CO 322 – Lab 01

Lab Report

E/18/180 – Kodithuwakku M.K.N.M

1. Output for the different size of arrays for three sorting algorithm as following. I used sizes of 100,1000,10000,100000 elements for test the performance.

Using best case dataset of 100 elements

Time taken for best Array of 100 elements to sort

Bubble Sort = 35100

Selection Sort = 460399

Inserion Sort = 6500

==========

Using Worst case dataset of 100 elements

Time taken for worst Array of 100 elements to sort

Bubble Sort = 670299

Selection Sort = 192701

Inserion Sort = 4300

==========

Using best case dataset of 1000 elements

Time taken for best Array of 1000 elements to sort

Bubble Sort = 60400

Selection Sort = 4673801

Inserion Sort = 73900

==========

Using Worst case dataset of 1000 elements

Time taken for worst Array of 1000 elements to sort

Bubble Sort = 7029200

Selection Sort = 315100

Inserion Sort = 50800

==========

Using best case dataset of 10000 elements

Time taken for best Array of 10000 elements to sort

Bubble Sort = 71200

Selection Sort = 28676799

Inserion Sort = 472100

==========

Using Worst case dataset of 10000 elements

Time taken for worst Array of 10000 elements to sort

Bubble Sort = 120663800

Selection Sort = 16769199

Inserion Sort = 404500

==========

Using best case dataset of 100000 elements

Time taken for best Array of 100000 elements to sort

Bubble Sort = 64000

Selection Sort = 1933058199

Inserion Sort = 2202700

==========

Using Worst case dataset of 100000 elements

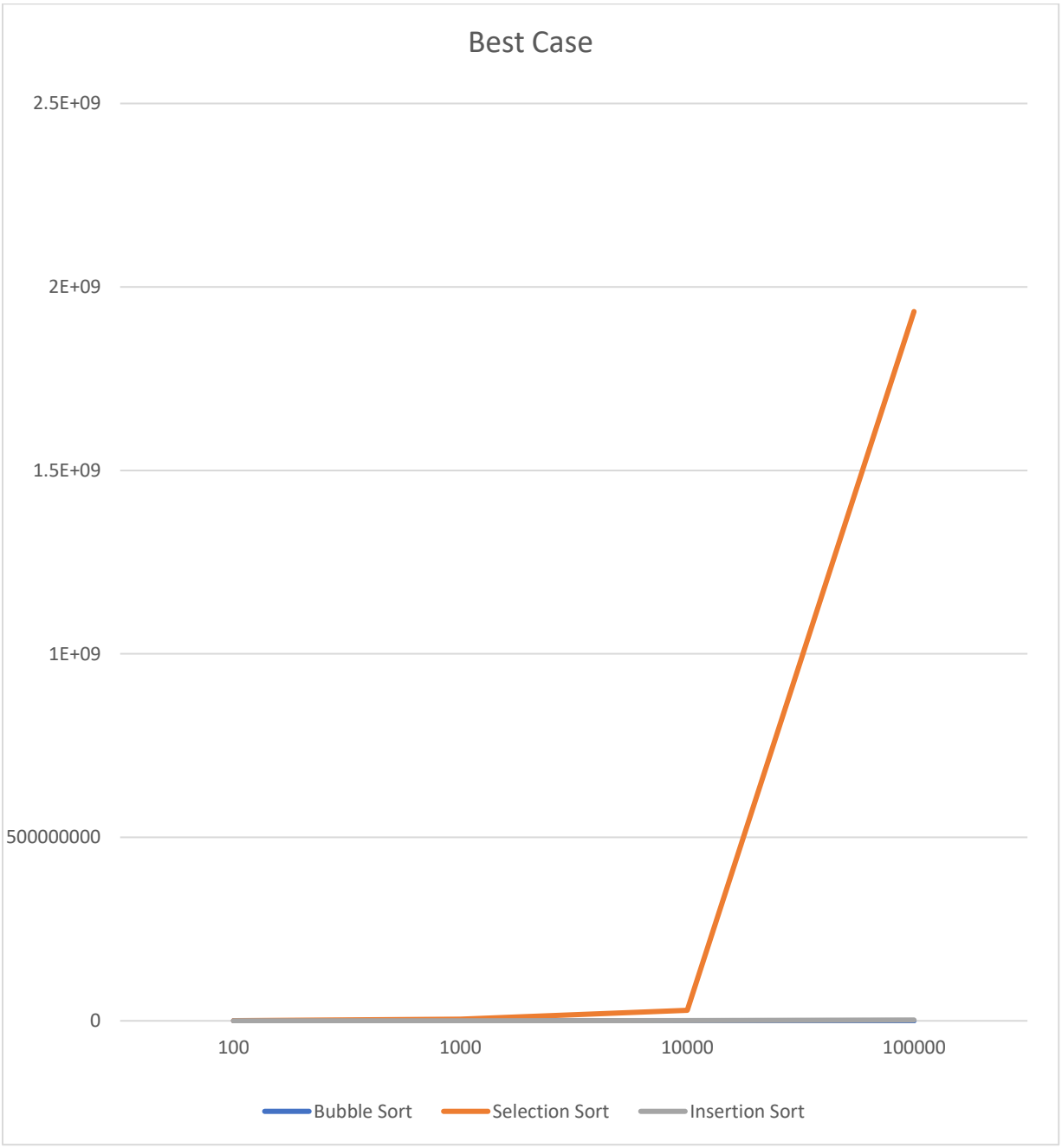
Time taken for worst Array of 100000 elements to sort

