

Chandrakanth Chittappa (001300076)

Ankita Senapati (001003695)

Venkata Raghu Teja Kumar Gollapudi(001529656)

Program Structures & Algorithms

Fall 2021

Team Project

◉ **Task (List down the tasks performed)**

- Your task is to implement MSD radix sort for a natural language which uses Unicode characters. You may choose your own language or (Simplified) Chinese. Additionally, you will complete a literature survey of relevant papers and you will compare your method with Tim sort, Dual-pivot Quicksort, Husky sort and LSD radix sort.
- We used Hindi language to implement the various sorting techniques. The hindi language uses Devanagiri script which typically follows the same lexical order as that of its corresponding Java Unicode values. Therefore, there is no need to use the Java Collator class for the comparison of Hindi characters.
- We used the various sorting methods available in the Info6205 repository as well the textbook Algorithms 4th Edition by Robert Sedgewick, Kevin Wayne.
- We implemented the BenchmarkTimer to benchmark all the sort methods and analyze their run time performances for various text sizes- 250k, 500k, 1M, 2M and 4M.
- We used a HindiWords generator utility class to generate the text files containing Hindi words.
- We have implemented the MergeHuskySort and PureHuskySort from the HuskySort repository by Professor Robin Hillyard.
- We have added test cases for edge cases as well as for the happy flow. All the test cases executed successfully.

⦿ Relationship Conclusion:

On comparing the run time performances of the various sort methods, the below relationship has been concluded:

1. Randomly sorted array:

LSD > Dual-Pivot Quicksort > Tim Sort > MSD > Husky Sort

2. Reverse sorted array:

LSD > Dual-Pivot Quicksort > Tim Sort > MSD > Husky Sort

3. Partially sorted array

LSD > Dual-Pivot Quicksort > Tim Sort > MSD > Husky Sort

The above relationship implies that LSD has the best run time performance and Husky sort has worst run time performance.

⦿ Evidence to support the conclusion:

1. Output (Snapshot of Code output in the terminal)

- LSD sort

```
BENCHMARKING USING LSD Radix Sort
-----
Benchmarking the randomly ordered array
|
Time : 444.0
Time : 395.0
Time : 401.0
Time : 370.0
Time : 472.0
Time : 414.0
Time : 449.0
Time : 402.0
Time : 364.0
```

- **MSD sort**

```
-----  
BENCHMARKING USING MSD Radix Sort  
-----  
  
Benchmarking the randomly ordered array  
  
Time : 741.0  
Time : 857.0  
Time : 664.0  
Time : 724.0  
Time : 644.0  
Time : 703.0  
Time : 729.0  
Time : 711.0  
Time : 700.0  
-----
```

- **Dual-pivot Quicksort**

```
-----  
BENCHMARKING USING Dual-Pivot Quick Sort  
-----  
  
Benchmarking the randomly ordered array  
  
Time : 350.0  
Time : 372.0  
Time : 357.0  
Time : 361.0  
Time : 332.0  
Time : 376.0  
Time : 347.0  
Time : 343.0  
Time : 531.0  
-----
```

- **Timsort**

```
-----  
BENCHMARKING USING Tim Sort  
-----  
  
Benchmarking the randomly ordered array  
  
Time : 521.0  
Time : 498.0  
Time : 539.0  
Time : 541.0  
Time : 501.0  
Time : 306.0  
Time : 520.0  
Time : 487.0  
Time : 549.0  
-----
```

