Program Structures and Algorithms
Spring 2023(SEC –01)

NAME: Ashi Tyagi
NUID: 002706544

Task:

Solve 3-SUM using the *Quadrithmic*, *Quadratic*, and (bonus point) *quadraticWithCalipers* approaches, as shown in skeleton code in the repository. There are hints at the end of Lesson 2.5 Entropy.

There are also hints in the comments of the existing code. There are a number of unit tests which you should be able to run successfully.

Submit (in your own repository--see instructions elsewhere--include the source code and the unit tests of course):

(a) evidence (screenshot) of your unit tests running (try to show the actual unit test code as well as the green strip);

(b) a spreadsheet showing your timing observations--using the doubling method for at least five values of N--for each of the algorithms (include cubic); Timing should be performed either with an actual stopwatch (e.g. your iPhone) or using the Stopwatch class in the repository.

(c) your brief explanation of why the quadratic method(s) work.

Relationship Conclusion:

Implementation of ThreeSum which follows the brute-force approach of testing every candidate
in the solution-space.The array provided in the constructor may be randomly ordered.
Construct a ThreeSumCubic on a.
param a :an array.
For Quadratic:
Implementation of ThreeSum which follows the approach of dividing the solution-space into N sub-spaces where each sub-space corresponds to a fixed value for the middle index of the three values.
Each sub-space is then solved by expanding the scope of the other two indices outwards from the starting point.
Since each sub-space can be solved in O(N) time, the overall complexity is O(N^2).
Construct a ThreeSumQuadratic on a.
@param a :a sorted array.
Get a list of Triples such that the middle index is the given value j.
@param j :the index of the middle value.
@return a Triple
For Quadratic with Calipers:

Implementation of ThreeSum which follows the approach of dividing the solution-space into N sub-spaces where each sub-space corresponds to a fixed value for the middle index of the three values.
Each sub-space is then solved by expanding the scope of the other two indices outwards from
the starting point.
Since each sub-space can be solved in O(N) time, the overall complexity is O(N^2).
The array provided in the constructor MUST be ordered.
Construct a ThreeSumQuadratic on a.
@param a: a sorted array.
Get a list of Triples such that the middle index is the given value i.
@param a : a sorted array of ints.
@param i : the index of the first element of resulting triples.
@param function : a function which takes a triple and returns the comparison of sum of the triple with zero.
@return a Tripl
It can be observed from the results of the benchmark test:
In the worst case scenario which happens when we generate all possible triplets and compare the sum of every triplet with the given value and therefore, runs in cubic time: O(n^3).
In the average and best case scenario which follows the approach of dividing the solution-space
into N sub-spaces where each sub-space corresponds to a fixed value for the middle index of the three values. Each sub-space is then solved by expanding the scope of the other two indices outwards from the starting point. The array provided must be sorted.Since each sub-space can be solved in O(N) time, the overall complexity is O(N^2).

Evidence to support that conclusion:

Value of n	No. of runs	Cubic raw time	Cubic Normalized Time(n^3)
250	100	11.48	0.73
500	50	57.66	0.46
1000	20	453.35	0.45
2000	10	3629.8	0.45
4000	5	28837	0.45
8000	3	NA	NA
16000	2	NA	NA

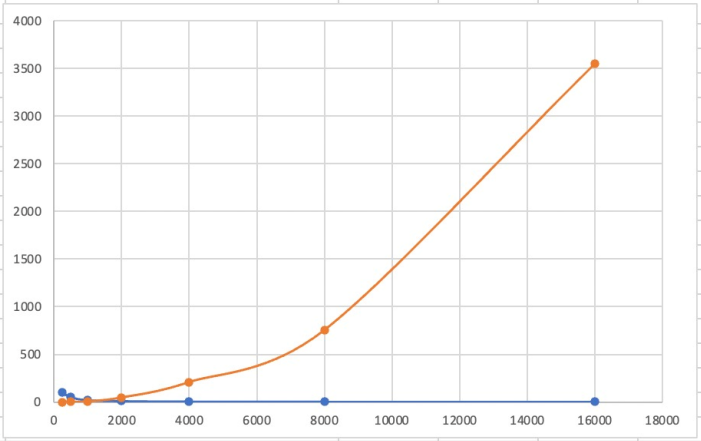
Value of n	No. of runs	Quadrithmic Raw Time	Quadrithmic Normalized Time(n^2 logn)
250	100	1.4	2.81
500	50	5.1	2.28
1000	20	17.5	1.76
2000	10	90.8	2.07
4000	5	380.8	1.99
8000	3	1632	1.96
16000	2	6781.5	1.9

Value of n	No. of runs	QuadrithmicWcaliper Raw Time	QuadrithmicWcalipers Normalized Time(n^2)
250	100	0.44	7.04
500	50	1.14	4.56
1000	20	7.85	7.85
2000	10	47.2	11.8
4000	5	209.6	13.1
8000	3	756.67	11.82
16000	2	3550.5	13.87

