Algorithms Practical 4 - Ruth Dooley 19300753 - Answers

Quick Questions

1. How many comparisons does insertion sort make on an input array that is already sorted?

Linear

2. What is the stable sorting algorithm?

A stable sorting algorithm ensures that if items are equal to each other, that they aren't reordered in the sorting process. If so the sorting algorithm would be considered unstable

3. What is an external sorting algorithm?

A. Algorithm that uses tape or disk during the sort.

4. Identify 6 ways of characterising a sorting algorithm?

- Does the sorting algorithm compare? A comparison type sort looks at two elements at a time whereas non-comparison sort do not compare two items at a time
- The time complexity? How much time is required to complete the tasks based on the size of the input
- Space complexity? How much memory is required for the algorithm to complete the task based on the size of the input
- Stability? Stable or non stable sorting. Does the sorting algorithm preserve the existing relative order of elements when comparing equal keys
- Internal or external? Is the sorting contained in the main memory or RAM or externally sorted where external memory is used
- Recursive or non-recursive? Recursive implements the divide or conquer approach splitting up the dataset into smaller tasks and calls itself over and over. Non - recursive does not use this technique to split the dataset up

Sorting Algorithm Results

* Time in nanoseconds
* Numbers to be sorted range from 0 - 100
* In each of the cells in the sorting algorithm columns in the below table there will be three results. Each of these represent the time taken for the task to be completed given the random input

No. of Ints	SelectionSort	InsertionSort	BogoSort
5	538786 1072055 815817	418062 1115220 743699	450392 811417 835431
10	549203 531830 1068745	631637 478278 794533	571479 522193 715638
50	592592 1466376 1417973	465528 1067449 985787	504171 966386 623932
100	816634 758002 1550124	491251 762880 972070	458896 629316 919282
500	3369843 3494941 3857677	568892 974172 721899	470422 775381 948565

1,000	8275088 9757907 7195210	544447 904514 397618	353671 985887 335536
5,000	38641169 43417604 40971067	581664 776637 777156	420350 619254 439871
10,000	133224931 139129277 125419995	685214 694807 867749	460439 451252 483926
20,000	523509284 515365311 533099208	1382097 919410 966850	1481949 976725 814116
25,000	767714616 790821378 802450264	1115568 1686338 1185088	715184 1251832 750394

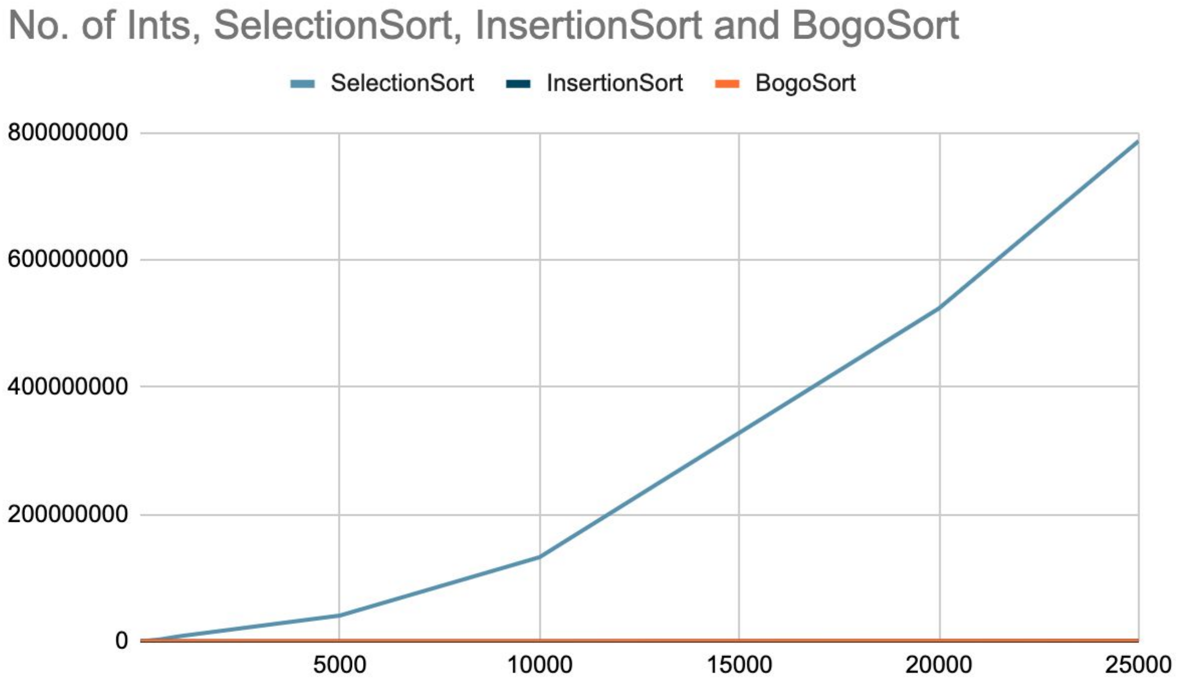
* After 25,000 elapsed time could not be calculated correctly and consistently

