Final Summary – Caleb Flegel

**Data Analysis**

**Average Visit Time vs Frequency**

The graph seemed to form an exponential curve, which was a bit surprising for me. I expected a bit more of a linear increase. I wouldn't have been surprised if there was a bit of an exponential curve, but the slope was much more than I expected. Another surprising thing was the data points on the 40 and 50 people per hour frequencies. I can't explain why both seem to deviate from the trendline so much. If anything, it seems that the 50 person frequency is closer to the line, and the 40 person datapoint is off somehow. The problem may lie in how I designed my algorithm that randomly selects people to become patients.

**Doctor and Nurse Comparisons-**

\*Note\*: frequency of visits was set to 30

1 Doctor, 1 Nurse: 1hr, 52min

1 Doctor, 2 Nurses: 0hr, 21min

2 Doctors, 1 Nurse: 0hr, 16min

The results of the doctor and nurse number comparison surprised me a bit. I expected that the 1 doctor, 2 nurse combination would be the most efficient as the nurses have a shorter average treatment time. But it seems like high-priority patients having to wait for one doctor outweighed the doctor's longer treatment times. I was also a bit surprised at just how much of a difference is between having two and three providers. I expected maybe a 20- or 30-minute difference, but a difference of over an hour shows just how far a little extra help can go.

**Project Reflection**

Overall, I think my project went okay, but I encountered big problems along the way. From my original design, I got rid of the resident class and added the time class. I got rid of the resident class because the residents in this project were only names and thus easily stored on a map. On the other hand, I added the time class because timekeeping was more complicated than originally thought. It was easy enough for me to use variables in my town class to keep track of the overall time, but when trying to get the average and total visit times, using a class proved a lot more practical. Also, I felt I had trouble understanding how the priority queue worked at first, which caused some problems later down the line. I didn’t realize you couldn’t use iterators so I eventually had to add variables that would store temporarily popped patients. In addition, I had many more odd build errors than almost all of my other projects. I ended up with many linker errors and include errors with code that should work. This made me either have to leave and make note of a couple of things that didn’t impact my program so much. Lastly, commitments outside of this class lined up to make for a busy week, not including the final, which made finishing the project on time very difficult. Next time, I would use a different class structure, likely based around Hospital rather than Town, which would hopefully cut down on static variables and linker errors. I would also do more research on the potential problems first, so I don’t spend a lot of time trying to figure out something like priority queues later. But even with its challenges, my simulation does work and it was fun to work on it.