

## Assignment #6 (An application of tree searching)

Goal: Given a sequence of numbers, determine those numbers that appear  $2n+1$  times, where  $n$  is an integer and  $n \geq 0$ .

Guide: 是指出現奇數次者

For example, in the sequence 3, 3, 5, 9, 5, 3, 4, 8, 5, 3, 10, 10, 8, the numbers that appear  $2n+1$  times are 5, 9, and 4, which are the required outputs. This problem can be solved with a binary search tree.

From left to right, you can examine each number in the sequence. For each number  $x$ , search  $x$  in the binary search tree. If  $x$  is already in the tree, which means that  $x$  has appeared for  $2n+1$  times, then the node  $x$  should be removed. Otherwise, a new node for  $x$  should be inserted into the binary tree, because  $x$  appeared  $2n$  times before ( $n$  may be zero). After you go through the entire sequence, those numbers that appear  $2n+1$  times will be stored in the binary search tree. Then, you have to output all numbers in the tree as the solution.

In the binary search tree, smaller numbers should always be stored in the left subtree, while larger ones are stored in the right subtree. To remove a node, there are three cases.

Case 1: To remove a leaf node: just remove it.

Case 2: To remove a nonleaf node that has only one son: you should remove this node, and then connect its son to its parent.

Case 3: To remove a nonleaf node  $p$  that has two sons: you should first find the left-most node  $q$  in the right subtree of  $p$  ( $q$  is called the inorder successor of  $p$ ). Then, node  $p$  should be replaced by node  $q$ . To finish the replacement, you have to remove node  $q$ , which only has a right son (note that this is case 2).

Input:

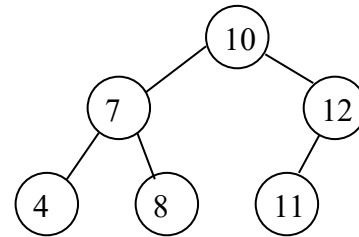
The input contains several test cases. Each case is a sequence of positive numbers (may be more than one line). Two numbers are separated by a blank space, and each case ends with a number -1 (-1 is not a part of the test case). Notice, if there is no other numbers following -1, then it means the end of the input. In other words, the end of file means the end of the input.

Output:

For each number in the sequence, you have to output the final binary search tree with the increasing order.

inorder  
leftChild  
rightChild

node	4	7	8	10	11	12
left	0	4	0	7	0	11
right	0	8	0	12	0	0



Here, we use 0 to represent a null node. In your output, the numbers should be separated by a blank space, and two cases should be separated by a new line. NOTICE, there are no new lines before the first test case, and after the last test case.

Sample Input : (If you type the input from the keyboard in Windows, press ctrl+z, which means the end of file.)

10 7 12 8 11 4 -1

3 3 5 9 5 3 4 8 5 3 10 10 8 -1

Sample Output :

node: 4 7 8 10 11 12

left: 0 4 0 7 0 11

right: 0 8 0 12 0 0

node: 4 5 9

left: 0 0 4

right: 5 0 0