



## Continuous Assessment Test – 2 – Answer Key

Course Name & code: Data Structures and Algorithms & BCSE202L

Programme Name : B.TECH

### Slot: B1+TB1

---

1] Given a linked list and a value  $x$ , partition it such that all nodes less than  $x$  come before nodes greater than or equal to  $x$ . Devise an algorithm to perform the above operation using the given input and illustrate its working in detail.

Input: 40, 10, 15, 29, 4, 33.

**Ans:** Below is the step to solve this problem: - 4 M

- Initialize first and last nodes of below three linked lists as NULL.
- Linked list of values smaller than  $x$ .
- Linked list of values equal to  $x$ .
- Linked list of values greater than  $x$ .
- Now iterate through the original linked list. If a node's value is less than  $x$  then append it at the end of the smaller list. If the value is equal to  $x$ , then at the end of the equal list. And if a value is greater, then at the end of the greater list.
- Now concatenate three lists.

Input: 40, 10, 15, 29, 4, 33

40 -> 10 -> 15 -> 29 -> 4 -> 33 (Should explain step by step) - 4M

Assume,  $x = 15$

4 -> 10 -> 15 -> 29 -> 33 -> 40 - 2M

2] Given an input of distinct integers.

Input: 10, 100, 93, 32, 35, 65, 80, 90, 94, 6

Use a binary search tree data structure to obtain an output by replace every element with the least greater element on its right or with -1 if there are no greater elements and illustrate the working in detail.

**Ans:** Construct BST form the given input - 4M

If the student identify the in-order successor procedure - 1M

Hint: A node's in-order successor is the node with the least value in its right subtree, i.e., its right subtree's leftmost child. If the right subtree of the node doesn't exist, then the in-order successor is one of its ancestors. To find which ancestors are the successor, we can move up the tree towards the root until we encounter a node that is left child of its parent. If any such node is found, then in-order successor is its parent; otherwise, in order successor does not exist for the node. At that time use -1 to replace the particular key value.

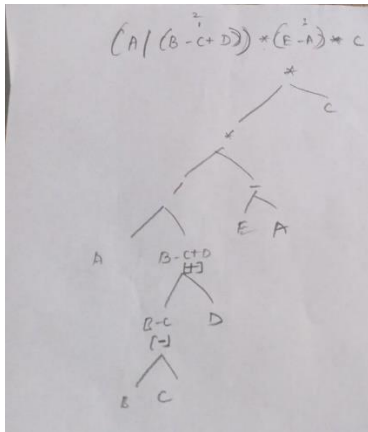
The idea is to traverse from left to right and find in-order successor for each node and replace with respective value.

(Should explain step by step) - 4M

Final tree - 1M

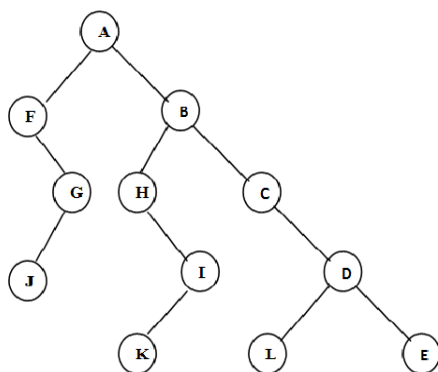
3] a. Construct the expression tree using the given input

Input:  $(A/(B-C+D))*(E-A)*C$



**Ans:** Step by Step explanation to be followed. 5m

b. For the given tree apply all traversal method with the time complexity of  $O(n)$ .



