Report

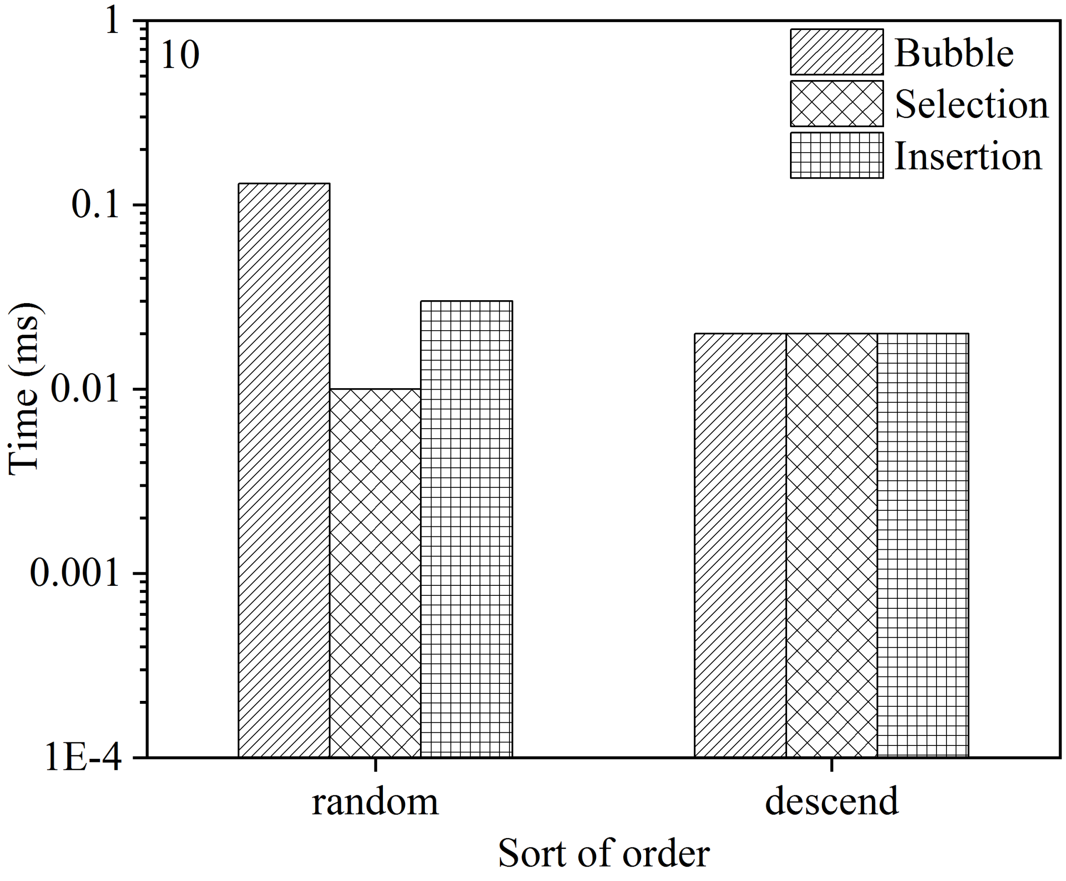
My code first generates a random number which implements bubble sort, insertion sort and selection sort algorithms for a random number. It also implements a descending padding of a specified array length and implements each sort algorithm to measure the average time to process, generate a random length array and sort the descending sort array.

