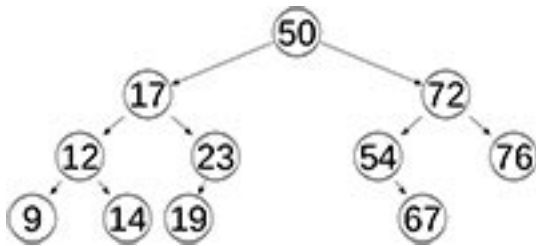


CS61B SPRING 2016 SECRET SECTION 3 WORKSHEET

Week of March 14, 2016

1. Warm up: Given a diagram of a BST, practice inserting and deleting nodes. (There may be more than one way to go about each of these.)

- Insert 22.
- Delete 12 and 72.
- How should you delete 50?

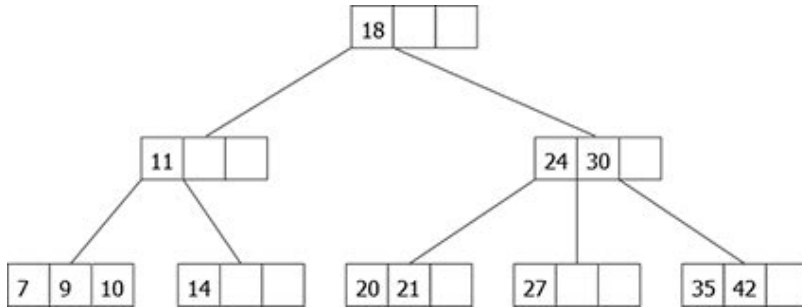


2. Implement `isBalanced`. A balanced tree assures that its left and right subtrees differ by no more than 1. The `getHeight` method will come in handy.

```
1 public static int getHeight(TreeNode root) {
2     if (root == null) return 0; // Base case
3     return Math.max(getHeight(root.left),
4                     getHeight(root.right)) + 1;
5 }
6
7 public static boolean isBalanced(TreeNode root) {
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27 }
```

3. Insert and delete some elements from the following 2-3-4 tree.

- Insert 13
- Delete 18
- Insert 43
- Insert 44



4. Convert the following 2-3 tree into a left-leaning red-black tree.

