Assignment 6 – Sorting

Method: I have created a while loop to loop through the array ten times. Clicking on the sort type and type of order. Then stating the size of the array. It then prints out the speed it took to sort the array. At the end it prints the average of the ten rounds.

In general over all of the sorts and orders the first round was generally slower and over the ten rounds

Key points found from results tables:

Selection Sort:

1. At size 100 reverse was the fastest
2. The sorted array at size >10000 takes for ever to compile
3. Not to much difference between the 3 order types
4. Slowest sort overall

Selection Sort 2:

1. Size > 20000 the array took a very long time to sort.
2. Selection 1 was significantly faster than selection 2;
3. I believe it wasn't faster because it sorted from front and back but didn't stop at the middle

Insertion Sort:

1. A ordered sort is significantly faster than random/reverse
2. Random is very slow

Insertion Sort 2:

1. I possibly didn't write the code correctly for this method, anything size > 10000 will take well over a minute to compile.
2. For 100 and 1000 similar results for insertion 1

Merge Sort:

1. Significantly faster than Selection and insertion sort.
2. Ordered is fastest as the size increases
3. Reversed and random are similar speeds

Merge Sort 2:

1. Faster in all results against merge 1.
2. The fastest sorting algorithm in all the of the results
3. Ordered is the winner, reverse and random are similar speeds

Quick Sort:

1. Similar results across the board, ordered is quicker

Quick Sort 2:

2. Very similar results to the first quick sort

Quick Sort 3:

3. I think something didn't quite work out with my code to find the median as it takes a very long time to sort now the results are as bad as selection or insertion sort.

Conclusion

Out of the altered sorts the merge sort was the only sort than made a significant increase in performance. The insertion and selection sorts struggled to sort large size arrays. The first sort was generally the slowest – the CPU/Java finds the path of least resistance. At a small size array 1000 or less the results where all pretty much instantaneous.

From this assignment I have a through understanding of how the sorts work. The speed it takes is also understandable. I have also googled and read up on other sorts. Altering the sorts was challenging but I got a great buzz when the array sorted for the first time. I am learning a lot each week and trying to make sure I attempt the challenges.

Casey Glover