Simulation

Olga Novichkova
CS2235

Idaho State University

I. Introduction

The project consists of creating a program which will help the Transportation Security Administration at Airports effectively manage their operational costs while having great customer service at the same time. The problem has been that at certain large and busy airports, there are not enough security lines open leaving people angry and frustrated for having to wait too long. While at other smaller airports the problem is that there are too many lines open which is costing the airport money. The goal of writing such a program is to help any airport management decide how many lines to open based on how busy they are and what their limitations on equipment are. The main hypothesis is to find out the average of how long a person would have to wait in line for each number of open lines.

II. Simulation Design

A. *Stacks, Queues, Dequeues*

First, the implementations of the classes linked stack, linked queue, and linked dequeue have to be implemented. The linked stack will be used to represent the lines at the airport.

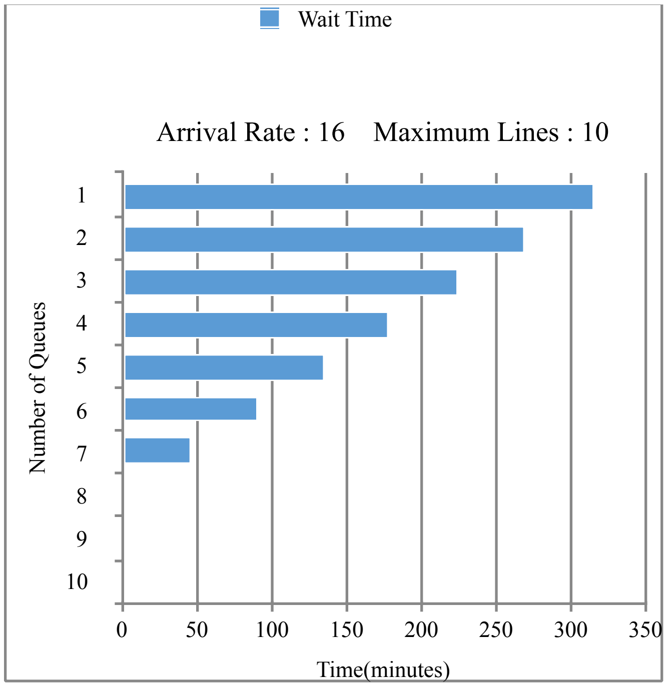
B. *Simulation*

There are two methods, one for calculating the wait time and the other to find the shortest line. The second method returns a linked queue out of the vector of queues which it accepts. A for loop runs through all the queues and checks the size. A temporary variable is used to save and return the shortest queue. In the first method, I create a new empty vector to store all the Queues. Then using a for loop, from one queue to the maximum number of queues, I create and insert the Queue into the vector. I create a variable to hold the number of people exited and a variable to hold the total number of average wait time. Then, another loop to run by the total test time(720 minutes) where all calculations will be made. For every minute the method that generates random number of people entering is called and saved to a variable. An inner for loop runs through every single person entering and adds their time of arrival to the shortest queue which is defined by calling the second method. This loop is then exited and another inner loop created for all the people exiting. It runs through all the queues in the vector and for each queue another for loop runs from 0 to the service rate(2 people).Inside this inner loop a persons entered time is recorded and saved to a variable. That same persons wait time is determined by taking their entered time and subtracting it from the total time and adding 1. This time is then added and saved to the average time variable. The last thing that occurs in this loop is the total number of people increases by 1 for each cycle. The last thing to do is exit all the loops and then take the added up average wait time and divide it by the total number of people exited which results in the goal of this problem; calculate the average wait time when using the input of the user based on their maximum number of queues and arrival rate per minute.
Analysis and Interpretation

III. Experimental Methods

The arrival rate will be determined by the number of people that enter the airport each minute. This will be done over a 12 hour period or 720 minutes. The program will take the input from the user and then for each minute out of the 12 hours generate a random number very close to the provided arrival rate. Then it will take each person, record their time of arrival, and then one by one place them into the line which is shortest at the moment. At the same time, for each minute the service rate stays at a constant of 2 people. In other words, for every minute a certain number of people arrive but only 2 people leave. When each person leaves, their time is recorded, this gives us the total amount of time each person waited.

IV. Results



Arrival Rate : 16 Maximum Lines : 10

Average wait time using 1 Queue	315 minutes
Average wait time using 2 Queues	269 minutes
Average wait time using 3 Queues	225 minutes
Average wait time using 4 Queues	179 minutes
Average wait time using 5 Queues	135 minutes
Average wait time using 6 Queues	90 minutes
Average wait time using 7 Queues	46 minutes
Average wait time using 8 Queues	3 minutes
Average wait time using 9 Queues	1 minute
Average wait time using 10 Queues	1 minute

These are the results for arrival rate of 16 and maximum number of open lines is 10. The more lines that are open the less time people have to wait but the difference between adding one more line makes a big difference. Also, noticing that if the waiting time drops all the way down to just one minute before the maximum number of queues, the rest remaining queues will have to be 1 minute because it is not possible for anyone to wait zero minutes even if there is absolutely no one in the line.

V. Analysis and Interpretation

The results are extremely important and critical for an airport business or any other similar business. For example, if an airport were to input 16 for the average number of people entering per minute and 10 for the maximum number of lines they can possibly open, the results for them would look like the chart/table above. Such results give the airport management an idea and help them choose the best option at the moment. Keeping in mind that they have to provide good customer service where the people wont be angry for having to wait too long but at the same time the airport is not losing money by having too many lines.

VI. Conclusion

Overall, this project could be extremely useful and it would have great impact on businesses by improving their stability and overall performance. The goal was to create a program such that would use arrival rates and maximum number of queues to print out average waiting times in lines. For this program to work, the implementation of a linked queue, linked stack, and linked dequeue were required. Then a simulation method had to be written which used the input from the user, the provided random method and the queue implementation to calculate the average wait times for each number of lines. The key findings are the wait times which guide the user towards making the right decision to work in favor for the users business. In the future, this program could be used to build off of and create more specific or similar concept problems and results.

References

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use "Ref. [3]" or "reference [3]" except at the beginning of a sentence: "Reference [3] was the first ..."

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors' names; do not use "et al.". Papers that have not been published, even if they have been submitted for publication, should be cited as "unpublished" [4]. Papers that have been accepted for publication should be cited as "in press" [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

[1] M. Goodrich, R. Tamassia and M. Goldwasser, *Data structures and algorithms in Java.*

[2] I. Griffith, *PA03*. 2019, pp. 1-4.