

Shuo Wang(001533020)

Zeen Wang(001082883)

Xiyue Suo(001348347)

Program Structures & Algorithms

Fall 2021

Final Project Report

◎ Task (List down the tasks performed in the Assignment)

We sorted array of Chinese names into pinyin order with five kinds of sort algorithms: MSD radix sort, LSD radix sort, Dual-Pivot Quick Sort, Tim sort and Husky sort. We wrote specific unit test with Chinese names and measured the performing time with benchmarks.

We firstly converted all Chinese names to lower case pinyin and each Chinese character followed by its tone (number 1, 2, 3, or 4) and separated by a space. We added tone because that two different Chinese character may have same pinyin. For example, “王硕” was converted to “wang2 shuo4.” Then we put all Chinese names and their pinyin to a HashMap. Then we got the order of pinyin with sort algorithms, now the corresponding Chinese names listed as values in the map.

Our benchmarks counted the running time of the sort process containing the mapping and sorting process. It gave an average time of 10 runs.

Ran tests with 5 different array sizes: 0.25million, 0.5 million, 1 million, 2 million and 4 million Chinese names.

◎ Implement

1. We finished the code of the MSD radix sort and adapted other sort to Chinese. We used the dependency which can convert the Chinese to pinyin called pinyin4j. To improve the

accuracy of the Chinese words with same pinyin, we add the tone at the end of every pinyin.

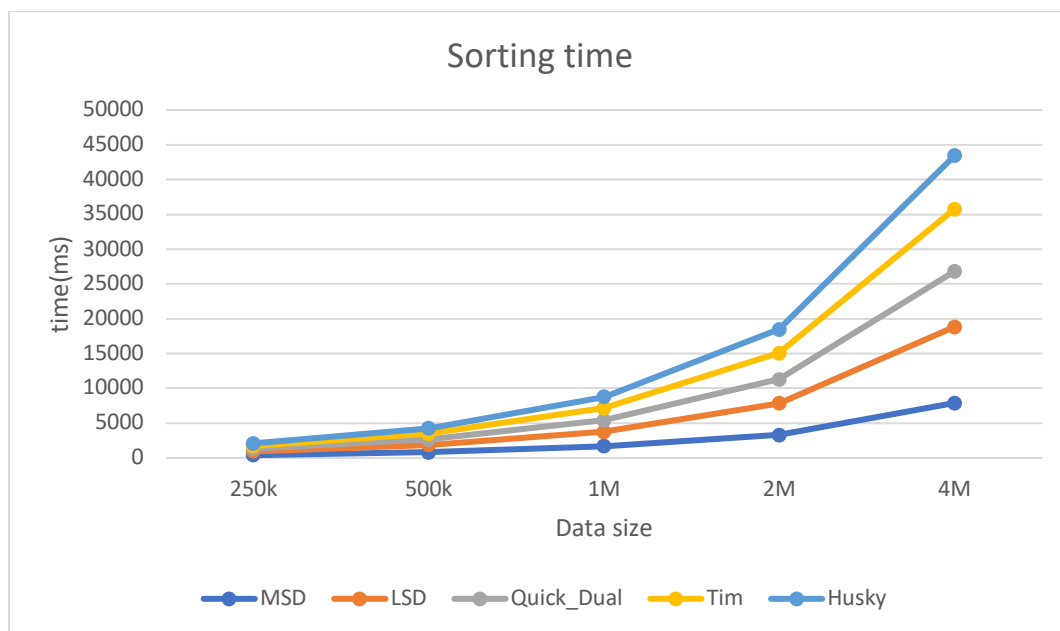
2. We used the Identity HashMap to set the different words in Chinese which have the same pinyin and tone. We did not use the array at MSD radix sort and LSD radix sort, to make the error of different sort as low as possible. Thus, we can compare the sorting efficiency without other effects.
3. And we added unique Chinese test in the unit test for every sort, to make sure our sorting method is correct.

