Assignment 2

CmpE 250, Data Structures and Algorithms, Fall 2018

Instructor: A. T. Cemgil
TA: Özlem Şimşek
SA: Burak Çetin, M. Emin AYAR

Due: 12/11/2018, 23:59 Sharp

1 Description

As a data scientist working at the CMPE Airlines, it is your job to always try new ideas in order to increase the efficiency. After working at the CMPE Airlines for a long time you realize that the passengers are really annoyed by the long waiting times for luggage hand-in and security checks. Some of the passengers are even missing their flights because of the long waiting times at the queues. You come up with a few ideas to reduce these waiting times, but before proposing your ideas to the management board you decide to actually test your design 'in silico' if the new proposals will work or not on the data you have.

In the current system there is a first come first serve queue before the L luggage hand-in counters. Whenever there is a free counter , first person in the queue goes to the free counter and gives their luggage. After handing in luggage, they go into a second queue for a security check. This is also a first come first serve queue and there are S security counters. Similarly, whenever a security counter is free, first person in the queue goes to that counter for the security check. This is visualized in the Figure 1.

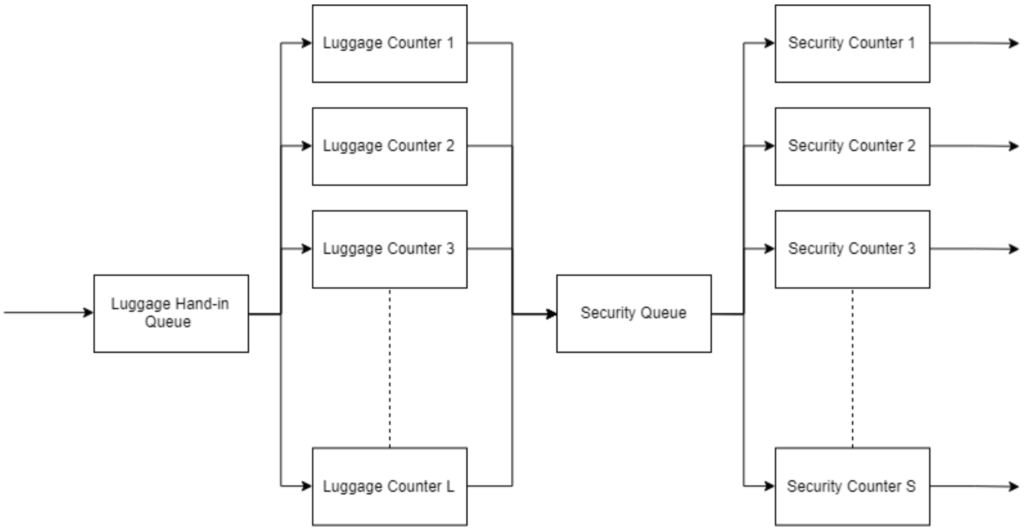

Figure 1: Initial queue structure

2 Changes To Implement

You have 3 different ideas to make these queues more efficient. You need to simulate the system and collect data for 8 possible scenarios where you implement each of these 3 ideas or not. The first data you need to collect is the average time between the arrival of a passenger to the terminal and her departure from the security check. Second data you need to collect is the number of passengers who are late to their flights.

1. Instead of making the queues first come, first serve you can make them first-to-fly, first serve. The person who is flight is earliest will move to the first place of the line. This applies to both luggage hand-in queue and security queue. You are hoping that this would reduce the number of people who are late to their flights.

2. CMPE Airlines has a special VIP Passenger program. Each passenger is either in this program or not. As a firm you are trusting these customers. You decide that you can make these VIP passengers skip the security queue entirely. You are hoping that this would have an impact on other people's waiting time as well since VIP passengers won't keep the security counters busy.

3. As some people do not have a luggage to hand-in, you can implement an online ticketing system. As a result the people who does not carry a luggage can skip the first queue entirely and go in line for the security

check immediately. You are hoping that this would have an impact on other people's waiting time as well, since passengers without a luggage won't keep the luggage hand-in counters busy.

You are asked to simulate and collect the statistics for each of the 8 cases in the order given by this table.

	First-to-Fly First Serve	**No Security Check for VIP**	**Using Online Ticketing**
Case 1:	✗	✗	✗
Case 2:	✓	✗	✗
Case 3:	✗	✓	✗
Case 4:	✓	✓	✗
Case 5:	✗	✗	✓
Case 6:	✓	✗	✓
Case 7:	✗	✓	✓
Case 8:	✓	✓	✓

3 Input/Output Format

For this project, time will be measured in minutes. Start of the day will be at 0 minutes. All of the given times in the input will be integers, indicating how many minutes from the start of the day. Output is also in minutes. As a result, you do not need to perform time conversions in your code.

