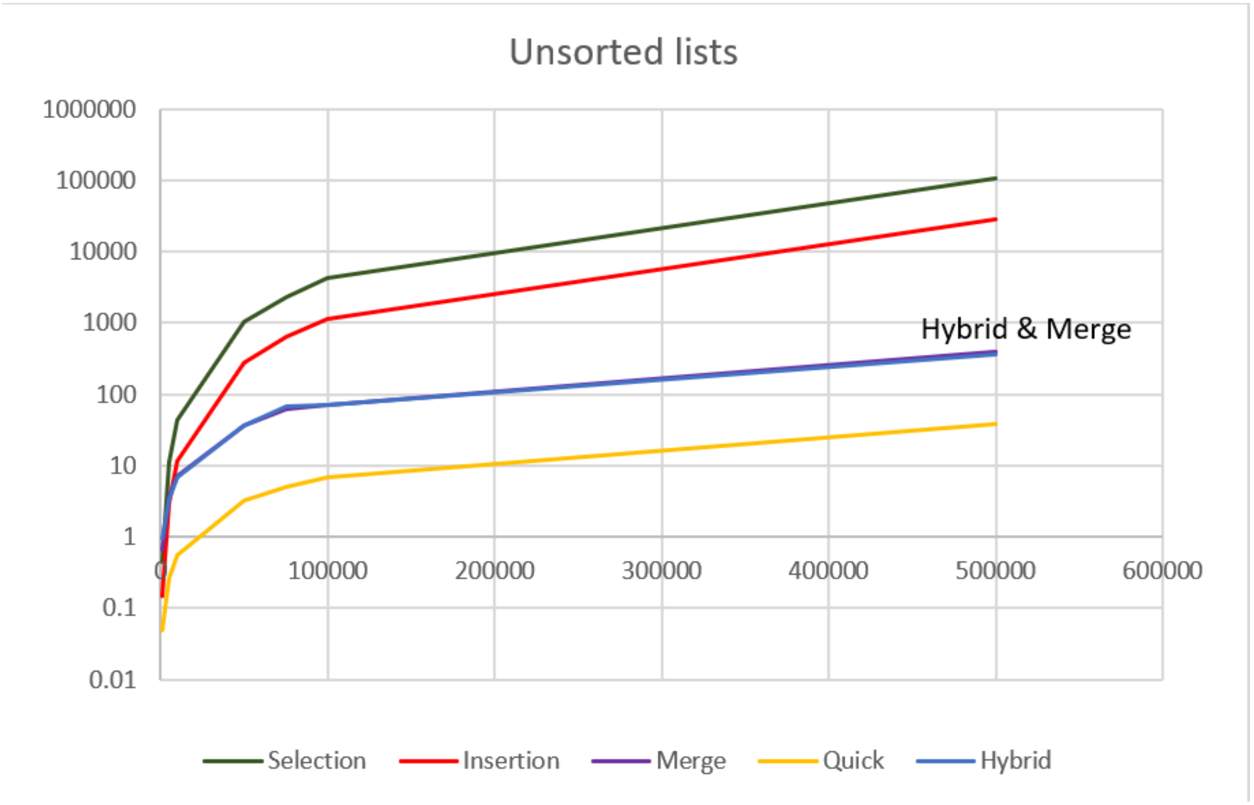
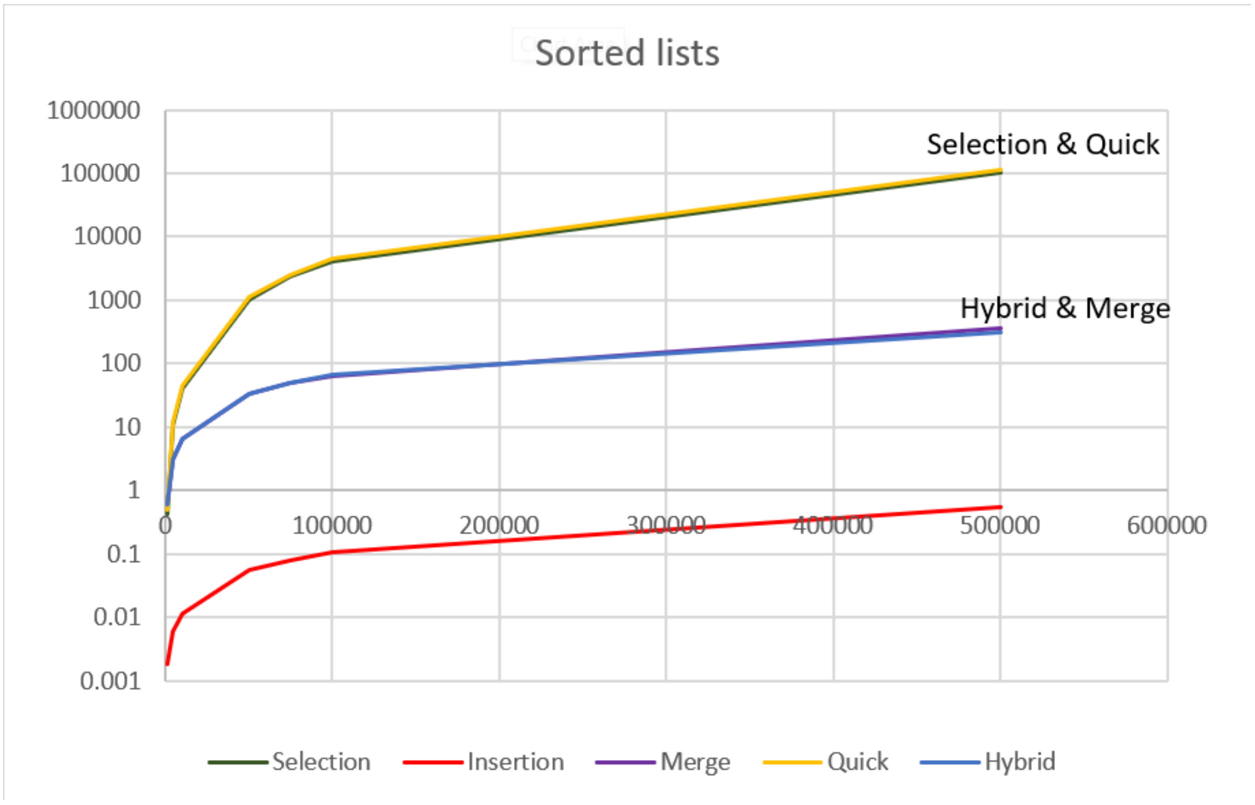
	Random					Sorted				
	Selection	Insertion	Merge	Quick	Hybrid	Selection	Insertion	Merge	Quick	Hybrid
N	Time (ms)	Time (ms)	Time (ms)	Time (ms)	Time (ms)	Time (ms)	Time (ms)	Time (ms)	Time (ms)	Time (ms)
1000	0.4389	0.1498	0.6532	0.05	0.9537	0.4232	0.0018	0.6143	0.4915	0.6145
5000	10.8455	2.9379	3.6663	0.259	3.3752	10.4395	0.0059	3.0945	11.678	3.1151
10000	43.042	11.3325	7.0813	0.5441	6.9103	41.411	0.0113	6.4428	45.3415	6.5195
50000	1034.79	280.61	35.6062	3.2174	36.4178	1035.54	0.0545	33.4297	1124.18	33.6842
75000	2332.35	632.589	61.9408	4.9989	65.887	2345	0.0802	48.359	2538.05	50.3232
100000	4149.84	1118.16	70.5234	6.6769	71.2176	4148.68	0.1055	64.7065	4524.49	66.3564
500000	105364	28597.9	394.13	38.2654	352.527	104852	0.5468	351.421	113545	316.872





