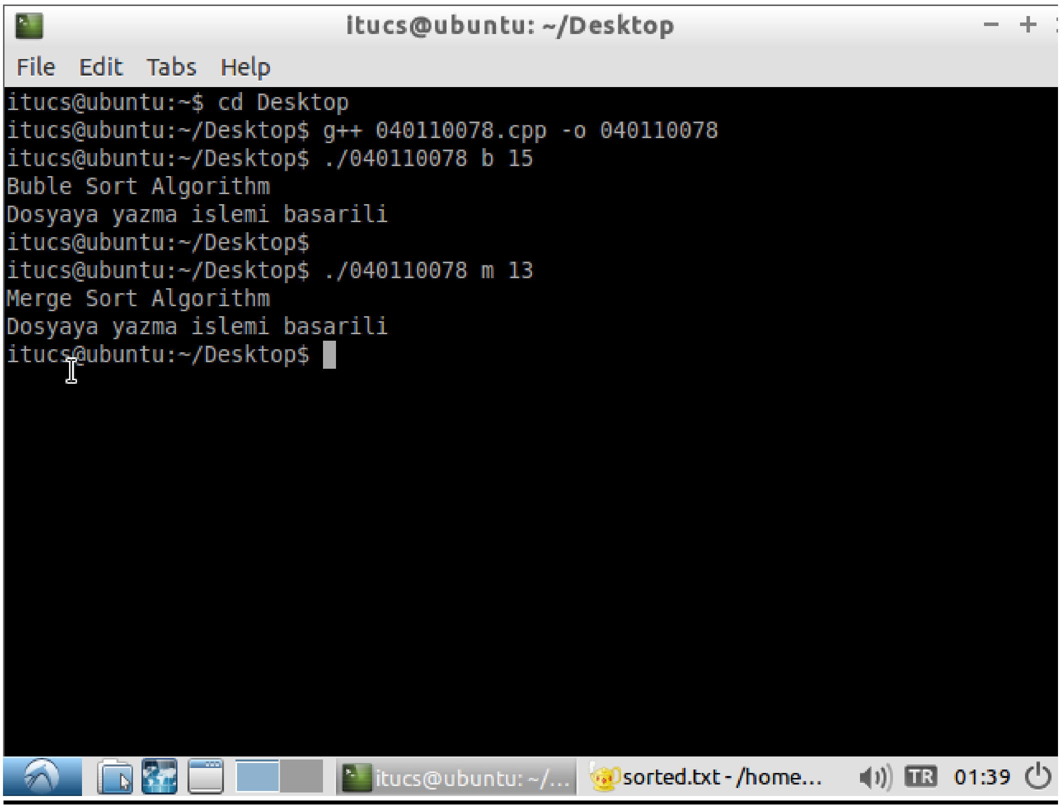
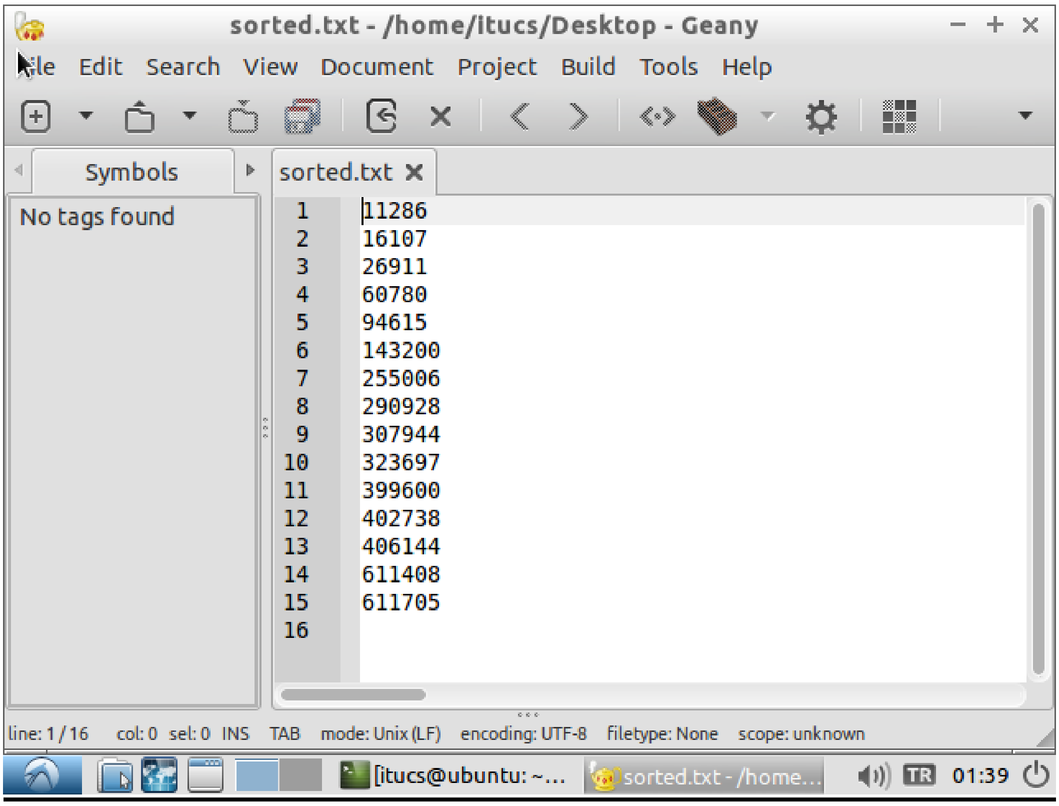
Part A (Outputs)

