Landon Holmes
lholmes7@my.athens.edu
A00116756
Assignment 10

```
Original: 5 2 9 1 5 6

QuickSort:
Sorted in 0.0006 ms
1 2 5 5 6 9

MergeSort:
Sorted in 0.0152 ms
1 2 5 5 6 9

BubbleSort:
Sorted in 0.0006 ms
1 2 5 5 6 9

Lambda Sort (std::function):
Sorted in 0.0022 ms
1 2 5 5 6 9
```

Output:

```
Simple / quicksort: 1 2 5 5 6 9
Functor / quicksort: 1 2 5 5 6 9
Functor / mergesort: 1 2 5 5 6 9
Functor / bubblesort: 1 2 5 5 6 9
```

In OOP, a strategy is chosen by giving an object another object that holds the behavior, and swapping the objects and algorithm at runtime. In FP, a strategy is selected by passing functions as values, so the behavior changes by giving a different function. Functions act as direct behavior, while objects wrap behavior inside methods. OCaml modules group related code, and functors create new modules from existing ones, giving them a role similar to class templates.