Divya Kulkarni (NUID # 001543678)

INFO 6205
Program Structures & Algorithms
Spring 2021

Task – Merge Sort Optimization

The purpose of this assignment is to implement the two Merge Sort optimization techniques, Insurance Test and No Copy in MergeSortBasic.java, modify config.ini file, implement the test cases, and Benchmark it.

I. Implementation

To implement this merge sort benchmarking case, I added insuranceOnOff and noCopyOnOff in the helper section of Config.ini file. Then added two boolean variables insurance, and noCopy to the constructor of MergeSortBasic.java.

Public method sortWithNoCopyAndInsurance:

Public entry for other methods to call the merge sort with two improvements. This method accepts an array to be sorted and returns sorted array by using the logic of merge sort with insurance and noCopy improvements implemented.

If noCopy is set to 'true' then aux array is pointing to the original array. If it is ‘false’, then original array elements are copied over to the auxiliary array.

Private method sortWithNoCopyAndInsurance:

I implemented this private method sortWithNoCopyAndInsurance to implement Insurance and NoCopy logic.

For small arrays (satisfying cut-off condition), I have implemented insertion sort.

sortWithNoCopyAndInsurance has been called to recursively sort left sub array and then same method is called to recursively sort the right sub array.

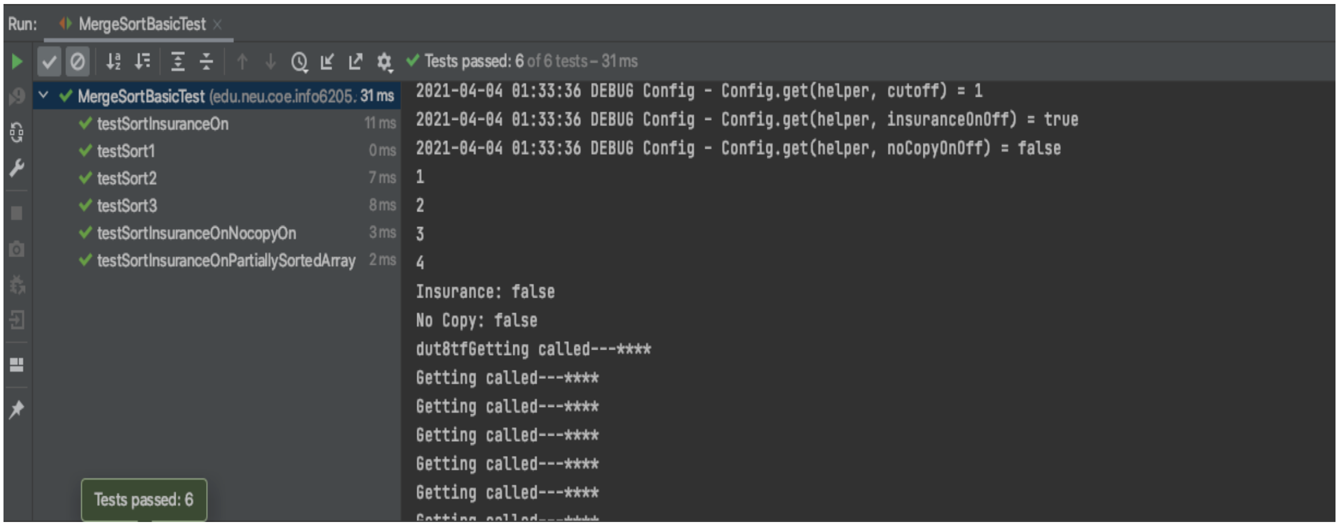
If insurance variable is true, then we compare the last element of the left sub-array with the first element of the right subarray. If the left element of the left sub-array is smaller than the first element of the right subarray, then we skip the merge.

Now, we call for the method mergeForNoCopy to merge the two sorted arrays.

For noCopy improvement as I have switched the roles of source array with destination array while recursively calling sortWithNoCopyAndInsurance.

II. Testing

To test the implemented improvement techniques, I created three test cases: testSortInsuranceOn, testSortInsuranceOnNoCopyOn, testSortInsuranceOnPartiallySortedArray



Run: MergeSortBasicTest

✓ Tests passed: 6 of 6 tests – 31 ms

```
✓ MergeSortBasicTest (edu.neu.coe.info6205. 31 ms
    ✓ testSortInsuranceOn                       11 ms
    ✓ testSort1                                  0 ms
    ✓ testSort2                                  7 ms
    ✓ testSort3                                  8 ms
    ✓ testSortInsuranceOnNocopyOn                3 ms
    ✓ testSortInsuranceOnPartiallySortedArray    2 ms
```

```
2021-04-04 01:33:36 DEBUG Config - Config.get(helper, cutoff) = 1
2021-04-04 01:33:36 DEBUG Config - Config.get(helper, insuranceOnOff) = true
2021-04-04 01:33:36 DEBUG Config - Config.get(helper, noCopyOnOff) = false
1
2
3
4
Insurance: false
No Copy: false
dut8tfGetting called---****
Getting called---****
Getting called---****
Getting called---****
Getting called---****
Getting called---****
Getting called---****
```

Tests passed: 6

III. Benchmarking

To run the different combinations, I have changed the values in the config.ini file directly.

To do the benchmarking, I had to modify the second constructor of MergeSortBasic class that takes in two parameters, n and config. Now this will extract insuranceOnOff, and noCopyOnOff to store it in the instance variables.

