Istanbul Technical University
Faculty of Computer and Informatics
Computer Engineering Department

BLG 335E
Assignment 2 Report

Ömer Malik Kalembaşı - 150180112

December 2nd, 2022

1 How Program Works?

Assignment2.cpp file that I present is uses heap data structure. Firstly, I created a class which is named as "sample". It holds the features data has such as date, time, gap, grp, v and gi.

In main function, I created a string vector "estimators" which holds the estimators written in the input file. According to this vector, program prints the estimated estimators in output.

After program read the estimators line by line and store them into "estimators" vector, it starts to read features according to command "add" that given in input file. Program reads the features and send them into "sample" heap data structure with the help of function "insert". This function calls function "heapify" for creating proper max heap.

Program continues to read lines and if it cross with the line "print", it going to call the function "heapSort". This function will help us to applying other functions which are calculate statistic values.

In this assignment, I used brute force for implementing calculate functions for 4 features: gap, grp, v and gi. I have wrote functions for each of those features. In general look, it looks insufficient but it does not effect the complexity in the end.

Algorithm complexity of heapify = o(logn)

Algorithm complexity of heap sort = o

Total complexity of the implementation = o(n*logn)

2 Statistics

2.1 Table

Table 1: Table of program efficiency

input file	time elapsed miliseconds	number of calls the data structure methods
input-1.txt	1653	62
input-firstq-10.txt	1541	104
input-firstq-100.txt	2566	11633
input-firstq-1000.txt	125010	1662462
input-firstq-10000.txt	14733076	212446222
input-firstq-100000.txt	1801260413	-764293186
input-max-10.txt	1495	78
input-max-100.txt	2852	12337
input-max-1000.txt	129055	1722842
input-max-10000.txt	14467331	210042913
input-max-100000.txt		
input-mean-10.txt	1201	123
input-mean-100.txt	2375	11442
input-mean-1000.txt	120460	1627470
input-mean-10000.txt	15168656	216383686
input-mean-100000.txt		
input-median-10.txt	1576	78
input-median-100.txt	4795	12949
input-median-1000.txt	146701	1614593
input-median-10000.txt	14616769	211667982
input-median-100000.txt		
input-min-10.txt	1420	50
input-min-100.txt	4777	13702
input-min-1000.txt	133939	1645206
input-min-10000.txt	14907108	210978017
input-min-100000.txt		
input-std-10.txt	1387	50
input-std-100.txt	17558	11510
input-std-1000.txt	3102558	1747631
input-std-10000.txt		
input-std-100000.txt		
input-thirdq-10.txt	1483	62
input-thirdq-100.txt	6589	11486
input-thirdq-1000.txt	170096	1662906
input-thirdq-10000.txt	13961548	202390618
input-thirdq-100000.txt		
input.txt	1564	62

2.2 Plots

Figure 1: firstq with time elapsed(miliseconds) and number of calls data structures