First of all my terminal screenshots are given below for both bubble sort and merge sort algorithm



```
itucs@ubuntu:~$ cd Desktop
itucs@ubuntu:~/Desktop$ g++ 040110078.cpp -o 040110078
itucs@ubuntu:~/Desktop$ ./040110078 b 15
Buble Sort Algorithm
Dosyaya yazma islemi basarili
itucs@ubuntu:~/Desktop$
itucs@ubuntu:~/Desktop$ ./040110078 m 13
Merge Sort Algorithm
Dosyaya yazma islemi basarili
itucs@ubuntu:~/Desktop$
```


sorted.txt - /home/itucs/Desktop - Geany
File Edit Search View Document Project Build Tools Help
Symbols
No tags found
sorted.txt
line: 1 / 16 col: 0 sel: 0 INS TAB mode: Unix (LF) encoding: UTF-8 filetype: None scope: unknown
[itucs@ubuntu: ~...
sorted.txt - /home...
TR 01:39

```
11286
16107
26911
60780
94615
143200
255006
290928
307944
323697
399600
402738
406144
611408
611705

```

As shown in the screenshot, both algorithms are proceed true.

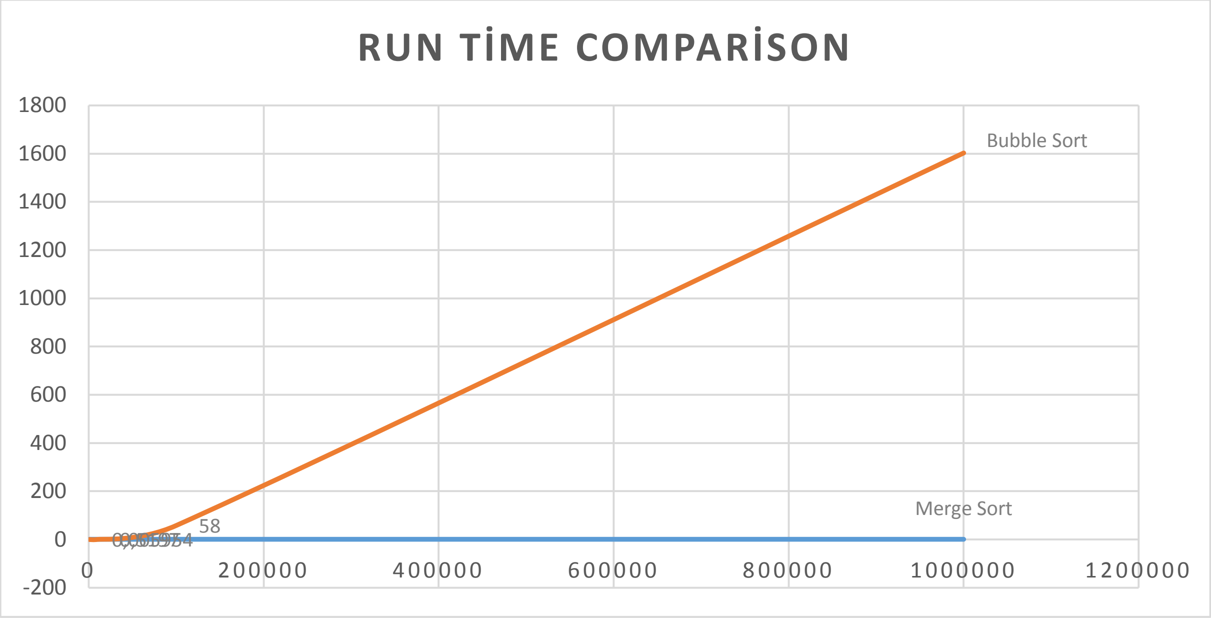
Part B

a. Theoretically asymptotic upper bound is defined O (n^2) for Bubble Sort and O (nlogn) for Merge Sort Algorithms. In my codes, it can easily seen that in Bubble Sort algorithm there are 2 for loop which makes the complexity O (n^2) whatever the coefficients are. Also Merge Sort Alg. Has recursive structure. It completely different from Bubble Sort. It also loops proportional with n. But, differently from Bubble Sort, in each it divides the problem into two sub problem. So logn where n is even effects on the complexity as a multiplier.

b &c. The results (in terms of runtime duration with unit 'second') for both algorithms related with different values of N is given below as a table.

