ITU Computer Engineering Department
BLG 223E Data Structures, Spring 2022
Homework #2
Due June 1, 2022 11:59pm

Problem Definition

In this homework, we ask you to create a software-based solution for the check-in procedure of an airline company. Different than the regular queues as we experience during the check-in procedure, the company aims to revolutionize it by prioritizing some of their customers. The solution will help the airline company to assign the passengers to the available check-in agents so that prioritized passengers would not need to wait in queue for a long duration. For this reason, the check-in procedure is performed by classifying and prioritizing the passengers based on the following tier statuses:

- **Platinum Tier**: The best tier in the airline company. A passenger is assigned to Platinum Tier if he has flown for more than 80,000 miles.
- **Gold Tier**: The second-best tier in the airline company. Can be assigned if a passenger had flown between 40,000 and 80,000 miles.
- **Silver Tier**: The third best tier in the airline company. Passengers need to fly for more than 25,000 and less than 40,000 miles to acquire that tier.
- **Classic Tier**: The basic tier which is applied to all passengers.

Apart from these tier statuses, the software-based solution should give the highest possible priority to the customers who are **disabled** or **pregnant**. Other than that, your implementation should consider the following factors:

- The passenger with the highest priority should be examined first (disabled/pregnant > platinum > gold > silver > classic).
- In case two passengers are in the same tier, the passenger who has flown further should be selected first.
- If more than one check-in agent is available instantly, the passenger is matched to a check-in agent with a lower ID.
- When a passenger is being served, the agent should complete its service with the passenger even though another passenger with a higher priority enters to the queue.
- Once a passenger is assigned to an agent, the agent immediately begins assisting the passenger and becomes unavailable until the end of the check-in.
- The maximum time allocation for the **completion** of the check-in procedure is limited to 120 minutes for domestic flights and 180 minutes for international flights. If the scheduler exceeds the timing constraints given the number of check-in agents, the program should issue an error and state how long the check-in procedure lasted.
- All data entries are composed of integers.

Data Format & Brief Explanation

The data is stored in a text file. The first line of the file contains the number of passengers along with the number of available check-in agents and whether the flight is domestic (0) or international (1). The following lines contain the passenger ID, miles flown, is pregnant and is disabled, enqueue time, and time for completing the check-in procedure. For instance, for the passenger with ID 1, the passenger has flown for 70,000 miles, is neither disabled nor pregnant, gets in to the queue at minute 1, and his check-in procedure takes 10 minutes to complete. You can assume that the passengers are sorted based on their queue entry times and are limited to 300 passengers.

You can find sample data, input.txt, and its associated schedule in Fig. 1. For this sample, three check-in agents are present to complete the check-in procedure of eight passengers on a domestic flight. The first passenger with ID 1 is enqueued at minute 1, and his check-in is completed at minute 11. Hence, the passenger with ID 1 is handled by agent 1. Then, at the third minute, three passengers are enqueued, and the passenger with ID 3 is assigned to the second agent since he is disabled. After that, we need to make a selection among passengers with IDs 2 and 4, where the former is a platinum tier, and the latter is a silver tier holder. Since ID 2 has priority over ID 4, the passenger with ID 2 is handled by agent 3. Up until minute 11, passenger IDs 5, 6, and 7 are enqueued. As ID 5 is a platinum tier holder, ID 6 is a classic tier holder, and ID 7 is pregnant, the queue should initially contain the passenger IDs in the order of (7,5,4,6). Also, agents 1 and 2 become available as a result of handling passenger IDs 1 and 3, and they ask for the passengers from the priority queue. As a result of that, passengers with IDs 7 and 5 are popped from the priority queue and assigned to agents 1 and 2, respectively. Due to agents 2 and 3 becoming available at minute 18, passenger IDs 4 and 6 are then handled by them in parallel. It is also important to keep in mind that the passenger with ID 4 is assigned to agent 2, since it has got more miles than passenger ID 6. Lastly, the last remaining passenger (with ID 8) is picked by agent 1 since it is the only free agent. As this is a domestic flight, and the check-in procedure has been completed in less than 120 minutes, we need to have the following output in Code Listing 2.

For the case of input2.txt, however, as this is an international flight and the second passenger arrives at minute 178, and his check-in procedure takes 3 minutes, the code should output as in Code Listing 4 since the check-in procedure has taken longer than 180 minutes.