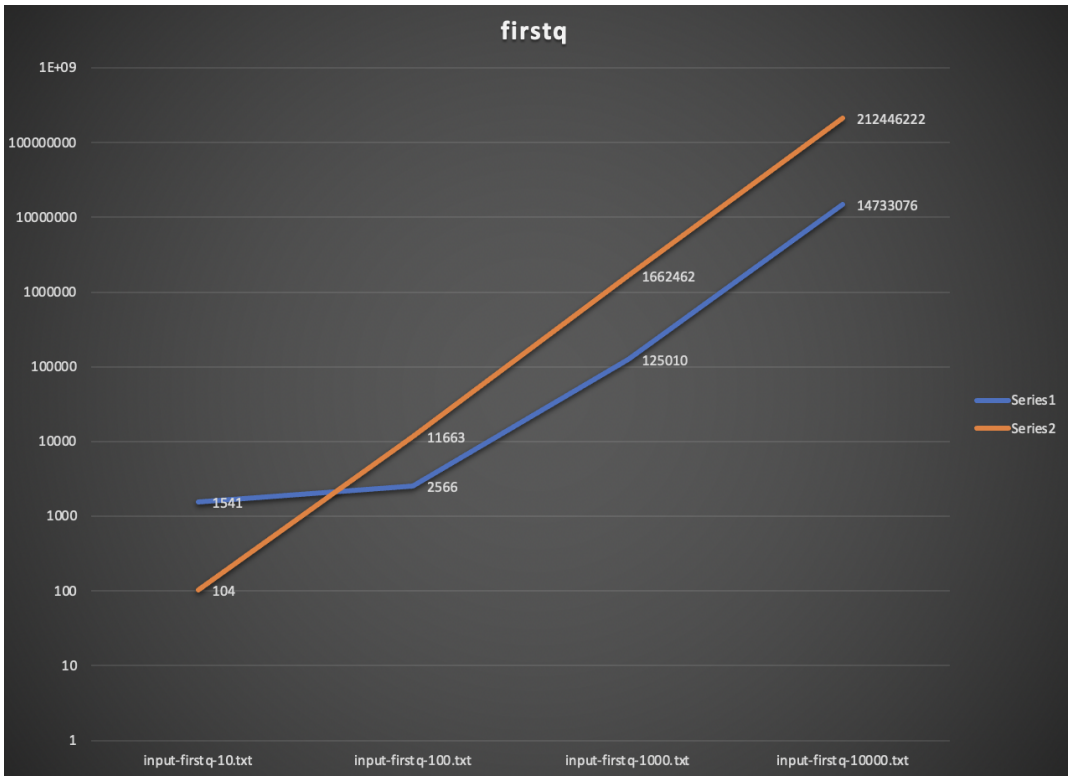

Figure 2: max with time elapsed(miliseconds) and number of calls data structures

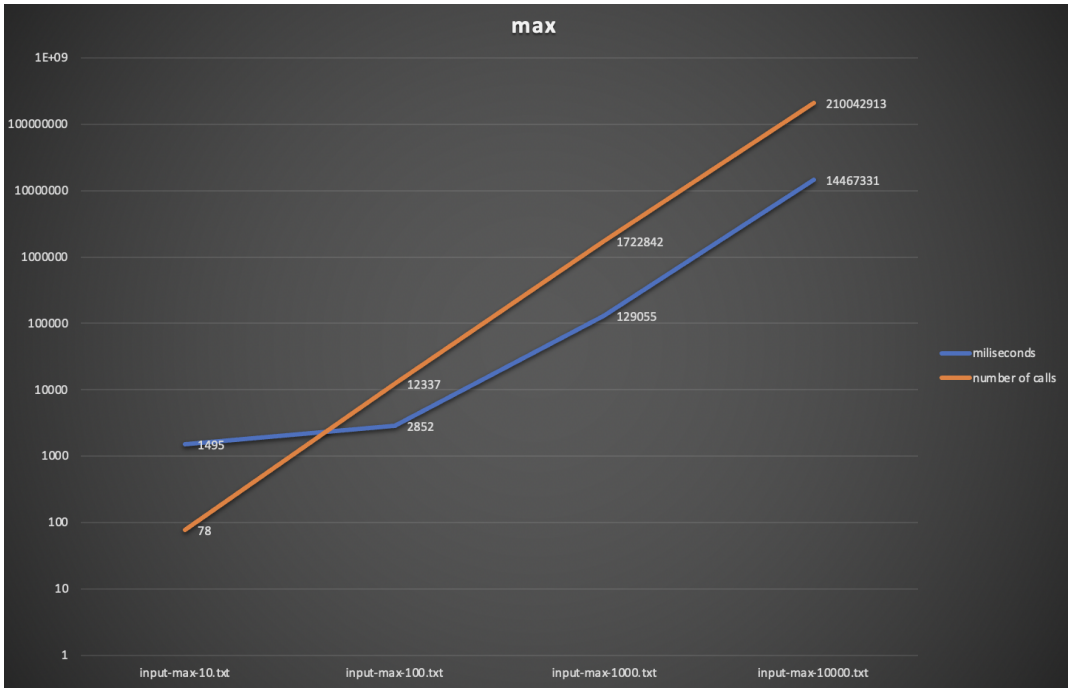

Figure 3: mean with time elapsed(miliseconds) and number of calls data structures

