IIT PALAKKAD

COMPUTER SCIENCE AND ENGINEERING
Indian Institute of Technology, Palakkad
CS2130: Data Structures and Algorithms Lab
Lab 10 (Binary Search Trees)

12 Oct, 2018

Time: 3 hrs

General Instructions

- You are expected to follow good programming practices (refer to the book titled *C How to Program* by Dietel and Dietel). Your programs should handle all kinds of test cases - positive, negative, corner cases/boundary conditions, good input, bad input and are never expected to crash on any input.
- You are also expected to provided a high level description of the functions used in your programs. There is no restriction on the format except that it should be concise and precise. Further, the running time and additional space (excluding the space for the arguments) used by the functions should be clearly stated and justified.
- **Please note that poor programming practices and improper documentation can attract a penalty of up to 20% of the marks allocated to a question.**

1. (30 points) Write a program to build a binary tree (with positive integer values in every vertex as keys) and print its preorder, inorder and postorder traversal sequences. Every node is associated with an index that gives the information regarding the order in which the nodes were inserted into the tree. For example, the root has index 1 and the second element that was inserted has index 2 and so on.

   Input-Output Format: The first line of the input consists of the integer value at the root of the binary tree. Each of the subsequent lines specify one of the following operations to be performed on the tree.

   INS x i l. Insert the element **x** as the left child of the node with index **i** into the tree. If there is no node with index **i** or the node with index **i** already has a left child, then print -1.
   INS x i r. Insert the element **x** as the right child of the node with index **i** into the tree. If there is no node with index **i** or the node with index **i** already has a right child, then print -1.
   PRE. Prints the preorder traversal sequence of the tree with every pair of consecutive elements separated by a space.
   PST. Prints the postorder traversal sequence of the tree with every pair of consecutive elements separated by a space.
   INO. Prints the inorder traversal sequence of the tree with every pair of consecutive elements separated by a space.
   OCC k. Prints the number of occurrences of the key **k** in the tree.

   ```
   Sample Input:
   500
   INS 20 1 l
   ```

Continued on next page...

```
INS 30 1 r
INS 10 2 l
INS 1 2 r
INS 2 3 r
INS 3 4 l
INS 1 7 l
PRE
PST
INO
OCC 1
INS 100 10 l
OCC 50
OCC 10
INO

Expected Output:
500 20 10 3 1 1 30 2
1 3 10 1 20 2 30 500
1 3 10 20 1 500 30 2
2
-1
0
1
1 3 10 20 1 500 30 2
```

2. (70 points) Write code to implement the standard operations of a binary search tree with positive integer values in every vertex as keys.

   Input-Output Format: Each line contains a specific operation to be performed on the tree.

   INS x. Insert **x** into the tree.
   PRE. Prints the preorder traversal sequence of the tree with every pair of consecutive elements separated by a space.
   PST. Prints the postorder traversal sequence of the tree with every pair of consecutive elements separated by a space.
   INO. Prints the inorder traversal sequence of the tree with every pair of consecutive elements separated by a space.
   DEL x. Deletes the vertex **x** from the tree. If **x** has 2 children, then replace **x** with its inorder successor. If the delete is unsuccessful, then print -1.
   SER k. Searches for the key **k** in the tree. If the search is successful print "**Y**", otherwise print "**N**".

```
Sample Input:
INS 20
INS 30
INS 10
```

Question 2 continues on the next page...

```
INS 1
INS 2
INS 3
INS 5
INS 100
INS 110
INS 90
INS 15
PRE
PST
INO
DEL 1
PRE
PST
INO
DEL 10
PRE
PST
INO
SER 48
SER 100
SER 248
SER 200
INO
INS 500
INO
DEL 550
```

```
Expected Output:
20 10 1 2 3 5 15 30 100 90 110
5 3 2 1 15 10 90 110 100 30 20
1 2 3 5 10 15 20 30 90 100 110
20 10 2 3 5 15 30 100 90 110
5 3 2 15 10 90 110 100 30 20
2 3 5 10 15 20 30 90 100 110
20 15 2 3 5 30 100 90 110
5 3 2 15 90 110 100 30 20
2 3 5 15 20 30 90 100 110
N
Y
N
N
2 3 5 15 20 30 90 100 110
2 3 5 15 20 30 90 100 110 500
-1
```