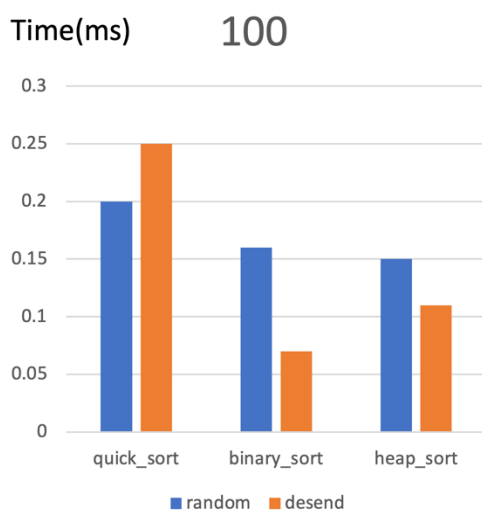
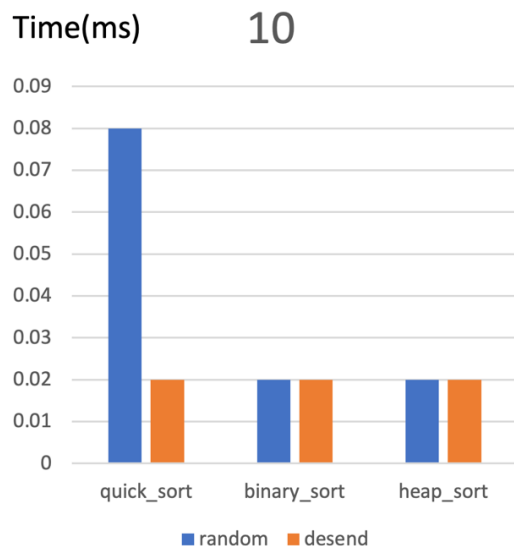


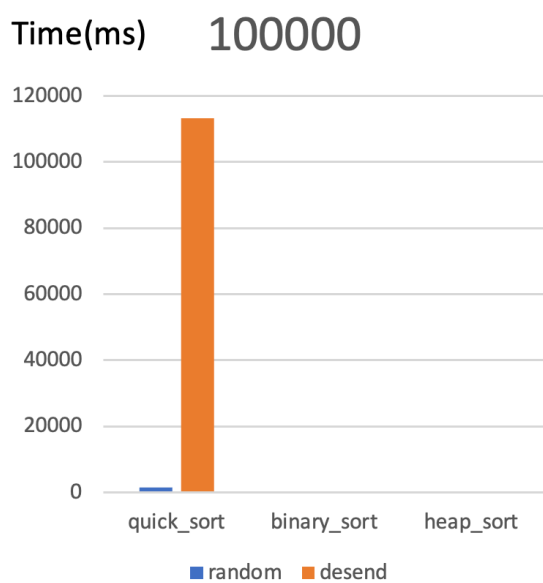
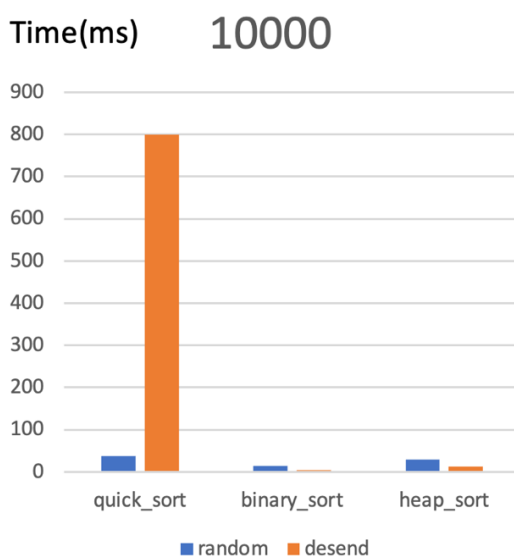
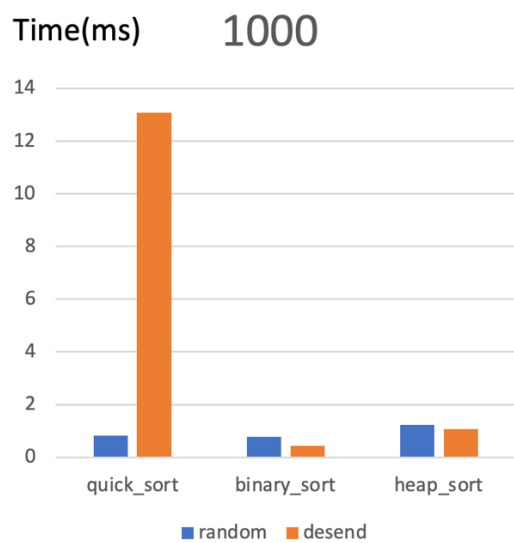
## Report

My code first generates a random number which implements quicksort, binary sort and heapsort 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 quicksort, the data to be sorted is divided into two independent parts by setting a pivot, and I set the pivot to the left most part. Then the two parts of data are sorted separately, and the whole sorting process can be done recursively, so that the whole data becomes an ordered sequence. About binary sort, the time complexity in the best case is  $O(n * k)$ , and the value is always stable. About heapsort, its time complexity is  $O(n \log n)$ . Among quicksort, binary sort, and heapsort, the binary sort and heapsort algorithms are stable, and the complexity of the quicksort algorithm increases with the increase of random numbers. It can be seen that quicksort is very unstable. This is because quicksort increases the time complexity as the position of pivot varies.

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





It can be visually found from the above picture that the sorting time of quicksort increases significantly incrementally with the increase of random number. And the difference between binary sort and heapsort sorting time is not much, even as the random number increases, compared with quicksort, the sorting time on the picture is so less that it cannot be shown on the picture, this is because the time of binary sort and heapsort sorting is so much less than quicksort that it cannot be shown on the last picture data.

I also compared quicksort, binary sort and heapsort with bubble sort, insertion sort and selection sort in the last experiment, and then found that, overall, quicksort, binary sort and heapsort sort would significantly faster than bubble sort, insertion sort and selection sort, but when the worst case of quicksort, it is similar to the time complexity of bubble sort. So, quicksort is the most unstable.