1) In this case, we start off with two lists. We want to sort them with merge sort.
List 1: [1, 25, 31, 16]
List 2: [-3, 0, 16, 27]

Step 1) Let's make a new list. This will be our final sorted list.

Step 2) Start by comparing the 0^{th} element in both lists.

In our first cade case, -3 is less than 1. So, we put -3 in the final list.
Then it's 1 and 0, 0 is smaller, so we put 0 in the final list..
then it's 1 and 16, 1 is smaller, so we put 1 in the final list..
then it's 25 and 16, 16 is smaller, so we put 16 in the final list..
then it's 25 and 27, 25 is smaller, so we put 25 in the final list..
then it's 31 and 27, 27 is smaller, so we put 27 in the final list..
And then we put 31, because it's the only number left.

Ok, now our final list is ready!
We get [-3, 0, 1, 16, 25, 27, 31] as out final list.

2) Insertion sort
[-1, -5, 67, -10, 21, 8, 4, 1]

Step 1) Compare index 1 with index 0. In this case, the smaller number should go to the 'left' side of the array. The left side of the array is the sorted array. -5 is smaller, so we insert -5 to the beginning.

Step 2) Now we compare index 1 with index 2, repeat the same process as step 1. After one more iteration, it's now [-5, -1, 67, -10, 21, 8, 4, 1], because 67 is in the correct place. Repeat this process until you get a completed, sorted list.

Step 3) Finished. The new list should be [-10, -5, -1, 1, 4, 8, 21, 67]

3) Quick sort
[-5, 42, 6, 19, 11, 25, 26, -3]

First, we have to choose a pivot. In this case I will go with -3 as the pivot.

Now we put the pivot in the correct place. I will swap -3 with 42, so now we get
[-5, -3, 6, 19, 11, 25, 26, 42]

Left List: [-5]

Right List: [6, 19, 11, 25, 26, 42]

I will choose 42 as the pivot. 42 is correct.

To the left of 42, I will make a new list: [6, 19, 11, 25, 26]

I will choose 26 as the pivot. 26 is correct.

Same with 25, 25 is correct.

Now we have to deal with [6, 19, 11].

I will choose 11 as the pivot.

Now I must switch 11 with 19. Now we get [6, 11, 19].

Now the list to the right of -3 is sorted. The list (the [-5]) to the left of -3 is already correct. So now we get [-5, -3, 6, 11, 19, 25, 26, 42].

4) Shell Sort
[15, 14, -6, 10, 1, 15, -6, 0]

Step 1) Compare numbers 4 steps apart and swap if you need to.

So after the first pass:

[1, 14, -6, 0, 15, 15, -6, 10]

Next, I'm going to use use a smaller gap (2). So the name as number 1, but with 2 steps instead of 4.

After the second pass, I got:
[-6, 0, 1, 14, -6, 10, 15, 15]

Finally, we use regular insertion sort.
So the final result should be: [-6, -6, 0, 1, 10, 14, 15, 15]

5)

1st place: Quick Sort
2nd place: Merge Sort
3rd place: Shell Sort
4th place: Insertion Sort

1st, 2nd, and 3rd place are all O(n log n), whereas 4th place is O(n^2).

Shell sort really just builds on insertion sort, but it can also vary depending on the gap sequence. Therefore, I put it in 3rd place, just above insertion sort.

I put quick sort 1st, because if you choose a good pivot, it may become even more effective than merge sort, though merge sort is also very effective and efficient.

9)

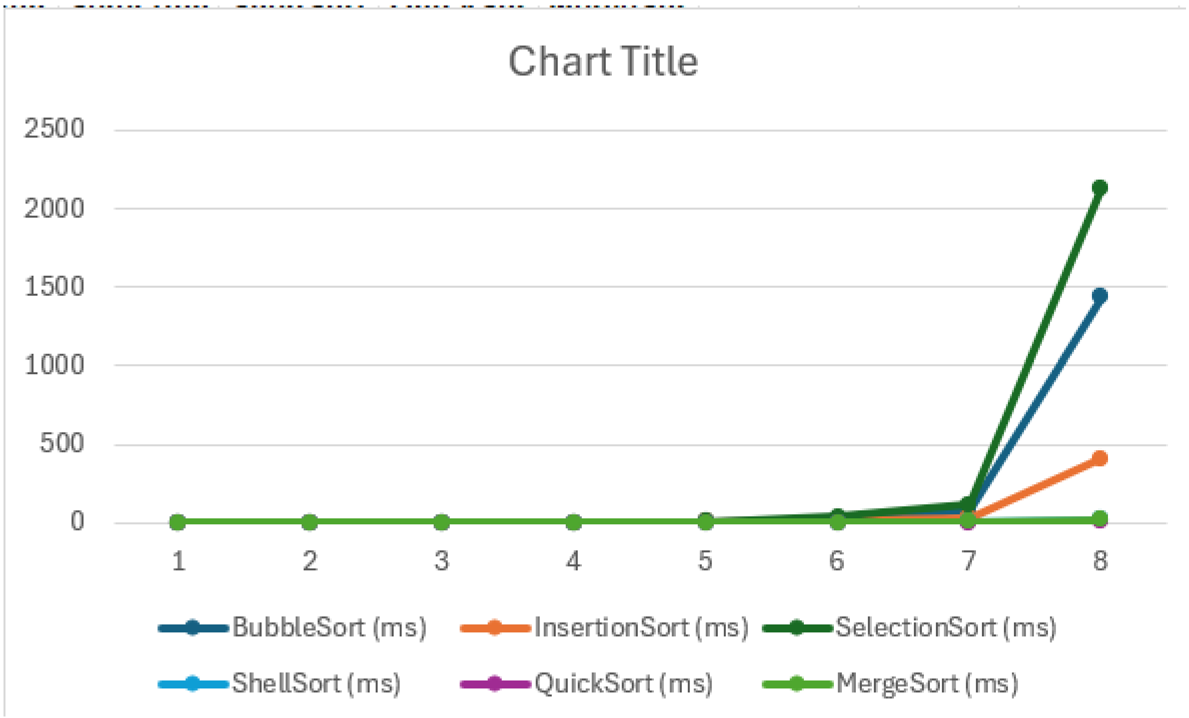
Chart Title
BubbleSort (ms)
InsertionSort (ms)
SelectionSort (ms)
ShellSort (ms)
QuickSort (ms)
MergeSort (ms)