Insertion sort is better than marge sort for small n

So, I start by making an algorithm that acts like a merge sort for the large n and insertion sort for small n

But the difficulty was in finding suitable number that the algorithm works above it as merge and bellow it as insertion I tried many numbers and at last, I found that 75 is better at most cases and for large n it becomes better than merge and insertion and selection.

<u>Unordered Lists:</u>

In the unordered lists the worst algorithms are insertion and selection as their complexities in the average case is O (n^2).

Merge sort is better than them average case is O (n log(n)).

Hybrid sort is better than Merge sort.

Quick sort is better than them all why?

That is because merge sort performs copies of the original list so that takes a lot of time that is why quick sort is better in the random data O (n log(n)).

<u>Ordered Lists:</u>

In this case the worst algorithms are the selection and the quick sort as the quick sort worst case when the data is sorted
O (n^2)

That's why quick sort is almost the same as the selection sort in the above graph.

Merge is better than them both.

Hybrid is better than merge.

But now the Insertion sort is the best as it's complexity when the data is sorted O (n)

<u>Here is a table for the algorithms ordered from the best in both cases (ordered and unordered lists) for large n.</u>

Unordered Lists:

1- Quick sort
2-Hybrid sort
3-Merge sort
4-Insertion sort
5-Selection sort

Ordered Lists:

1-Insertion sort
2-Hybrid sort
3-Merge sort
4-Selection sort
5-Quick sort