(1) Create a BST by inserting the following values in this order: **R, U, A, T, W, P, I  
For each node, be sure to represent value of its N attribute.**

(2) When “reading” the letters from left to right in any level of a BST, do they always appear in ascending order? Provide counter-example or explain why.

(3) Time analysis of recursive functions

**frequency** **cost**

**public** Key min() { **return** min(root).key; } \_1\_ \_D+1\_

**private** Node min (Node parent) {

**if** (parent.left == **null**) { **return** parent; } \_1\_ \_1\_

**return** min(parent.left); \_1\_ \_D\_

}

**public** Key nonRecursiveMin() {

Node n = root; \_1\_ \_1\_

**while** (n.left != **null**) { \_D\_ \_1\_

n = n.left; \_D\_ \_1\_

}

**return** n.key; \_1\_ \_1\_

}

Note: Above, D is the Maximum Depth of the Tree reached by following left references.

(4) Given initial tree you created above, what is **floor(R)**?

(5) Given initial tree you created above, what is **floor(S)**?

(6) What is resulting BST after calling **deleteMin**()?

(7) What is result of preorder traversal of BST created above?