◎ Results and Observations

- **Table 1 Sorting Time for Different Array Size:**

Size/Time	MSD	LSD	Quick_Dual	Tim	Husky
250k	392.865875	543.119088	382.740971	398.757271	375.495275
500k	815.630338	1041.99552	798.450225	830.409767	778.58295
1M	1681.25228	2081.67405	1630.39014	1759.8964	1604.40378
2M	3288.14192	4564.3655	3466.55106	3740.55863	3401.49164
4M	7891.54064	10954.4722	7973.06744	8977.23849	7653.35618

- **Chart 1 Sorting Time of 5 Sorting Algorithms**



- **Table 2 Sample Output Order**

1	First 1000 Output					
2	MSD	LSD	DP-QS	PureHusky	Timsort	07
3	阿安	阿安	阿安	阿安	阿安	
4	阿斌	阿斌	阿彬	阿彬	阿斌	
5	阿滨	阿滨	阿滨	阿滨	阿滨	
6	阿彬	阿彬	阿斌	阿斌	阿彬	
7	阿冰	阿冰	阿兵	阿兵	阿冰	
8	阿兵	阿兵	阿冰	阿冰	阿兵	
9	阿冰冰	阿冰冰	阿冰冰	阿冰冰	阿冰冰	
10	阿婵	阿婵	阿婵	阿婵	阿婵	
11	阿超	阿超	阿超	阿超	阿超	
12	阿朝	阿朝	阿朝	阿朝	阿朝	
13	阿琛	阿琛	阿琛	阿琛	阿琛	
14	阿臣	阿臣	阿辰	阿晨	阿臣	
15	阿晨	阿晨	阿臣	阿辰	阿晨	
16	阿辰	阿辰	阿晨	阿臣	阿辰	
17	阿称	阿称	阿称	阿称	阿称	
18	阿诚	阿诚	阿诚	阿诚	阿诚	
19	阿澄	阿澄	阿澄	阿澄	阿澄	
20	阿弛	阿弛	阿弛	阿弛	阿弛	
21	阿弛	阿弛	阿弛	阿弛	阿弛	

- Sample Chinese Names in Pinyin Version

```
Run: MSDStringSort
ai4 ning2 ning2
ai4 nong2
ai4 nuo4
ai4 ou1
ai4 ou1
ai4 pan1
ai4 pan4
ai4 pan4 pan4
ai4 pei2
ai4 pei2 pei2
ai4 pei4
ai4 pei4 pei4
ai4 peng2
ai4 peng2
ai4 peng2
ai4 pi1 shan4
ai4 ping2
ai4 ping2
ai4 ping2 ping2
ai4 ping2 ping2
ai4 pu2
ai4 pu2
```

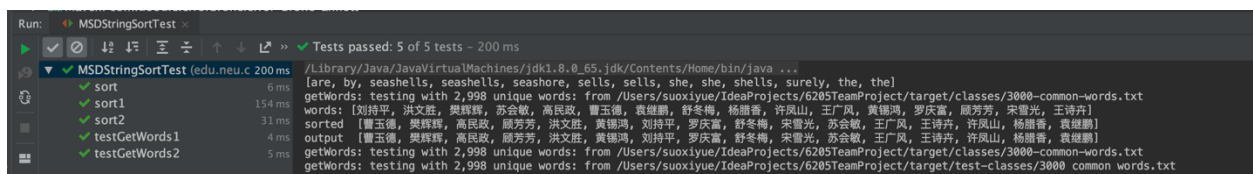
◎ Conclusions

- From table 2 output, we compared the positions of sorted Chinese names with same pinyin and tone, such as "阿斌 577658", "阿滨 790796" and "阿彬 945627" with their original order on input, we can see MSD and LSD radix sorts, and Huskysort maintine dual-pivot quick sort and PureHuskuy sort are not.
- Husky sort is faster than dual-pivot quick sort and Timsort when sorting objects because it reduces the array access, here we are sorting Strings, this is consistent with theoretical conclusion from Professor Robin's paper Huskysort.
- Based on our benchmark results, Huskysort gives best performance, comparing to MSD/LSD radix sorts, Timsort, and dual-pivot quick sort. LSD took most time in sorting Chinese names among those 5 sorts.
- The sorting times are linearithmic. ($\sim N \log N$)

