National University of Computer & Emerging Sciences (NUCES) Islamabad,
Department of Computer Science

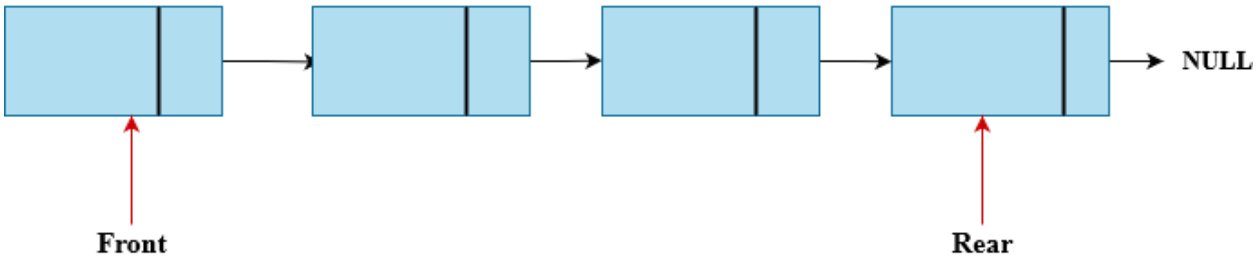
DATA STRUCTURES – FALL 2021

LAB 07

Learning Outcomes

In this laboratory, you will implement the Queue using linked list.

This lab requires you to implement your own Class Queue, its operations and using those operations to perform different tasks. Keep in mind that this Queue class is built upon linked list so first you have to create Node which is the building block of linked list.



TASK 1

- Implement Class Node having its data members, getters and setters.
- Implement Class Queue, its data members, getters, setters and operation listed below.

Queue ()
A non-parameterized constructor that creates an empty queue. Where should the front and rear of an empty queue point to?

enqueue ()
Inserts the element at the rear of queue.

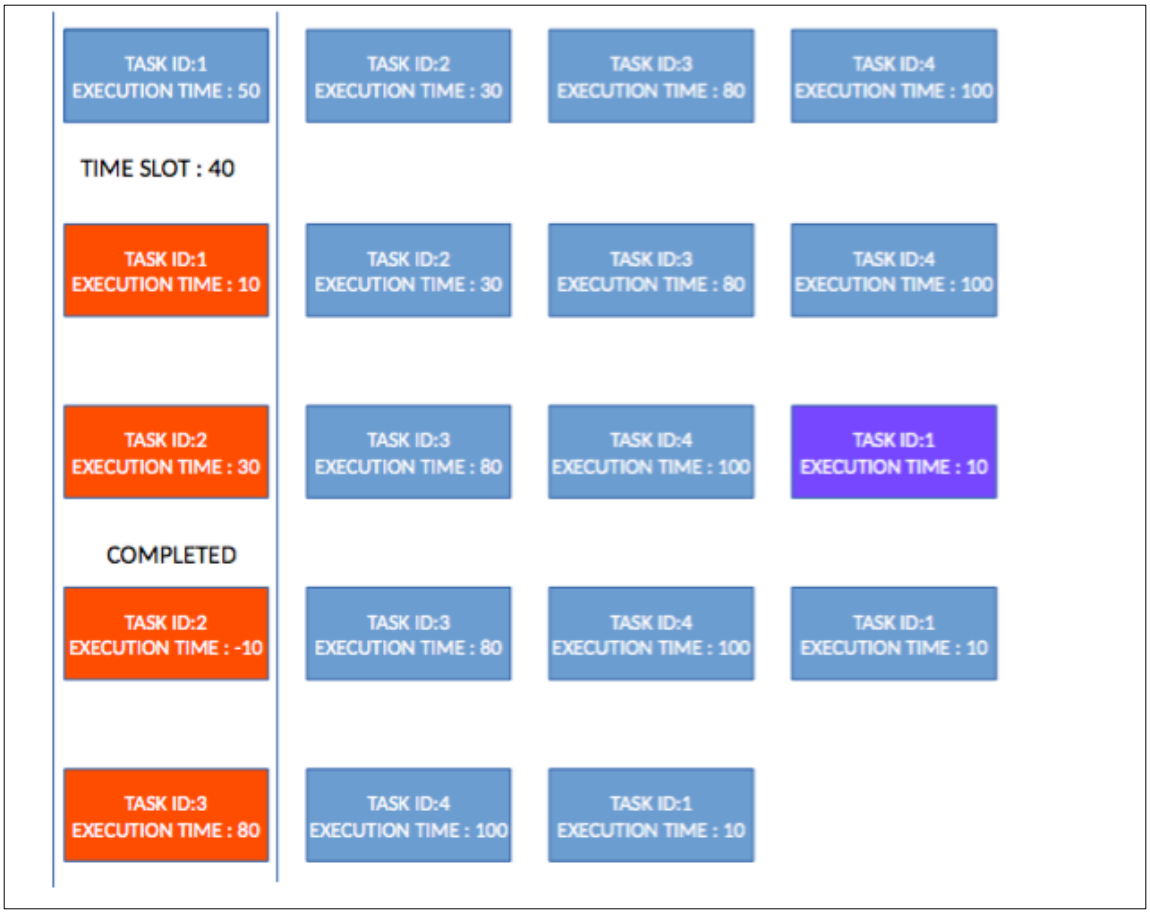
dequeue ()
Removes the element from the front of queue.

