Title: Sorting Analysis Report
Author: Hilalnur Beral
ID: 11549136358
Section: 01
Assignment: 2
Description: Report

Experimental Setup

- First of all, I created 3 arrays. These arrays are ordered ascending, descending and with random integers relatively. In the jar file there were 5 sorting algorithms that we don't know the name of them. That's why, I tried to find the execution time of each of these five sorting algorithms. Then I predicted the names of the algorithms from the elapsed time.

- For time calculation, I used System.nanoTime().
  To calculate every sorting algorithm's execution time, I assigned a variable to System.nanoTime(). Then I called the corresponding sorting algortihm and then I called System.nanoTime() again and subtracted the variable from it.

For example:

```
start1 = System.nanoTime();

SortingAlgorithms.sort1(a, 1234567801L);

time = System.nanoTime() - start1;
```

Procedure

I created:

1.) Arrays in ascending order

2.) Arrays in descending order.

3.) Array that includes randomly generated inputs.

I used sorting algorithms on to these array by one by then I compared the results. You can see my compared results below.

Experimental Results

Conclusion

1.) In the **descending ordered array** fastest algortihm is sort4 that's why it could be an quick sort. Also in the **ascending ordered array** slowest algortihm is sort1 That' why it could be an quick sort algorithm. Another point is that sort1 could be selection sort because it is the slowest algorithm in **randomly generated input arrays.**

2.) In the **descending ordered array** second fastest algorithm is sort 5. That's why it could be merge sort. The reason for this is that , arrays which include less elements and in **descending ordered array**, quick sort is the fastest algortihm and merge sort is the second fastest algorithm. Also in our experiment , our array size is small that's why we get these results. But also, In the **ascending ordered array**, fastest algorithm is sort5. That'why it could be bubble sort.The reason for this is that bubble sort is the fastest algorithm if the array is ordered in ascending order.

3.) In the **ascending ordered array**, second fastest algorithm is sort2. That'why it could be the insertion sort.

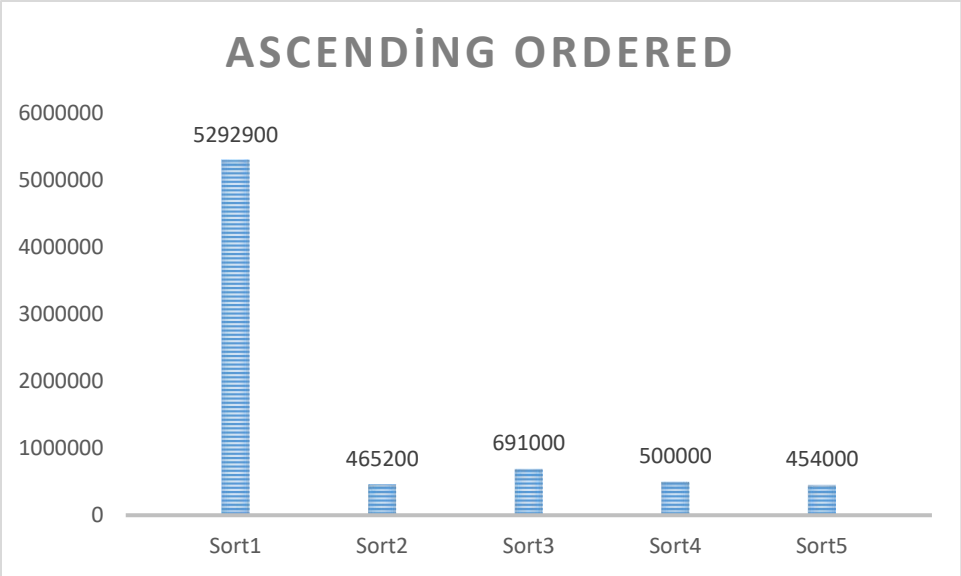
4.) Sort3 could be the selection sort because it is not as fast as bubble sort, not as slow as quick sort in the **ascending ordered array**. Bubble sort is fast when array is already sorted in ascending order.