◎ Output

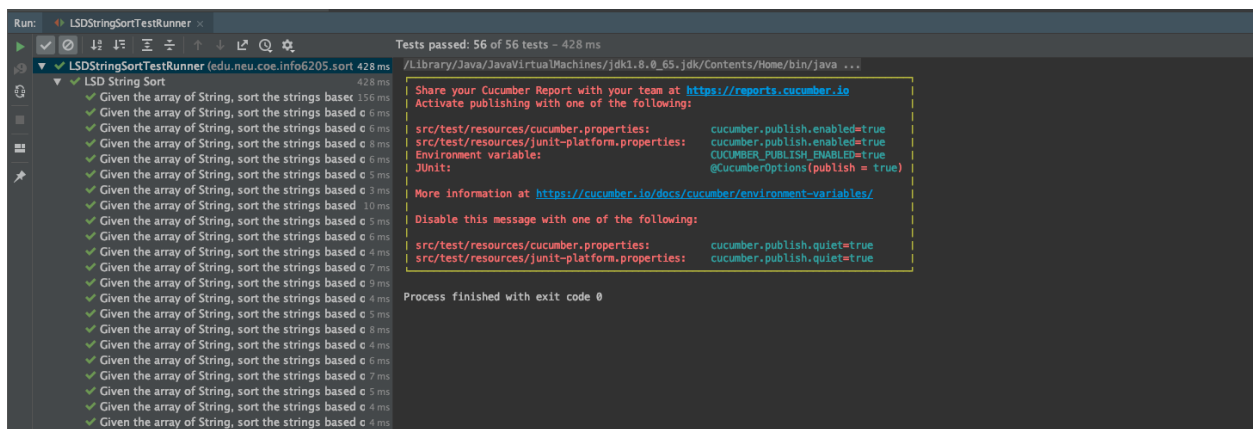
• Unit Tests

MSD sort:



The screenshot shows the test results for MSDStringSortTest in an IDE. The test suite passed 5 out of 5 tests in 200 ms. The tests listed are sort (6 ms), sort1 (154 ms), sort2 (31 ms), sorted (4 ms), and testGetWords2 (5 ms). The output of the tests shows the sorting of a list of Chinese names and the retrieval of words from a file.

LSD sort:



The screenshot shows the test results for LSDStringSortTestRunner in an IDE. The test suite passed 56 out of 56 tests in 428 ms. The tests listed are LSD String Sort (428 ms), Given the array of String, sort the strings based c (156 ms), Given the array of String, sort the strings based c (6 ms), Given the array of String, sort the strings based c (8 ms), Given the array of String, sort the strings based c (6 ms), Given the array of String, sort the strings based c (5 ms), Given the array of String, sort the strings based c (3 ms), Given the array of String, sort the strings based c (10 ms), Given the array of String, sort the strings based c (5 ms), Given the array of String, sort the strings based c (6 ms), Given the array of String, sort the strings based c (4 ms), Given the array of String, sort the strings based c (7 ms), Given the array of String, sort the strings based c (3 ms), Given the array of String, sort the strings based c (4 ms), Given the array of String, sort the strings based c (3 ms), Given the array of String, sort the strings based c (4 ms), Given the array of String, sort the strings based c (7 ms), Given the array of String, sort the strings based c (5 ms), Given the array of String, sort the strings based c (4 ms), and Given the array of String, sort the strings based c (4 ms). The output of the tests shows the sorting of a list of Chinese names and the retrieval of words from a file. A message box is also displayed, encouraging the user to share their Cucumber report with their team at <https://reports.cucumber.io> and providing instructions on how to activate publishing.

```
Run: LSDStringSortTestRunner x
Tests passed: 2 of 2 tests - 37 ms
/Library/Java/JavaVirtualMachines/jdk1.8.0_65.jdk/Contents/Home/bin/java ...
LSDStringSortTestRunner (edu.neu.coe.info6205.sort) 37 ms
  sort1 35 ms
  sort2 2 ms
words: [刘持平, 洪文胜, 樊辉辉, 苏会敏, 高民政, 曹玉德, 袁继鹏, 舒冬梅, 杨腊香, 许凤山, 王广凤, 黄锦鸿, 罗庆富, 廖芳芳, 宋雪光, 王诗卉]
sorted [曹玉德, 樊辉辉, 高民政, 廖芳芳, 洪文胜, 黄锦鸿, 刘持平, 罗庆富, 舒冬梅, 宋雪光, 苏会敏, 王广凤, 王诗卉, 许凤山, 杨腊香, 袁继鹏]
output [曹玉德, 樊辉辉, 高民政, 廖芳芳, 洪文胜, 黄锦鸿, 刘持平, 罗庆富, 舒冬梅, 宋雪光, 苏会敏, 王广凤, 王诗卉, 许凤山, 杨腊香, 袁继鹏]
words: [阿斌, 阿斌, 阿斌, 阿兵, 阿明, 阿冰冰, 阿婵, 阿安, 阿超, 阿瑞, 阿滨, 阿臣, 阿晨, 阿辰, 阿祿, 阿斌, 阿斌]
sorted [阿安, 阿斌, 阿斌, 阿彬, 阿冰, 阿兵, 阿冰冰, 阿婵, 阿超, 阿朝, 阿瑞, 阿臣, 阿晨, 阿辰, 阿祿, 阿斌, 阿斌]
output [阿安, 阿斌, 阿斌, 阿彬, 阿冰, 阿兵, 阿冰冰, 阿婵, 阿超, 阿朝, 阿瑞, 阿臣, 阿晨, 阿辰, 阿祿, 阿斌, 阿斌]
Process finished with exit code 0
```

