Project 3 Report

Ekaterina Miakotina, Evan Wenzel

[kmiakotina@csu.fullerton.edu](mailto:kmiakotina@csu.fullerton.edu) , [xevanx2002@csu.fullerton.edu](mailto:xevanx2002@csu.fullerton.edu)

Algorithm 1, Rentals, calculates the minimum number of laptops needed so students can access one during the time intervals without any overlap. The program uses a greedy algorithm with a min-heap, which is priority queue, in order to track the laptop availability. First, the intervals are sorted by start time, using min-heap in order to keep track of when laptops will be available. For each time interval, the program checks if laptop is free. If yes – reuse it, and if not – add a new laptop. The length of the heap represents the total number of laptops needed. The program takes an input (start,end), converts it into tuples, and prints the final output. The time complexity of the algorithm is O(n log n) because of sorting and heap operations. The space complexity is O(n) in the worst-case scenario when all intervals overlap and require separate laptops.

Algorithm 2, Network Delay Time, was a tricky algorithm to implement once arriving at the coding section of the process. Creating the pseudocode for the algorithm proved to be easy as the program itself checks for how much delay there is within the network when selecting a source node. A simple for loop that loops through specifically all the node entries to compare the source node to the signal node and keep track of the total delay when this match occurs. Start the counter at -1 to be sure that if there are no matches to occur, then return -1 to show that no nodes received the signal. With the implementation of a single for loop, the time complexity comes out to be O(n) or linear time based on the number of nodes present within the network. Space complexity O(1) or constant as there are no large scaling forms of data being created and just a counter that is used to keep track of the total delay.