- Husky Sort

```

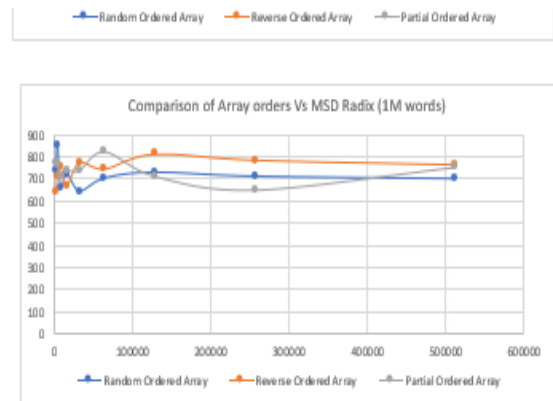
BENCHMARKING USING Husky Sort
-----
Benchmarking the randomly ordered array

Time : 794.0
Time : 711.0
Time : 824.0
Time : 895.0
Time : 770.0
Time : 746.0
Time : 762.0
Time : 781.0
Time : 779.0
-----

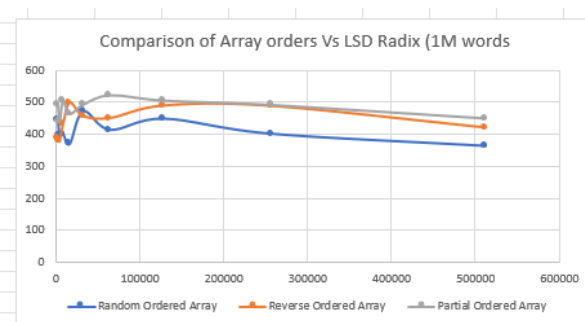
```

2. Graphical Representation(Observations from experiments should be tabulated and analyzed by plotting graphs(usually in excel) to arrive on the relationship conclusion)

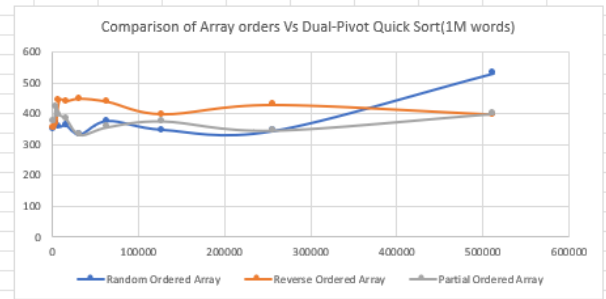
| MSD RADIX SORT (1M Words) | | | |
|---------------------------|----------------------|-----------------------|-----------------------|
| No. of Elements | Time Taken(ms) | | |
| | Random Ordered Array | Reverse Ordered Array | Partial Ordered Array |
| 2000 | 741 | 640 | 776 |
| 4000 | 857 | 714 | 762 |
| 8000 | 664 | 755 | 712 |
| 16000 | 724 | 671 | 739 |
| 32000 | 644 | 775 | 739 |
| 64000 | 703 | 746 | 825 |
| 128000 | 729 | 815 | 713 |
| 256000 | 711 | 787 | 649 |
| 512000 | 700 | 766 | 753 |



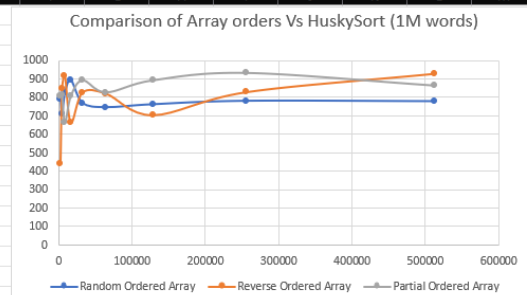
| LSD RADIX SORT (1M Words) | | | |
|---------------------------|----------------------|-----------------------|-----------------------|
| No. of Elements | Time Taken(ms) | | |
| | Random Ordered Array | Reverse Ordered Array | Partial Ordered Array |
| 2000 | 444 | 389 | 493 |
| 4000 | 395 | 380 | 440 |
| 8000 | 401 | 431 | 504 |
| 16000 | 370 | 497 | 463 |
| 32000 | 472 | 456 | 493 |
| 64000 | 414 | 450 | 523 |
| 128000 | 449 | 488 | 506 |
| 256000 | 402 | 487 | 492 |
| 512000 | 364 | 421 | 449 |



