**Experiment Log**

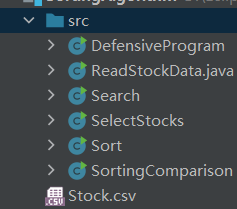
**Environment:**

Compiler: idea

Jdk environment: openjdk-17

**Directory Structure**:

Note that the src file is located in the same directory as Stock.csv



Part1： Q1 Sort.java Q2: SortingComparison.java Q3:Search

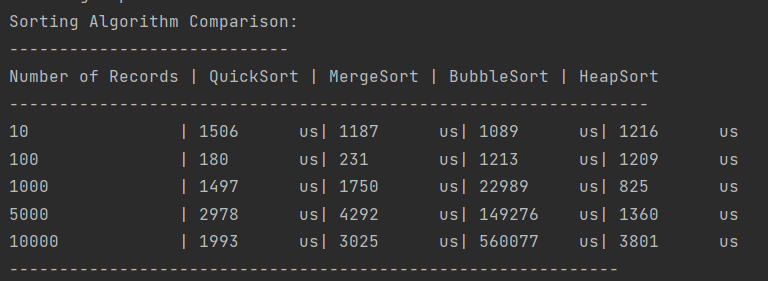
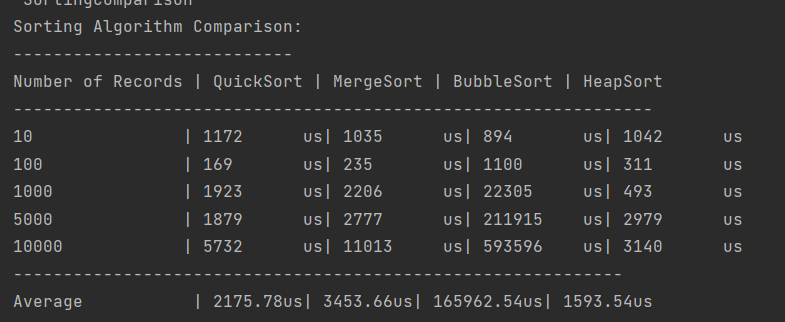
Part2: DefensiveProgram.java

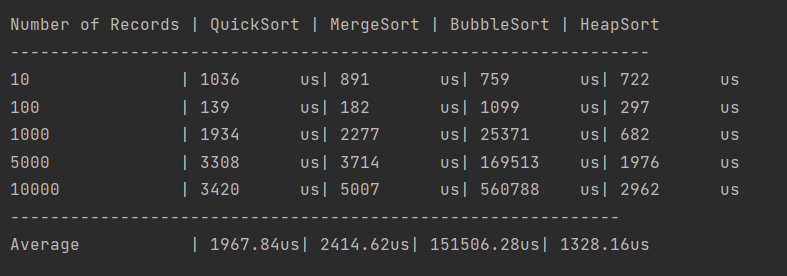
Part3： SelectStocks.java

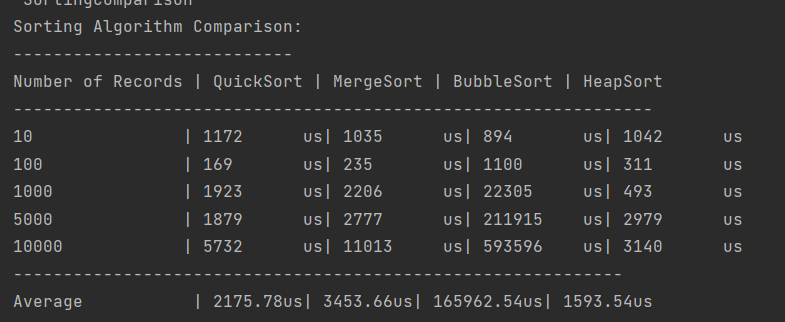
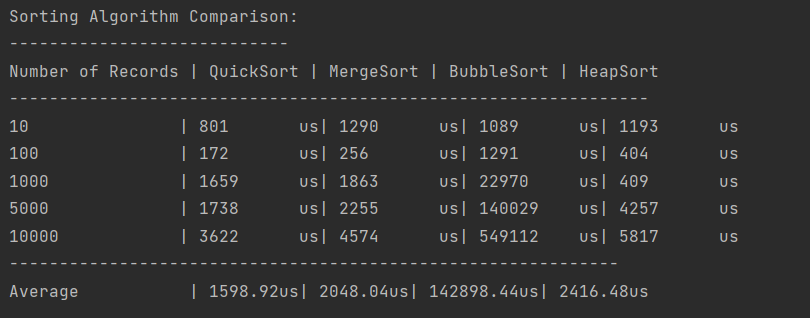
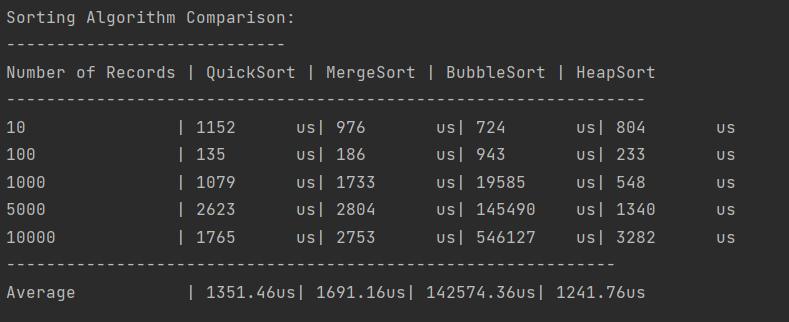
**Other notes:**

1. Sorting:

I tried four sorting methods, namely quick sort, merge sort, heap sort and bubble sort. The time complexity of the first three algorithms is O(nlogn), and because of their low time complexity, they are the main sorting methods I chose. Bubble sort is O(n^2), which can be used for comparison and reference.

I used random sampling for 10-1000 data subsets and conducted multiple tests (as shown below). Considering the time complexity and space complexity, I concluded that quick sort is more suitable for sorting this 10,000 stock.csv data set.





According to the data, the following brief analysis is made:

Quick sort is the fastest of the four sorting methods. On average, the time complexity of quick sort is O(nlogn), but in the worst case, the time complexity becomes O(n^2). Therefore, it can be seen that when the amount of data is small, there will be a longer sorting time, resulting in an increase in the average sorting time. However, for 10,000 data, quick sort is still the best.

Merge sort also has a time complexity of O(nlogn), but performs slightly worse than quick sort in the test data. Merge sort requires additional memory space to merge subarrays, which may affect its performance, especially when sorting large amounts of data.

For heap sort, it can be seen that when processing small amounts of data (10 and 100 records), the speed of heap sort is sometimes better than quick sort, but as the amount of data increases, the performance of heap sort begins to lag significantly behind quick sort.

In summary, quick sort is selected as the sorting method.

2. Part 3 implementation method:

The problem is to choose which goods can be loaded onto the truck to obtain the maximum profit given a limited truck weight limit. To solve this problem, the algorithm first sorts all the optional goods according to the profit ratio, which is the ratio of the profit of the goods to their weight, indicating the profit per unit weight of the goods. Then it starts to traverse from the sorted list of goods, selecting goods that fit the truck's load limit until the limit is exceeded or all goods have been considered. In this way, it can ensure that the total profit of the selected goods is maximized and does not exceed the truck's weight limit.