Tim sort:

```
Run: TimSortTest x
Tests passed: 2 of 2 tests - 108 ms
/Library/Java/JavaVirtualMachines/jdk1.8.0_65.jdk/Contents/Home/bin/java ...
TimSortTest (edu.neu.coe.info6205.sort.linearithmic) 108 ms
  sort 106 ms
  sort2 2 ms
words: [刘持平, 洪文胜, 樊辉辉, 苏会敏, 高民政, 曹玉德, 袁继鹏, 舒冬梅, 杨腊香, 许凤山, 王广凤, 黄锦鸿, 罗庆富, 廖芳芳, 宋雪光, 王诗卉]
sorted [曹玉德, 樊辉辉, 高民政, 廖芳芳, 洪文胜, 黄锦鸿, 刘持平, 罗庆富, 舒冬梅, 宋雪光, 苏会敏, 王广凤, 王诗卉, 许凤山, 杨腊香, 袁继鹏]
output [曹玉德, 樊辉辉, 高民政, 廖芳芳, 洪文胜, 黄锦鸿, 刘持平, 罗庆富, 舒冬梅, 宋雪光, 苏会敏, 王广凤, 王诗卉, 许凤山, 杨腊香, 袁继鹏]
words: [阿斌, 阿斌, 阿斌, 阿兵, 阿明, 阿冰冰, 阿婵, 阿安, 阿超, 阿瑞, 阿滨, 阿臣, 阿晨, 阿辰, 阿祿, 阿斌, 阿斌]
sorted [阿安, 阿斌, 阿斌, 阿彬, 阿冰, 阿兵, 阿冰冰, 阿婵, 阿超, 阿朝, 阿瑞, 阿臣, 阿晨, 阿辰, 阿祿, 阿斌, 阿斌]
output [阿安, 阿斌, 阿斌, 阿彬, 阿冰, 阿兵, 阿冰冰, 阿婵, 阿超, 阿朝, 阿瑞, 阿臣, 阿晨, 阿辰, 阿祿, 阿斌, 阿斌]
Process finished with exit code 0
```

Dual-pivot sort:

```
Run: QuickSortDualPivotTest x
Tests passed: 13 of 13 tests - 109 ms
/Library/Java/JavaVirtualMachines/jdk1.8.0_65.jdk/Contents/Home/bin/java ...
QuickSortDualPivotTest (edu.neu.coe.info6205.sort.l) 109 ms
  testSort 41 ms
  testPartition1 2 ms
  testPartition2 0 ms
  sort 28 ms
  sort2 3 ms
  testSortWithInstrumenting0 2 ms
  testSortWithInstrumenting1 5 ms
  testSortWithInstrumenting2 9 ms
  testSortWithInstrumenting3 8 ms
  testSortWithInstrumenting4 3 ms
  testSortWithInstrumenting5 3 ms
  testPartitionWithSort 2 ms
  testSortDetailed 3 ms
words: [刘持平, 洪文胜, 樊辉辉, 苏会敏, 高民政, 曹玉德, 袁继鹏, 舒冬梅, 杨腊香, 许凤山, 王广凤, 黄锦鸿, 罗庆富, 廖芳芳, 宋雪光, 王诗卉]
sorted [曹玉德, 樊辉辉, 高民政, 廖芳芳, 洪文胜, 黄锦鸿, 刘持平, 罗庆富, 舒冬梅, 宋雪光, 苏会敏, 王广凤, 王诗卉, 许凤山, 杨腊香, 袁继鹏]
output [曹玉德, 樊辉辉, 高民政, 廖芳芳, 洪文胜, 黄锦鸿, 刘持平, 罗庆富, 舒冬梅, 宋雪光, 苏会敏, 王广凤, 王诗卉, 许凤山, 杨腊香, 袁继鹏]
words: [阿斌, 阿斌, 阿斌, 阿兵, 阿明, 阿冰冰, 阿婵, 阿安, 阿超, 阿瑞, 阿滨, 阿臣, 阿晨, 阿辰, 阿祿, 阿斌, 阿斌]
sorted [阿安, 阿斌, 阿斌, 阿彬, 阿冰, 阿兵, 阿冰冰, 阿婵, 阿超, 阿朝, 阿瑞, 阿臣, 阿晨, 阿辰, 阿祿, 阿斌, 阿斌]
output [阿安, 阿斌, 阿斌, 阿彬, 阿冰, 阿兵, 阿冰冰, 阿婵, 阿超, 阿朝, 阿瑞, 阿臣, 阿晨, 阿辰, 阿祿, 阿斌, 阿斌]
Instrumenting helper for quick sort dual pivot with 128 elements
StatPack {hits: 1,636; copies: 0; Inversions: 4,224; swaps: 402; fixes: 4,750; compares: 1,000}
compares: 1000, worstCompares: 1242
Process finished with exit code 0
```

