CSC 212 Midterm 2 Solution - Fall 2014

College of Computer and Information Sciences, King Saud University
Exam Duration: 90 Minutes

07/12/2014

Question 1 [35 points]

1.

```
public static <T> void moveAfter(Stack<T> st1, Stack<T> st2, int i) {
        Stack<T> tmp = new Stack<T>();
        int c = 0;

        while (!st1.empty()) {
                T x = st1.pop();
                tmp.push(x);
                if (i == c)
                        while (!st2.empty())
                                tmp.push(st2.pop());
                c++;
        }

        while (!tmp.empty())
                st1.push(tmp.pop());
}
```

2.

```
public static <T> int countEquals(Stack<T> st, T e) {
        Stack<T> tmp = new Stack<T>();
        int count = 0;

        while (!st.empty()) {
                T x = st.pop();
                tmp.push(x);
                if (x.equals(e))
                        count++;
        }

        while (!tmp.empty())
                st.push(tmp.pop());

        return count;
}
```

Question 2 [20 points]

1. (a) Preorder: ”/ × + 6 4 - 5 3 - + 7 1 12”.
   (b) Inorder: ” 6 + 4 × 5 - 3 / 7 + 1 - 12”.
   (c) Postorder: ”6 4 + 5 3 - × 7 1 + 12 - /”.

2. ”3 4 9 × + 4 6 × 7 × - 3 -”.

3. Trace:

8	6 8	14	7 14	5 7 14	2 14	28
6 28	8 6 28	14 28	7 14 28	7 28	4	

Question 3 [35 points]

1. Using the Binary Search Tree in Figure 1, insert the following:

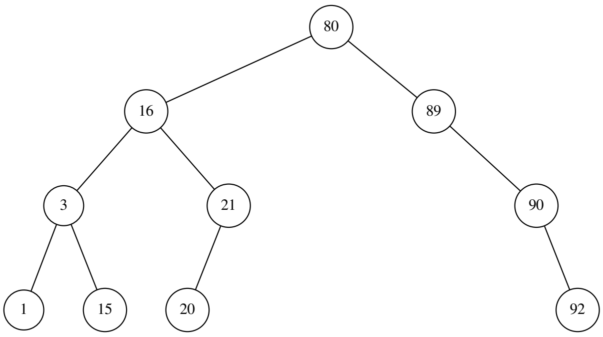
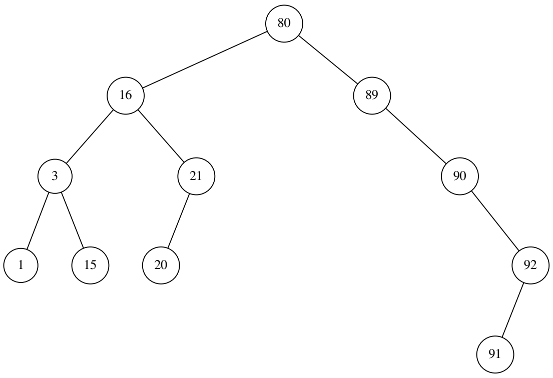
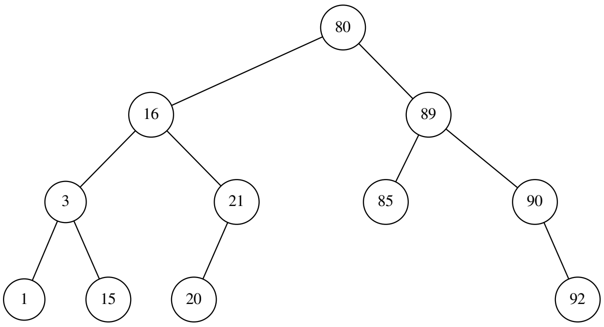


Figure 1: A BST.

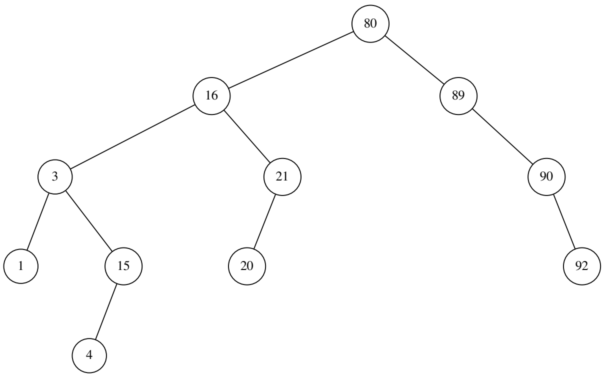
(a) 91 into the **Original tree**.



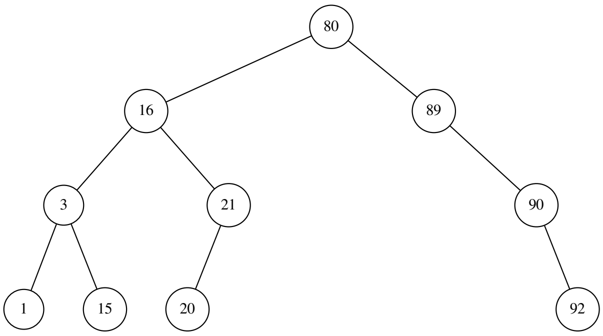
(b) 85 into the **Original tree**.



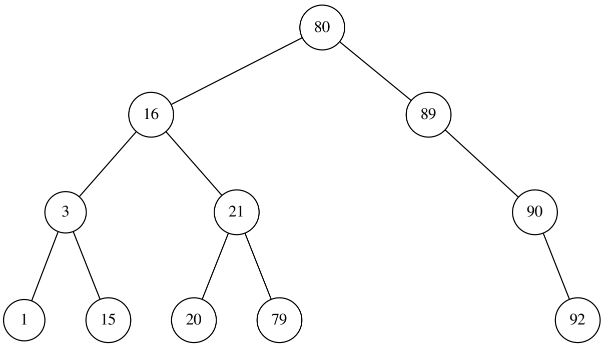
(c) 4 into the **Original tree**.



