**Quiz Lab 13**

**CS 2001 – Data Structures (CS)**

**Fall 2024**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Q:1. Transform the provided Binary Search Tree into an AVL tree. (Do rough work on backside of this paper). In situations where there are multiple options for performing rotations, prioritize the rotations in the following order:**

**1. LR Rotation**

**2. RL Rotation**

**3. RR Rotation**

**4. LL Rotation**



**Q: 2. Draw the updated AVL tree after deleting the root node of the above derived AVL tree in (Question 01).**

**Q:3. Write the missing code in provided boxes.**

**//Rotate right**

Node\* rightRotate (Node\* y){

Node\* x = y->left;

Node\* T2 = x->right;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x-> |  | = |  |  |
| y-> |  | = |  |  |

y->height = max(height (y->left), height (y->right)) +1;

x->height = max(height (x->left), height (x->right)) +1;

return x;

}

**Q:4.** Design a function that takes two values as input from the user and returns true if the subtree of a given AVL tree, rooted between those two values, is itself an AVL tree. Otherwise, it returns false. (You can use integer tree for it)

**Q:5**. Take n inputs from the user and construct an AVL tree using those values. Next, take an integer k from the user and delete the kth largest value node from the constructed AVL tree.