

Algorithmics

Correction Final Exam #2 (P2)

UNDERGRADUATE 1st YEAR S2# – EPITA

8 May 2019 - 11 : 00

Solution 1 (How many? – 3,5 points)

Specifications:

The function `nb_inter(B, a, b)` computes the number of values of the binary search tree B in the interval $[a, b[$

```
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 (BST → AVL – 4,5 points)

Specifications:

The function `makeAVL(B)` builds a copy of the binary tree B with the balance factors specified in each node.

```
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 - Add 0 – 5 points)**Specifications:**

The function `add0(A)` inserts the value 0 in the AVL A (that contains only non-zero naturals). It returns a pair: the tree after insertion and a boolean that indicates whether the tree height has changed.

```

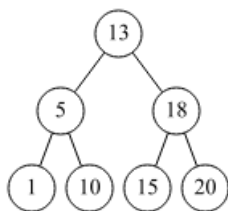
1 def __add0_AVL(A):
2     if A == None:
3         return (avl.AVL(0, None, None, 0), True)
4     else:
5         (A.left, dh) = __add0_AVL(A.left)
6         if not dh:
7             return (A, False)
8         else:
9             A.bal += 1
10            if A.bal == 0:
11                return (A, False)
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

```

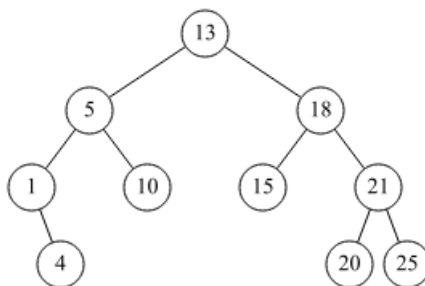
Solution 4 (AVL – 3 points)

Final AVL from the list [13, 20, 5, 1, 15, 10, 18, 25, 4, 21, 7, 12, 23].

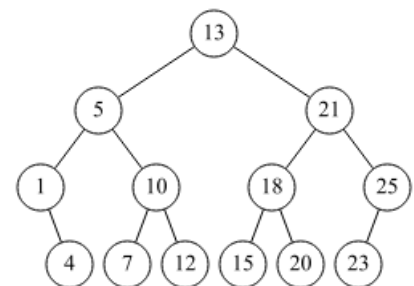
13, 20, 5, 1, 15, 10, 18:



+ 25, 4, 21:



+ 7, 12, 23:



Solution 5 (2.4-tree → Red-black Tree – 2 points)

1. Red-black tree associated with the 2-4 tree of the subject:

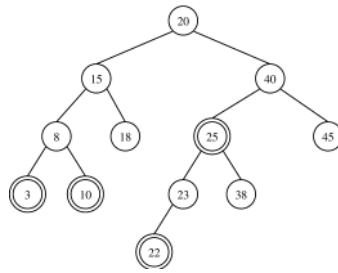
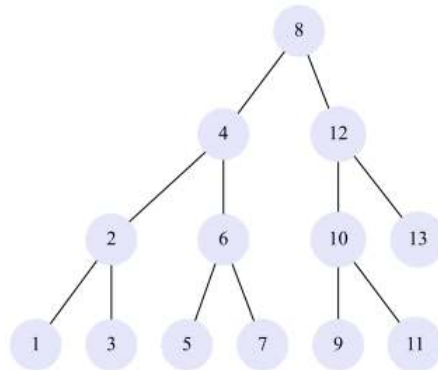


Figure 1: Arbre bicolore

2. The result tree is not an AVL.
The node that contains 40 has a balance-factor of 2 (the tree is not h-balanced).

Solution 6 (Trees and mystery – 3 points)

1. Tree built by `makeTree(13)`:



2. Properties of the tree built by `makeTree(n)` ($n > 0$):

- (a) Complete tree
- (b) Binary search tree