Front()
Returns the value of the element at front of the queue.

isEmpty ()
Returns True if queue is empty else returns False.

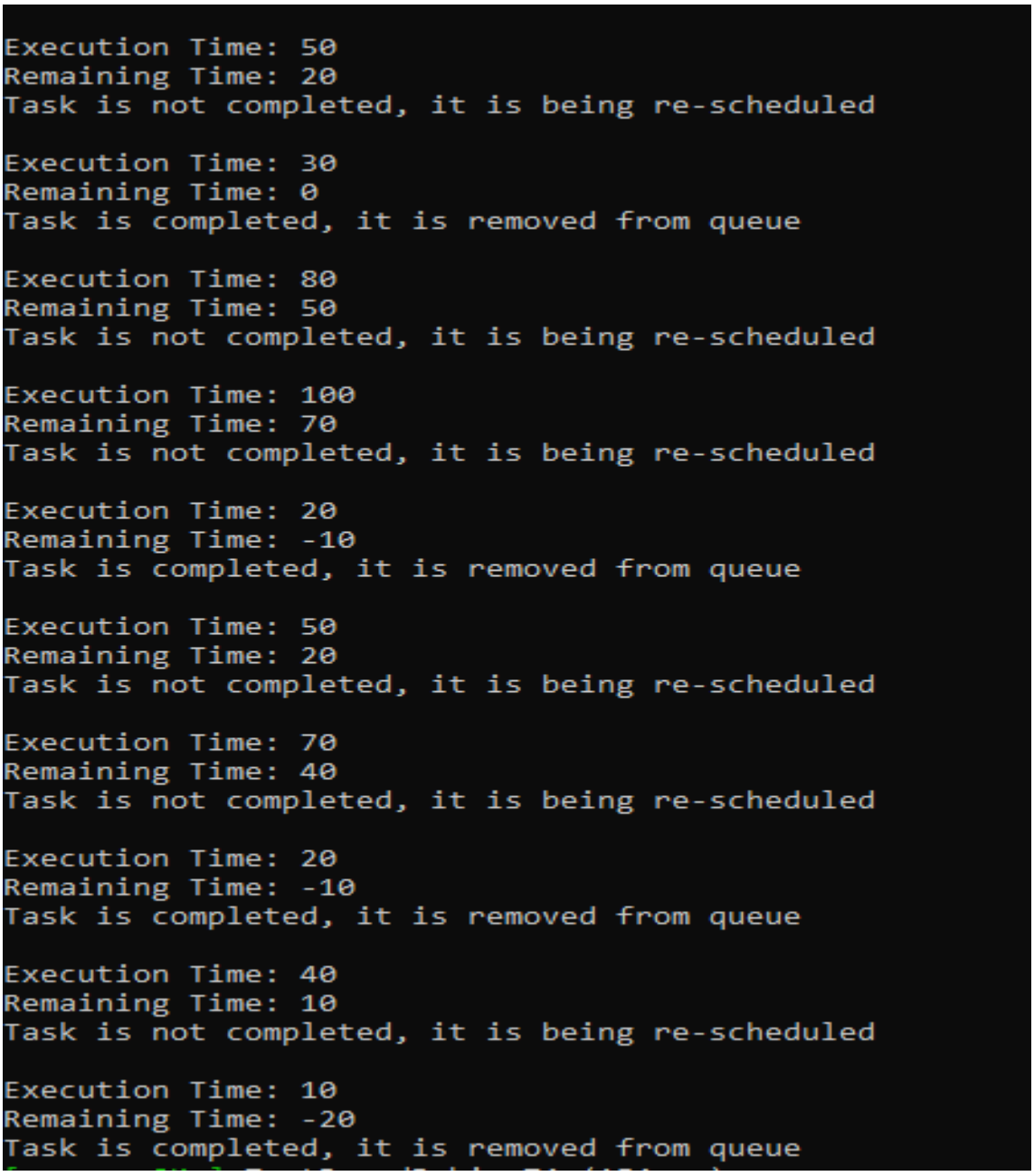
TASK 2

Round robin is a scheduling algorithm that an operating system uses to time share computational resources of a processor between tasks. Each task is given a specific time slot **(Quantum)** to execute on a processor (CPU Time), once this time slot expires and the task is not yet completed, it is preempted **(dequeue)** and added to the back of the queue **(enqueue)** with its **Remaining Execution Time**. Then the next task (**front)** in the queue is selected and this processes continues until all tasks have finished execution.



Your task is to simulate this process using a queue. Implement a function **roundRobin()** that takes input a **Queue of execution time and a time quantum**. Then it simulates the process of task execution until the queue gets empty.

Example of output is displayed below



TASK 3

In this Exercise, you have to take a single string as input. Using this input string, you have to create multiple queues in which each queue will comprise of separate word appeared in input string. At the end, you will again concatenate all queues to a single queue and return it to user.

Example:

String = “Data Structure and

Algorithms”

Q1 = D → a → t → a

Q2 = S → t → r → u → c → t → u → r

→e

Q3 = a → n → d

Q4 = A → l → g → o

At the end concatenate all queues.

Q1→Q2→Q3→Q4