

- I. Selection, bubble, and insertion sort only move elements of the array when elements of lower value are after elements of a higher value. In the best case, this does not happen so there are no moves.
- II. The bubble and insertion sort algorithms only compare two items at a time. They traverse the entire array and compare every element with the element before it. This would make the number of comparisons $n-1$. In our case, n is 1000, so the number of comparisons is 999.
- III. The selection sort algorithm results in 1500 moves when sorting the worst case array because the maximum number of moves for an array with 1000 elements would be 3000 moves. In the descending order, selection sort inherently swaps the smallest element with the largest element. This prevents the algorithm from switching elements a second time, as they are already in the correct place.
- IV. Merge sort splits up the array to single elements, then relies on comparisons to put the array back in order. Even if the array is in its best case, the algorithm still moves all the elements each time.
- V.

For selection sort, the cost of moves is small compared to the other algorithms. However, the number of comparisons is very high.

For bubble sort, the number of moves for the best case is zero. However, when it is not the best case, bubble sort requires a lot of moves. It also requires a lot of comparisons when the array is not already sorted. Bubble sort requires the most amount of moves out of all the algorithms.

Insertion sort is similar. In the average case, however, insertion sort performs better than bubble sort. It does have less moves than bubble sort however.

Merge sort is the most efficient algorithm in terms of moves and comparisons as it requires the least of both. However, merge sort does require more memory than the other algorithms as it instantiates a temporary array while sorting.

Quick sort is similar to insertion sort but requires more comparisons for the best case. It also requires moves for the best case, where insertion does not require moves.

sort.txt

Moves Comp

SELECTION SORT

--BST--

0 501500

--AVG--

2985 501500

--WST--

1500 501500

BUBBLE SORT

--BST--

0 999

--AVG--

751074 496944

--WST--

1495503 499500

INSERTION SORT

--BST--

0 999

--AVG--

251355 251357

--WST--

499499 499500

MERGE SORT

--BST--

19952 5044

--AVG--

19952 8744

--WST--

19952 4941

QUICK SORT

--BST--

6 1999

--AVG--

133347 134099

--WST--

496515 497509

Excel Graphs

