Summary Report

CS665
Zhuo Liu
zl9901

For large amount of data, quicksort is among one of the fastest algorithms and the insertion sort is used for some kind of list which is already sorted. In my report, I combine two methods together which can be effective and can reduce the time complexity.

At first we use quicksort algorithm. Quicksort takes O(log n) time to sort the list but the worst case can take O(n^2) time complexity. In this worst case, if we use the insertion sort algorithm to sort the list which is almost ordered, the problem will be quite simple and easy to deal with.
When the quicksort works, it divides the list into many small parts. When the list turns into small one, the insertion sort can be used and efficient.

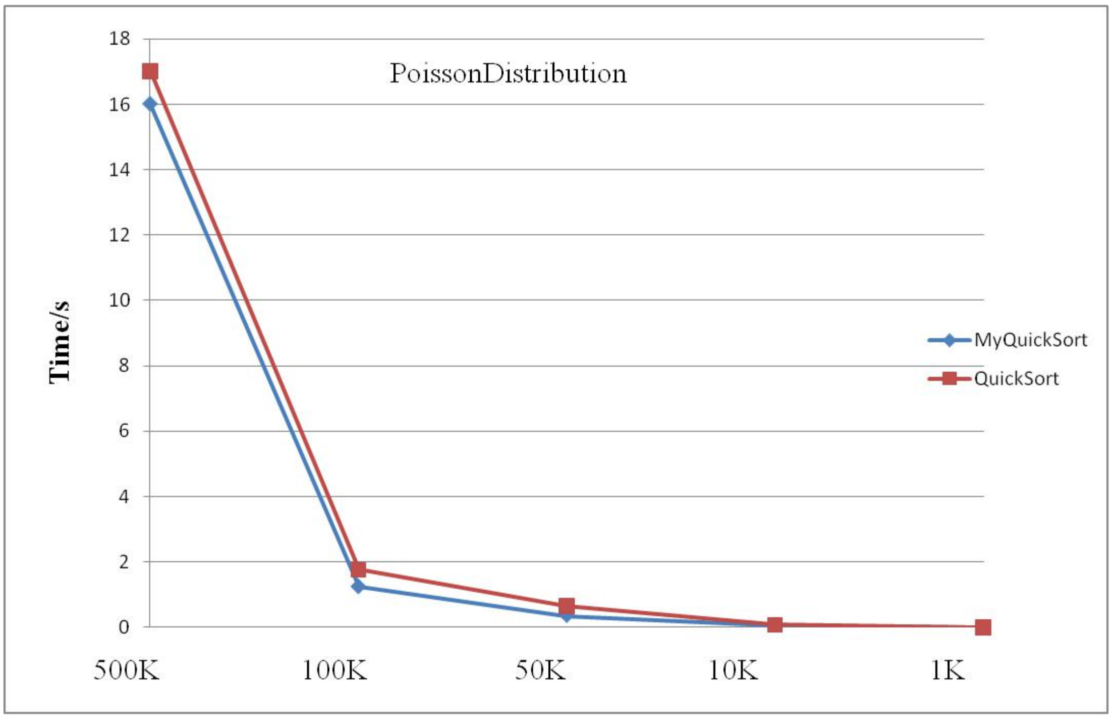

Figure 1. Poisson Distribution comparison of Quicksort and MyQuicksort

As the figure shown above, we can see that when the size of data becomes smaller, the algorithm will take less time. In the meanwhile, MyQuickSort can be more efficient than the original QuickSort.

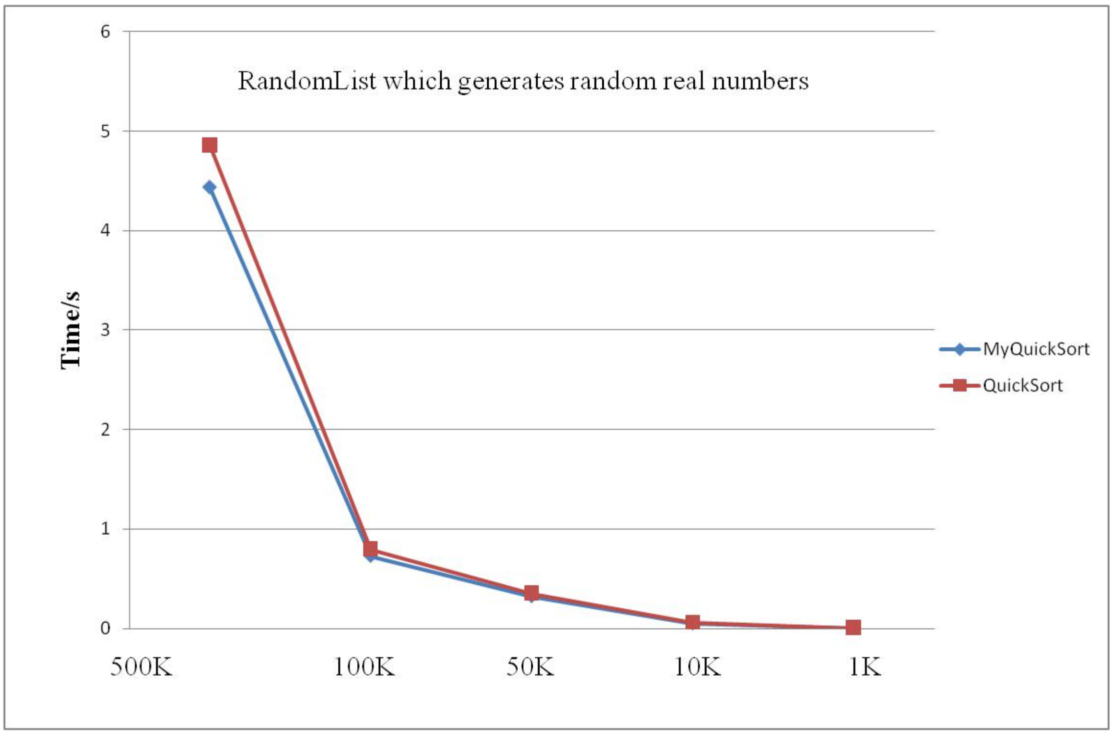

Figure 2. RandomList comparison of Quicksort and MyQuicksort

As the figure shown above, we can see that when the size of data becomes smaller, the algorithm will take less time as well. In the meanwhile, MyQuickSort can be more efficient than the original QuickSort.
Numbers which are in the RandomLists are all real numbers.

In conclusion, using two data sets, we can see from the figure, the new algorithm tends to be more efficient than the original QuickSort algorithm.