Program Structures & Algorithms

Fall 2021

Final Report - Group 18

- **Team members:**

Danlin Lu (1567176)

Wenqi Yang (1586011)

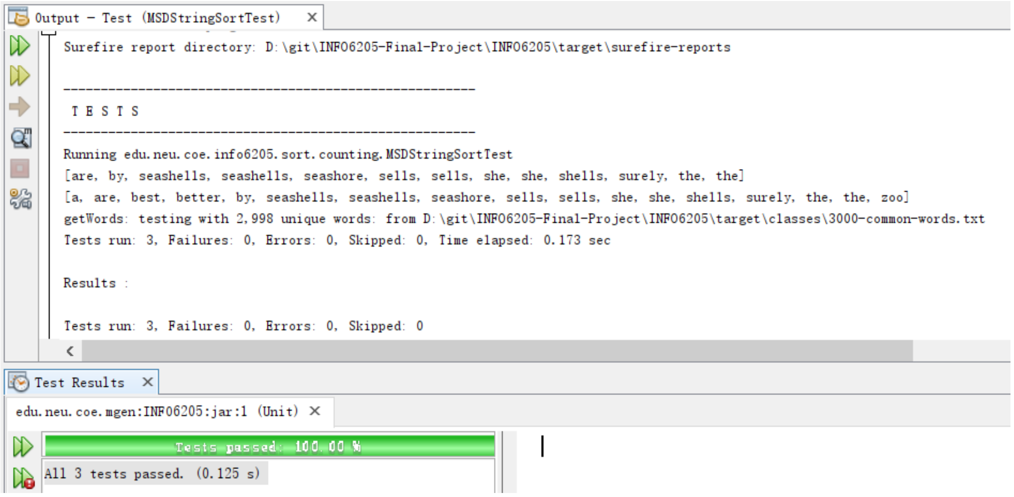
Kunhui Zhang (1563549)

- **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)**

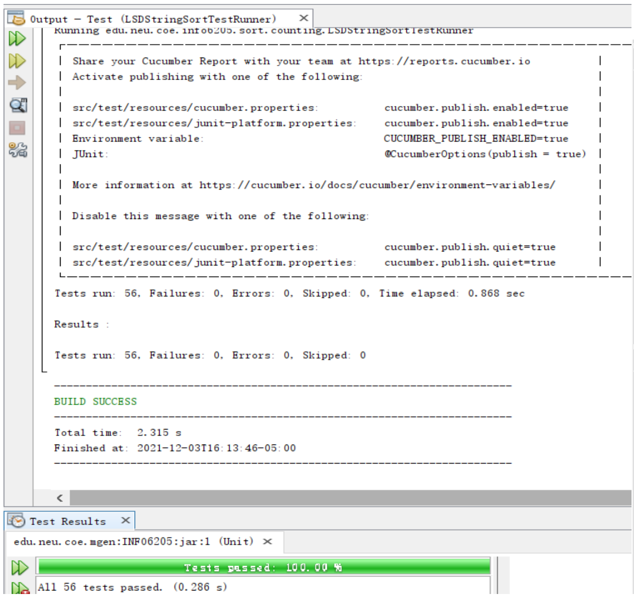
1. Firstly, we built 5 sorting methods and used unit tests to test feasibility of the methods and then modified the methods to make sure every method is correct.



```
Surefire report directory: D:\git\INFO6205-Final-Project\INFO6205\target\surefire-reports

-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running edu.neu.coe.info6205.sort.counting.MSDStringSortTest
[are, by, seashells, seashells, seashore, sells, sells, she, she, shells, surely, the, the]
[a, are, best, better, by, seashells, seashells, seashore, sells, sells, she, she, shells, surely, the, the, zoo]
getWords: testing with 2,998 unique words: from D:\git\INFO6205-Final-Project\INFO6205\target\classes\3000-common-words.txt
Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.173 sec

Results :

Tests run: 3, Failures: 0, Errors: 0, Skipped: 0
```

```
All 3 tests passed. (0.125 s)
```

