CALIFORNIA STATE UNIVERSITY, FRESNO
DEPARTMENT OF COMPUTER SCIENCE

Class:	**Algorithms & Data Structures**	Semester:	**Fall 2023**
Points		Document author:	**Pratham Aggarwal**
		Author’s email:	**Pratham_aggarwal@mail.fresnostate.edu email**
		Laboratory number:	**Lab 3**

1. Statement of Objectives

This lab compares and implements the execution times of the two sorting algorithms: Selection Sort and Merge Sort. The objective of the lab is to determine how long it takes for these algorithms to run under the following listed conditions: sorted, half-sorted, and reversed.

2. Experimental Procedure

1. First generate an array of integers with a specified size (either 1000 or 20, depending on the version of the code). When the size was 20 the time was coming out to be 0 and the same for size 100 as well, so I used random generator for size 1000 to perform the operations.
2. Sort the original array using Selection Sort and measure the time taken for sorting.
3. Reset the array to its original state.
4. Sort the original array using Merge Sort and measure the time taken for sorting.
5. Repeating steps 2-4 for a sorted array, a half-sorted array, and a reversed array.
6. I have commented the part where I am not using the random generator for the array in the end.

3. Analysis

1. Unsorted array
    - Selection Sort: 2990 microseconds
    - Merge Sort: 959 microseconds
2. **Sorted Array**:
    - Selection Sort: 2990 microseconds
    - Merge Sort: 996 microseconds
3. **Half-Sorted Array**:
    - Selection Sort: 2992 microseconds
    - Merge Sort: 997 microseconds
4. **Reversed Array**:
    - Selection Sort: 3992 microseconds
    - Merge Sort: 1035 microseconds

- Selection Sort consistently takes a longer time to sort arrays compared to Merge Sort.
- This behavior is consistent with the time complexity of Selection Sort, which is O(n^2) whereas for merge sort is O(nlogn) which remain efficient even with different initial orders of the array.

Screenshot in the end of the report.

