

Quiz 9 (Nov 15)

By taking this quiz, you agree to adhere to the honor code of the class.

Name:

netid:

Name:

netid:

Write an efficient method `void raiseDeepest()` in `AVLTree` that performs rotations to make the deepest node in the tree the root (you can assume the tree is balanced and the heights of all nodes are all correct). Don't worry about keeping the tree balanced or updating heights. Recall that `Node` has `int getHeight()`, `Node getLeft()`, and `Node getRight()`. `AVLTree` has `void rotate(Node node)`. You can assume the tree has at least 2 entries. There are two parts of the problem: a) find the deepest node, b) rotate it to the root (4pt each). What is the asymptotic computational complexity of this method (2pt)?

Hint: for part a, you want to traverse to the deepest node. There are four cases at each node, considering the existence of left and right child. For part b, `rotate(node)` is a good function to use. How does `rotate(node)` affect the depth of `node`?

```

Void raiseDeepest() {
    Node current = root;
    While (current.getLeft() != null || current.getRight() != null) {
        If (root.getLeft() == null) {
            current = root.getRight();
        }
        Else if (root.getRight() == null) {
            current = root.getLeft();
        }
        Else {
            If (current.getRight().height() > current.getLeft().height())
                Current = current.getRight();
            else
                Current = current.getLeft();
        }
    }

    While (current != root) {
        rotate(current);
    }
}

```