CMPE 365 Lab4: Greedy Algorithm

Cache Angus

20000629

October 12, 2018

# Question 1:

When solving for the gate problem, a greedy algorithm for scheduling needed to be used. This matter was to import and sort the arrival and departure time for planes. After importing them, they were then placed into a list and sorted in ascending order using the start time. This list when then put into the scheduling algorithm.

This algorithm had an empty list where planes were put into once they were to ‘arrive’. All of the planes were iterated to and the next plane to arrive was checked against those that were in the checked list. If the arrival time of the new plane was greater than the departing time of the old planes in the list, the planes were taken out of the arrived list and the number of gates in use were subtracted by one. This ensured that planes that had already ‘departed’ were accounted for. If a plane was still in the way then the number of gates increased.

For the first set there were 20 gates at a maximum that needed to be used and for the second case there were 22.

# Question 2:

The same code from the first problem was able to be used through adding an additional function. The function added simply took random indexes from a list and put a delay on the time at that index. The delays were also random and ranged from 0 minutes to 1 hour. When adding this delay to both the arrival and departure times, the number of gates changed each time the program ran. Because of the randomness in the delay, there was typically an increase in the number of gates used, though never that many more. The additional delays cause the starting times to have to further compete with the newly delayed departure times. However, there were times that the number of gates needed decreased. In a certain order, a delay would allow for there to be a delay in the arrival time of a plane that would cause it to no longer conflict with the previously incoming flight.