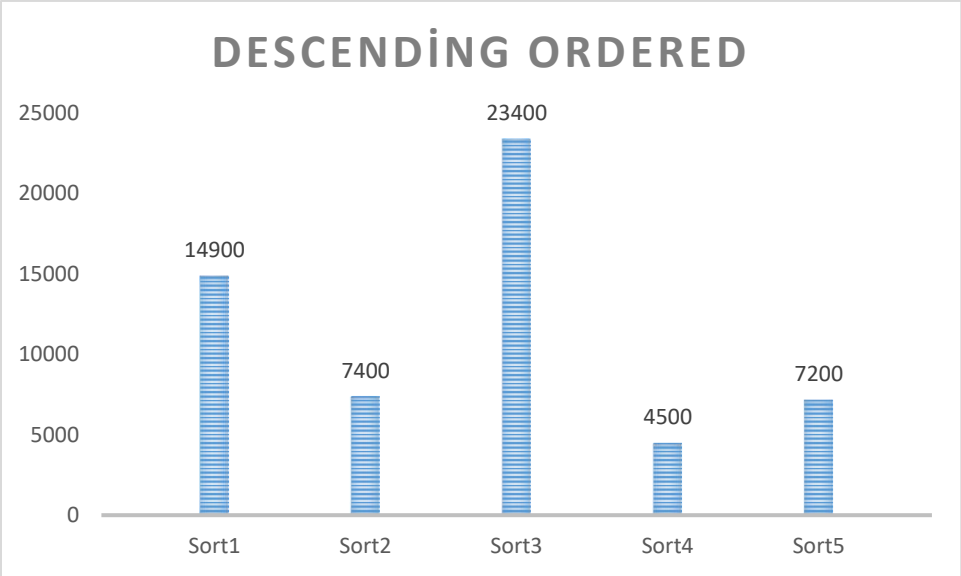
Of course these results may differ. The reason for this is that, the speed of these algorithms may vary from computer to computer, even programs running in the background can affect speed.

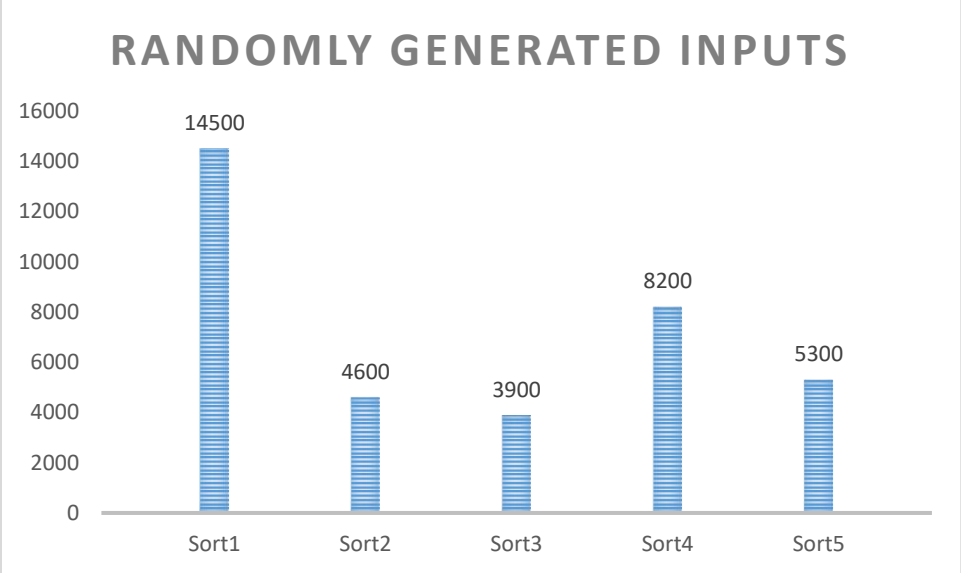
Before I started this assignment, I thought everyone would get the same results. However,when I talked to my friends later, I realized that everyone can have different results. For example, I said that sort1 is could be quick sort but my another friend said that sort1 cannot be quick sort in my experiment because I didnot get any such value.

Another important point is that one sorting algortihm algortihm can be related to more than one sorting algortihms. What I mean is that for example sort1 could be either quick sort algorithm or selection sort algorithm beause in **ascending ordered arrays** and **randomly generated input arrays** there are different situations for this experiment.

	Sort1	Sort2	Sort3	Sort4	Sort5
Ascending Ordered	5292900	465200	691000	500000	454000
Descending Ordered	14900	7400	23400	4500	7200
Randomly Generated Inputs	14500	4600	3900	8200	5300


ASCENDİNG ORDERED
6000000
5000000
4000000
3000000
2000000
1000000
0
5292900
465200
691000
500000
454000
Sort1
Sort2
Sort3
Sort4
Sort5


DESCENDİNG ORDERED
25000
20000
15000
10000
5000
0
14900
7400
23400
4500
7200
Sort1
Sort2
Sort3
Sort4
Sort5


RANDOMLY GENERATED INPUTS
16000
14000
12000
10000
8000
6000
4000
2000
0
14500
4600
3900
8200
5300
Sort1
Sort2
Sort3
Sort4
Sort5