“From the unit test 2, we notice the dual pivot have different order for Chinese names with exactly same pinyin and tone, comparing to previous sorts. Will compare which sort follows the original order, this will give which sorting algorithm is in place.”

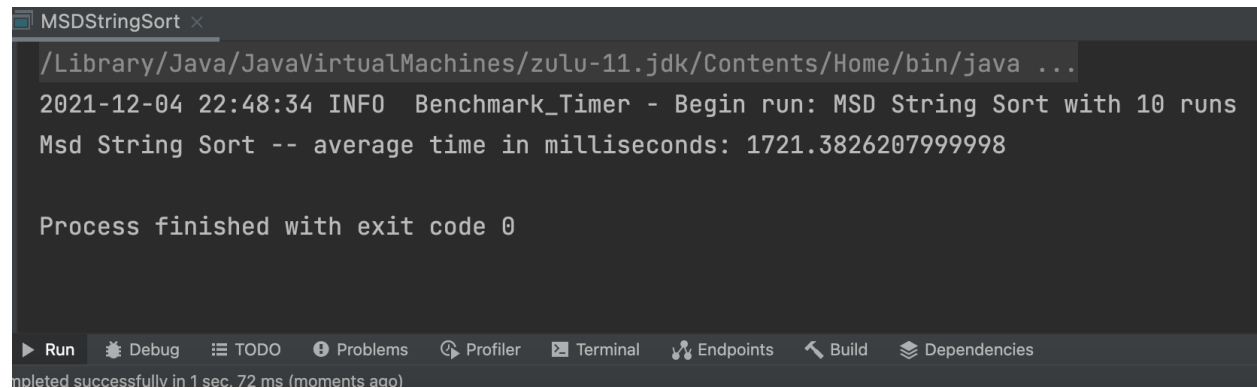
Husky sort:

```
Run: PureHuskySortTest x
Tests passed: 9 of 9 tests - 166 ms
/Library/Java/JavaVirtualMachines/jdk1.8.0_65.jdk/Contents/Home/bin/java ...
PureHuskySortTest (edu.neu.coe.huskySort.sort.hus) 166 ms
  testSortString1 147 ms
  testSortString2 4 ms
  testSortString3 6 ms
  testSortString4 0 ms
  testSortString5 0 ms
  testSortString6 6 ms
  testFloorLog 1 ms
  testWithInsertionSort 1 ms
  testInsertionSort 1 ms
pure husky sort unit test 1 Chinese names
xs input words: [刘持平, 洪文胜, 樊辉辉, 苏会敏, 高民政, 曹玉德, 袁继鹏, 舒冬梅, 杨腊香, 许凤山, 王广凤, 黄锦鸿, 罗庆富, 廖芳芳, 宋雪光, 王诗卉]
sorted [曹玉德, 樊辉辉, 高民政, 廖芳芳, 洪文胜, 黄锦鸿, 刘持平, 罗庆富, 舒冬梅, 宋雪光, 苏会敏, 王广凤, 王诗卉, 许凤山, 杨腊香, 袁继鹏]
output [曹玉德, 樊辉辉, 高民政, 廖芳芳, 洪文胜, 黄锦鸿, 刘持平, 罗庆富, 舒冬梅, 宋雪光, 苏会敏, 王广凤, 王诗卉, 许凤山, 杨腊香, 袁继鹏]
Process finished with exit code 0
```