Average Times

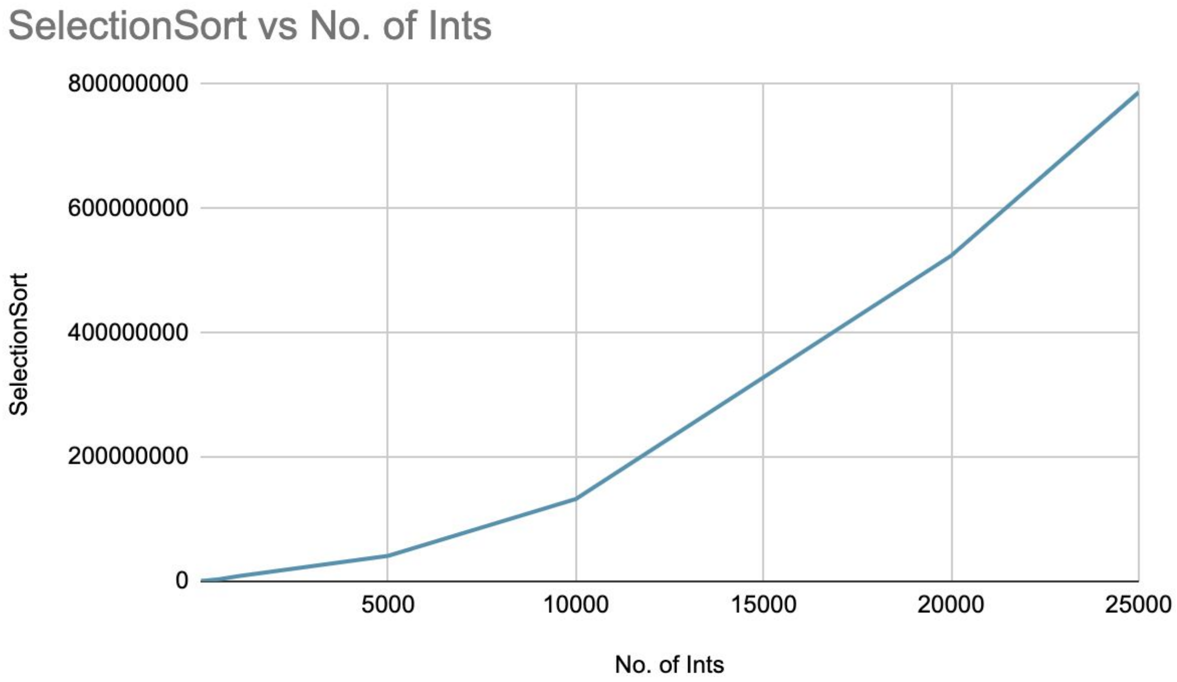
* Given the results above the average elapsed time for the task to be completed is calculated