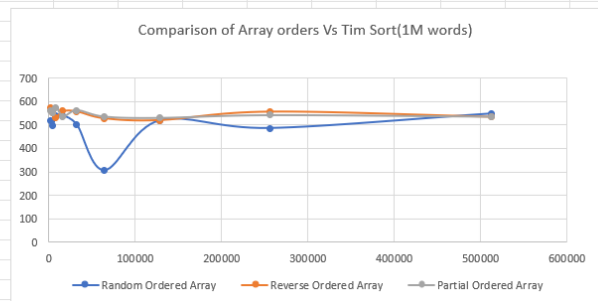
| DUAL-PIVOT QUICK SORT (1M Words) | | | |
|----------------------------------|----------------------|-----------------------|-----------------------|
| Time Taken(ms) | | | |
| No. of Elements | Random Ordered Array | Reverse Ordered Array | Partial Ordered Array |
| 2000 | 350 | 352 | 374 |
| 4000 | 372 | 367 | 420 |
| 8000 | 357 | 443 | 397 |
| 16000 | 361 | 440 | 382 |
| 32000 | 332 | 448 | 331 |
| 64000 | 376 | 438 | 356 |
| 128000 | 347 | 398 | 375 |
| 256000 | 343 | 428 | 345 |
| 512000 | 531 | 397 | 399 |



| HUSKY SORT (1 Million Words) | | | |
|------------------------------|----------------------|-----------------------|-----------------------|
| Time Taken(ms) | | | |
| No. of Elements | Random Ordered Array | Reverse Ordered Array | Partial Ordered Array |
| 2000 | 794 | 443 | 809 |
| 4000 | 711 | 847 | 817 |
| 8000 | 824 | 920 | 665 |
| 16000 | 895 | 664 | 807 |
| 32000 | 770 | 828 | 893 |
| 64000 | 746 | 822 | 828 |
| 128000 | 762 | 704 | 893 |
| 256000 | 781 | 830 | 933 |
| 512000 | 779 | 930 | 865 |

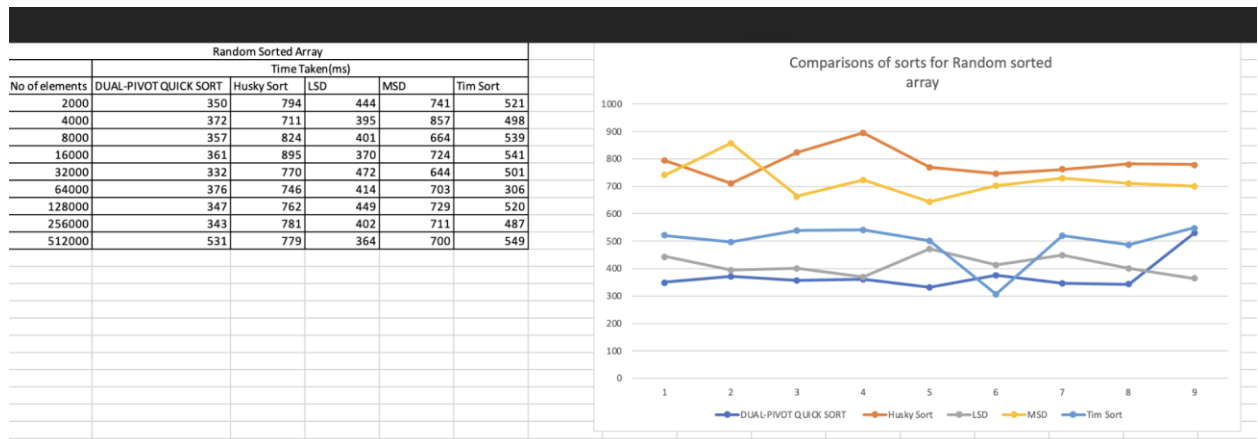


| TIM SORT (1M Words) | | | |
|---------------------|----------------------|-----------------------|-----------------------|
| Time Taken(ms) | | | |
| No. of Elements | Random Ordered Array | Reverse Ordered Array | Partial Ordered Array |
| 2000 | 521 | 573 | 563 |
| 4000 | 498 | 551 | 555 |
| 8000 | 539 | 533 | 576 |
| 16000 | 541 | 560 | 538 |
| 32000 | 501 | 559 | 563 |
| 64000 | 306 | 528 | 537 |
| 128000 | 520 | 520 | 532 |
| 256000 | 487 | 558 | 543 |
| 512000 | 549 | 535 | 537 |