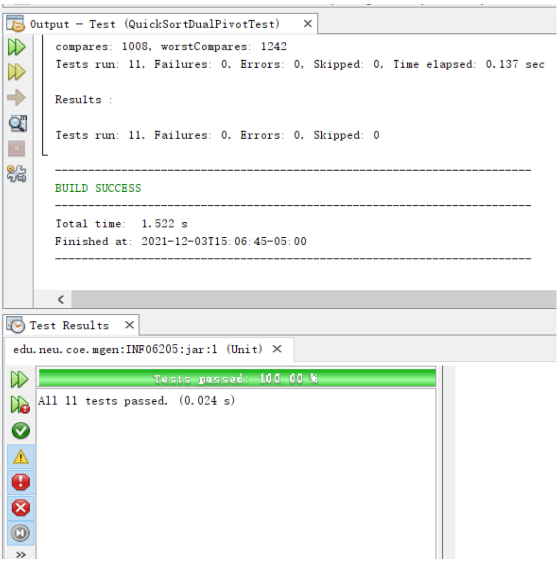
Picture 1. Unit tests of MSD radix sort



```
Running edu.neu.coe.info6205.sort.counting.LSDStringSortTestRunner
┌──────────────────────────────────────────────────────────────────────────────
│ Share your Cucumber Report with your team at https://reports.cucumber.io     │
│ Activate publishing with one of the following:                                │
│                                                                               │
│ src/test/resources/cucumber.properties:          cucumber.publish.enabled=true │
│ src/test/resources/junit-platform.properties:    cucumber.publish.enabled=true │
│ Environment variable:                            CUCUMBER_PUBLISH_ENABLED=true │
│ JUnit:                                           @CucumberOptions(publish = true) │
│                                                                               │
│ More information at https://cucumber.io/docs/cucumber/environment-variables/  │
│                                                                               │
│ Disable this message with one of the following:                               │
│                                                                               │
│ src/test/resources/cucumber.properties:          cucumber.publish.quiet=true   │
│ src/test/resources/junit-platform.properties:    cucumber.publish.quiet=true   │
└──────────────────────────────────────────────────────────────────────────────
Tests run: 56, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.868 sec

Results :

Tests run: 56, Failures: 0, Errors: 0, Skipped: 0

------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  2.315 s
Finished at: 2021-12-03T16:13:46-05:00
------------------------------------------------------------------------
```

```
All 56 tests passed. (0.286 s)
```

Picture 2. Unit tests of LSD radix sort



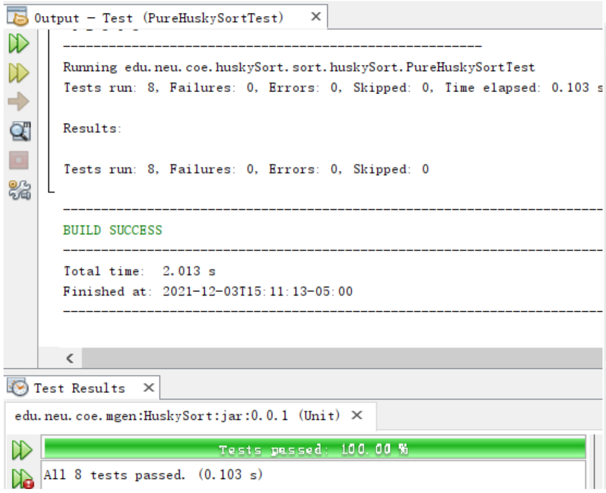
Output - Test (QuickSortDualPivotTest)

```
compares: 1008, worstCompares: 1242
Tests run: 11, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.137 sec

Results :

Tests run: 11, Failures: 0, Errors: 0, Skipped: 0

------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  1.522 s
Finished at: 2021-12-03T15:06:45-05:00
------------------------------------------------------------------------
```

Test Results
edu.neu.coe.mgen:INFO6205:jar:1 (Unit)
Tests passed: 100.00 %

```
All 11 tests passed. (0.024 s)
```

