

Report Lab 1

LE Quoc Hung

24/09/2024

C Programming and Unix

Introduction

We already know some slow sorting algorithms such as bubble sort, insertion sort and selection sort. However, which is the best one? Which takes less time to sort compared to others? What is the relation between the execution time and the number of elements in an array? We will conduct some experiments with programming language C.

Methodology

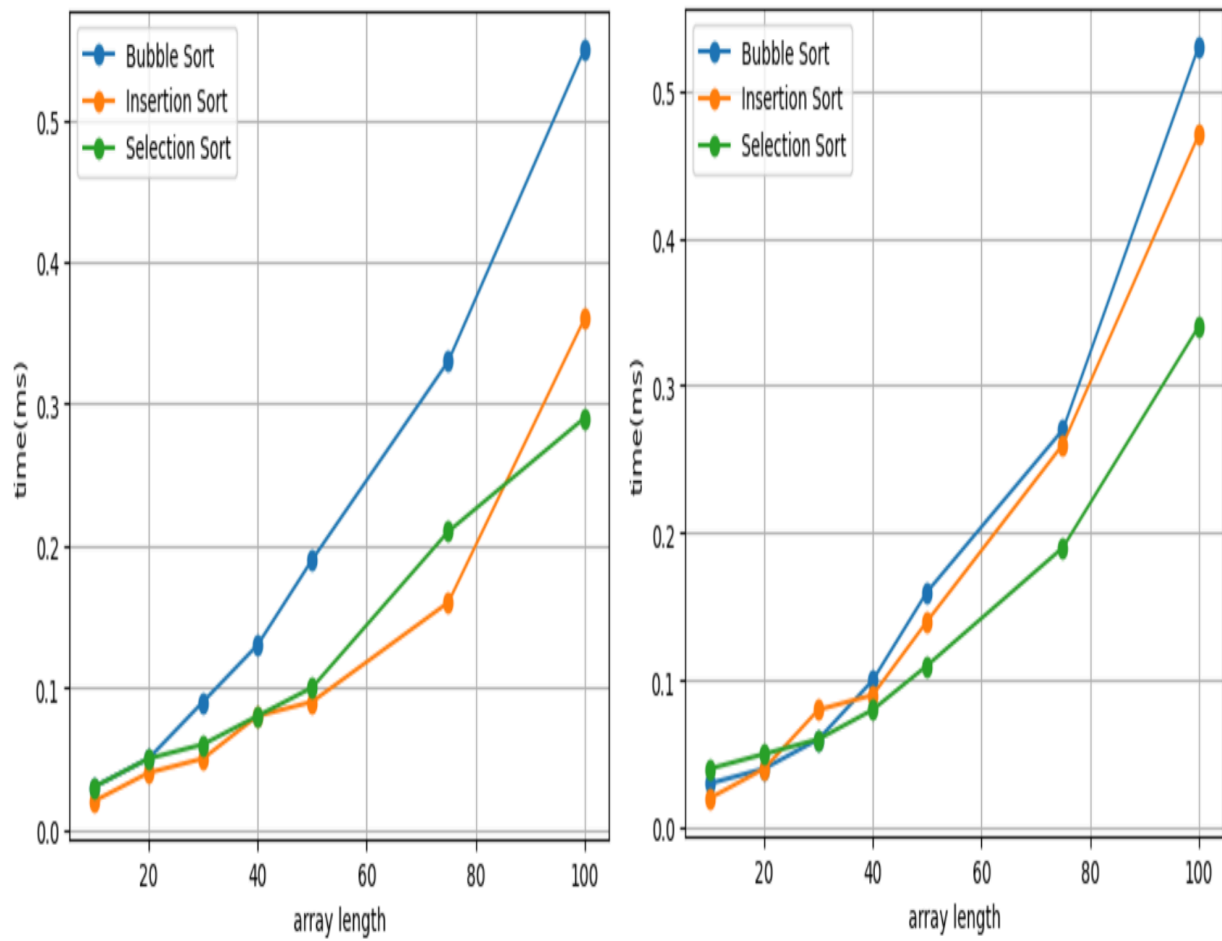
We compile the C file named `sorting.c` by gcc, change 3 lines consequently and see the results.

Link to my github: <https://github.com/hunglqdz/c-unix>

Results

For array filled with random numbers (left figure):

Array length	10	20	30	40	50	75	100
Bubble Sort	0.03	0.05	0.09	0.13	0.19	0.33	0.55
Insertion Sort	0.02	0.04	0.05	0.08	0.09	0.16	0.36
Selection Sort	0.03	0.05	0.06	0.08	0.1	0.21	0.29



For array filled with random numbers but in reverse (descending) order (right figure):

Array length	10	20	30	40	50	75	100
Bubble Sort	0.03	0.04	0.06	0.1	0.16	0.27	0.53
Insertion Sort	0.02	0.04	0.08	0.09	0.14	0.26	0.47
Selection Sort	0.04	0.05	0.06	0.08	0.11	0.19	0.34

Conclusion

In both cases (in terms of the order of random numbers), the bubble sort algorithm has the highest execution time among the three. Although there are some intersections, we can predict that with higher and higher array length (about 1000), the selection sort algorithm will be better than the insertion sort algorithm.