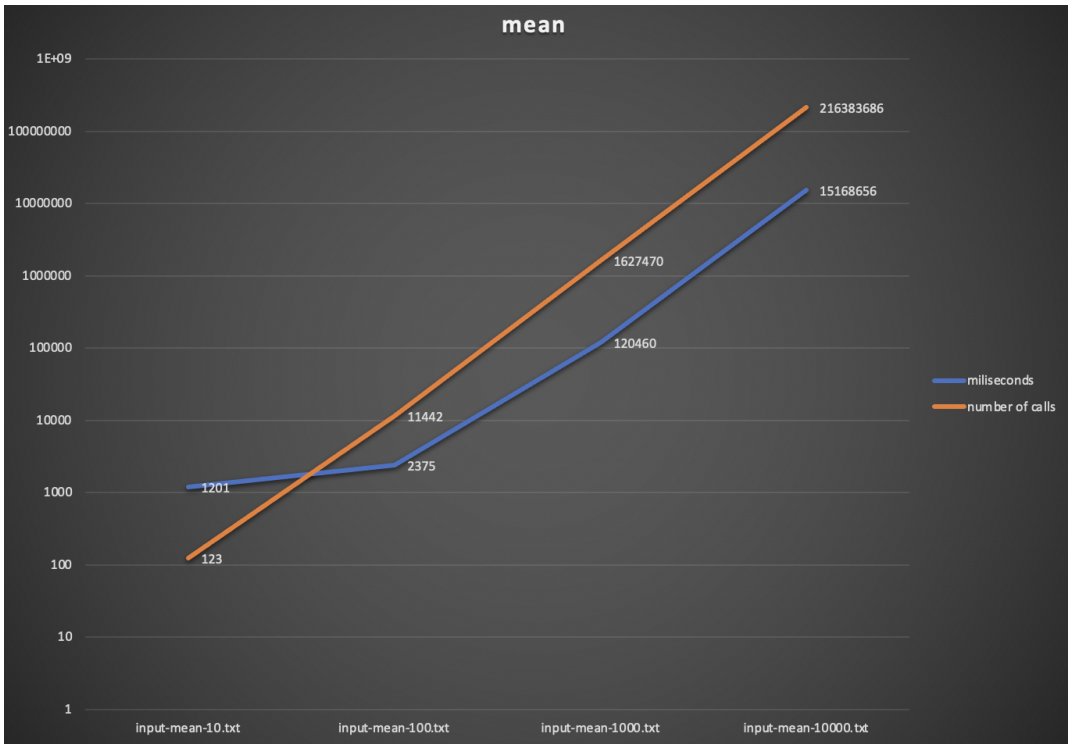

Figure 4: median with time elapsed(miliseconds) and number of calls data structures

