

CSC 212 Midterm 2 solution - Fall 2017

College of Computer and Information Sciences, King Saud University

Exam Duration: 90 Minutes

30/11/2017

Question 1 [30 points]

1. Answer[12]

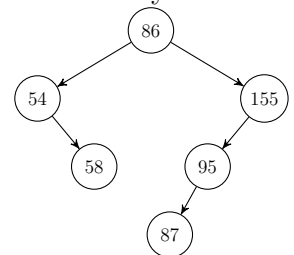
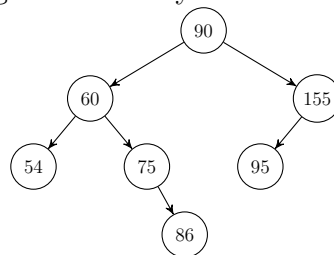
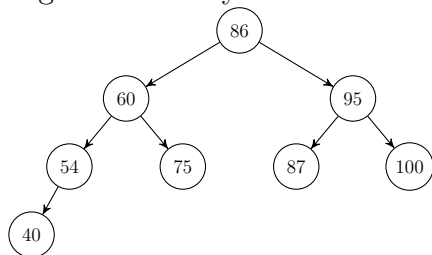
				3					2		1				
	4		7	7	21		4	4	8	8	7				
8	8	2	2	2	2	-19	-19	-19	-19	-19	-19	-12			
/						*	-	*				-	+		

2. Answer[6]

Preorder	Inorder	Postorder
2 F R D I S M L A O H	4 D I R M S L F A H O	1 I D M L S R H O A F

3. Answer[12]

Figure 1: Binary Search Tree a. Figure 2: Binary Search Tree b. Figure 3: Binary Search Tree. c



Question 2 [35 points]

(1) [18]

```
public static <T> void insert (Stack<T> st1, Stack<T> st2, int n)
{
    int count =0;
    T e;
    Stack<T> temp= new LinkedStack<T> ();

    for(int i=1;i<=n;i++)
        temp.push(st2.pop());

    while(!st1.empty())
    {
        temp.push(st1.pop());
        count++;
    }
    while (!temp.empty())
    {
        e=temp.pop();
        if(count-->0)
            st1.push(e);
        st2.push(e);
    }
}
```

(2) [17]

```
public static <T> boolean insertAtEnds(BT<T> bt, T e, Relative rel){
    boolean left=false, right=false;

    bt.find(Relative.Root);
        while (bt.find(Relative.LeftChild));

    left=bt.insert(rel,e);

    bt.find(Relative.Root);
        while (t.find(Relative.RightChild));

    right=bt.insert(rel,e);

    return left && right;
}
```

Question 3 [35 points]

(1) [17]

```
private T findMin(BTNode<T> t) {
    T min = t.data;
    if (t.left != null) {
        T e = findMin(t.left);
        if (min.compareTo(e) > 0) {
            min = e;
        }
    }
    if (t.right != null) {
        T e = findMin(t.right);
        if (min.compareTo(e) > 0) {
            min = e;
        }
    }
    return min;
}
```

(2) [18]

```
private void swapMaxMin(int k)
{
    BSTNode<T> p = root;           //find key
    while (p != null && p.key!=k)
    {
        if (k < p.key)
            p = p.left;
        else if (k > p.key)
            p = p.right;
    }

    if(p==null) return; //k not exist

    if(p.left==null || p.right== null) return; //less than two children

    //find max on left subtree
    BSTNode<T> maxLeft= p.left;
    while (maxLeft.right!=null)
        maxLeft=maxLeft.right;

    //find min on right subtree
    BSTNode<T> minRight= p.right;
    while (minRight.left!=null)
        minRight=minRight.left;

    //Swap
    T temp=minRight.data;
    minRight.data= maxLeft.data;
    maxLeft.data=temp;
}
```