**Program Structures & Algorithms**

**Fall 2021**

**Final Report - Group 18**

* **Team members:**

Danlin Lu (1567176)

Wenqi Yang (1586011)

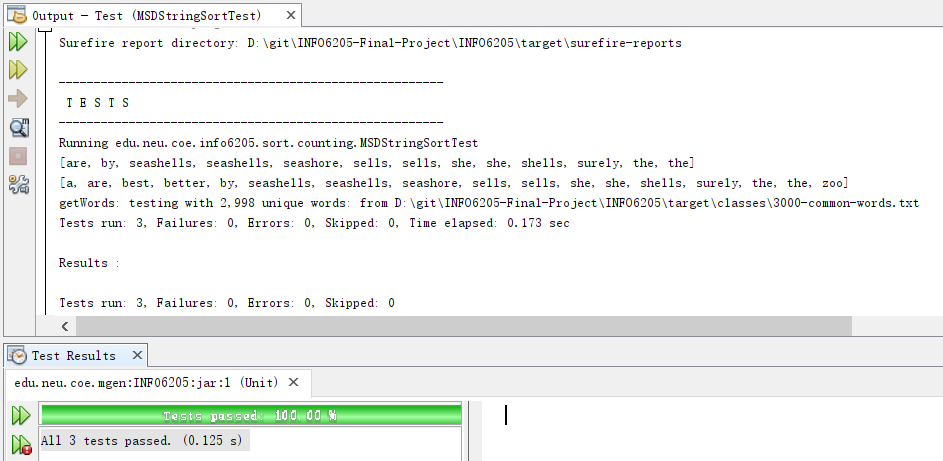
Kunhui Zhang (1563549)

* **Tasks**

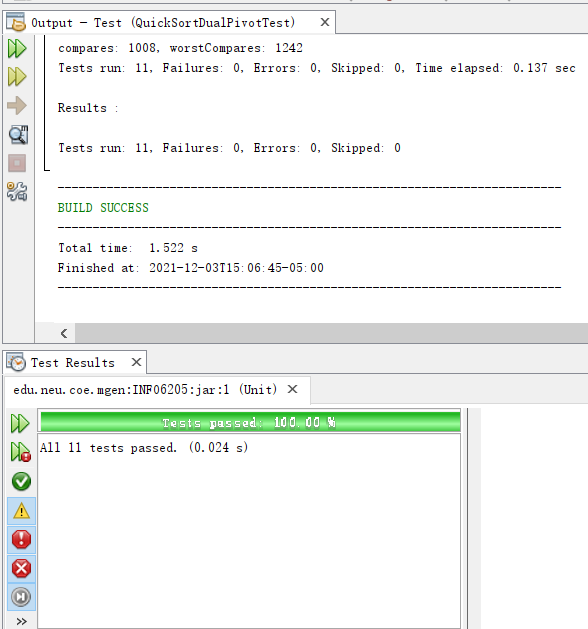
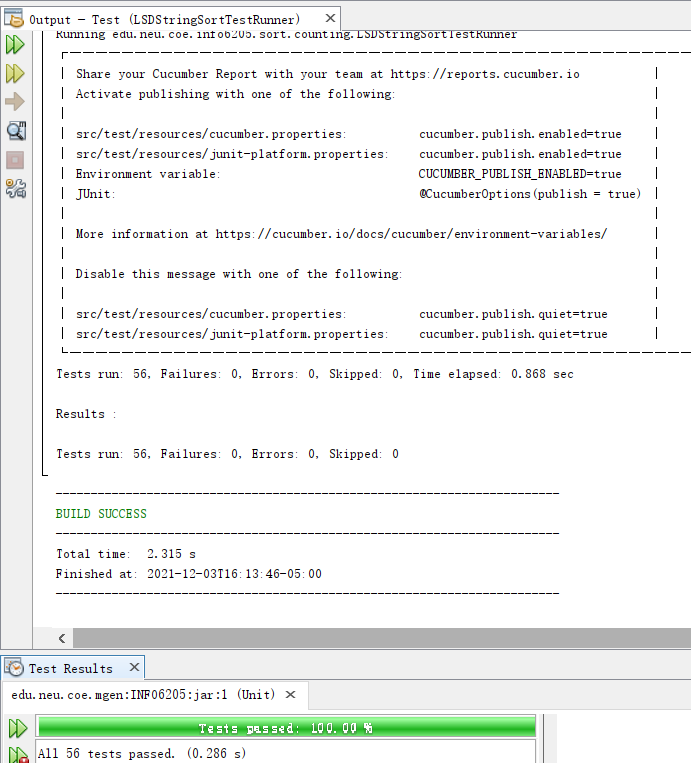
1. Implement MSD radix sort for Simplified Chinese which uses Unicode characters (UTF-8). (The conventional order for Chinese is according to the English order of the Pinyin.)
2. Compare MSD radix sort which we implemented with Timsort, Dual-pivot Quicksort, Huskysort, and LSD radix sort.
3. Show the trend of MSD radix sort’s running time under different data sizes. Show the comparisons among results of five sorts in a graph.