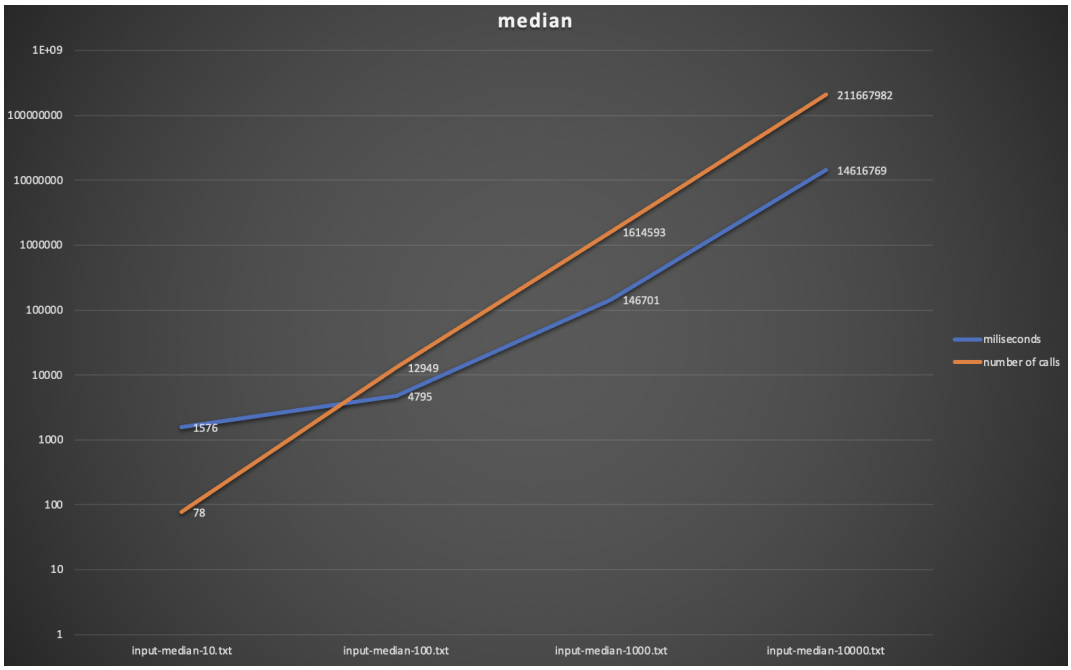

Figure 5: min with time elapsed(miliseconds) and number of calls data structures

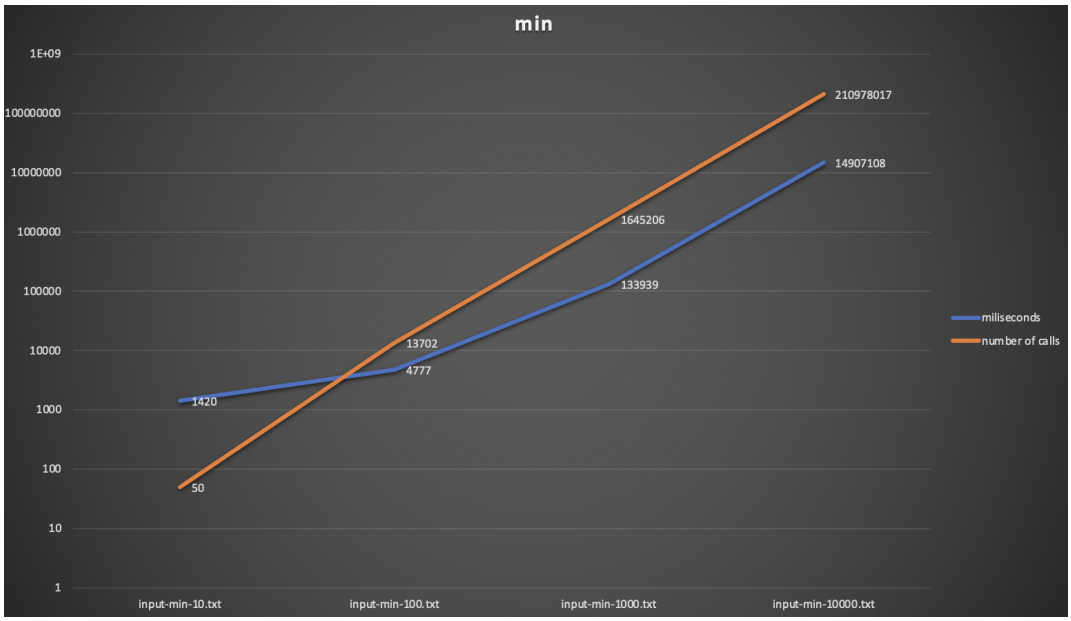

Figure 6: std with time elapsed(miliseconds) and number of calls data structures

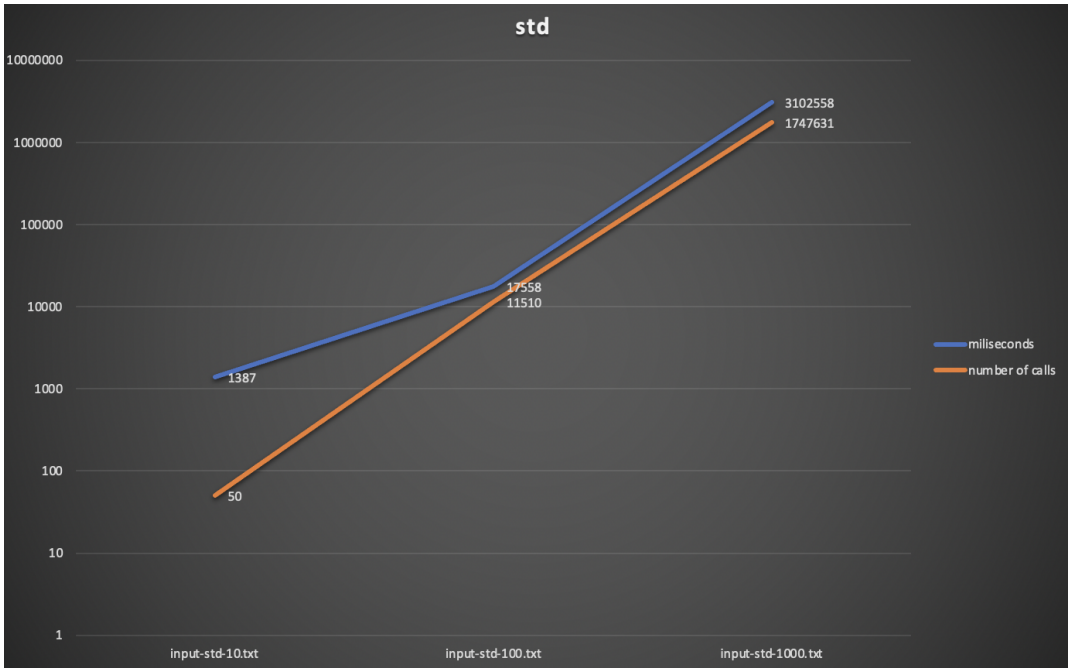

Figure 7: thirdq with time elapsed(miliseconds) and number of calls data structures