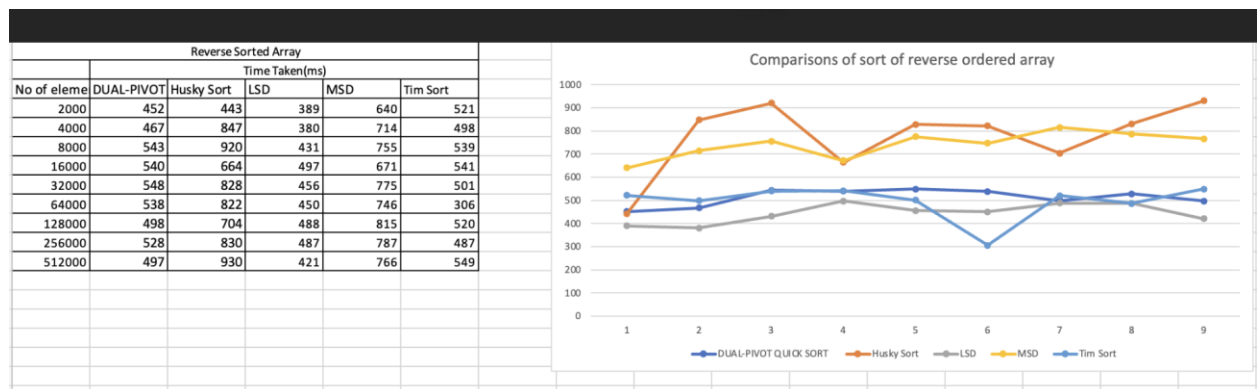


Screenshot supporting the conclusion:

