International Institute of Information Technology, Bangalore.
EG 102 L Data Structures and Algorithms. April 2020

1. Given two BST's , write a linear time algorithm to merge them in to a Balanced Binary Search Tree .

2. Your are given binary tree such that if you swap two nodes of the tree, you get a BST. Implement an efficient algorithm to find a BST, given the tree.

3. Given Pre order Traversal of the BST, implement an efficient algorithm to build the BST.

4. Given Level order traversal of the BST, implement an efficient algorithm to build the BST.

5. In the Interval Tree, we assumed that all the left end point of intervals are distinct, implement an Interval to allow same left end point of the intervals.

6. Given Post order Traversal of the BST, implement an efficient algorithm to build the BST.

7. Given a binary tree with some keys at each node, implement an efficient algorithm to compute the largest subtree that is BST.

8. Given a BST, implement an efficient algorithm to compute the largest subtree that satisfies the AVL Property at each node.

9. Given a binary tree with some keys at each node of the tree, implement an efficient algorithm to check if the tree is a BST.

10. Any sequenceAof sizenis calledB-sequenceif: $A1 < A2 < ... < Ak > Ak + 1 > Ak + 2 > ... > An$where1 $\leq k \leq n$. That is, a sequence which is initially strictly increasing and then strictly decreasing (the decreasing part may or may not be there). All elements inAexcept the maximum element comes atmost twice (once in increasing part and once in decreasing part) and maximum element comes exactly once. All elements coming in decreasing part of sequence should have

come once in the increasing part of sequence. You are given a B-sequenceSandQoperations. For each operation, you are given a value-val. You have to insertvalinSif and only if after insertion,Sstill remains a B-sequence . After each operation, print the size ofS. After all the operations, print the sequenceS.

11. Design a data structure to answer the following queries efficently

    (a) Insert an integer to the list.

    (b) Given an integer x, you're about to find an integer k which represent x's index if the list is sorted in ascending order. Note that in this problem we will use 1-based indexing.

    The first line contains an integer Q, which denotes how many queries that follows. The next Q lines will be one of the type queries which follow this format:

    1 x means insert x to the list

    2 x means find x's index if the list is sorted in ascending order.

    For each query type 2, print a line containing an integer as the answer or print "no" if the requested number does not exist in the current lis. Example: Input:

    10

    10

    1 100

    1 74

    2 100

    2 70

    1 152

    1 21

    1 33

    2 100

    2 21

    21

Output:

2

no

4

1

no

12. You are given a sequence of n distinct intergers $a_0, a_1, \ldots a_{n-1}$. In each iteration you pick the minimum number and delete it, the cost deteting the minimum number is the number of numbers to the right of it. Repeat this n number of times. Given a_i's implemet $O(n \log n)$ algorithm to compute the total cost of n iterations. For example

$$A : 6, 2, 8, 4, 9, 3$$
$$4 + 0 + 1 + 2 + 1 + 0 = 8$$

13. You are given a sequence of n intergers $a_0, a_1, \ldots a_{n-1}$ (both positive and negative) in the non decreasing order. You are given m queries, each query consisting of indicies i and j, $1 \leq i \leq j \leq n$. For each query, determine the most frequent value among the integers $a_i, \ldots a_j$ You are given a sting of length n containg charater A, B, C, D.

    You have m queries, each query is of the type k X, for each querry delete the kth ocscurence of character X. If the number of X left in the string is less than k, do nothing.

    Output: Print the string after m queries.

    Example: $ABCDBCAAB$ and querries are

    2 A

    1 C

    1 D

    3 B

    2 A

    Output:ABBC

14. You are given a sequence of n distinct intergers $a_0, a_1, \ldots a_{n-1}$. In each iteration you pick the maximum number and delete it, the cost deteting the maximum number is the number of numbers to the left of it. Repeat this n number of times. Given a_i's implemet $O(n \log n)$ algorithm to compute the total cost of n iterations. For example

$$A : 6, 2, 8, 4, 9, 3$$

$$4 + 2 + 0 + 1 + 1 + 0 = 8$$

15. Given two nodes in the tree A, B, there is only one path form A to B in the BST.

    Implement an efficient algorithm to find the maximum and minimum values on the path form A to B in the BST.