3.1 Input Format

The first line of the input file holds 3 integers, P, L and S, number of passengers, number of check-in counters and number of security gates.

In the following P lines, the passenger data is given, one passenger per line. Each line holds 4 integers and 2 characters, the time at which the passenger will arrive at the airport, the time at which the passenger has to board her flight, the time it will take for her to hand in her luggage, the time her security control will take and the VIP membership and luggage state of that passenger. If the passenger is a VIP member the first character will be a 'V', otherwise it will be a 'N'. Similarly, if the passenger has a luggage to hand in, the second character will be a 'L', otherwise it will be a 'N'.

3.2 Output Format

Your program should calculate and print out the two statistics for each case given in the table in eight separate lines.

- Average waiting time for passengers.
- Number of passengers who get out of the procedure after their boarding time.

3.3 Example Input

```
8 2 2
1 10 6 3 N L
3 16 3 6 N N
4 11 2 3 V N
5 15 2 5 N N
6 9 2 1 V N
16 19 1 1 N L
17 26 3 5 N L
18 22 1 3 N L
```

- There are 8 passengers, 2 luggage counters and 2 security counters.
- The 1st passenger comes to the terminal at 1 minutes. Her flight is at 10 minutes, she will spend 6 minutes on the luggage counter and she will spend 3 minutes on the security counter. She is not a VIP passenger and she has a luggage (She has to go to the luggage counter even if online ticketing is implemented.).
- The 5th passenger comes to the terminal at 6 minutes. Her flight is at 9 minutes, she will spend 2 minutes on the luggage counter and she will spend 1 minutes on the security counter. She is a VIP passenger and has no luggage. As a result, if online ticketing is implemented she will skip the luggage counters and go straight into the security queue. If both online ticketing and no security check for VIP passengers is implemented she will directly board on her flight, waiting for 0 minutes in the procedure.

3.4 Example Output

```
7.625 3
7.375 3
6.25 1
6.125 0
5.75 2
5.5 1
4.5 1
4.5 1
```

- 8 lines for each separate simulation cases.
- As an example, compare lines 3 and 4. These are the cases where VIP security skip is implemented. These lines differ by the queuing rules. You can see that making the queues first-to-fly, first serve we are effectively reducing the average waiting time and preventing a passenger from being late to their flight. You can see a similar trend on lines 5 and 6, where online ticketing is implemented.
- If you compare the lines 4 and 8, you can also see that implementing all the options can cause more people to miss their flights. This happens because the passengers 2 and 4 skip the luggage queue and fill both security counters. They end up blocking passenger 1 from proceeding with the security check and cause her to miss her flight.

4 Grading

Grading of this project is based on the success of your code in test cases. Your score will be the sum of collected points from each test case. Each test case will have equal weight. Maximum score is 100.

For each test case total points will be divided to 8 equal parts for each of the simulation cases. **You can still get a partial score by successfully implementing a subset of these 8 simulation cases. If you decide to do this, make sure that you print "0 0" for the cases you did not calculate.**

5 Implementation Details

1. Variable limits are as follows:
   - $1 \leq P \leq 5{,}000$
   - $1 \leq L, S \leq 20$
   - $1 \leq$ All the times given in minutes $\leq 50{,}000$
2. Execution time limit is 1 seconds. **If your code runs more than 1 seconds on a test case you will get 0 points from that test case.**
3. Even if a passenger is already late to their flight they will still **stay in the queues**. Do not put people out of the procedure based on this.
4. You can safely assume that no two passengers will arrive at the terminal at the same time or leave luggage counters at the same time.
5. If two different events, say a passenger leaving a luggage counter and another passenger leaving a security counter, is happening at the same time, the person leaving the luggage counter can occupy the security counter that was emptied by the other passenger in that instant. Same thing applies for a passenger arriving at the terminal and another passenger leaving a luggage counter at the same time. **Hint:** You can ensure this by making sure that you simulate simultaneous events in this order:
   (a) A passenger leaving a security counter.
   (b) A passenger leaving a luggage counter.
   (c) A passenger arriving at the terminal.
6. Your program will be compiled with **cmake CMakeLists.txt && make** command and executed with **./project2 inputFile outputFile** command.

6 Warnings

1. Make sure an executable is created after executing **cmake CMakeLists.txt && make** commands. Otherwise, no executable will be created and your code will fail in grading.