150015-Semester II - 5781

Data Structures I

**Homework Assignment #4**

**Question 1**

Explain how to implement two stacks using one array A[1..n] in such a way that there will only be an overflow when the number of elements in both stacks together is greater than n.

**Question 2**

1. Implement efficiently a stack using two queues. Determine the run-time complexity of the basic stack functions that you wrote.
2. Can a stack be implemented using only one queue and only O(1) additional memory? If yes, implement and determine the run-time complexity of the functions, if not, explain why.

**Question 3**

The school secretariat has **frenzied**, **urgent** and **regular** tasks. Frenzied tasks are carried out first, followed by urgent tasks and then ordinary tasks.

The secretariat has a computerized system, Manager.

**The assignment of tasks to the Manager** is done according to the following rules:

- A new frenzied task will be inserted before all the frenzied, urgent, and regular tasks currently in Manager

- A new urgent task will be inserted after all the frenzied tasks and before the urgent and regular tasks that are currently in Manager

- A new regular task will be inserted after all the frenzied, urgent, and regular tasks that are currently in Manager.

**Task Executing from Manager** is done in the order created in Manager.

The Task class has two properties:

content - a string that is a task description,

code - which is an integer representing the type of task:

1 represents a frenzied task;

2 represents an urgent task;

3 Represents a regular task.

A. Suggest a data structure that is appropriate for the implementation of Manager so that the complexity of the input operation to Manager and the removal of an operation from Manager is O (1). The structure has to consist of a combination of **a stack(s)/queue(s)** and auxiliary variables as needed.

B. Write a function in the pseudo-code that accepts a task and inserts it into the Manager according to the rules described at the beginning of the question.

C. Write an algorithm in the pseudo-code for an action that removes out the next task to execute and returns it. If there is no Task in Manager, the action returns null.