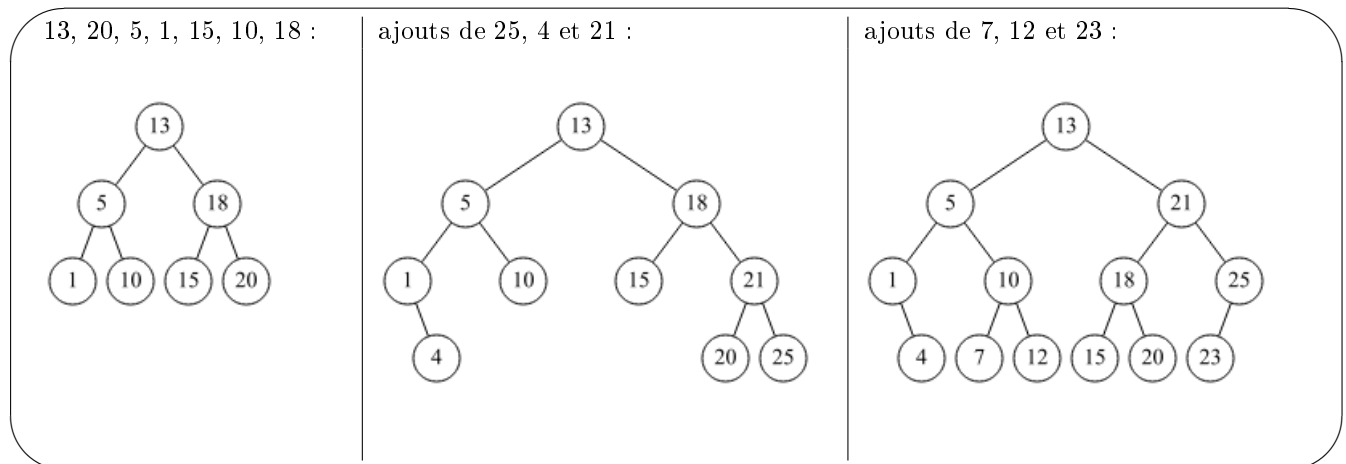
```

1 def __add0_AVL(A):
2     if A == None:
3         return (avl.AVL(0, None, None, 0), True)
4     else:
5         (A.left, dh) = __add0_AVL(A.left)
6         if not dh:
7             return (A, False)
8         else:
9             A.bal += 1
10            if A.bal == 0:
11                return (A, False)
12            elif A.bal == 1:
13                return (A, True)
14            else: # A.bal == 2
15                A = rr(A)
16                return (A, False)
17
18 def add0_AVL(A):
19     (A, _) = __add0_AVL(A)
20     return A

```

Solution 4 (AVL – 3 points)

AVL résultat depuis la liste [13, 20, 5, 1, 15, 10, 18, 25, 4, 21, 7, 12, 23].



Solution 5 (Arbre 2.3.4 → Arbre bicolore – 2 points)

1. Transformation de l'arbre 2.3.4 du sujet :

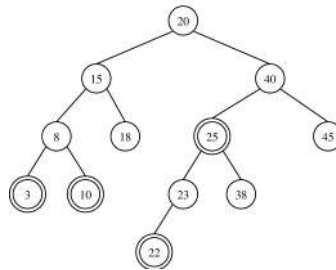
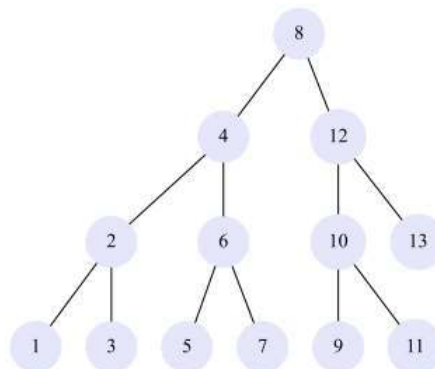


FIGURE 1 – Arbre bicolore

2. L'arbre obtenu n'est pas un AVL.
Le nœud contenant 40 a un déséquilibre de 2 (l'arbre n'est pas h-équilibré).
-

Solution 6 (Arbres et mystère – 3 points)

1. Arbre construit par `makeTree(13)` :



2. Propriétés de l'arbre construit par `makeTree(n)` ($n > 0$) :

- (a) Arbre parfait
 - (b) Arbre binaire de recherche
-