N	Merge Algorithm	Bubble Sort Algorithm
1000	0.000484	0.005954
10.000	0.004875	0.519700
100.000	0.060988	57.921828
1.000.000	0.489495	1602.151597

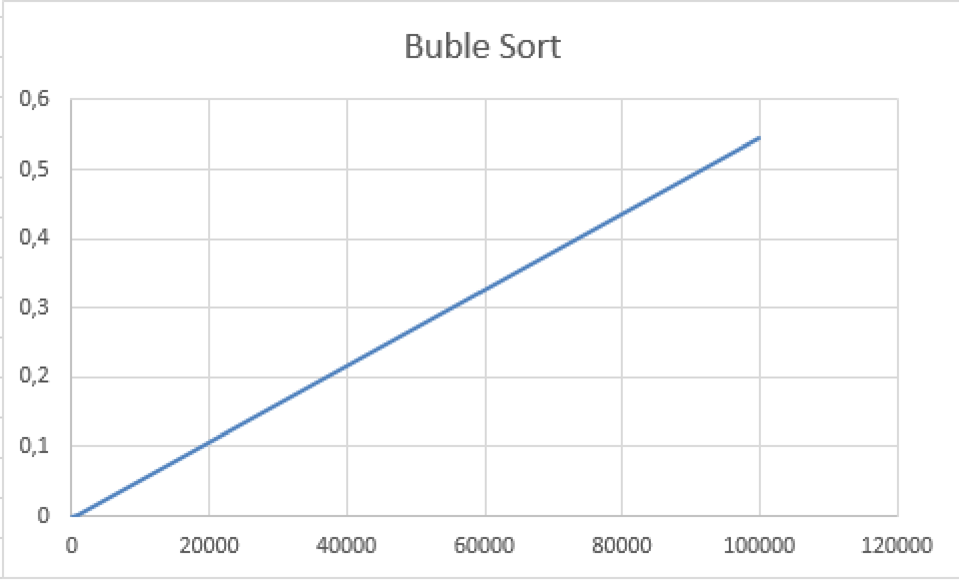
The data in the graph is illustrated in below graph.

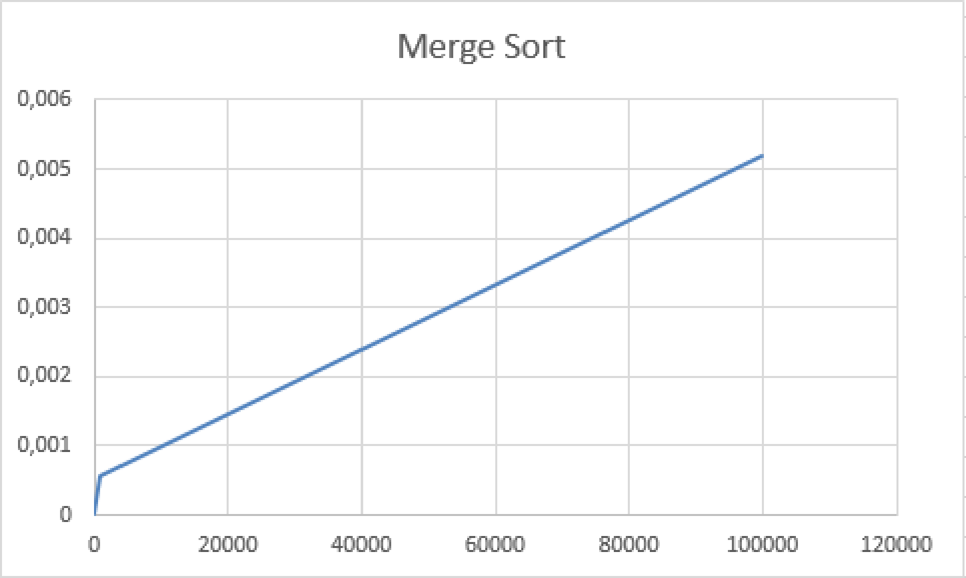


Bubble Sort algorithm has O (n^2) which means the worst case complexity. In order to explain this graph we can say that it is not the worst case. (Bubble Sort algorithm complexity depends on the input)

Merge Algorithm has the complexity of nlogn. The results in the graph are same as our expectations.

The results for N=100, N=1000 and N=10000 are given below graphs individually for both algorithms.





Conclusion

I would prefer bubble sort for very small data sets such as N=30. For smaller values of N, it is very fast algorithm. But practically, our need is more than 30. N can take really big values. In these type of cases, we should prefer Merge Sort in order to make the operation faster.

+Merge Sort algorithm has a good worst-case behavior

-Merge-sort needs an extra space to sort which means more memory access

Order is not important for Merge Sort. It can be beneficial depending on the case.