4. Encountered Problems

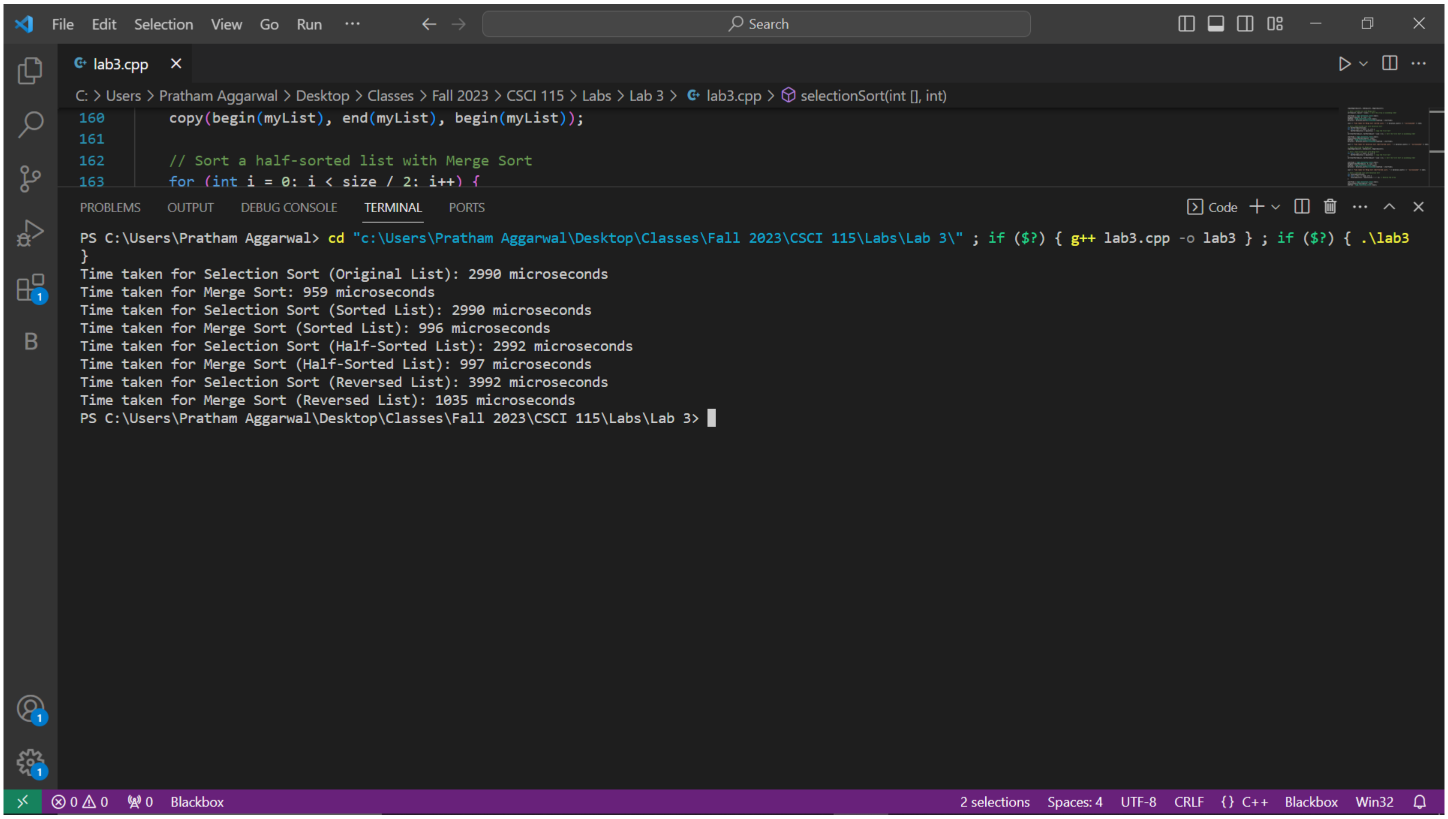
Firstly, I encountered problems in the pseudo codes of the sorting algorithm and for the merge function. Then, when I was trying an array of size 20, it was resulting in 0 microseconds execution times. So, to obtain meaningful results, I increased the array size to 1000 and used random generator to get the array. Then to get the other parts of the question I had to see the Sort() function, which sorted the original array into the given conditions for the question. I had to find out how to reset the array to the original state after performing different operations on it. Lastly, I had to look up how to use time function to calculate the time at each step. I took help of tutors and other coding platforms and website to fix these errors.

5. Conclusions

Selection Sort is a straightforward sorting algorithm with a time complexity of O(n^2), making it less efficient for large datasets.
Merge Sort is a more efficient sorting algorithm with a time complexity of O(n log n), making it suitable for larger datasets.
Selection Sort is significantly impacted by the original order of the array's elements, with reversed and half-sorted arrays taking far longer to sort than randomly generated and sorted arrays. Whereas merge sort, remains consistent throughout.

6. References

N/A



```
    copy(begin(myList), end(myList), begin(myList));

    // Sort a half-sorted list with Merge Sort
    for (int i = 0; i < size / 2; i++) {
```

```
PS C:\Users\Pratham Aggarwal> cd "c:\Users\Pratham Aggarwal\Desktop\Classes\Fall 2023\CSCI 115\Labs\Lab 3\" ; if ($?) { g++ lab3.cpp -o lab3 } ; if ($?) { .\lab3
}
Time taken for Selection Sort (Original List): 2990 microseconds
Time taken for Merge Sort: 959 microseconds
Time taken for Selection Sort (Sorted List): 2990 microseconds
Time taken for Merge Sort (Sorted List): 996 microseconds
Time taken for Selection Sort (Half-Sorted List): 2992 microseconds
Time taken for Merge Sort (Half-Sorted List): 997 microseconds
Time taken for Selection Sort (Reversed List): 3992 microseconds
Time taken for Merge Sort (Reversed List): 1035 microseconds
PS C:\Users\Pratham Aggarwal\Desktop\Classes\Fall 2023\CSCI 115\Labs\Lab 3>
```