CSC 212 Midterm 2 Solution - Spring 2015

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

23/04/2015

Question 1 [35 points]

1. Answer:

```java
// =========== Non-Recursive ================
public static boolean isIncreasing(Stack<Integer> st){
        Stack<Integer> st2 = new Stack<Integer>();
        boolean result = true;
        Integer cur = null;
        Integer pre = null;
        while(!st.empty()){
                cur = st.pop();
                if(pre != null)
                        if(pre > cur)
                                result = false;
                pre = cur;
                st2.push(cur);
        }
        while(!st2.empty())
                st.push(st2.pop());
        return result;
}
// ============= Recursive ==================
public static boolean isIncreasing(Stack<Integer> st){
        if(st.empty())
                return true;
        Integer X = st.pop();
        boolean result = recIsIncreasing(st, X);
        st.push(X);
        return result;
}
private static boolean recIsIncreasing(Stack<Integer> st, Integer
   pre){
        if(st.empty())
                return true;
        Integer X = st.pop();
        if (X < pre){
                st.push(X);
                return false;
        }
        boolean result = recIsIncreasing(st, X);
```
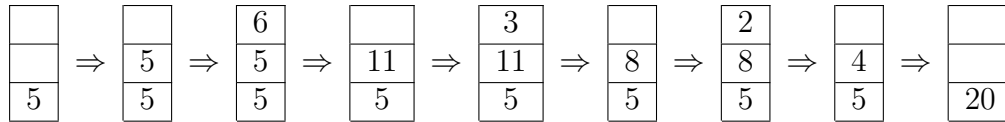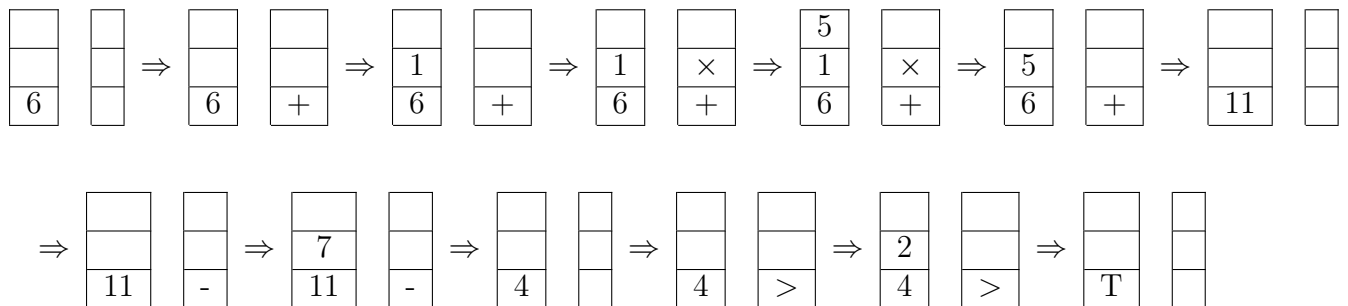
```
        st.push(X);
        return result;
}
```

2. Answer:

		6		3		2		
	5	5	11	11	8	8	4	
5	5	5	5	5	5	5	5	20

$$\Rightarrow \quad \Rightarrow \quad \Rightarrow \quad \Rightarrow \quad \Rightarrow \quad \Rightarrow \quad \Rightarrow \quad \Rightarrow$$

3. Answer:

(stack trace diagram)

First row:
$6 \Rightarrow 6, + \Rightarrow 1/6, + \Rightarrow 1/6, \times/+ \Rightarrow 5/1/6, \times/+ \Rightarrow 5/6, + \Rightarrow 11$

Second row:
$\Rightarrow 11, - \Rightarrow 7/11, - \Rightarrow 4 \Rightarrow 4, > \Rightarrow 2/4, > \Rightarrow T$

Question 2   [30 points]

1. Answer:

```
private void printLeaf(BTNode<T> t){
        if(t == null)
                return;
        if((t.left == null) && (t.right == null)){
                System.out.println(t.data);
                return;
        }
        printLeaf (t.left);
        printLeaf (t.right);
}
```

2. Answer:

```
public int maxKey (int k){
        BSTNode<T> p =root;
        while (k != p.key) {
                if(k < p.key)
                        p = p.left;
                else
                        p = p.right;
        }
        while (p.right != null) {
                p = p.right;
        }
        return p.key
}
```

Question 3   [35 points]

1. Preorder: 5, Inorder 2, Postorder: 4

2. 1: e, 2: c, 3: d

3. 1: e, 2: c, 3: d