BLG-335E
ANALYSIS OF ALGORITHM

Furkan GÜVENÇ
150160149

1- Introduction

In this project, we were asked to code the event scheduler using the min-heapshort algorithm. By doing this we have to use C++ language and this project should be compilable in g++ compiler. Since the course is an algorithm analysis, the differences between the calculation of complexity and algorithms are examined. The aim of this project is to implement the heapshort algorithm and understanding how it works.

2- Development and Operating Environments

This project is coded using CLion IDE on Linux operating system. G ++ will be used to compile. In order for the program to run, it is necessary to enter the txt file as the command line argument from the terminal. The txt file should be written with a space between each event name, start time, and end time, as specified in the assignment definition, with one event per line. In the project there are two library was included, iostream and fstream.

3- Data Structures and Variables

A struct named node is defined in the project. The algorithm was implemented on the array by using this struct that holds the event name, whether it is start or end. The newNode function creates and returns a node, taking the event name, type, and time. The swap function takes two nodes as arguments and replaces them. In the adder function, the array, node and order are taken as arguments and the node is added to the array. The Rearray function takes array with deleted root and returns the array by heapify.

4- Program Flow

The program first opens txt in the main function. Counts the number of events, then closes the file. Allocate array by number of events. It then opens txt again and creates two nodes for each row by sending it to the newNode function. One of these nodes holds the start time of the event and the end time. Nodes created with the newNode function are sent to the adder function and added to the array. After all nodes have been added, root is deleted and a print message is written. The remaining array is then sent to the Rearray function and heapified.

5- Conclusion

To conclute, in c ++, the min-heapify algorithm was implemented and executed. The ability to implement the algorithm through the array instead of using a binary tree is what makes this algorithm powerful. The complexity of this algorithm, which is logn long and controlled n times, is nlogn. It is one of the most efficient algorithms among those without memory complexity.