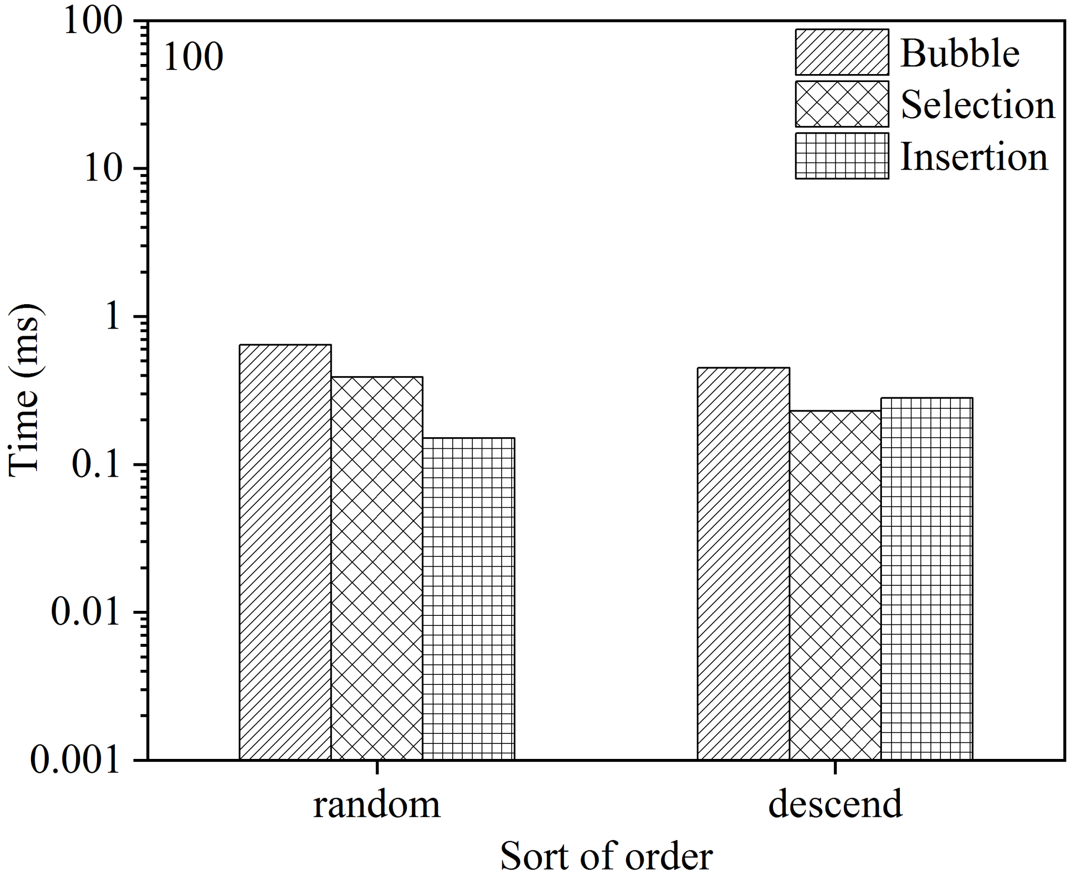
About bubble sort, the range of bubble sort is 0 to N-1. It is to compare the first number with the second number, and if the previous number is greater than the next exchange the two numbers, otherwise not. Then continue to compare the second number and the third number, according to the previous rule in turn. When the sorting range becomes 0 to N-2, the second largest number in the array is placed in the penultimate position in the array. After another swap, it finally becomes an ordered number.

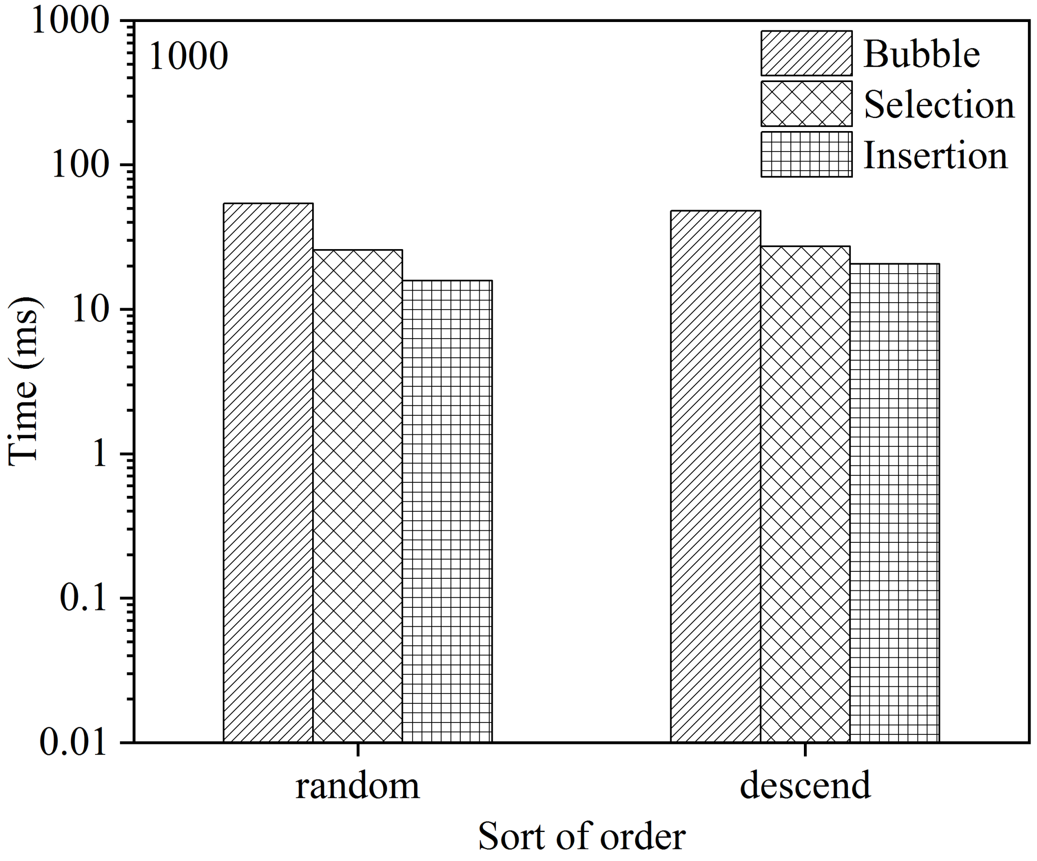
About insertion sort, insertion sort is a sort from smallest to largest, where the first number of an array is considered as an ordered array, and the ordered array is scanned from back to front or front to back. And the remaining N-1 numbers in the array are inserted into the ordered array, depending on the size of the values until all the numbers in the array are ordered.