**Ans:** in-order, pre-order and post-order time complexity –  $O(n)$  – 2m

Inorder: FJGAGKIBCLDE – 1m

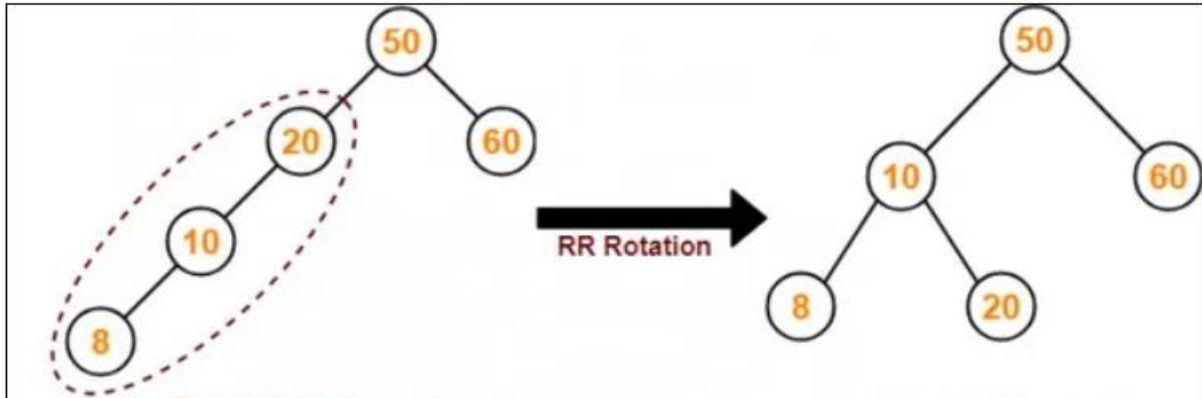
Preorder: AFGJBHIKCDLE – 1m

Postorder: JGFKIHLEDCBA – 1m

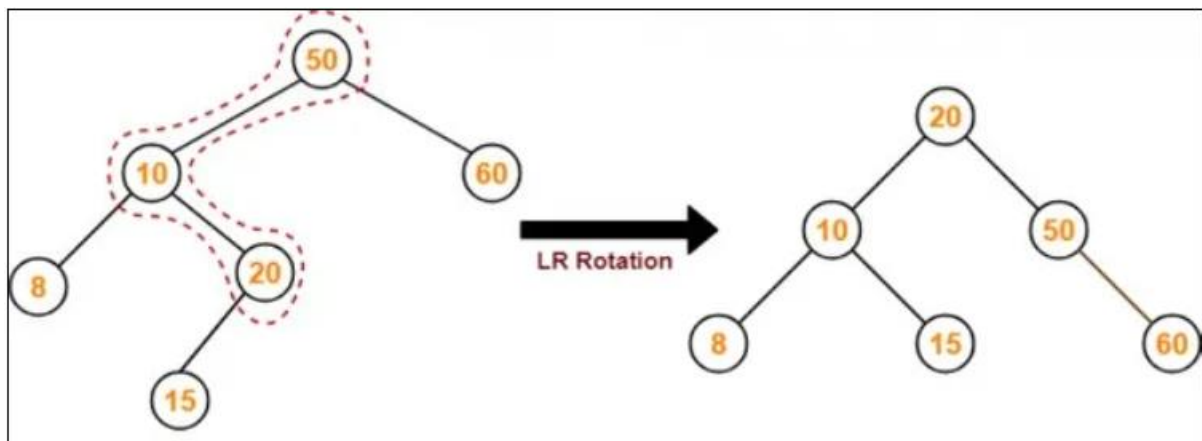
4] 50, 20, 60, 10, 8, 15, 32, 46, 11, 48

**Ans:** 10 Steps – 10M

First 5 steps insertion each element till element 8 then, apply RR rotation to balance the tree.

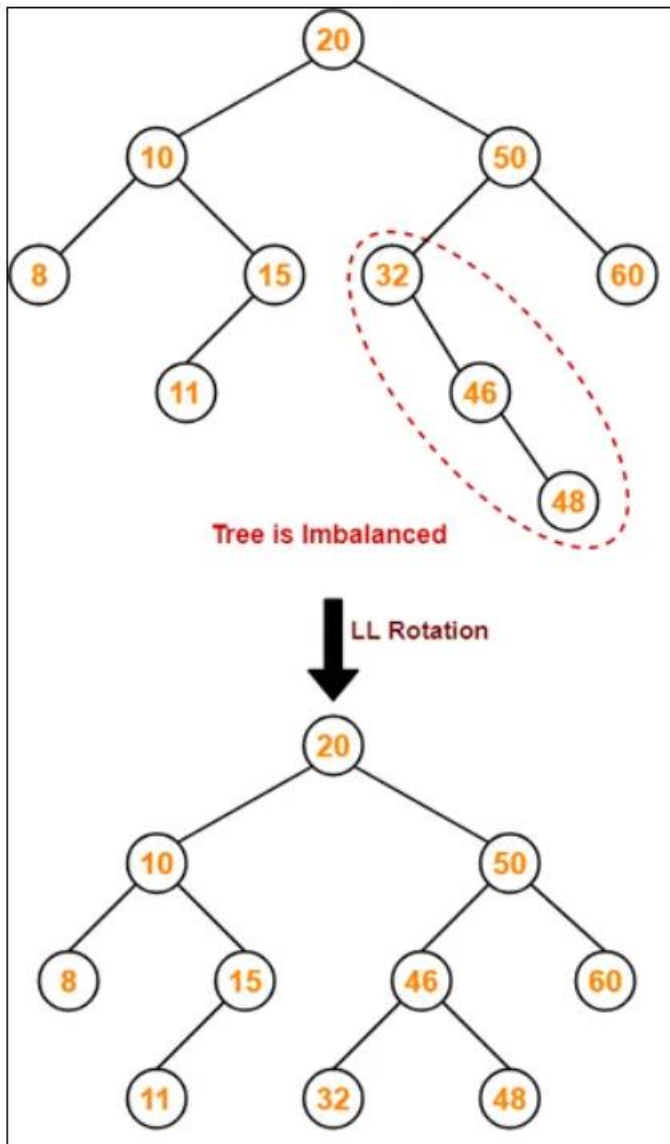


Step 6 insert element 15 then, apply LR rotation to balance.



Step 7 insert 32, Step 8 insert 46, Step 9 insert 11

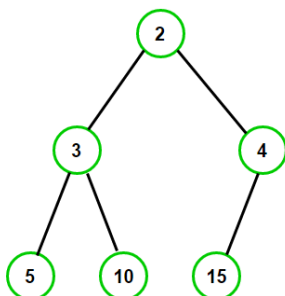
Step 10 insert 48 then, apply LL rotation.



5] Given an array of input as Set 1 = [2,3,4,5,10,15] and Set 2 = [2,10,4,5,3,15]. Check if an array belongs to the types of heap. If exist then justify how these inputs belong to the types of heap along with its working in detail. If does not exist then apply suitable operation to satisfy heap property.

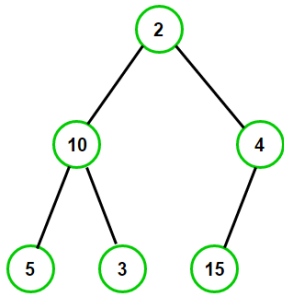
**Ans:** Set 1 – satisfy the min heap property. -2 M

Illustrate – 2M



Set 2 – not satisfy – 2M

Illustration – 2M



Apply min/Max heap – 2M