BST Implementation

Implement the BST class which includes following functions -

1.insert - Given an element, insert that element in the BST at the correct position. If element is equal to the data of the node, insert it in the left subtree.
2.delete - Given an element, remove that element from the BST. If the element which is to be deleted has both children, replace that with the minimum element from right sub-tree.
3.search - Given an element, find if that is present in BST or not. Return true or false.
4.printTree (recursive) - Print the BST in in the following format -
For printing a node with data N, you need to follow the exact format -

N:L:x,R:y
where, N is data of any node present in the binary tree. x and y are the values of left and right child of node N. Print the children only if it is not null.
There is no space in between.
You need to print all nodes in the recursive format in different lines.

Input Format

First line number of queries
From Second line start giving queries

Output Format

Results of each query in different line

Sample Input 0

```
6
1 2
1 3
1 1
4
2 2
4
```

Sample Output 0

```
2:L:1,R:3
1:
3:
3:L:1,
1:
```

Sample Input 1

```
6
1 2
1 3
1 1
3 2
2 2
3 2
```

Sample Output 1

```
true
false
```

Contest ends in 23 days

Submissions: 61
Max Score: 10
Difficulty: Medium

Rate This Challenge:

More

Java 7

```java
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

class node{
    int data;
    node left;
    node right;
    node(int data){
        this.data=data;
        this.left=null;
        this.right=null;
    }
}

```

```
public class Solution {


    static node ins(int d,node root){
        if(root==null){
            return new node(d);
        }
       if(d<=root.data){
            root.left=ins(d,root.left);
        }
        else{
            root.right=ins(d,root.right);
        }
        return root;
    }

    static node del(node root,int d){
         if(root==null) return null;
        if(d<root.data){
            root.left=del(root.left,d);
        }
        else if(d>root.data){
            root.right=del(root.right,d);
        }
        else{
          if(root.left==null && root.right==null) return null;
          else if(root.left!=null && root.right==null) return root.left;
          else if(root.left==null && root.right!=null) return root.right;
          else{
              node t=root.right;
              while(t.left!=null) t=t.left;
              root.data=t.data;
              root.right=del(root.right,t.data);
              return root;
              }
        }
        return root;
    }
```

```
56
57      static boolean ser(int d,node root){
58          if(root==null){
59              return false;
60          }
61          if(root.data==d){
62              return true;
63          }
64          if(d<root.data){
65              return ser(d,root.left);
66          }
67          else{
68              return ser(d,root.right);
69          }
70
71      }
72
73      static void disp(node root){
74          if(root==null) return;
75          System.out.print(root.data+":");
76          if(root.left!=null){
77              System.out.print("L:"+root.left.data+",");
78          }
79          if(root.right!=null){
80              System.out.print("R:"+root.right.data);
81          }
82          System.out.println();
83          disp(root.left);
84          disp(root.right);
85
86      }
87
88      public static void main(String[] args) {
89          node root=null;
90          Scanner sc=new Scanner(System.in);
91          int n=sc.nextInt();
92          int op,val;
93          for(int i=0;i<n;i++){
```

```
94                op=sc.nextInt();
95                if(op==1){
96                    val=sc.nextInt();
97                    root=ins(val,root);
98                }
99                else if(op==2){
100                   val=sc.nextInt();
101                   root=del(root,val);
102               }
103               else if(op==3){
104                   val=sc.nextInt();
105                   System.out.println(ser(val,root));
106               }
107               else if(op==4){
108                   disp(root);
109               }
110           }
111       }
112   }
```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code