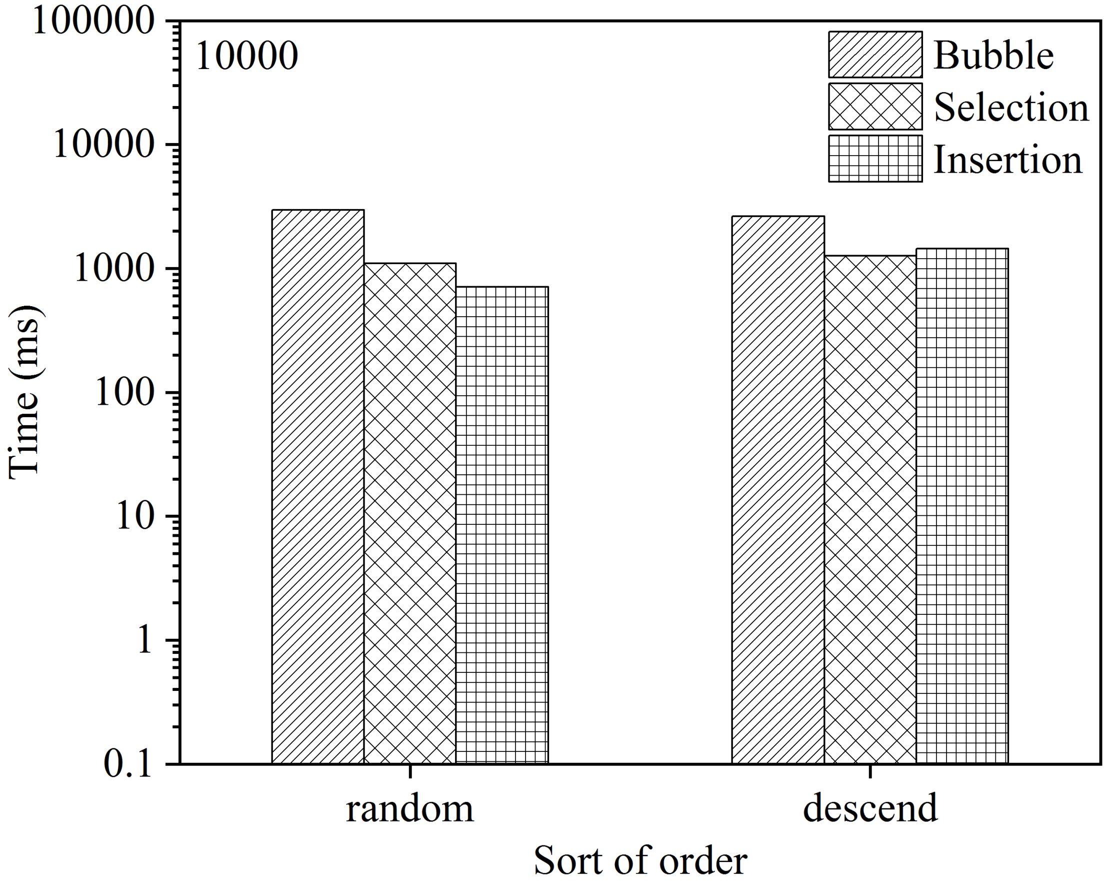
About selection sorting, the selection sort is a smallest-to-largest sort. At first select a minimum value from the interval 0 to N-1 and place it on position 0, then select the minimum value on the range 1 to N-1 and place it on position 1. Repeat the process until the last element is left, and the array is ordered.

