Bank Simulation Report

I. Motivation
- The structure of the project interested me, and it is applicable to other aspects of Computer Science
- Some of the other project suggestions triggered my urge to gamble
- I used to work at a bank

II. Problem Statement
- Typically, going to a bank requires waiting in line for a long period of time
- Not everyone wants to or has the time to wait in line
- Using First in First Out and priority queues, this simulation will attempt to answer in what ways can this be optimized, such as adding more windows, or prioritizing certain customers

III. Background Material
- Background material
  - The majority of banks will have at a time of ten teller windows, with the goal of having eighty work units per day.
  - We all know that there is a uniform line when it comes to arriving in a bank, so there will be a uniform distribution of customers over the eight hour work day.
  - The first part of the events is the external events that are not controlled by the bank setting, because there are different times where customers arrive, whether its at the beginning or the end of the work day.
  - The second part is the internal events that are in the banks control, where tellers are either busy or just sitting idle.
  - There are times where customers are just waiting in line, being helped by a teller, or the off chance where they are not being helped at all.

IV. Approach
- We must first look over our priority queue, in which its used for the elements that will have different priorities over other elements. This

illustrates that some customers will have a higher priority than others, allowing the priority queue to remove those who have a higher priority to get help from the tellers.This will show that those who don't have a high priority, will end up having to wait a longer time, compared to those who do have the higher priority.

- Next part is the FIFO(First In, First Out) approach of where it would just be a standard line that we use in our everyday lives. When there is a person who is first in line, they always are the first person to leave that line, followed by the next person replacing them in the first position spot.

V. Experiment set-up

- Number of iterations (days) n = 5000
- Performance metrics
    - Arrival distribution
    - Customers served and not served
    - Customer wait times
    - Add and subtract 1 teller
        - Simulated with and without a priority queue

VI. Results and Discussion

- We first look at the results of the arrival distribution, we compare the normal and the priority queue, which illustrates from the graphs in the slide that arrival times were better in the normal distribution. When we look at the next results of how many customers were served, we analyze it from the average served, which only gave us a 1 person difference between normal and priority. Then the focus is turned towards how many customers were not served, which priority had the best results, as there was only a nine person difference on how many were not served per day.
- When we review the wait times, the priority queue has the better times at 23 minutes, compared to the wait time without the priority queue at 36 minutes. The wait time for when we add vs removing a window, when we add an extra window, we have a wait time of 11 minutes, while when we remove a window, the wait time becomes 37 minutes. After that comes the

results for the customers not served, while testing the results of adding or removing a window. There is no priority queue for the window tests, when adding an extra window, there are around 21 people that are not served on average. Then comes the results from removing a window, which is around 34 people. Now we go to the results of there being one extra teller for normal and priority queue, allowing us to view both the wait times and the amount of people that were not served. The results for wait times came out as 4 minutes for the normal line, while there was the wait time on average of 11 minutes for priority queue. The opposite came from when normal lines had the average of 21 people not being served, while the priority line had around 10 people not being served.

VII. Contributions and Conclusions

- The overall results illustrated how we can figure out the best way to clear out lines for banks, while viewing the overall best and worst cases for the results. Best case came from having an extra window in priority line, with the average wait time of 11 minutes and the average amount of unserved customers being 10 per day. Worst case is the 9 teller windows for a normal line, that has wait time averages of 36 minutes and an average of unserved customers at 34 per day. From the simulation standpoint, priority queues do work, by having the reduction in wait time, which makes it easier on the customers. The obvious removal or addition of windows drastically affects the wait times and how many customers are not served.
- The code was made by Romeo and revised by other team members. The report was written by Carlos and revised by other team members.