

CSC 212 Midterm 2 - Spring 2014

College of Computer and Information Sciences, King Saud University
Exam Duration: 2 Hours

04/05/2014

Question 1 [25 points]

1. Write the static method *peek* (user of Stack ADT), that takes a stack *s*, and returns the top element. The stack elements should not change after calling this method. The method signature is: *public static <T> T peek(Stack <T> s)*.
2. Write the static method *print* (user of Stack ADT), that takes a stack *s* containing data of type *String*, and prints the elements inside it from top to bottom. The method signature is: *public static void print(Stack<String> s)*.
3. Trace the evaluation of the following postfix expression: $4\ 2\ * \ 7\ + \ 5\ -$

.....

Question 2 [25 points]

1. Give the *preorder*, *inorder* and *postorder* traversals of the binary tree shown in Figure 1.

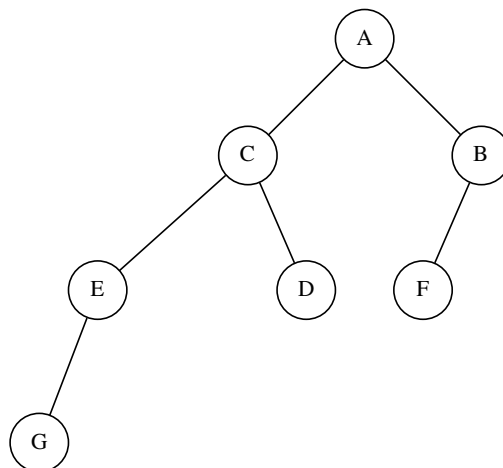


Figure 1: A binary tree

2. The method *func* shown below is a member of the class *BT* (Binary Tree). Give the list of visited nodes (popped from the stack) for the call *func("B")* on the tree shown in Figure 1. List the nodes in order of visit.

```

public boolean func(T e){

    if(root == null)
        return false;

    Stack<BTNode<T>> st= new Stack<BTNode<T>>();
    st.push(root);

    while( !st.empty() ){

        BTNode<T> p= st.pop();

        if(e.equals(p.data))
            return true;

        if(p.right != null)
            st.push(p.right);

        if(p.left != null)
            st.push(p.left);
    }
    return false;
}

```

.....

Question 3 [25 points]

1. Insert the following keys into an empty Binary Search Tree (BST): 21, 13, 7, 20, 34, 19, 90, 8, 13.
2. Remove the following keys from the final tree in part 1: 20, 34, 21.
3. If we wish to print the keys in increasing order, then which traversal method should we use?

.....

Question 4 [25 points]

1. Write the method *range* member of the class *BST* that returns the range of the binary search tree, which is defined as the difference between the maximum key and the minimum key. Assume that the tree is not empty. The method signature is: *public int range()*.

Example 4.1. *The range of the BST shown in Figure 2 is $87-5= 82$.*

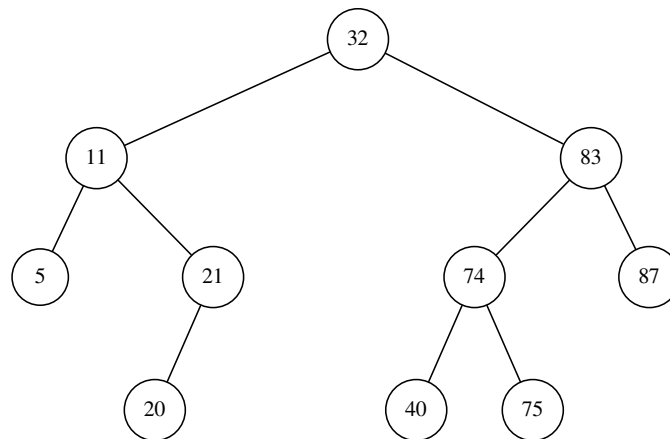


Figure 2: A binary search tree

2. Write the **recursive** method *nbTwoChildren*, a private member method of the class *BT* (binary tree) that takes as input a node *t* and returns the number of nodes in the subtree rooted at *t* having exactly two children. The method signature is: *private int nbTwoChildren(BTNode <T> t)*.

Example 4.2. In the tree shown in Figure 1, the call to *nbTwoChildren* with the node containing data *C* as parameter returns 1, whereas the call with the node containing data *A* as parameter returns 2.

.....

A ADT Stack Specification

- Push (Type *e*): **requires:** Stack *S* is not full. **input:** Type *e*. **results:** Element *e* is added to the stack as its most recently added elements. **output:** none.
- Pop (Type *e*): **requires:** Stack *S* is not empty. **input:** none. **results:** the most recently arrived element in *S* is removed and its value assigned to *e*. **output:** Type *e*.
- Empty (boolean flag): **input:** none. **results:** If Stack *S* is empty then flag is true, otherwise false. **output:** flag.
- Full (boolean flag): **requires:** none. **input:** none. **results:** If *S* is full then Full is true, otherwise Full is false. **output:** flag.