

C1 Review Test

Course: Advanced Data Structures and Algorithms (ADA)

M.Tech. (IT) – 1st Semester ; Session : August-December 2018

Course Instructor: Dr. T. Pant & Dr. S. Maity

Full Marks: 35

Time: 1 hour

1. Suppose the following numbers are inserted into a Binary Search Tree (BST) one-by-one in the sequence as they are shown below.

11, 6, 43, 41, 50, 14, 24, 19, 34, 27

Show the **final state** of the above BST if we delete its root node.

3 Marks

2. In an AVL tree, following algorithm is used to balance the tree if it gets unbalanced after an insertion. Let the newly inserted node be w . Let z be the youngest ancestor to become unbalanced. Let y and x are the child and grandchild of z that comes on the path from w to z . Following are the possible 4 arrangements:

- a) y is left child of z and x is left child of y (Left Left Case)
- b) y is left child of z and x is right child of y (Left Right Case)
- c) y is right child of z and x is right child of y (Right Right Case)
- d) y is right child of z and x is left child of y (Right Left Case)

Following are the operations to be performed in each of the above 4 cases to balance the subtree rooted at z .

Left Left Case: Step 1. Right Rotate (z)

Left Right Case: Step 1. Left Rotate (y); Step 2. Right Rotate(z)

Right Right Case: Step 1. Left Rotate (z)

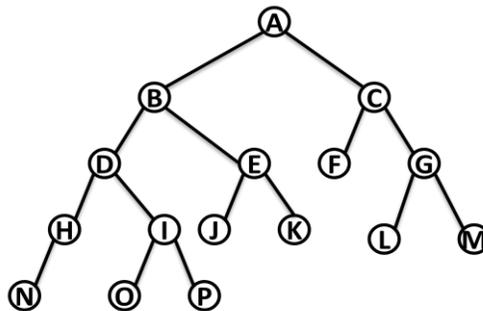
Right Left Case: Step 1. Right Rotate (y); Step 2. Left Rotate(z)

Suppose the following numbers are inserted into an AVL tree one-by-one in the sequence as they are shown below. Show the final state of the AVL tree.

29, 8, 35, 5, 21, 34, 41, 2, 17, 25, 13.

5 Marks

3. The following Scapegoat tree is given. Now, a new node is inserted whose value 'X' lies between 'D' and 'O'. Draw the final state of the Scapegoat tree after this insertion.



5 Marks

4. Write the most efficient algorithm to find the inorder predecessor of a node p in a binary search tree. Demonstrate your algorithm with suitable examples. Analyze the time complexity of your algorithm.

5 Marks

5. Write an efficient algorithm to convert a prefix expression into postfix using stack (non-recursive). Demonstrate your algorithm on the following string:-

$- + * + A B C / D + E * F G H$

5 Marks

6. Write an efficient algorithm to concatenate two linear lists with Header Nodes having the 'head' and the 'tail' pointers.

2 Marks

7. Write an efficient algorithm to reverse the elements of a singly linked list without using a temporary list. Demonstrate your algorithm with suitable examples.

5 Marks

8. Insert into an initially empty red-black tree, the following: 4 7 12 15 3 5 14 18 16 17. Perform all the necessary steps (coloring/rotations) of Red Black Tree after inserting each node and shape the tree. Write R to represent Red color and B to represent Black color along with the node. (e.g if node 7 is red then write 7R and if node 7 is black write 7B).

5 Marks