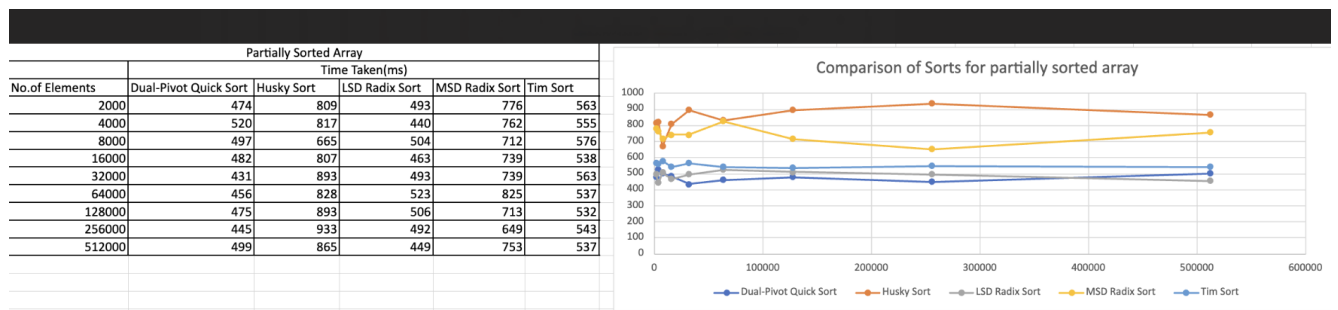
➤ Random Sorted Array analysis



➤ Reverse sorted Array analysis



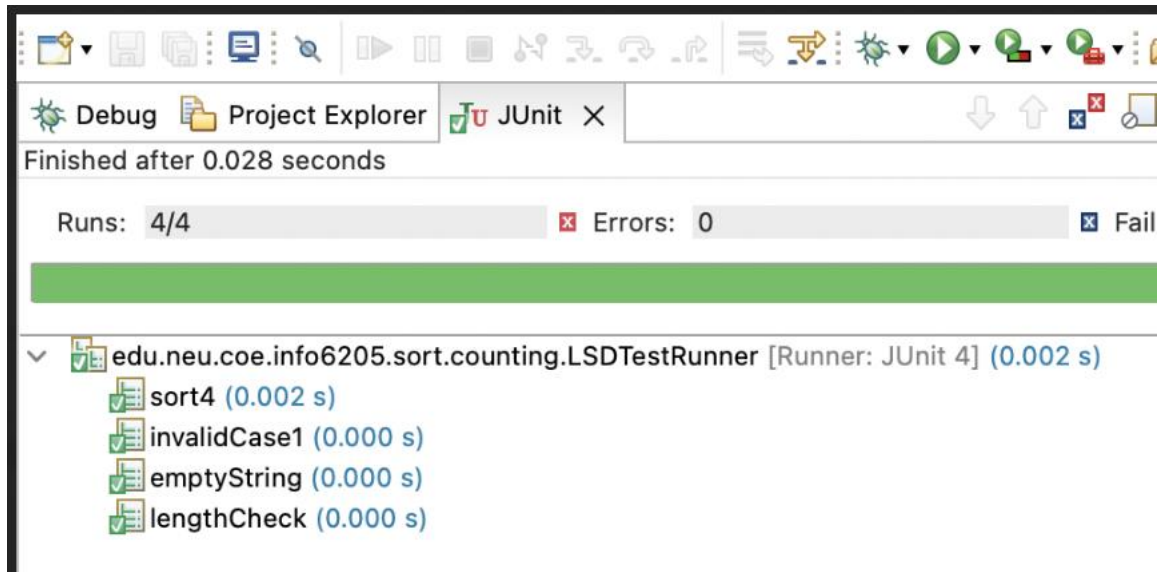
➤ Partially sorted array



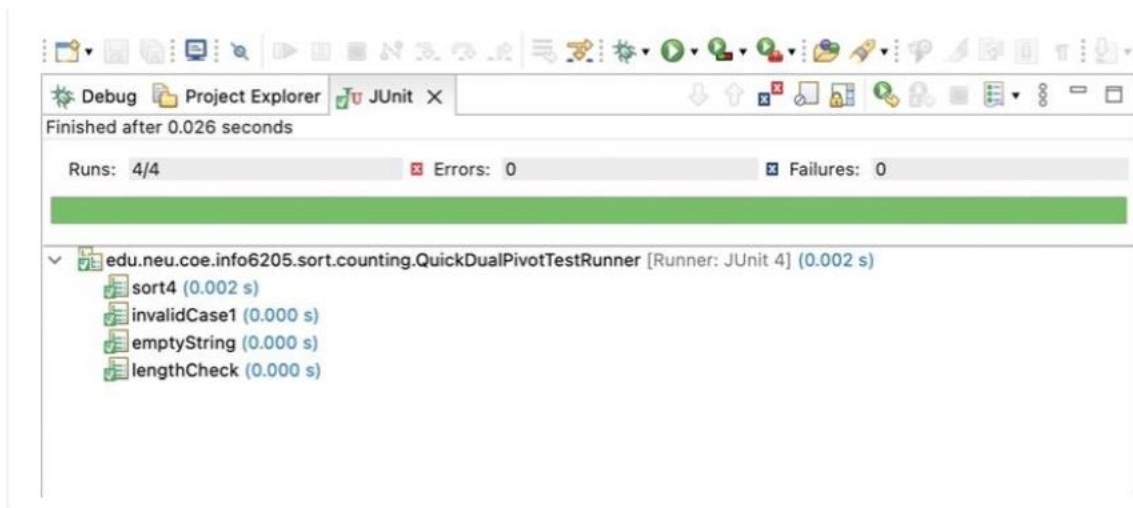
- **Unit tests result:(Snapshot of successful unit test run)**

Below are the screenshots of the unit tests run for each sorting method.