* **Process (What we did)**

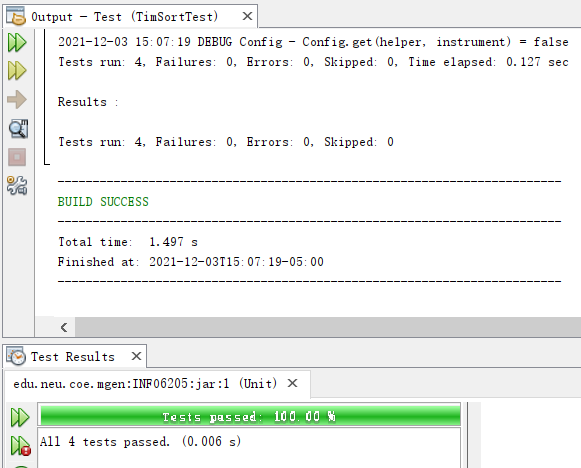
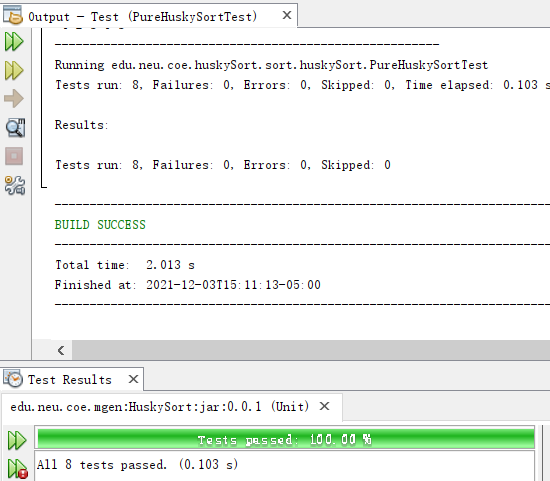
1. Firstly, we built 5 sorting methods and used unit tests to test feasibility of the methods and then modified the methods to make sure every method is correct.



Picture 1. Unit tests of MSD radix sort



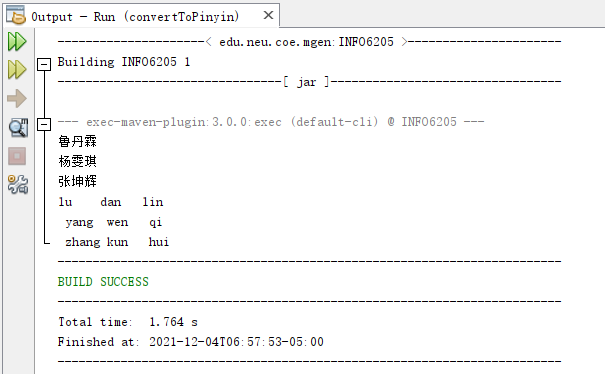
Picture 2. Unit tests of LSD radix sort Picture 3. Unit tests of QuickSort\_DualPivot



Picture 4. Unit tests of Pure HuskySort Picture 5. Unit tests of Tim Sort

The above unit tests mean that these sort methods are all avaiable and usable.

