Analysis of Algorithms I Project I

MUHAMMMED SAİD DİKİCİ

150140007

CRN:10983

a.

Asymptotic upper bound for merge sort is O($\mathrm{nlog}_2\, n$) . And we can see that from running time for merge sort for 100000 elements by price is 1.205 seconds and for 800000 elements by price is 11.089 seconds it approximately $8 * \log_2 8$=24 times larger than the first one.

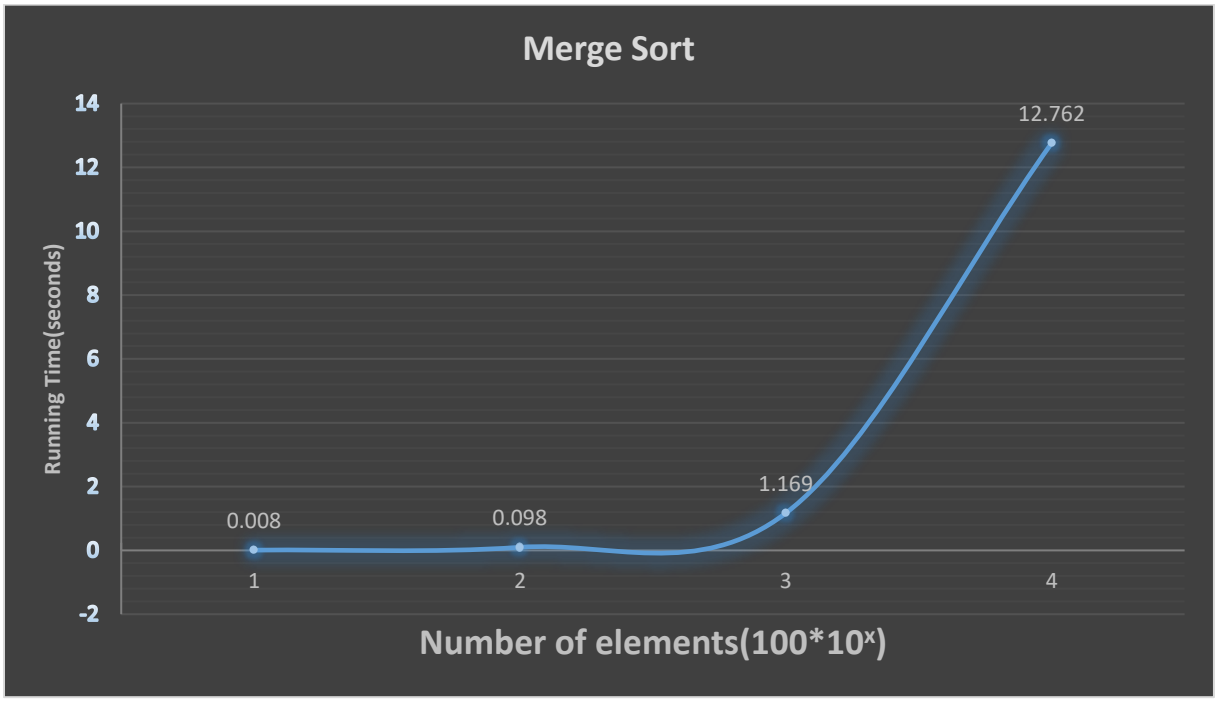
For insertion sort O(n^2) we also can see that running time for insertion sort for 10000 elements by price is 5.703 seconds and for 100000 elements is 751.508 seconds its about 100 times larger than the first one.

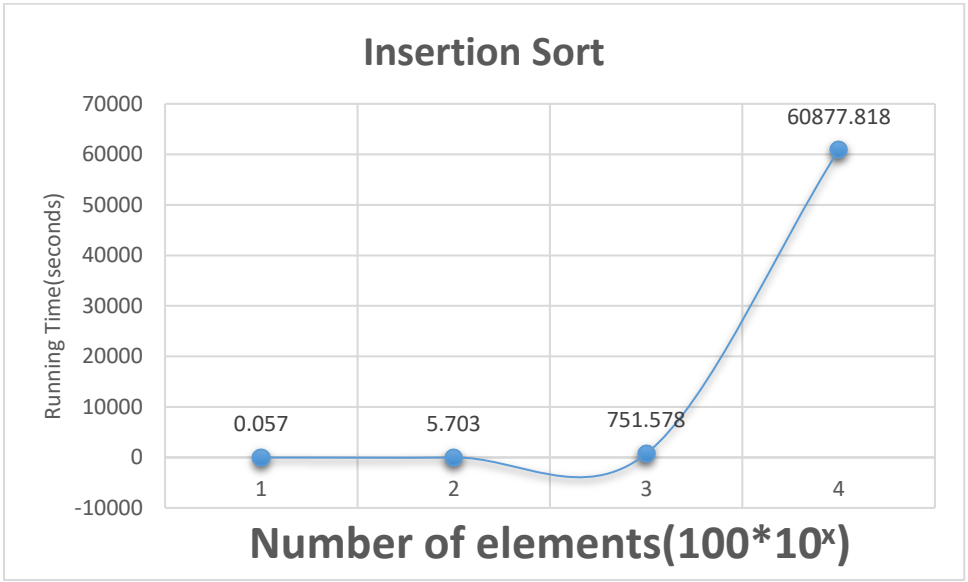
b.

Execution Time(seconds)	Merge Sort by price 1000 elements	Merge Sort by price 10000 Elements	Merge Sort by price 100000 Elements	Merge Sort by Price 900000 Elements(approximately)
1.	0.008	0.094	1.073	12.916
2.	0.009	0.092	1.203	12.307
3.	0.008	0.096	1.179	12.893
4.	0.008	0.096	1.154	12.804
5.	0.008	0.099	1.167	12.761
6.	0.009	0.103	1.181	12.784
7.	0.009	0.100	1.160	12.875
8.	0.008	0.102	1.221	12.693
9.	0.009	0.100	1.163	12.744
10.	0.008	0.099	1.185	12.843
Average	0.008	0.098	1.169	12.762

Execution Time(seconds)	Insertion Sort by price 1000 elements	Insertion Sort by price 10000 Elements	Insertion Sort by price 100000 Elements	Insertion Sort by Price 900000 Elements(approximately)
1.	0.057	5.533	699.968	56697.408
2.	0.058	5.734	704.172	57037.932
3.	0.058	5.934	717.326	58103.406
4.	0.055	5.508	761.978	61720.218
5.	0.055	5.506	780.404	63212.724
6.	0.057	5.827	746.486	60465.366
7.	0.055	5.669	784.360	63533.160
8.	0.057	5.709	798.181	64652.661
9.	0.056	5.809	806.452	65322.612
10.	0.057	5.798	716.455	58032.855
Average	0.057	5.703	751.578	60877.818

c.





At the first table we can say that line of merge sort is pretty similar with graphic of $\mathrm{nlog}_2\, n$ so it reflects the characteristics of O($\mathrm{nlog}_2\, n$).

And at the second table we see that line of insertion sort is similar to n^2 and it also reflects the characteristics of $O(n^2)$.

d.

We know that for insertion sort at the best case(data is already sorted) running time is Θ (n) but for merge sort it is $\Theta(n\log_2 n)$ for best, worst and average cases so we need to choose insertion sort algorithm for data that already sorted. We can confirm that with these;

Insertion sort with 900000 elements by timestamp takes 0.027 seconds and for merge sort 900000 elemnts by timestamp takes 10.981 seconds…