Picture 3. Unit tests of QuickSort_DualPivot



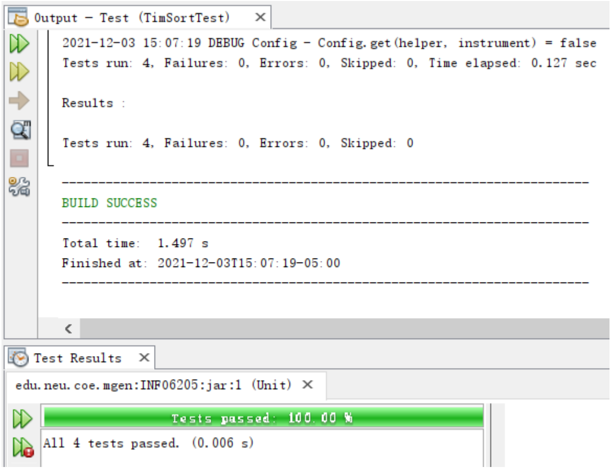
Output - Test (PureHuskySortTest)

```
-------------------------------------------------------
Running edu.neu.coe.huskySort.sort.huskySort.PureHuskySortTest
Tests run: 8, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.103 s

Results:

Tests run: 8, Failures: 0, Errors: 0, Skipped: 0

------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  2.013 s
Finished at: 2021-12-03T15:11:13-05:00
------------------------------------------------------------------------
```

Test Results
edu.neu.coe.mgen:HuskySort:jar:0.0.1 (Unit)
Tests passed: 100.00 %

```
All 8 tests passed. (0.103 s)
```

Picture 4. Unit tests of Pure HuskySort



Output - Test (TimSortTest)

```
2021-12-03 15:07:19 DEBUG Config - Config.get(helper, instrument) = false
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.127 sec

Results :

Tests run: 4, Failures: 0, Errors: 0, Skipped: 0

------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  1.497 s
Finished at: 2021-12-03T15:07:19-05:00
------------------------------------------------------------------------
```

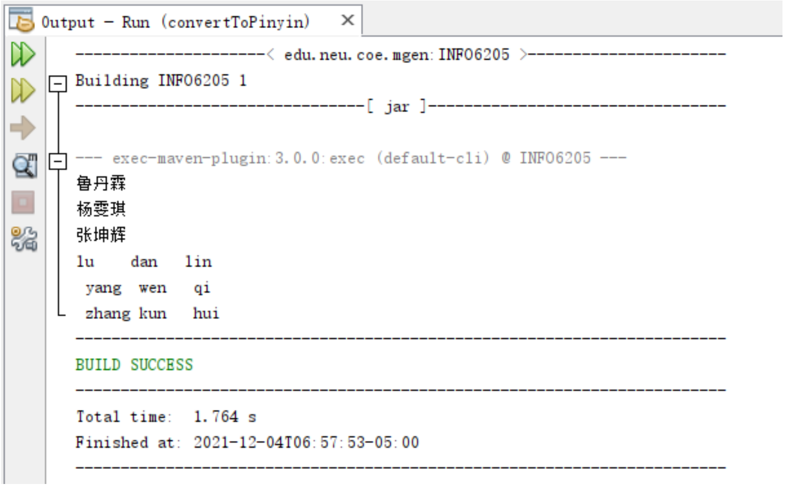
Test Results
edu.neu.coe.mgen:INFO6205:jar:1 (Unit)
Tests passed: 100.00 %

```
All 4 tests passed. (0.006 s)
```

Picture 5. Unit tests of Tim Sort

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

2. 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.

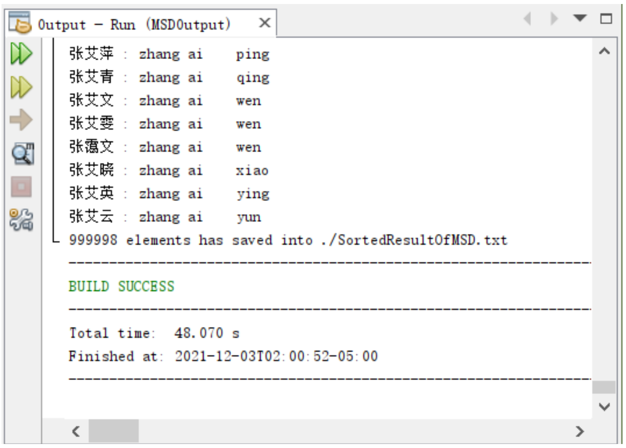


```
--------------------< edu.neu.coe.mgen:INFO6205 >---------------------
Building INFO6205 1
--------------------------------[ jar ]--------------------------------

--- exec-maven-plugin:3.0.0:exec (default-cli) @ INFO6205 ---
鲁丹霖
杨雯琪
张坤辉
lu    dan   lin
 yang  wen   qi
 zhang kun   hui
-----------------------------------------------------------------------
BUILD SUCCESS
-----------------------------------------------------------------------
Total time:  1.764 s
Finished at: 2021-12-04T06:57:53-05:00
-----------------------------------------------------------------------
```

Picture 6. Convert Chinese to Pinyin

This means we converted the Simplified Chinese into Pinyin successfully.