I modified the existing **public X[] sort** methods to call the newly implemented sortWithNoCopyAndInsurance just for the purpose of Benchmarking.

I have created Benchmark_Mergesort.java file to benchmark and analyze the performance of Insurance test on partially sorted array.

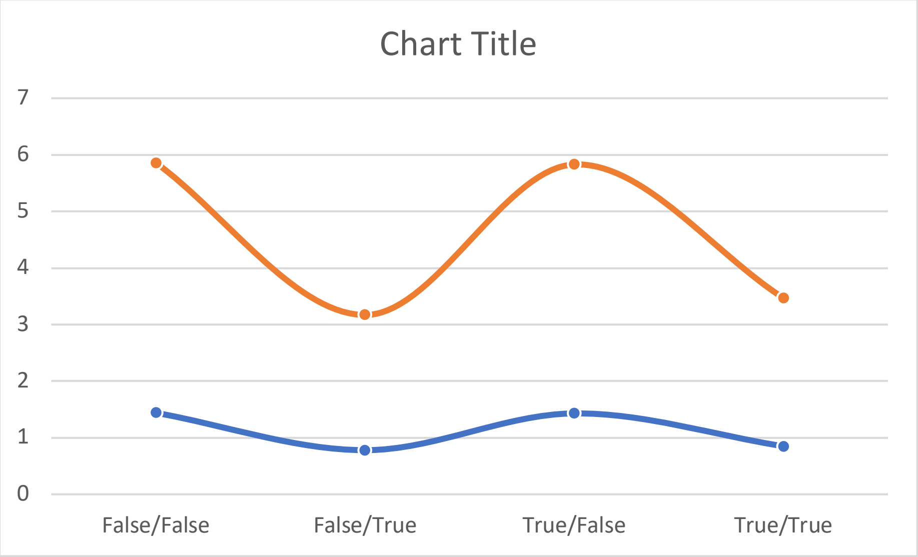
IV. Observation and Conclusion

Ran the file SortBenchmark.java. I analyzed that when both insurance and noCopy is set to True, time taken by the algorithm with implemented improvement is much lesser than without improvements.

Divya Kulkarni (NUID # 001543678)

Insurance: true
No Copy: true
2021-04-04 01:46:45 INFO SorterBenchmark - run: sort 5,000 elements using SorterBenchmark on class java.lang.String from 24,017 total elements and 1,000 runs using sorter: MergeSort

Insurance/NoCopy	Raw (mSec)	Normalized (n log n)
False/False	1.44	4.42
False/True	0.78	2.39
True/False	1.43	4.4
True/True	0.85	2.62



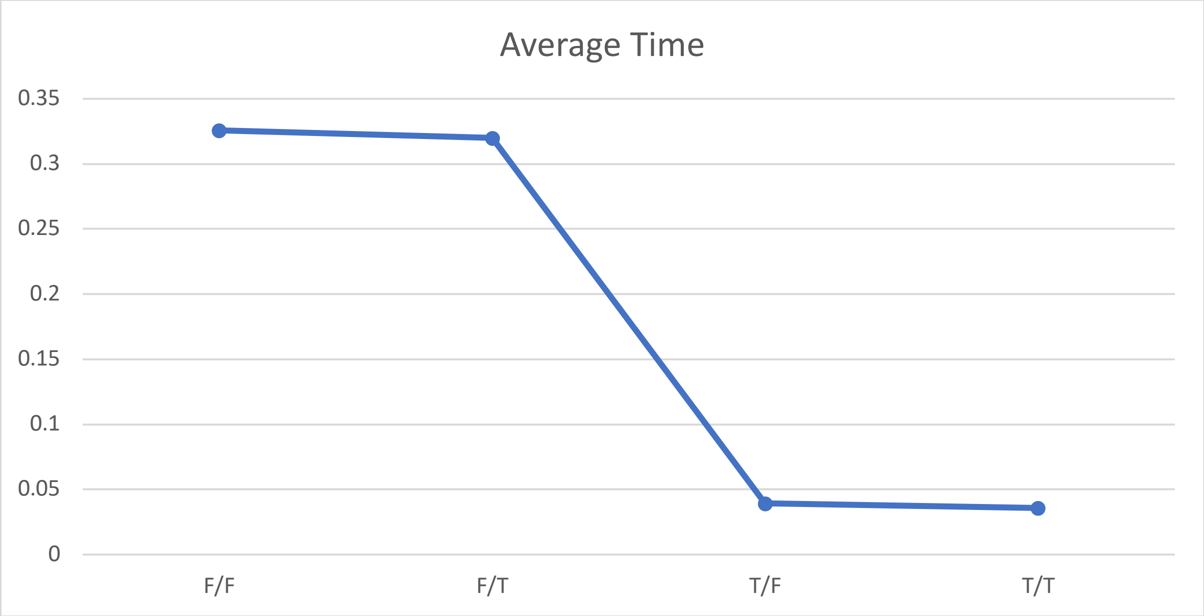
Observation for Insurance Test

I have Implemented benchmarking logic in Benchmark_Mergesort.java. I ran the experiment 1000 times for partially sorted array of 5000 elements. It is observed that when Insurance test was set to true, average time was much lesser. (Enclosed Raw_Data_PartiallySortedArray.xls file)

Insurance/NoCopy	Average Time(mSec)
F/F	0.325789782
F/T	0.319862937

Divya Kulkarni (NUID # 001543678)

T/F	0.039177847
T/T	0.035781925


Average Time
F/F
F/T
T/F
T/T