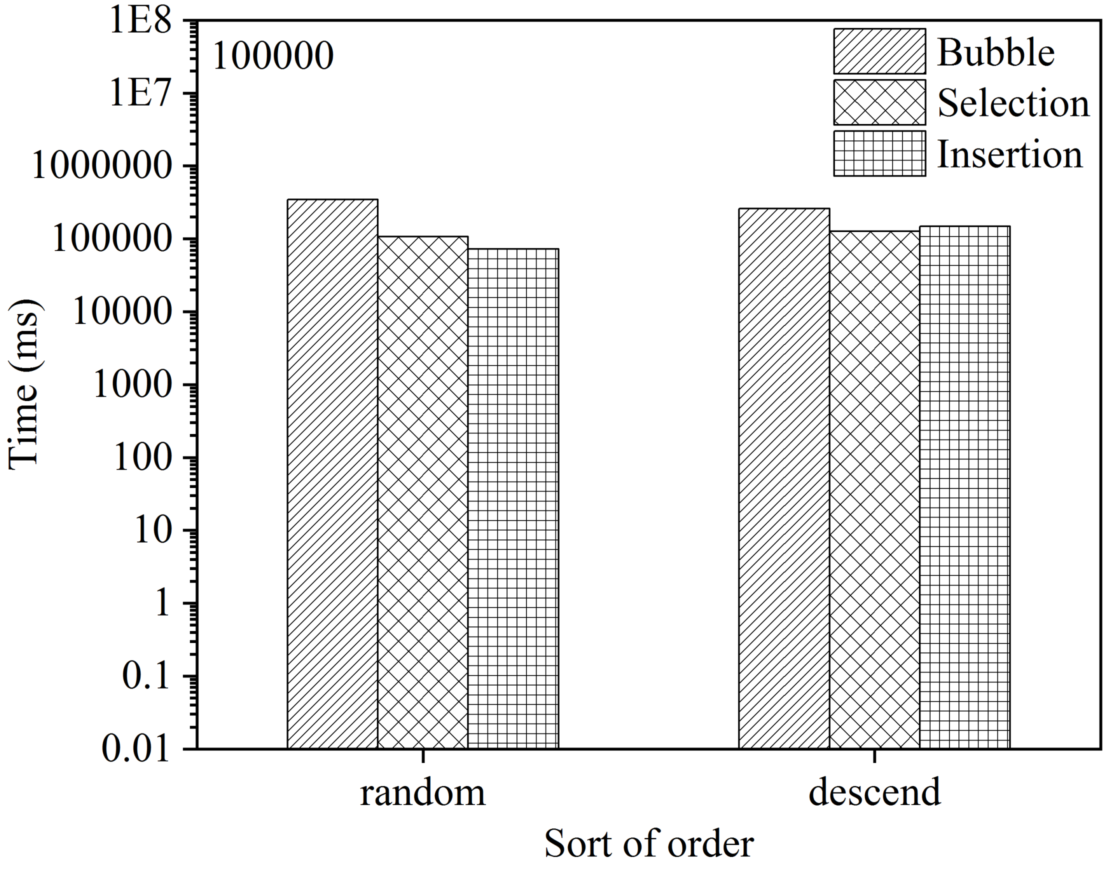
We can visualize the efficiency of different sorting algorithms by using the data in the following graphs.




100
Bubble
Selection
Insertion
Time (ms)
100
10
1
0.1
0.01
0.001
random
descend
Sort of order


1000
Bubble
Selection
Insertion
Time (ms)
1000
100
10
1
0.1
0.01
random
descend
Sort of order


10000
Time (ms)
100000
10000
1000
100
10
1
0.1
Bubble
Selection
Insertion
random
descend
Sort of order


100000
Time (ms)
1E8
1E7
1000000
100000
10000
1000
100
10
1
0.1
0.01
Bubble
Selection
Insertion
random
descend
Sort of order

From the above data, it can be seen that among the random generated numbers. Generally speaking the bubble sort takes the most time and the insertion sort takes the least time. In descend order, generally speaking the time difference between the three kinds of sort is not much. The larger the number the more general it is. I will take 100000 as an example, the time for bubble sort in random generated numbers is 4.78 times longer than the time for insertion sort. The time for bubble sort is 3.23 times longer than the time for selective sort. In descend order, the time of bubble sort is 1.75 times that of insertion sort and the time of bubble sort is 2.04 times that of selection sort. We can conclude that the time difference between the three sorts is not too big, and the bubble sort is the relatively slowest sorting algorithm because the bubble sort takes longer as it repeats the comparison each time. Insertion sort inserts the values in a sequence into an already sorted sequence until the end of that sequence. It is faster than bubble sort, but it is generally not suitable for situations where the amount of data is relatively large or where there are more repetitions of data. Insertion sort is faster than bubble sort and insertion sort. The time complexity of all three sorts is O(n^2) in the random case, and O(n^2) in the descend case.