Code Listing 1: input.txt

```
8 3 0
1 70000 0 0 1 10
3 500 1 0 3 8
2 80001 0 0 3 15
4 30000 0 0 3 20
5 200000 0 0 5 7
6 20000 0 0 6 8
7 0 0 1 11 12
8 50000 0 0 20 7
```

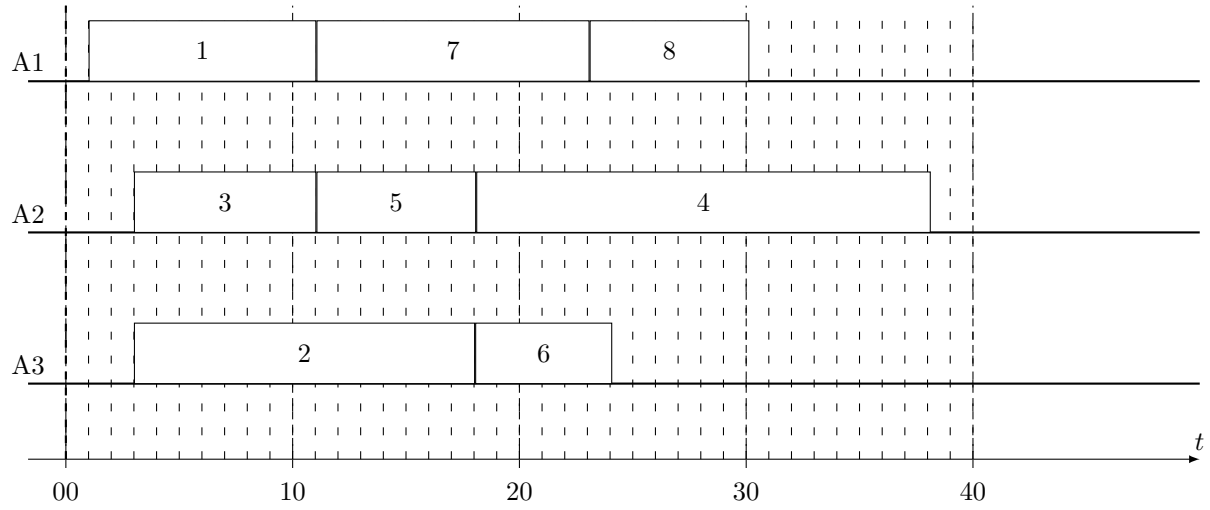

Figure 1: Scheduling diagram for the passenger flow in input.txt.

Code Listing 2: output.txt

```
test@d9d38dbf70fc:~/hostvolume/homework2/$ ./bin/homework2 cases/input.txt

Agent 1 takes passenger 1 at minute 1
Agent 2 takes passenger 3 at minute 3
Agent 3 takes passenger 2 at minute 3
Agent 1 takes passenger 7 at minute 11
Agent 2 takes passenger 5 at minute 11
Agent 2 takes passenger 4 at minute 18
Agent 3 takes passenger 6 at minute 18
Agent 1 takes passenger 8 at minute 23

Check-in is complete on time in 38 minutes.
```

Code Listing 3: input2.txt

```
2 1 1
1 10000 0 1 10 20
2 5000 0 0 178 3
```

Code Listing 4: output2.txt

```
test@d9d38dbf70fc:~/hostvolume/homework2/$ ./bin/homework2 cases/input2.txt

Check-in failed to be completed on time, it took 181 minutes.
```

Submission Rules

- Don't miss writing your ID and name to the top of your each of the files including main.cpp.
- Make sure that your submission is working in our Docker container and passing the basic test cases of Calico. An empty template project file has been shared to ease that.
- We expect you to write your own test cases for making sure that your code is handling edge cases.
- Use comments wherever necessary in your code to for a better understanding. Also, make sure that your code is not eye-bleeding, use clang-format where appropriate.

- We expect you to implement the priority queue with heaps.
- **You cannot use the STL library utilities for your heap implementation.**
- Your program will be checked by using **Calico(https://bitbucket.org/uyar/calico)** automatic checker.
- Do not share any code or text that can be submitted as a part of an assignment (discussing ideas is okay).
- Only electronic submissions through Ninova will be accepted no later than deadline.
- You may discuss the problems at an abstract level with your classmates, but you should not **share or copy code** from your classmates or from the Internet. You should submit your **own, individual** homework.
- Academic dishonesty, including cheating, plagiarism, and direct copying, is unacceptable.
- Note that **YOUR CODES WILL BE CHECKED WITH THE PLAGIARISM TOOLS!**
- If you have any question about the recitation, you cand send e-mail to Caner Özer (ozerc@itu.edu.tr).