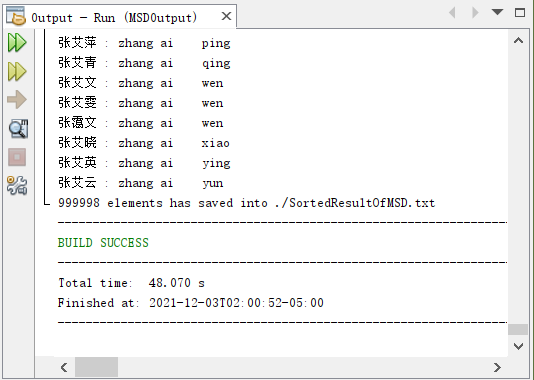
1. The second step is converting the Simplified Chinese into Pinyin format to be sorted by five sorting methods. We imported the package called “pinyin4j-2.5.0” to help us convert the Simplified Chinese into Pinyin format. Here we added some blank spaces to separate Pinyin of each Chinese character to avoid the situation that Pinyin of first name is compared with that of last name.



Picture 6. Convert Chinese to Pinyin

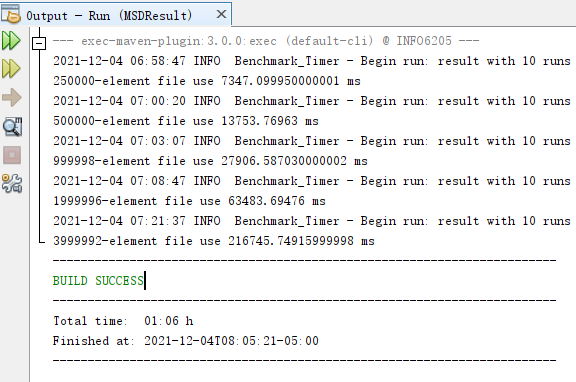
This means we converted the Simplified Chinese into Pinyin successfully.

1. Thirdly, we checked the sorted results before implementing next step. We printed all sorted results and outputted them into text files. Here are the outputs of 1M-size data by MSD radix sort method (you could see outputs of 1000-size data by all sort methods in repository):



Picture 7. Output of MSD radix sort

1. Finally, we implemented benchmark on all 5 sorting methods (MSD radix sort, Timsort, Dual-pivot Quicksort, Huskysort, and LSD radix sort) for 250k, 500k and 1M, 2M, 4M Chinese names (Here we only show the result of MSD radix sort, results of the other sort methods are in the repository).

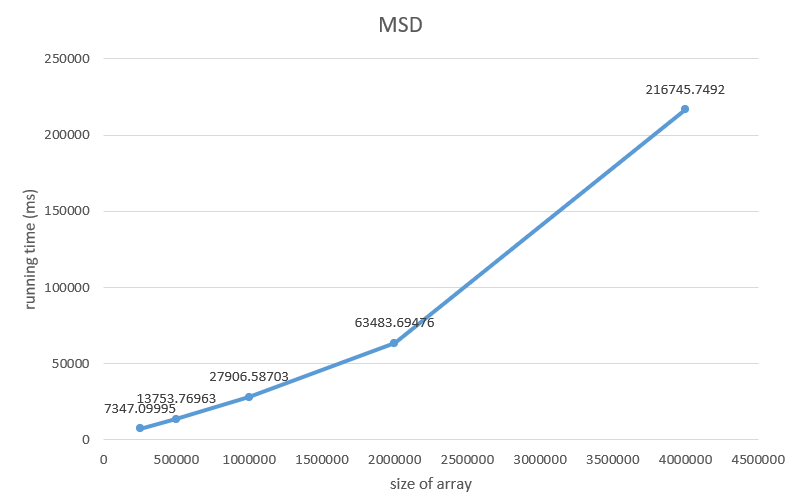


Picture 8. Result of MSD radix sort

According to the results, we can draw graphs to observe the growth and trend of sort methods.

