

COSC 2P03 Advanced Data Structures: Assignment 1

Instructor: Yifeng Li^{*1}

¹Department of Computer Science, Brock University

September 15, 2022

1 Waiting Room Management in a Vaccine Shot Clinic

Suppose you are a software developer to help a COVID-19 vaccine shot clinic develop a queue management system for the waiting room. The clinic opens at 9am. Priority queue should be used to manage patients in the waiting room. The priority of a patient is determined by his/her arrival time, age, occupation, and health condition. It takes 15 minutes to get the vaccine done for each patient. In this assignment, you are expected to finish the following tasks.

2 Your Tasks (Total: 8 marks)

You should define classes named **Patient**, **Node** (for the doubly-linked list), **Clinic**, and **WaitQueue** with the following requirements (feel free to define extra variables, classes, and methods if needed).

1. Define a method named **readData** under the **Clinic** class to load all the provided information in the text file to the class. Each item should be stored in an array of **Patient** instances. The array (name it as **patients**) here should be a data attribute of the **Clinic** class. (2 marks)
2. The **WaitQueue** class should have methods **removeMax**, **insert**. This class has a doubly-linked list as its data attribute for the queue. Each node of the list should contain a **Patient** instance, and other necessary information. An instance (named **wq**) of this **WaitQueue** class is a data attribute under **Clinic**. (2.5 marks)

Priority of a patient is decided by the following rules:

- Everyone has a base priority 0 when added to the queue.
 - For a patient with age ≥ 60 , 1 is added to his/her priority.
 - For a patient who is a teacher, nurse, or care giver, 1 is added to his/her priority.
 - For a patient with pregnancy, cancer, diabetes, asthma, primary immune deficiency, or cardiovascular disease health condition, 1 is added to his/her priority.
 - From the above rules, a patient can have accumulated priority greater than 1. Tie of priority is broken using time of arrival.
3. The **Clinic** class should have a method named **monitor** to start running the clinic at 9am. In this function, the **data** attribute is checked every minute such that a patient can be added to the priority queue **wq**. Meanwhile, function **monitor** should also remove a patient from the queue for vaccine shot once the previous one is finished. Print out the name of a patient when called for vaccine, and the corresponding time. (2 marks)
 4. In the **main** function, instance of the **Clinic** should be created. Methods of this instance should be called properly. (0.5 mark)
 5. Your code should be well commented. (1 mark)

*E-mail address: yli2@brocku.ca

3 Hints

You may consider defining a `Timer` class for convenient operations involving time. For example, the following structure may help.

- Class name: `Timer`.
- Data attributes/variables:
 - `int hour`;
 - `int minute`; Reset for every 60 minutes.
 - `int vxtime`; Reset for every 15 minutes.
- Methods:
 - The construction/initialization function of this class can set the start time based on the arrival time of the first patient. It can set `vxtime` to 1.
 - `increase()`: this function increases `minute` by 1. If `minute==60`, increase `hour` by 1 and reset `minute` to 0. Also increase `vxtime` by 1; if `vxtime==15`, reset it to 0 and return 1 or `true` from this function to indicate whether it is time to remove a patient from the queue.
 - `compare(timeofArrival)`: this function compares string `str(hour)+":"+str(minute)` with the `timeofArrival` obtained from the array of patients. Basically, here you are comparing two strings. Note: when `hour==9` and `minute==4` for instance, you will need to convert it to "09:04" rather than "9:4". This function return -1 (less than), 0 (equal), or 1 (greater than).
- An instance of this class can be created as a data attribute of the `Clinic` class and apply the above functions in the monitor function within a for loop.

4 Submission

- Your source code.
- A PDF printout of your source code.
- The text file with result required in Task 3.
- Compress the above files in a zipped folder named `COSC2P03_A1_Firstname_Lastname_StudentNumber.zip` and submit it through Brightspace before indicated due time.
- Late submissions will not be accepted.

5 Academic Integrity

This assignment should be tackled individually. Outsourcing or teamwork is not allowed. Violation of this requirements will be seriously processed in accordance with university policies.