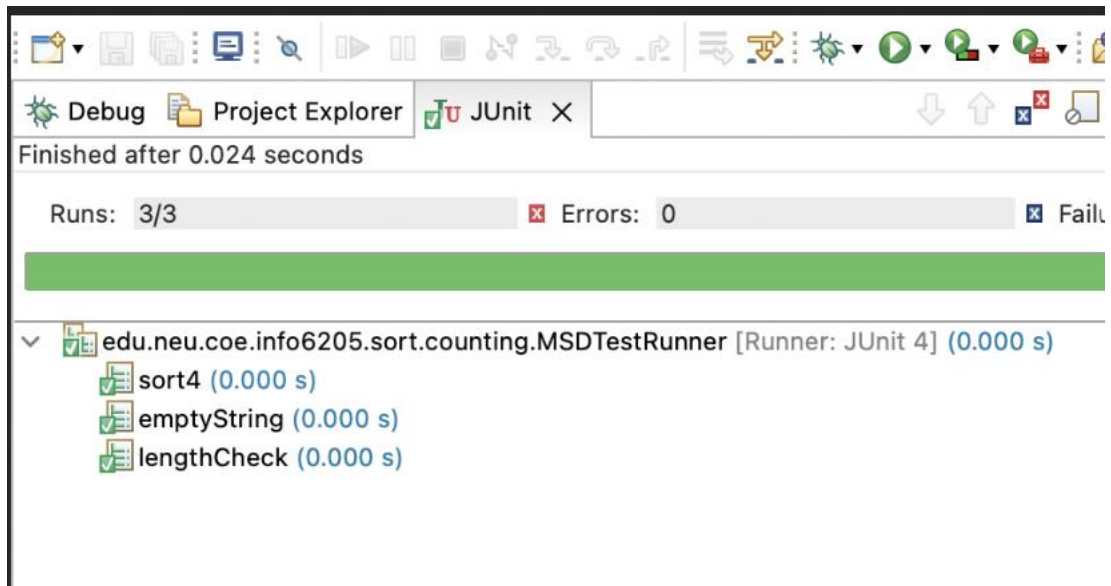
- LSD



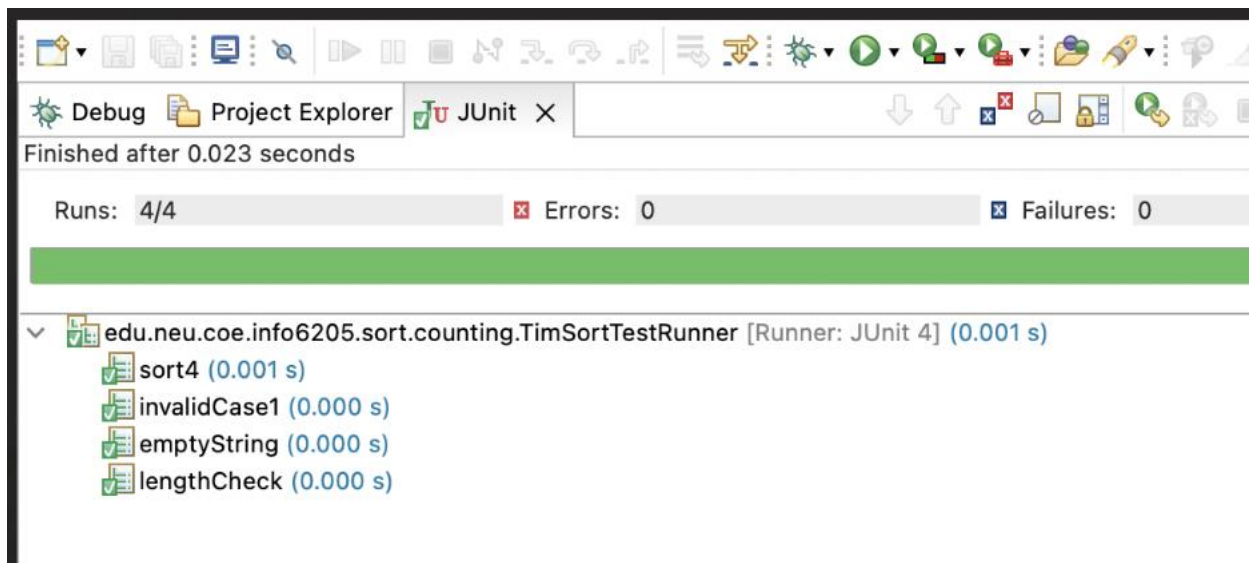
- Dual Pivot Quicksort



- MSD



➤ Timsort



➤ Husky Sort