* **Conclusion:**

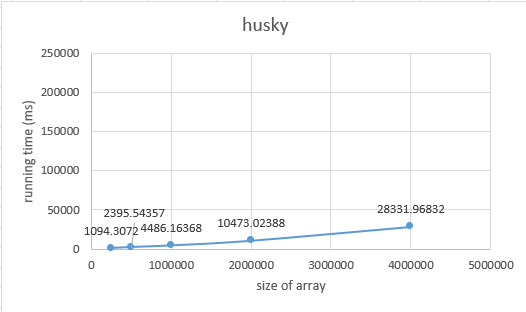
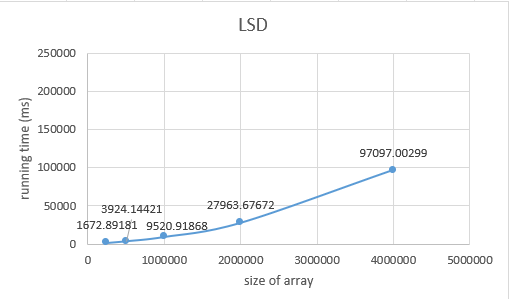
1. Firstly, observe the growth of MSD radix sort method:



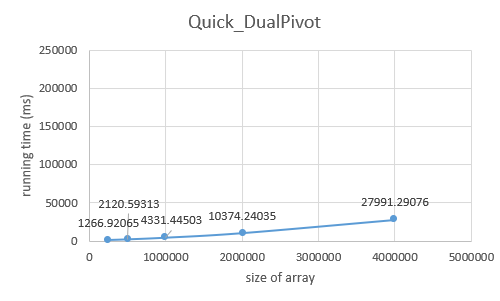
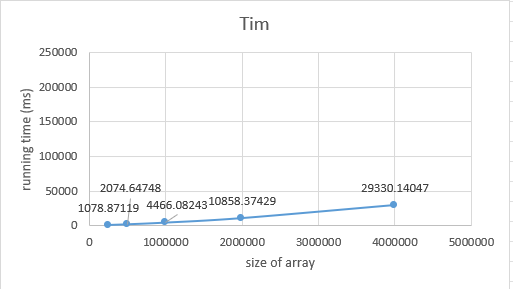
Picture 9. Growth of MSD radix sort

From the growth, the slope of running time is getting larger and larger. Therefore, the growth rate of MSD is gradually getting larger with the larger data size.

1. These are the time growths of the other 4 sorting methods:

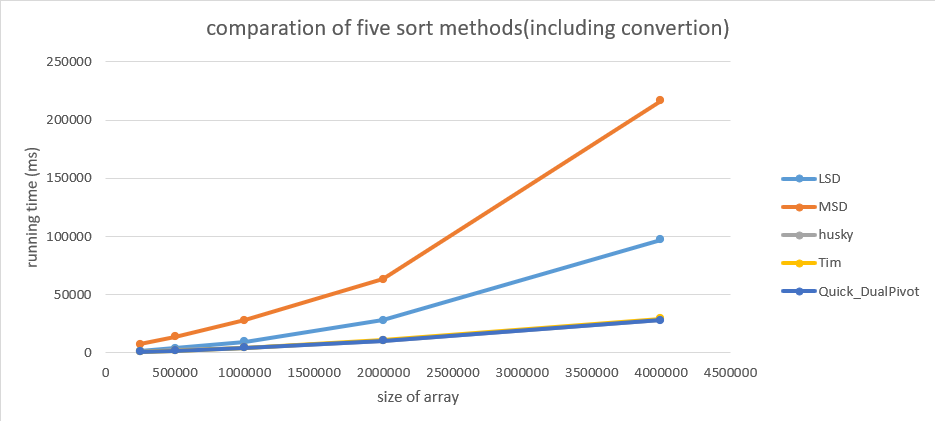


Picture 10. Growth of LSD radix sort Picture 11. Growth of HuskySort



Picture 12. Growth of Tim Sort Picture 13. Growth of QuickSort\_DualPivot

1. This is the graph of comparison among MSD radix sort, Timsort, Dual-pivot Quicksort, Huskysort, and LSD radix sort:



Picture 14. Comparation of five sort methods

According to the above graphs, we can find that MSD spends the most time to sort, and much more than the other methods. LSD took less time than MSD maybe because of using less space. Compared to MSD, HuskySort, TimSort and QuickSort\_DualPiovt nearly spend same running time, much smaller than MSD.