

Jim Farrell

## SWDV 610 Week 5

1. Consider the following list of integers: [1,2,3,4,5,6,7,8,9,10]. Show how this list is sorted by the following algorithms:

- bubble sort
- selection sort
- insertion sort

All three sort functions are  $O(n^2)$ . With the list of integers in the already sorted state no swapping had to take place during the sort. This would be considered “best case” for each of the functions.

I then reversed the list to [10,9,8,7,6,5,4,3,2,1]. I included an outer counter and inner counter for each of the loops. For both the bubble sort and the selection sort the outer and inner counter numbers remained the same for a sorted list and an un-sorted list. The insertion sort behaved differently. For the sorted list the inner loop was not visited. The insertion sort did increment for the reversed list the same number of times as for the selection sort.

.....Outer Iterations Inner Iterations Number of Swaps

Sorted Bubble	9	81	0
Unsorted Bubble	9	81	45
Sorted Selection	10	45	0
Unsorted Selection	10	45	45
Sorted Insertion	9	0	0
Unsorted Insertion	9	45	45

For each outer loop pass the bubble sort works through the list comparing adjacent items. If the item on the left is smaller than the item on the right it swaps the two items. It continues this until it reaches the end of the list and then repeats this on the next outer loop.

Bubble Sort starting with unsorted list.

[9, 10, 8, 7, 6, 5, 4, 3, 2, 1]

[9, 8, 10, 7, 6, 5, 4, 3, 2, 1]

[9, 8, 7, 10, 6, 5, 4, 3, 2, 1]

[9, 8, 7, 6, 10, 5, 4, 3, 2, 1]

[9, 8, 7, 6, 5, 10, 4, 3, 2, 1]

[9, 8, 7, 6, 5, 4, 10, 3, 2, 1]

[9, 8, 7, 6, 5, 4, 3, 10, 2, 1]

[9, 8, 7, 6, 5, 4, 3, 2, 10, 1]

[9, 8, 7, 6, 5, 4, 3, 2, 1, 10]

[8, 9, 7, 6, 5, 4, 3, 2, 1, 10]

[8, 7, 9, 6, 5, 4, 3, 2, 1, 10]

[8, 7, 6, 9, 5, 4, 3, 2, 1, 10]

[8, 7, 6, 5, 9, 4, 3, 2, 1, 10]

[8, 7, 6, 5, 4, 9, 3, 2, 1, 10]

[8, 7, 6, 5, 4, 3, 9, 2, 1, 10]

[8, 7, 6, 5, 4, 3, 2, 9, 1, 10]

[8, 7, 6, 5, 4, 3, 2, 1, 9, 10]

[7, 8, 6, 5, 4, 3, 2, 1, 9, 10]

[7, 6, 8, 5, 4, 3, 2, 1, 9, 10]

[7, 6, 5, 8, 4, 3, 2, 1, 9, 10]

[7, 6, 5, 4, 8, 3, 2, 1, 9, 10]

[7, 6, 5, 4, 3, 8, 2, 1, 9, 10]

[7, 6, 5, 4, 3, 2, 8, 1, 9, 10]

[7, 6, 5, 4, 3, 2, 1, 8, 9, 10]

[6, 7, 5, 4, 3, 2, 1, 8, 9, 10]

[6, 5, 7, 4, 3, 2, 1, 8, 9, 10]

[6, 5, 4, 7, 3, 2, 1, 8, 9, 10]

[6, 5, 4, 3, 7, 2, 1, 8, 9, 10]

[6, 5, 4, 3, 2, 7, 1, 8, 9, 10]

[6, 5, 4, 3, 2, 1, 7, 8, 9, 10]

[5, 6, 4, 3, 2, 1, 7, 8, 9, 10]

[5, 4, 6, 3, 2, 1, 7, 8, 9, 10]

[5, 4, 3, 6, 2, 1, 7, 8, 9, 10]

[5, 4, 3, 2, 6, 1, 7, 8, 9, 10]

[5, 4, 3, 2, 1, 6, 7, 8, 9, 10]

[4, 5, 3, 2, 1, 6, 7, 8, 9, 10]

[4, 3, 5, 2, 1, 6, 7, 8, 9, 10]

[4, 3, 2, 5, 1, 6, 7, 8, 9, 10]

[4, 3, 2, 1, 5, 6, 7, 8, 9, 10]

[3, 4, 2, 1, 5, 6, 7, 8, 9, 10]

[3, 2, 4, 1, 5, 6, 7, 8, 9, 10]

[3, 2, 1, 4, 5, 6, 7, 8, 9, 10]

[2, 3, 1, 4, 5, 6, 7, 8, 9, 10]

[2, 1, 3, 4, 5, 6, 7, 8, 9, 10]

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

The selection sort finds the largest value in the list and puts it in order towards the end of the list.

Selection Sort starting with un-sorted list:

[1, 9, 8, 7, 6, 5, 4, 3, 2, 10]

[1, 2, 8, 7, 6, 5, 4, 3, 9, 10]

[1, 3, 2, 7, 6, 5, 4, 8, 9, 10]

[1, 3, 2, 4, 6, 5, 7, 8, 9, 10]

[1, 3, 2, 5, 4, 6, 7, 8, 9, 10]

[1, 3, 2, 4, 5, 6, 7, 8, 9, 10]

[1, 3, 2, 4, 5, 6, 7, 8, 9, 10]

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

The insertion sort assumes a portion of the list is already sorted. It inserts at the front of the list list[0] and shifts the other items in the front of list.

Insertion Sort starting with un-sorted list:

[9, 10, 8, 7, 6, 5, 4, 3, 2, 1]

[8, 9, 10, 7, 6, 5, 4, 3, 2, 1]

[7, 8, 9, 10, 6, 5, 4, 3, 2, 1]

[6, 7, 8, 9, 10, 5, 4, 3, 2, 1]

[5, 6, 7, 8, 9, 10, 4, 3, 2, 1]

[4, 5, 6, 7, 8, 9, 10, 3, 2, 1]

[3, 4, 5, 6, 7, 8, 9, 10, 2, 1]

[2, 3, 4, 5, 6, 7, 8, 9, 10, 1]

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

I created a larger sorted list with 1000 items and included the Timer library. I ran some timing tests with 50 iterations of the 1000 item list. The insertion sort was noticeably faster than the bubble sort and selection sort.

BubbleSort with large sorted list 8.39991725072909 seconds

SelectionSort with large sorted list 9.117338706527338 seconds

InsertionSort with large sorted list 0.01062940525689271 seconds