

How many swaps does it take to get to the end of a sort?

This project is to implement three sorting algorithms within the Sorter class. Selection, Insertion, and Merge Sort.

Task 1: Create a main file and figure out how to instantiate Sorter objects. Read the runSorts() function to see how everything will come together once you have finished this project. You will only be able to test your project as you work if you understand how to use what has been given to you.

Task 2: Modify the Sorter class to implement the following sorting algorithms: Selection, Insertion & Merge. I know this is probably your first time seeing Comparators; for our purposes they are simple objects that evaluate to a boolean value. For example, if we have comp<-std::less<int>(), then comp(0, 1) will evaluate to true. Please reread the implementation of getPosOfMinOrMax() to see how we use comparators.

Hints: Use getPosOfMinOrMax() within selectionSort(), please choose position 1 as your pivot in partition(), and comment runSorts() out of the .hpp and .cpp as you do this task. You can make displayArr() a public function for testing purposes but please make it private once you move on to the following task.

Task 3: Modify selectionSort(), insertionSort() to increment their respective private count variables when a swap has taken place. Also, modify mergeSort() to increment its private count variable when a comparison between elements within the array it is processing is made. Once this has been completed, runSorts() should not crash your test file and should give you some deterministic output.

You will submit the following files: Sorter.cpp

Here is an example of what the output of runSorts() should look like if you finished all tasks correctly:

```
Sorter abc(RANDOM, 10);
def.runSorts(INCREASING);
```

```
***** Selection Sort *****
```

```
Original Array:
```

```
12 39 2 58 74 26 81 63 36 62
```

```
Sorted Array:
```

```
2 12 26 36 39 58 62 63 74 81
```

```
Number of swaps: 7
```

```
*****
```

```
***** Insertion Sort *****
```

```
Original Array:
```

```
91 24 29 37 77 19 54 11 16 75
```

```
Sorted Array:
```

```
11 16 19 24 29 37 54 75 77 91
```

```
Number of swaps: 27
```

```
*****
```

```
***** Merge Sort *****
```

```
Original Array:
```

```
37 9 75 60 1 78 89 88 6 73
```

```
Sorted Array:
```

```
1 6 9 37 60 73 75 78 88 89
```

```
Number of comparisons: 34
```

```
*****
```