

CS2002301 & EC2002302 Data Structures

Homework #2

Due Date: 11/25/2020 23:55

Announcements

1. This assignment is weighed 10 points.
2. Submit your code to the OJ system (<http://nlp.csie.ntust.edu.tw:2020/>).
3. Also, submit a report in PDF format to the Moodle system (please attach your source code in the report)

1. (5 pts) Thanos Finger Snap

In the film “*The Avengers*”, when Thanos snap his finger, half of population in the Earth will disappear. Let’s assume Thanos is the most senior software engineer in the world. When he snaps his finger, he will perform the following operations:

- a. He randomly numbers all citizens in the world, and thus build a binary search tree.
- b. Since he has to guarantee that he will not be deleted, he will locate himself at the root of the tree.
- c. He first deletes the members of the Avengers.
- d. He tries his best to *horizontally* cut the binary tree in half (that is, the difference between the number of nodes in the upper part and the number of nodes in the lower is the smallest), and kill all people in the lower part of the binary tree.

Take figure 1 for example, Thanos will delete all nodes that are lower than the red line (9, 10, 21, 19, 28, 36, 59, 65, 60).

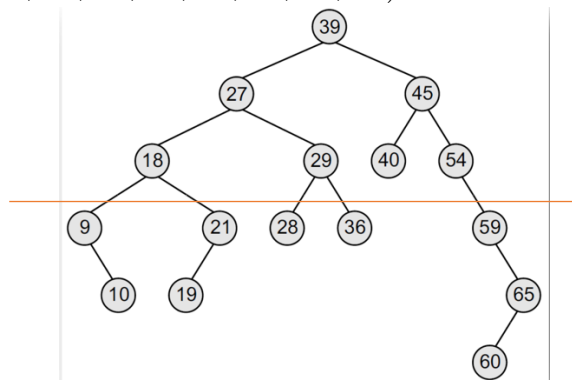


Figure 1

Please follow the operations below:

1. We will first give you Thanos' node number N (39 in figure 1), and then give you the numbers of all citizens. Please insert the numbers into the BST in order, then output the *post-order* traversal of the current BST.
2. We will give you the numbers of the Avengers. Please delete them in the BST. Replace the node's value with its in-order predecessor.
3. Cut the BST. We denote the upper half of the BST as S (the survivors), and denote the lower half of the BST as V (the victims). We guarantee that there is only one way to cut the BST for all test cases.
4. Please output the *in-order* traversal of S , and then output the *pre-order* traversal of V .

Of course, V is not a BST; instead, it is a forest. To traverse a forest F , please follow the steps below:

- (1) If F is empty then return.
- (2) Visit the root of the first tree of F .
- (3) Traverse the subtrees of the first tree in forest pre-order.
- (4) Traverse the remaining trees of F in forest pre-order.

For example, the pre-order traversal of figure 2 is:

A B C D E F P Q X Y Z

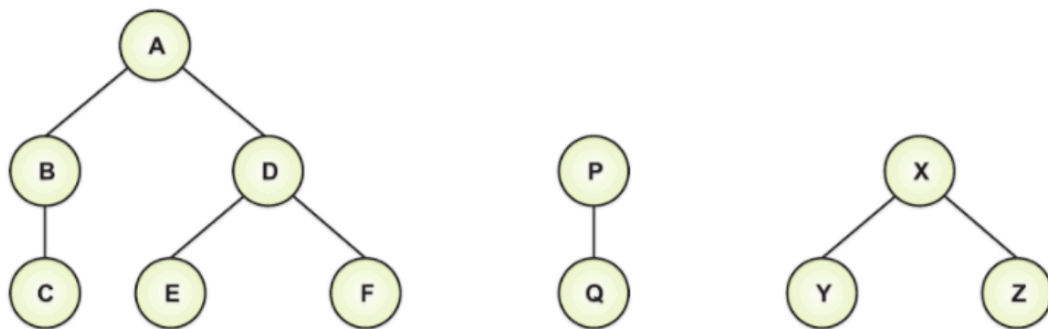
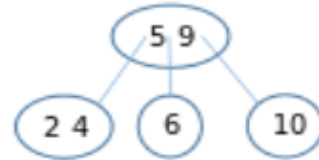
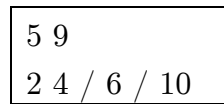


Figure 2

- * Sample I/O and I/O constraints are showed on the OJ system.
- * In this problem, only 60% of the test cases will be showed on OJ system, while the remaining 40% are hidden. Note that we will use your **latest submission** on OJ system to test the hidden cases, so make sure your latest submission is the best version.
- * In the report, briefly explain the way you solve the problem.

2. (5 pts) 2-3 Tree

- a. (3 pts) Please implement a 2-3 tree. Your program will read a number sequence, please insert the numbers into the 2-3 tree in order. When printing the following 2-3 tree, please output:



- * Sample I/O and I/O constraints are showed on the OJ system.
 - * In this problem, there are no hidden cases.
- b. (2 pts) Given a number sequence : 20, 45, 30, 50, 100, 70, 40, 10, 87, 1. Please show all steps for building a 2-3 tree, and explain them with your code. That is, the execution flow of your program when processing those numbers.