I. Bubble Sort

Bubble Sort is a simple sorting algorithm that repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order. The algorithm repeats this process until the list is sorted.The best case occurs when the list is already sorted. In this case, the algorithm will pass through the entire list once, but no swaps will be made. However, the algorithm still needs to check if any swaps are needed, so it takes O(n) time, where n is the number of elements.The worst case occurs when the list is in reverse order. The algorithm needs to compare and swap each pair of elements multiple times. In this case, the time complexity is O(n^2) because there are two nested loops: one for iterating over the elements, and another for comparing each adjacent pair.In the average case, the elements are randomly ordered. On average, each element needs to be compared and swapped multiple times, leading to O(n^2) time complexity.

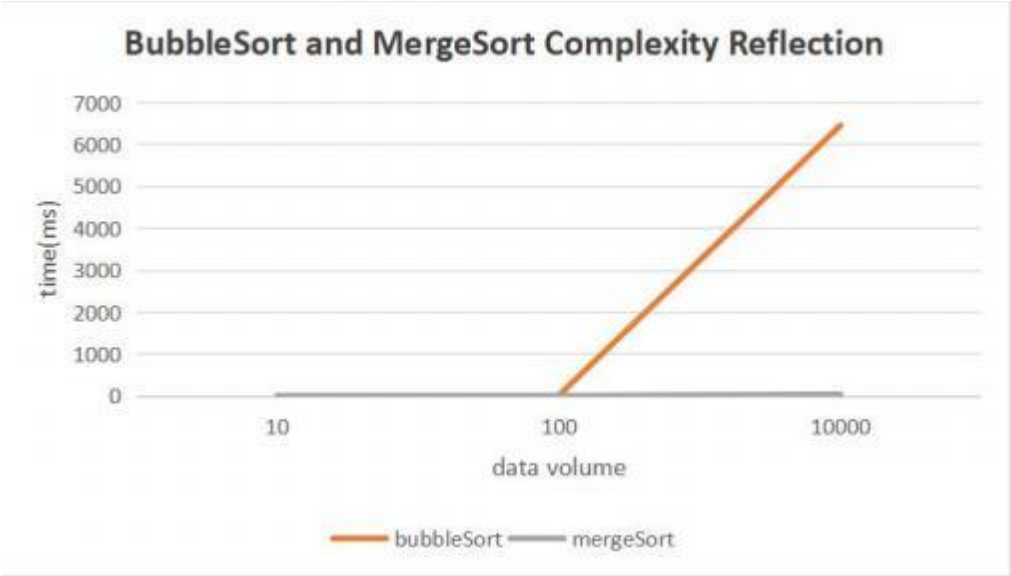
II. Merge Sort

No matter the best case and the worst case ,the time complexity is O(n log n). The array is divided into two halves recursively, which takes O(log n) levels of recursion. Next is the merge operation.At each level, the merging of two halves requires O(n) time. Since this merging happens at every level of recursion, and there are log n levels, the total time complexity is O(n log n). For the space complexity,merge Sort also requires additional space for the temporary arrays used during the merge step, which makes its space complexity O(n).

III.Experiment

We put different volume of data which contains 10, 100, 10000 card data respectively.Following are the time caused by each algorithm.

	10	100	10000
BubbleSort	1	12	6449
MergeSort	0	1	33



IV. BubbleSort and MergeSort Complexity Reflection

It can be easily found that the difference between mergesort and bubble sort is very small when the amount of data is very small.For example,when processing “sort10.txt” ,the time cost by two algorithm is nearly the same.Bubblesort cost 1ms and mergesort cost 0ms(ecause this is a nanosecond secretary). Similarly, when processing “sort100.txt”,bubblesort costs 12ms and mergesort costs 1ms.Merge sort is 11 milliseconds faster than bubble sort.However when the amount of data become big,the difference between mergesort and bubblesort become huge.For example ,when the amount of data come to 10000,time cost by bubblesort is over 6s,but mergesort only costs

0.033s,the difference value between mergesort and bubblesort is over 6s.Different time complexity of different algorithm is the reason of this.The time complexity of bubblesort is O(n2) ,the time complexity of mergesort is O(n log n).As the data volume increasing,the time complexity of bubblesort increases sharply.However,compared to bubblesort, the time complexity of mergesort increases slowly and smoothly.So, let's draw a conclusion ,although both bubblesort and mergesort can implement sort task,but it shows different performance when they handle different amount of data.When the amount of data is very small , two algorithm are almost same,but when the amount of data increase ,we'd better choose mergesort because it has better performance.

to 10000,time cost by bubblesort is over 6s,but mergesort only costs 0.033s,the difference value between mergesort and bubblesort is over 6s. Different time complexity of different algorithm is the reason of this.The time complexity of bubblesort is O(n^2) ,the time complexity of mergesort is O(n log n).As the data volume increasing,the time complexity of bubblesort increases sharply.However,compared to bubblesort, the time complexity of mergesort increases slowly and smoothly.So, let's draw a conclusion ,although both bubblesort and mergesort can implement sort task,but it shows different performance when they handle different amount of data.When the amount of data is very small , two algorithm are almost same,but when the amount of data increase ,we'd better choose mergesort because it has better performance.