National University of Computer and Emerging Sciences

Laboratory Manual

for

Data Structures Lab

Course Instructor	Ms. Syeda Tayyaba
Lab Instructor(s)	Samia Akhter Sonia Anum
Section	BDS-3B
Date	3-October-2022
Semester	Fall 2022

Department of Computer Science

FAST-NU, Lahore, Pakistan

Objectives:

In this lab, students will practice:

- Stack Implementation using a Singly Linked List
- Queue Implementation using a Fix-sized Circular Linked List

Question No. 01:

a) Implement a stack using a singly linked list. The required member methods are:

1. **int size()**: returns the count of total element stored in the stack.
2. **bool isEmpty()**: returns true if the stack is empty else false.
3. **bool top (&)**: returns, but does not delete, the topmost element from the stack via the parameter passed by reference. It returns false via a return statement if there is no element in the stack, else it returns true and assigns the top most element to the parameter passed by reference.
4. **void pop()**: deletes the top most element from the stack. If there is no element, return some error.

b) Now implement the following problem of stack using link list

- Sort stack using temporary stack
- Delete middle element of stack
- Check stacks IsPalindrome?
- Find two element in stack whose sum is K

c) Calculate time complexity of all functions mentioned above.

Question No. 02:

Implement a queue using a fixed-sized circular linked list. The required member methods are:

1. **int size()** : returns the count of total element stored in the queue.
2. **bool isEmpty()**: returns true if the queue is empty else false.
3. **bool front(&)**: returns, but does not delete, the front element from the queue via the parameter passed by reference. It returns false via a return statement if there is no element in the queue, else it returns true and assigns the front element of the queue to the parameter passed by reference.
4. **void dequeue()**: deletes the front element from the queue. If there is no element, return some error.
5. **void enqueue(const& e)**: inserts the element “e” at the back of the queue if there is some space available. Otherwise it returns some error.

Calculate time complexity of all functions mentioned above.