Hacettepe University Department of Computer Engineering BBM204

Name:Sergen Topçu Number:21328499

March 15, 2018

1 Introduction

In this experiment, we wanted to select 3 of the 7 different sorting algorithms. I have chosen 3 algorithms; Bubble Sort, Merge Sort and Heap Sort. Five different input files have been used to test these algorithms. These input files consist of 100,1000,50000,100000 and 250000 data. Using these algorithms, tables were created according to their working time and complexity.

2 Problems

I have many problem. First of all, time function. My time function is that,

```
double startTime=System.nanoTime();
double endTime = System.nanoTime();
double duration = (endTime - startTime)/1000000000;
```

Time function is starting before the function is called. Time function calculated the terms of nanosecond. Time has divide 1000000000 so times term is second.

3 Algorithm Analyses and Graphical Representation

3.1 Algorithm Running Time Table

Algorithm	TrafficFollow100	TrafficFollow1000	TrafficFollow50000	TrafficFollow100000	TrafficFollowAll
Bubble Sort	0,0004	0,11	372	1616	9206
Merge Sort	0,0006	0,003	0,18	0,41	1,26
Heap Sort	0,0023	0,005	0,27	0,66	1,88

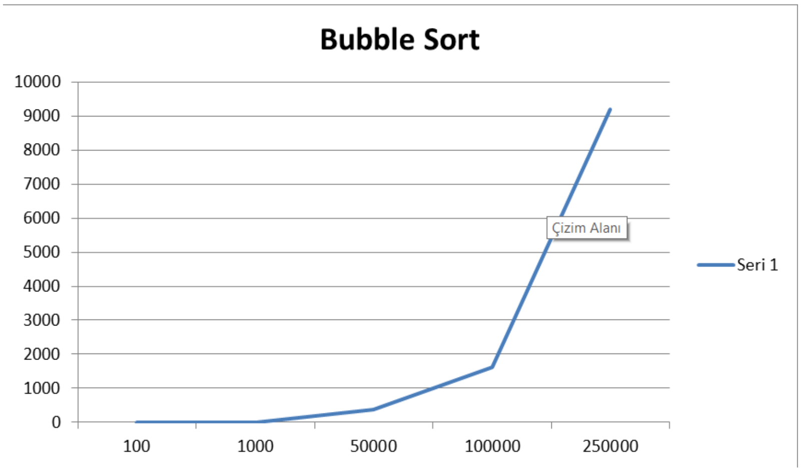

Figure 1: This is the bubble sort graph.

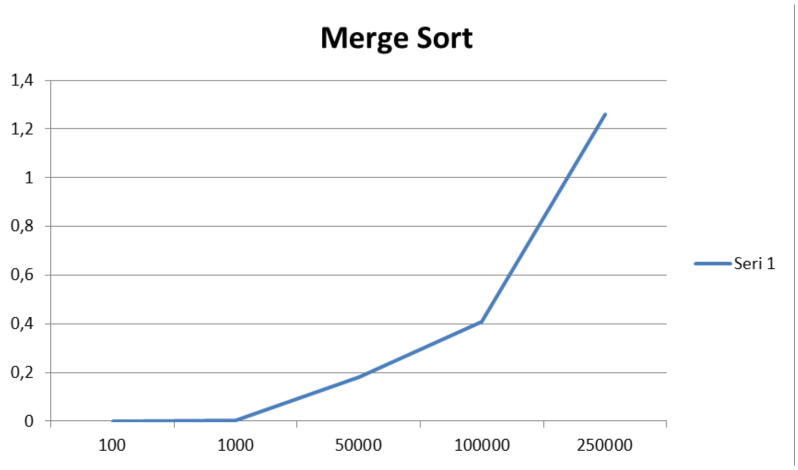

Figure 2: This is the merge sort graph.

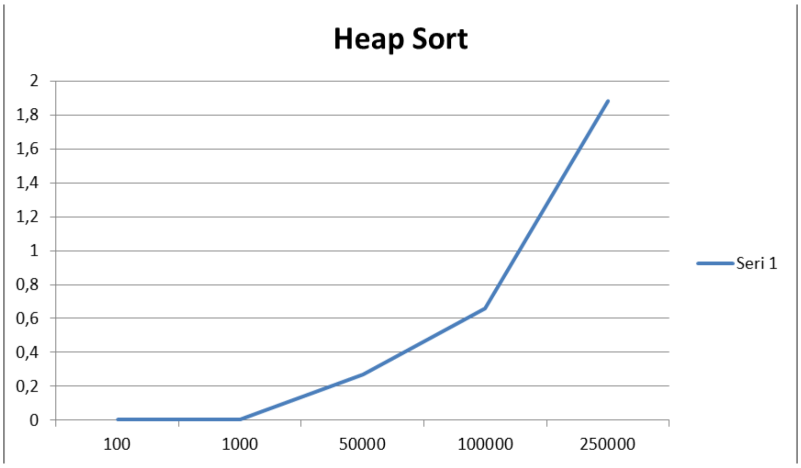

Figure 3: This is the heap sort graph.

4 Discussion

In this project, 3 different sort algorithms were used. These are Heap Sort, Merge Sort, and Bubble Sort. According to these 3 algorithms, working times were calculated using 5 different input files. It was observed that the Merge Sort and Heap Sort algorithms work much faster than the Bubble sort algorithm. The biggest reason is the complexity of the algorithm.

When we look at the memory used by these 3 algorithms, we can see that most uses memory merge sort algorithm. The reason for this is that the merge sort algorithm divides the data into 2 different sequences and performs the sorting algorithm. The memory used by the other 2 algorithms is much less than that of the Merge Sort. The Merge Sort algorithm is faster in some cases than in Heap Sort, but slower in some cases. But in general, times are very close together.

If I wanted to choose one of these 3 algorithms, I would prefer the Heap Sort algorithm to work faster and with less memory.