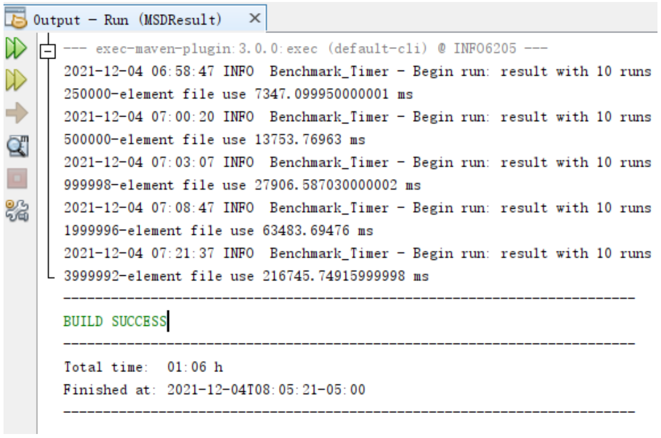
3. Thirdly, we checked the sorted results before implementing next step. We printed all sorted results and outputted them into text files. Here are the outputs of 1M-size data by MSD radix sort method (you could see outputs of 1000-size data by all sort methods in repository):



```
张艾萍 : zhang ai    ping
张艾青 : zhang ai    qing
张艾文 : zhang ai    wen
张艾雯 : zhang ai    wen
张霭文 : zhang ai    wen
张艾晓 : zhang ai    xiao
张艾英 : zhang ai    ying
张艾云 : zhang ai    yun
999998 elements has saved into ./SortedResultOfMSD.txt
------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------
Total time:  48.070 s
Finished at: 2021-12-03T02:00:52-05:00
------------------------------------------------------------------
```

Picture 7. Output of MSD radix sort

4. 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).



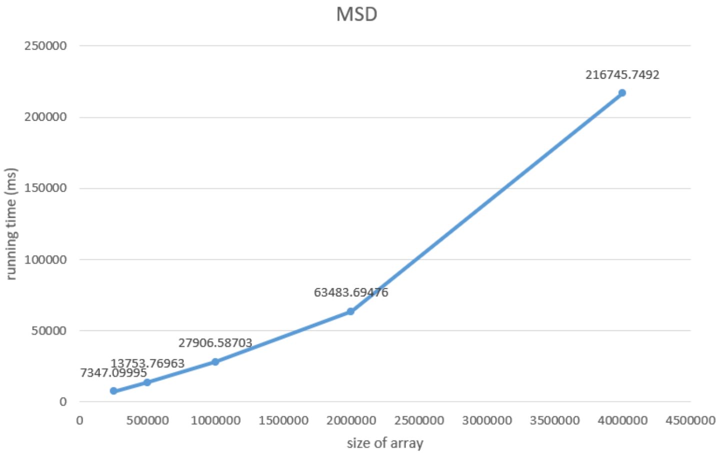
```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ INFO6205 ---
2021-12-04 06:58:47 INFO  Benchmark_Timer - Begin run: result with 10 runs
250000-element file use 7347.099950000001 ms
2021-12-04 07:00:20 INFO  Benchmark_Timer - Begin run: result with 10 runs
500000-element file use 13753.76963 ms
2021-12-04 07:03:07 INFO  Benchmark_Timer - Begin run: result with 10 runs
999998-element file use 27906.587030000002 ms
2021-12-04 07:08:47 INFO  Benchmark_Timer - Begin run: result with 10 runs
1999996-element file use 63483.69476 ms
2021-12-04 07:21:37 INFO  Benchmark_Timer - Begin run: result with 10 runs
3999992-element file use 216745.74915999998 ms
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  01:06 h
Finished at: 2021-12-04T08:05:21-05:00
------------------------------------------------------------------------
```

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:

