Name: نورهان رضا عبد الواحد احمد

Sec:2

BN:30

0.Selection Sort

N	Time for random data	Time for sorted data
1000	0	0
5000	12	11
10000	48	45
50000	1163	1149
75000	2608	2606
100000	4667	4640
500000	121748	118035

1.Insertion Sort

N	Time for random data	Time for sorted data
1000	0	0
5000	9	0
10000	22	0
50000	575	0
75000	1293	0
100000	2306	0
500000	57970	1

2. Merge Sort

n	Time for random data	Time for sorted data
1000	0	0
5000	1	0
10000	3	3
50000	20	16
75000	31	23
100000	40	27
500000	221	193

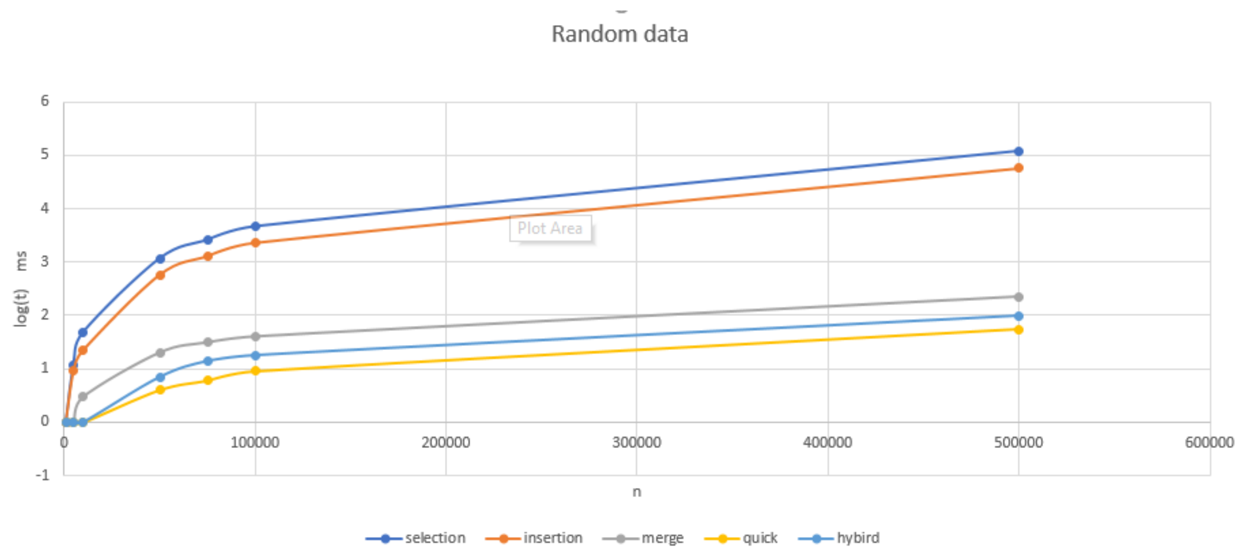
3. Quick Sort

N	Time for random data	Time for sorted data
1000	0	1
5000	0	17
10000	1	70
50000	4	1891
75000	6	4014
100000	9	7447
500000	55	326701

4. Hybrid Sort

n	Time for random data	Time for sorted data
1000	0	0
5000	0	1
10000	1	1
50000	7	3
75000	14	4
100000	18	9
500000	99	49

Graph for random data



Graph for sorted data

How I created hybrid sorting algorithm

from lectures

For small n, Insertion sort is better while Merge sort is better for

large n

•For n=2,10,100, Insertion sort is faster

•For n=1K,10K,…, Merge sort is faster

I took a threshold value which is 55 In between [2,100] in which insertion sort works better

So if the size of array is larger than 55 I divide the array and sort each 55

And then merge them together

And also the space complexity of insertion sort is better so it will be better for this algorithm

Insertion sort:

- Time complexity T(n)=$O(n)$
- Space complexity (in place) S(n)=Θ(1)

Merge sort:

- Time complexity T(n)=Θ($n$log$n$)
- Space complexity S(n)=Θ(n)