(d) 15 into the **Original tree**.

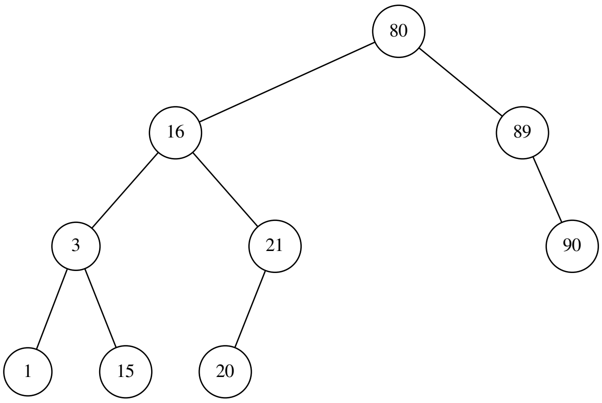


(e) 79 into the **Original tree**.

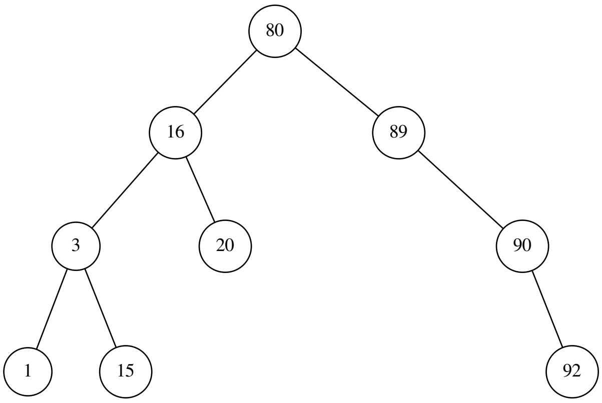


2. Using the Binary Search Tree in Figure 1, delete the following:

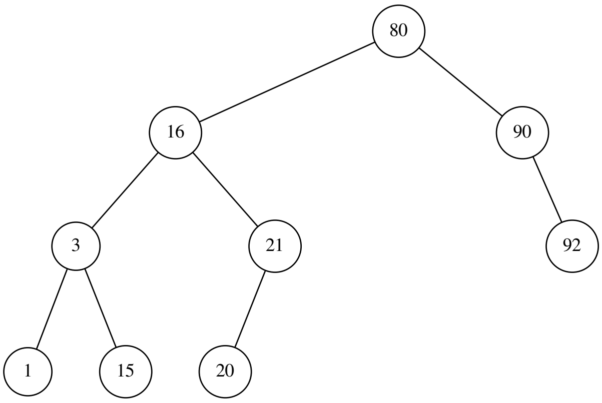
   (a) 92 from the **Original tree**.



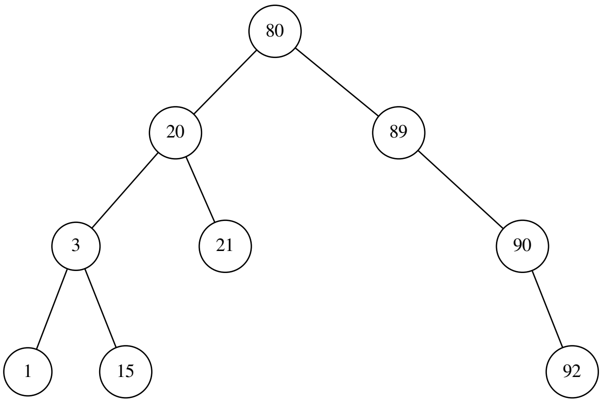
(b) 21 from the **Original tree**.



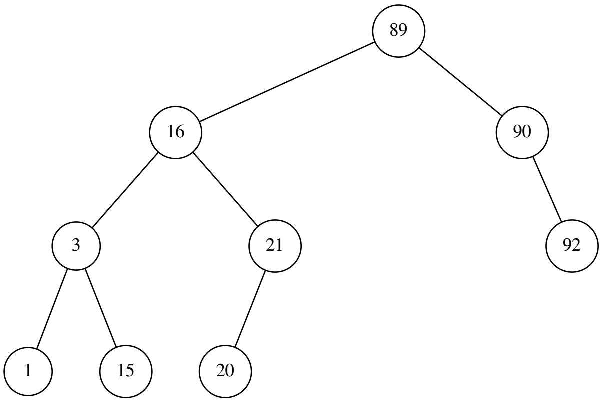
(c) 89 from the **Original tree**.



(d) 16 from the **Original tree**.



(e) 80 from the **Original tree**.



3.

```
public boolean updateChildrenData(int k, T e1, T e2) {
        BSTNode<T> p = root;
        while (p != null && p.key != k) {
                if (p.key > k)
                        p = p.left;
                else
                        p = p.right;
        }
        if (p == null)
                return false;
        if (p.left == null || p.right == null)
                return false;
        p.left.data = e1;
        p.right.data = e2;
        return true;
}
```

Question 4 [10 points]

```
public boolean isMirror(BT<T> bt) {
        return recIsMirror(root, bt.root);
}

private boolean recIsMirror(BTNode<T> t1, BTNode<T> t2) {

        if ((t1 == null) != (t2 == null))
                return false;
        if (t1 == null)
                return true;
        return (t1.data.equals(t2.data) && recIsMirror(t1.left, t2.
           right) && recIsMirror(t1.right, t2.left));
}
```