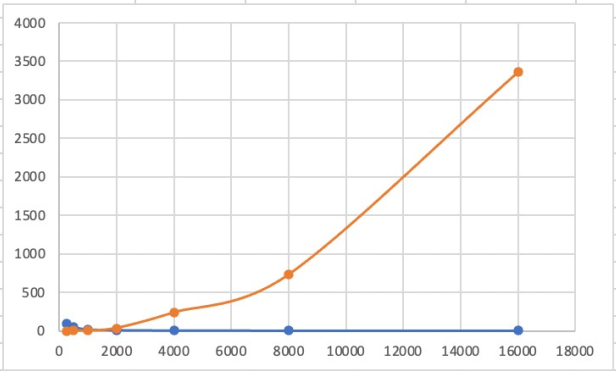
Value of n	Cubic raw time	Quadritic Raw Time	Quadrathmic Raw Time	QuadrithmicWcaliper Raw Time
250	11.48	0.7	1.4	0.44
500	57.66	1.8	5.1	1.14
1000	453.35	4.45	17.5	7.85
2000	3629.8	34.3	90.8	47.2
4000	28837	236	380.8	209.6
8000	NA	728.33	1632	756.67
16000	NA	3359.5	6781.5	3550.5

Graphical Representation:

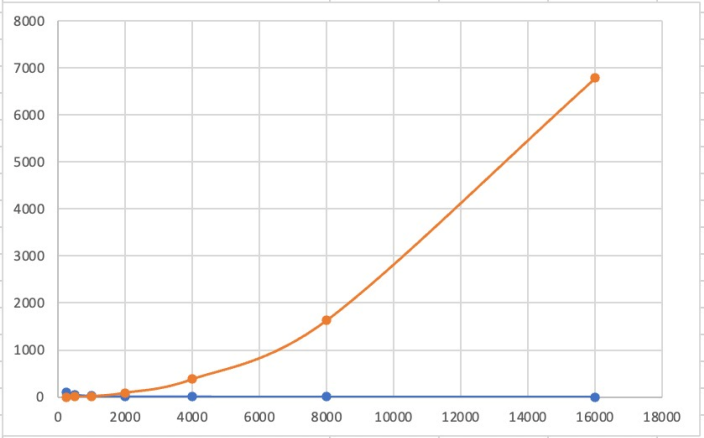
Value of n	No. of runs	QuadrithmicWcaliper Raw Time	QuadrithmicWcalipers Normalized Time(n^2)
250	100	0.44	7.04
500	50	1.14	4.56
1000	20	7.85	7.85
2000	10	47.2	11.8
4000	5	209.6	13.1
8000	3	756.67	11.82
16000	2	3550.5	13.87



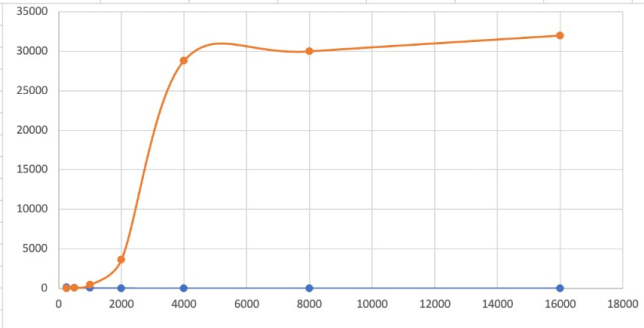
Value of n	No. of runs	Quadritic Raw Time	Quadritic Normalized Time(n^2)
250	100	0.7	11.2
500	50	1.8	7.2
1000	20	4.45	4.45
2000	10	34.3	8.57
4000	5	236	14.75
8000	3	728.33	11.38
16000	2	3359.5	13.12