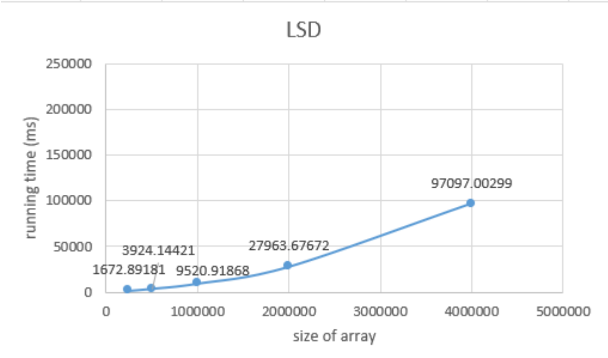
1. Firstly, observe the growth of MSD radix sort method:

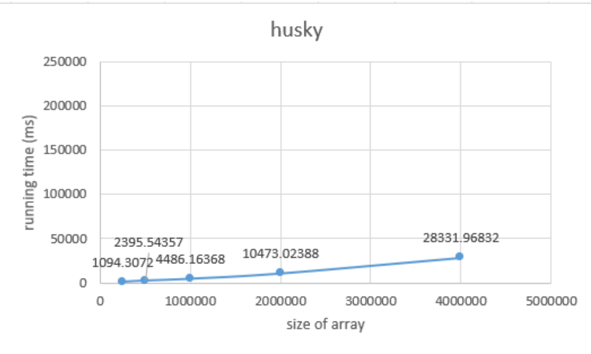


Picture 9. Growth of MSD radix sort

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.

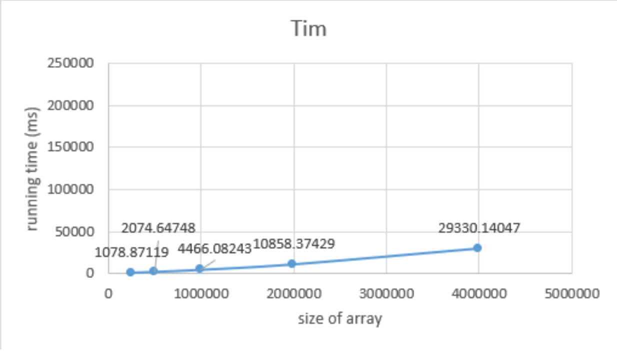
2. These are the time growths of the other 4 sorting methods:

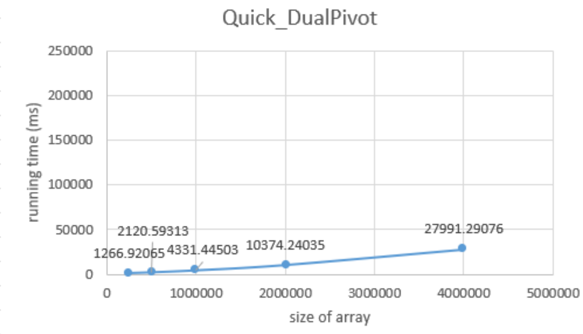




Picture 10. Growth of LSD radix sort

Picture 11. Growth of HuskySort

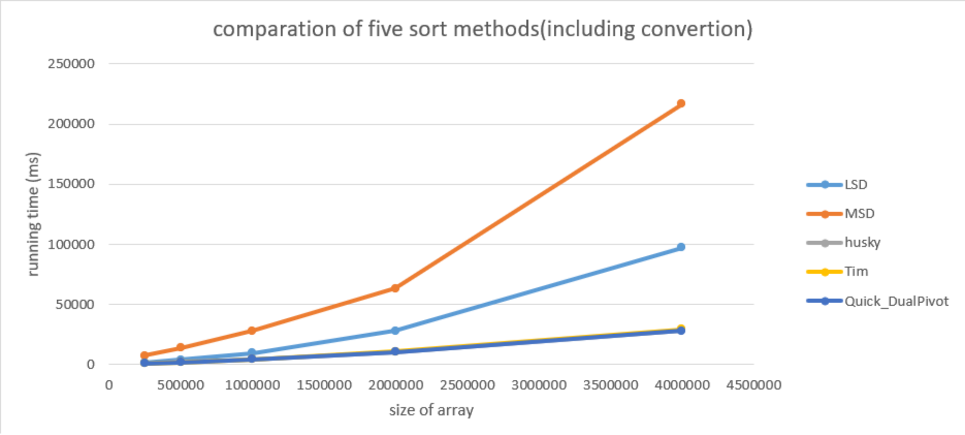




Picture 12. Growth of Tim Sort

Picture 13. Growth of QuickSort_DualPivot

3. 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.