No. of Ints	SelectionSort	InsertionSort	BogoSort
5	808886	758993.66667	699080
10	716592.66667	634816	603103.33333
50	1158980.3333	839588	698163
100	1041586.6667	742067	669164.66667
500	3574153.6667	754987.66667	731456
1,000	8409401.6667	615526.33333	558364.66667
5,000	41009946.667	711819	493158.33333
10,000	132591401	749256.66667	465205.66667
20,000	523991267.67	1089452.3333	1090930
25,000	786995419.33	1328998	905803.33333

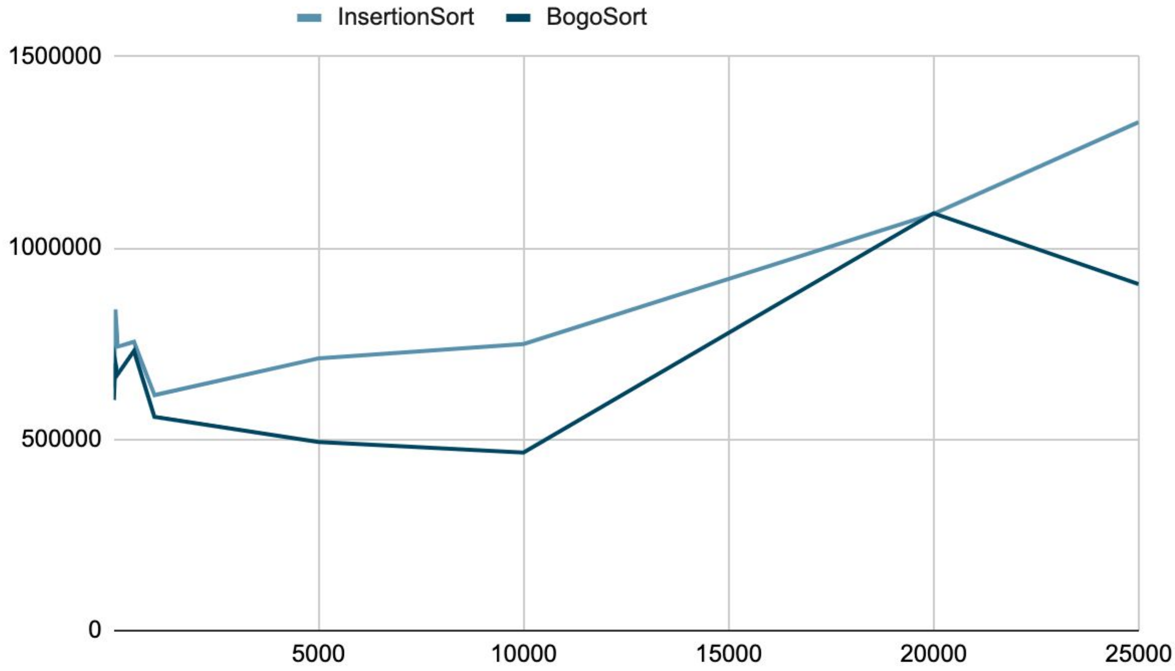
Total Number of Ints vs Time (All 3 Algorithms)



Total Number of Ints vs Time Selection Sort Algorithm



Total Number of Ints vs Time Remaining Algorithms



Selection Sort O(n2): Selection sort took the most time consistently to run as visualised in the first diagram. It can be clearly seen to take on a linear path, increasing in similar incrimements. THis is why the big O notation for this algorithm is n2 as the average slope for this graph is ½.

Insertion Sort O(n^2): The insertion sort seems to be taking on the beginning of a n * n curve. From around the midsection of the graph the curve appears to increase more and more steeply.

Bogo Sort O(n*n!)(On average): Best case scenario → O(n): Worst case scenario → O(∞): Although in this experiment with the small range of numbers from 0 - 99, the bogo sort did surprisingly well for the randomness of its sorting. It can be seen to perform better or equally as well to the insertion sort for every run. Having said this it is still an extremely unpredictable sorting algorithm and given a larger range of numbers and a larger input of numbers a sorted algorithm may never get reached.