Abraham Austin

Jan 30, 2023

The question I chose to address for my experiment was: “ Does the initial order of the elements in the array affect the runtime of the sorting algorithms?”. The methodology used to answer this question was to run each sorting algorithm (bubble, selection, insertion, shell, merge, and quick) on three different arrays: one randomized, one in reverse order, and one in ordered order. The run time of each algorithm was recorded and compared to determine any differences. I did this by looping through three different arrays where each had a different order and storing the time taken by each iteration and sorting algorithm.

The results showed that the run time of the algorithms varied depending on the initial order of the array. The bubble, selection, and quick algorithms showed the greatest differences in run times. Bubble was able to reduce the runtime for the ordered array, but for the randomized and reversed arrays it was also the most inefficient. The time taken for the randomized and reversed arrays were also pretty similar, but the randomized array was always slightly longer. Insertion sort was able to reduce the runtime for the ordered array significantly, but the reversed array caused a significant increase in runtime. The quick sort had an increased run time for the reverse and randomized arrays, but the reversed array had nearly double the run time of the randomized array. The shell, merge, and selection algorithms all had similar run times, regardless of the initial array order.

In conclusion, the initial order of the array does affect the run time of the sorting algorithms, but the magnitude of this effect depends on the algorithm. Bubble and insertion, and quick algorithms are particularly affected by the initial order of the array, while insertion, shell, and merge algorithms are not as affected.

data:

Initial Position	bubble	selection	insertion	shell	merge	quick
Randomized:	276	109	77	2	1	2
Reverse:	247	95	155	1	1	53
Ordered:	0	108	0	0	1	144