- **Benchmarks:**

Benchmarks with 1M Chinese names:

MSD sort:



```
MSDStringSort x
/Library/Java/JavaVirtualMachines/zulu-11.jdk/Contents/Home/bin/java ...
2021-12-04 22:48:34 INFO  Benchmark_Timer - Begin run: MSD String Sort with 10 runs
Msd String Sort -- average time in milliseconds: 1721.3826207999998

Process finished with exit code 0

Run Debug TODO Problems Profiler Terminal Endpoints Build Dependencies
Completed successfully in 1 sec, 72 ms (moments ago)
```

LSD sort:

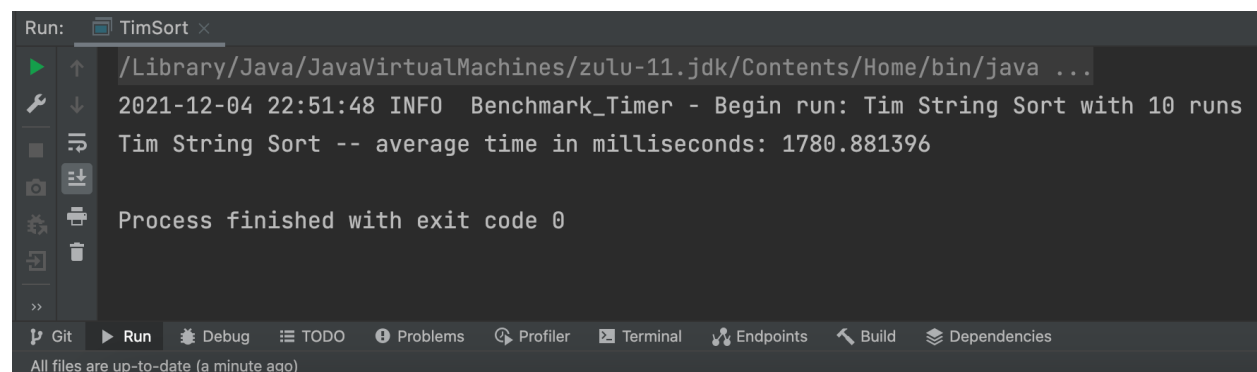


```
LSDStringSort x
/Library/Java/JavaVirtualMachines/zulu-11.jdk/Contents/Home/bin/java ...
2021-12-04 22:50:06 INFO  Benchmark_Timer - Begin run: LSD String Sort with 10 runs
Lsd String Sort -- average time in milliseconds: 2182.9351875

Process finished with exit code 0

Run Debug TODO Problems Profiler Terminal Endpoints Build Dependencies
All files are up-to-date (a minute ago)
```

Tim sort:



```
Run: TimSort x
/Library/Java/JavaVirtualMachines/zulu-11.jdk/Contents/Home/bin/java ...
2021-12-04 22:51:48 INFO  Benchmark_Timer - Begin run: Tim String Sort with 10 runs
Tim String Sort -- average time in milliseconds: 1780.881396

Process finished with exit code 0

Run Debug TODO Problems Profiler Terminal Endpoints Build Dependencies
All files are up-to-date (a minute ago)
```

Dual-pivot sort:

```
Run: QuickSort_DualPivot x
/Library/Java/JavaVirtualMachines/zulu-11.jdk/Contents/Home/bin/java ...
2021-12-04 22:51:05 DEBUG Config - Config.get(helper, instrument) = false
2021-12-04 22:51:05 INFO Benchmark_Timer - Begin run: QCD String Sort with 10 runs
QCD String Sort -- average time in milliseconds: 1628.8370123999998

Process finished with exit code 0
```

Husky sort:

```
Run: PureHuskySort x
/Library/Java/JavaVirtualMachines/zulu-11.jdk/Contents/Home/bin/java ...
2021-12-05 14:07:19 INFO Benchmark_Timer - Begin run: PHusky String Sort with 10 runs
PHusky String Sort -- average time in milliseconds: 1539.3529376000001

Process finished with exit code 0
```

Git Run TODO Problems Profiler Terminal Endpoints Build Dependencies

Build completed successfully in 1 sec, 25 ms (moments ago)