Student ID:
Name:
Signature:

COMP 203 DATA STRUCTURES AND ALGORITHMS

QUIZ 2 Date: 09.01.2024 10:30 am

Total: 100 points

1. Stack and SLL (Total: 55 points)

a. Write a pseudocode for a Node class, SinglyLinkedList class and SSLStack and their constructors that you need in your implementation. Node class has "char data and Node next" variables. **(5x3=15pt)**

b. Write a pseudocode for a method with the name "Node push(char addedValue)" to add a node with the value *addedValue* in the stack. This function returns the head node of the singly linked list. This will be implemented in SSLStack class. **(10pt)**

> **Example:** This is the stack after the operations of push(A), push(B), push(C), push(D).
>
> A->B->C->D->null (top of the stack is D here, A is the head node here.)
>
> After push (M) it will be;
>
> A->B->C->D->M->null returns the head node of the stack.

c. Explain the Big-O complexity of your code for "Node push(char addedValue)". **(5pt)**

d. Write a pseudocode for a method with the name "char pop()" to remove element from the stack. This function returns the deleted node value. **(10pt)**

> **Example:** This is the stack after the operations of push(A), push(B), push(C), push(D).
>
> A->B->C->D->null (top is D here)
>
> After pop(), it will be;
>
> A->B->C->null and returns D.

e. Explain the Big-O complexity of your code for "char pop()". **(5pt)**

f. Write a pseudocode for a test case in main to see if your push and pop functions work. You may assume that you have printStack() function. **(10pt)**

Student ID:
Name:
Signature:

2. Binary Search Tree and Node (Total: 45pt)

Write a pseudocode for Binary Search Tree abstract data structure using Node data structure. Implement the following classes and methods:

1. Write a pseudocode for Node class (to use in Binary Seach tree) and its constructor with the parameter “int data”. **(5pt)**
2. Write a pseudocode for Integer type BinarySearchTree class with its class variables and a constructor with no parameter. **(5pt)**
3. Write a pseudocode for insert(Node root, int value) method that adds a new node with the given *value* to the given tree **recursively. (10pt)** What is the time complexity of this function?**(2pt)**
4. Write a pseudocode for inOrderTraversal(Node root) method that prints the given tree with in-order traversal **recursively**. **(10pt)** What is the time complexity of this function? **(2pt)**
5. Write a pseudocode to test your methods in the main by creating the following binary search tree. (Creating the given tree: **5 pt**, testing 2 methods: **6 pt**)