Bubble Sort = 8248141800

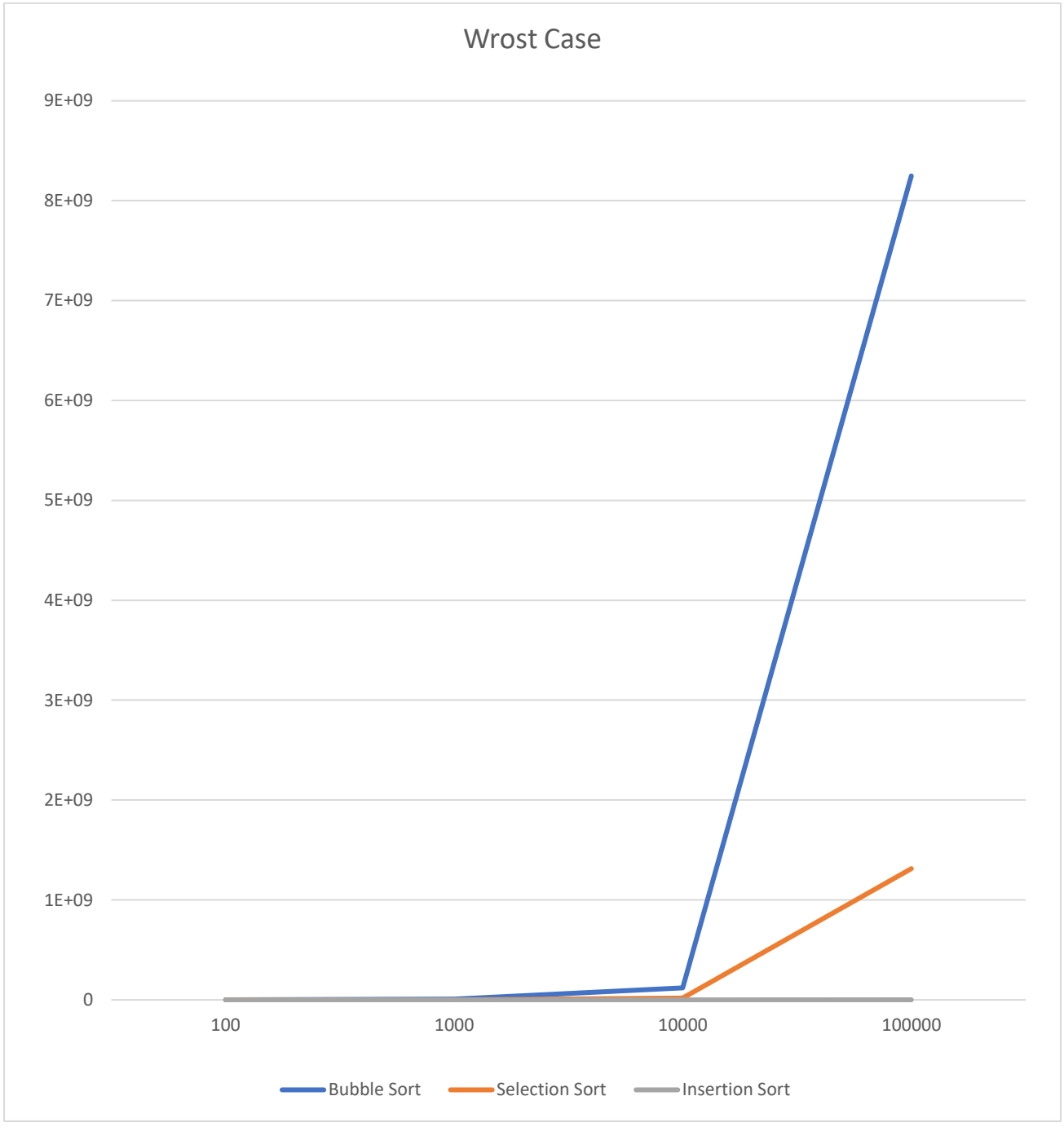
Selection Sort = 1313366599

Inserion Sort = 597100

- Considering the best-case scenario for all three algorithms



- Considering the worst-case scenario for all three algorithms



Observations:

- Time was measured using nanoseconds.
- For Best case scenarios both Bubble sort and insertion sort performed well except for selection sort for all array sizes.
- When array size increases for average scenarios bubble sorting took much time.
- For worst-case scenarios selection and insertion sort is given the best results for small array sizes.
- When array size increases in worst cases all three sorting algorithms took much time.

2. Considering the empirical results and theoretical results.

- Theoretical results.

Algorithm	Time Complexity	
	Best	Worst
Bubble Sort	$\Omega(n)$	$O(n^2)$
Selection Sort	$\Omega(n^2)$	$O(n^2)$
Insertion Sort	$\Omega(n)$	$O(n^2)$

According to the above table we can see the time complexities of all three algorithms are somewhat same as the theoretical results.

For the bubble sort algorithm, the worst case has a time complexity of big O of N2 theoretically. The practical results are not exactly that, but somewhat similar. In theory, when N gets doubled, running time should be 4 times the previous. But this is not the exact result but very similar for that. For the best case, bubble sort has N time complexity but in the practical scenario, this is not the case. In a practical scenario, the running time is high for smaller inputs.

If the selection sort is considered, it can see that the practical results almost agree with the theoretical results. As the theory suggests, the practical graph takes N2 behavior for both the best and worst-case scenarios.

For the insertion sort algorithm, the worst case has a time complexity of big O of N2 . The theoretical results are not exactly that, but somewhat similar. The only difference with the theoretical result is that the graph tends to be constant for medium input sizes. For the best-case bubble, sort has N time complexity but in the practical scenario, this is not the case.

3.

For the testing we should identify that the all algorithms can sorting any of array correctly and also all algorithms can be sorted arrays with any of size. For that we should test our algorithms according to various sizes of input arrays. For the correctness of the sorting we can use the issorted() function in java.

If the selection sort is considered, it can see that the practical results almost agree with the theoretical results. As the theory suggests, the practical graph takes N2 behavior for both the best and worst-case scenarios. For the insertion sort algorithm, the worst case has a time complexity of big O of N2 . The theoretical results are not exactly that, but somewhat similar. The only difference with the theoretical result is that the graph tends to be constant for medium input sizes. For the best-case bubble, sort has N time complexity but in the practical scenario, this is not the case.