Mini Luo

COP3530

In my sorting analysis I'll be comparing three different sorts: bubble sort, insertion sort and quick sort. Each sort had to sort through four arrays of different sizes (10000, 20000,50000, 100000) and each of those arrays had four versions (random, ascending, descending). In total, there were 12 different arrays to sort through. In order to get a more accurate time, I did each of the array 5 times and then took the average the 5 times. Therefore, each sorting algorithm was tested 60 times.

Bubble sort is a quadratic sorting algorithm with a time complexity of $O(n^2)$. With the number of items sorted being on the x-axis and the time being on the y-axis, it can be concluded that as the size increases, the time increases quadratically. The order of the data affects the sorting time. Randomization causes the sorting to take a longer time, compared to being already in order takes the shortest time.

Insertion sort is also a quadratic sorting algorithm, which means it also has a time complexity of $O(n^2)$ and is indicated by the graph. The number of items being sorted and the run time has a quadratic relationship and it can be confirmed by the graphs. It also follows the trend with random data taking the longest time and the data in order taking the shortest.

Quick sort is non-quadratic sorting algorithm with the best-case time complexity of $O(nlogn)$ and indicated by the graph. Quick sort requires a pivot position to compare the data to and each graph will look different depending on which pivot position is chosen. I selected the last position as the pivot position and recursively used quick sort to sort either side of the array. Quick sort follows a different trend compared to the quadratic sorts, quick sort has the shortest time when sorting an array of random numbers, and takes the longest time when sorting through ascending order numbers.

All three methods are acceptable ways to sort through an array of integers. However, between these three sorting algorithms, quick sort takes the least amount of time overall, but especially when encountering random integers in no specific order.

bubble sort (random)

Number of items sorted	Time (microseconds)					
	Average	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
10000	351013.8	366032	345586	382896	341937	318618
20000	1016778.4	996932	1013510	1031080	1013920	1028450
50000	8391730	8723060	8254370	8284190	8350110	8346920
100000	35129380	34413300	33822900	36485800	37210700	33714200

bubble sort (ascending)

Number of items sorted	Time (microseconds)					
	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Average
10000	142444	142500	151628	149970	138619	145032.2
20000	582674	593412	586126	580281	598540	588206.6
50000	3484480	3441010	3449300	3438350	3462330	3455094
100000	13730500	13731000	13796400	13883200	13737000	13775620

bubble sort (descending)

Number of items sorted	Time (microseconds)					
	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Average
10000	252807	251046	256612	256718	258356	255107.8
20000	1046090	1087410	1116410	1040020	1047470	1067480
50000	6621570	6210420	6253040	6190990	6255450	6306294
100000	24794200	24878200	24872200	24727700	24804800	24815420

Bubblesort
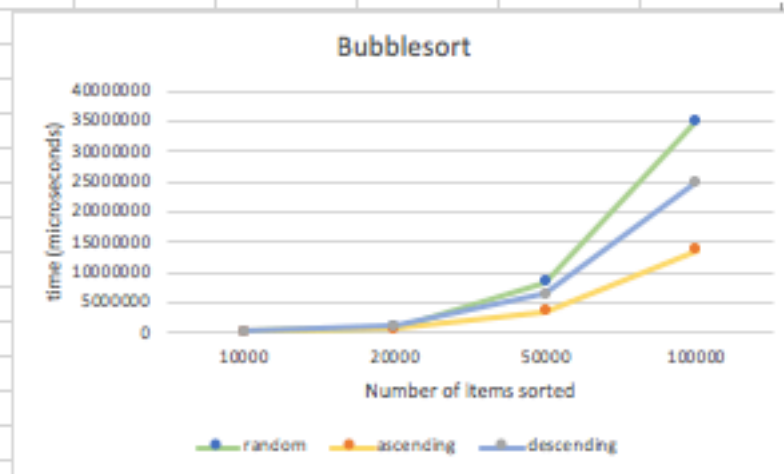
insertion sort (random)
Time (microseconds)

Number of it	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Average
10000	378481	352106	337000	338368	362345	353660
20000	1101230	994799	1035742	995956	998547	1025255
50000	8508390	8437380	8493829	8501293	8473827	8482944
100000	34977700	34814300	34918283	34829391	34920388	34892012

insertion sort (ascending)
Time (microseconds)

Number of it	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Average
10000	149311	147916	147365	147253	148572	148083.4
20000	567754	532989	536640	539283	549283	545189.8
50000	3286940	3284170	3281928	3287491	3281746	3284455
100000	13458600	13114500	13492833	13482731	13293748	13368482

insertion sort (descending)
Time (microseconds)

Number of it	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Average
10000	294226	307597	291820	298736	272887	293053.2
20000	1257680	1086460	1204938	1109050	1182737	1168173
50000	6608140	6524180	6592837	6686170	6538273	6589920
100000	26308500	27392837	26918273	27083774	26837283	26908133

Insertionsort — time (microseconds) vs Number of items sorted; series: random, ascending, descending
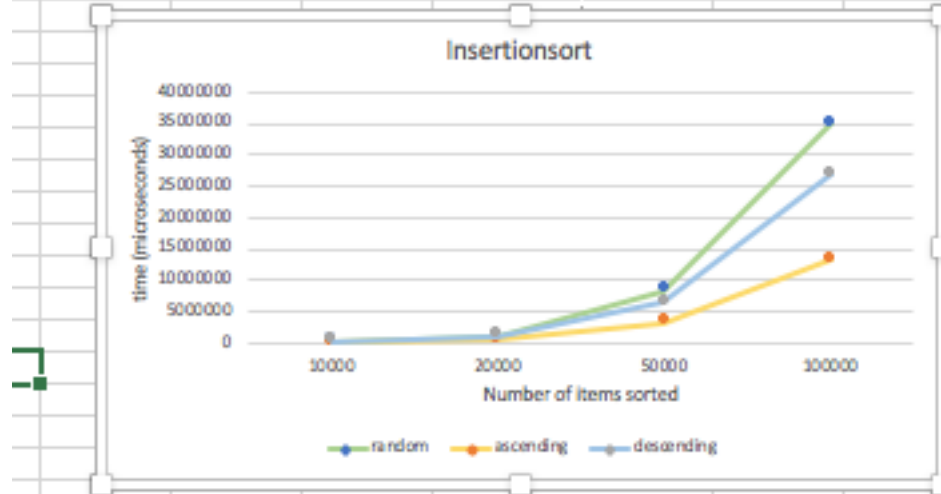
quicksort (random)						
Time (microseconds)						
Number of it	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Average
10000	1381	1357	1394	1402	1384	1383.6
20000	2759	2847	2793	2749	2733	2776.2
50000	6972	6582	6842	6684	6853	6786.6
100000	16291	15317	15948	16029	15849	15886.8

quicksort (ascending)						
Time (microseconds)						
Number of it	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Average
10000	252521	299891	283948	234819	248391	263914
20000	1294620	1081600	984750	1039283	992837	942701
50000	5855160	5859720	5847203	5837429	5832192	5846340.8
100000	22997000	23367700	22309849	23048293	2306700	18805908

quicksort (descending)						
Time (microseconds)						
Number of it	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Average
10000	220044	206543	203818	210381	221929	212543
20000	816046	776276	782948	794827	803928	794805
50000	4609090	4568180	4658271	4561123	4569182	4593169.2
100000	18571500	18447700	18273618	18337726	18478322	18421773

Quicksort