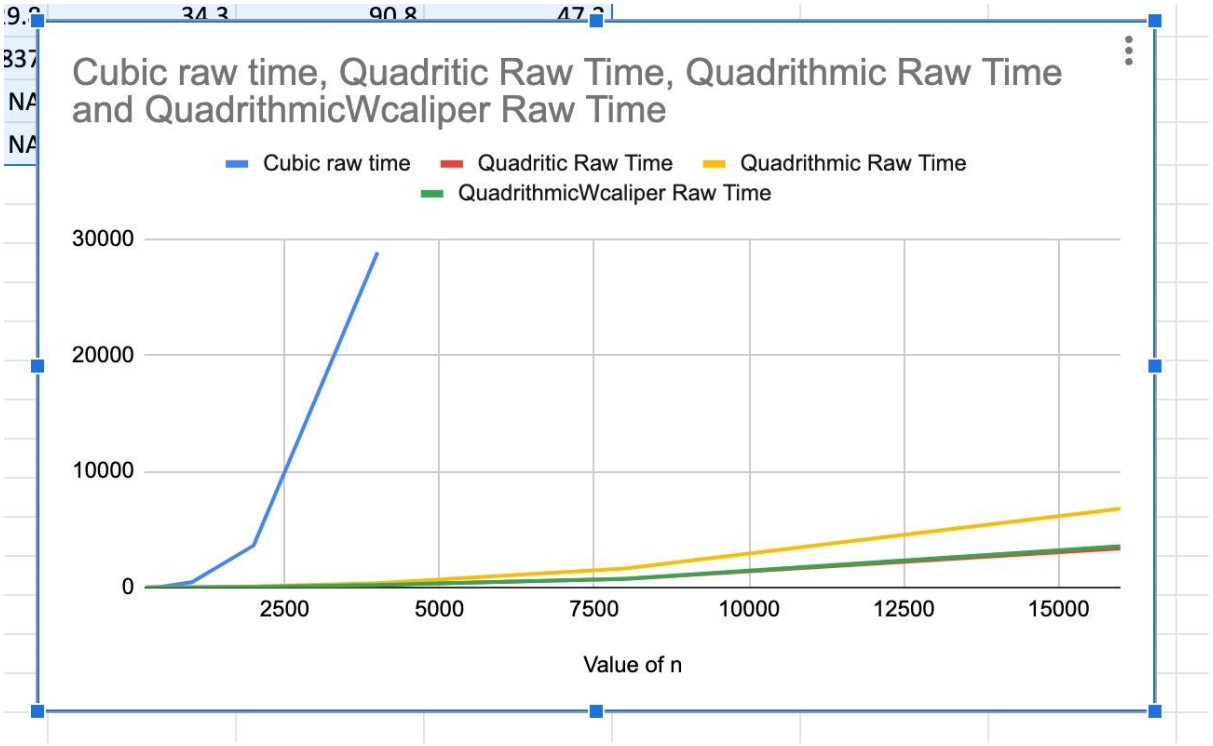


Value of n	No. of runs	Quadrithmic Raw Time	Quadrithmic Normalized Time(n^2 logn)
250	100	1.4	2.81
500	50	5.1	2.28
1000	20	17.5	1.76
2000	10	90.8	2.07
4000	5	380.8	1.99
8000	3	1632	1.96
16000	2	6781.5	1.9

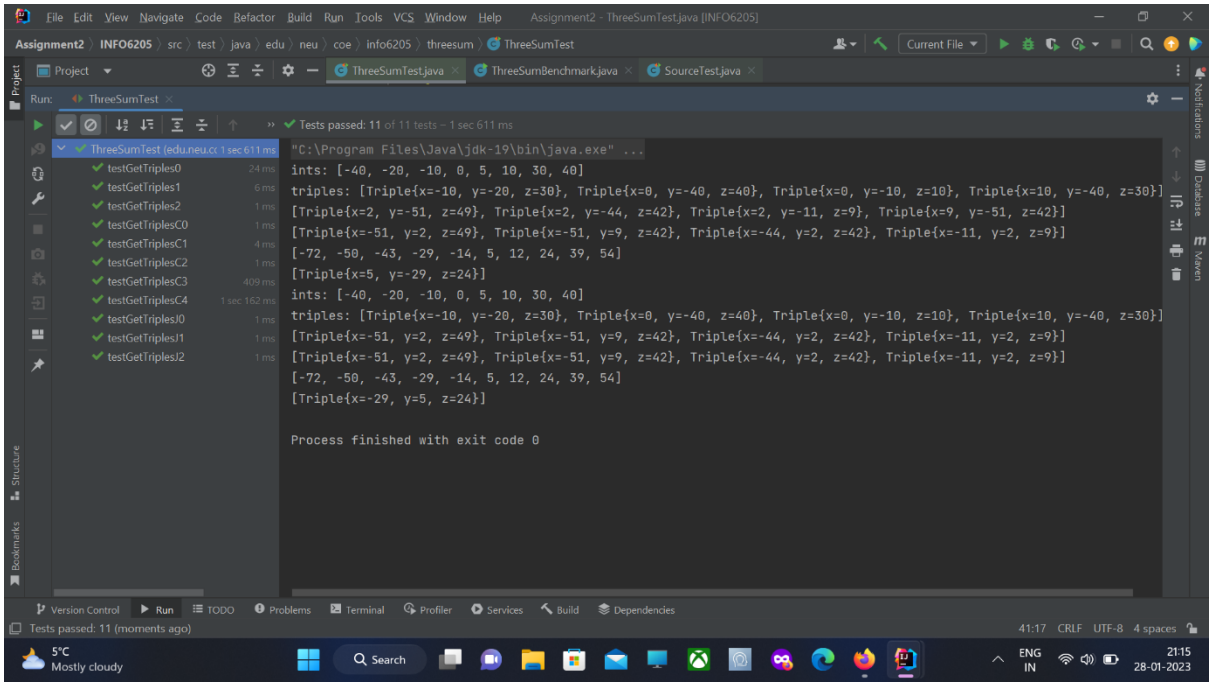


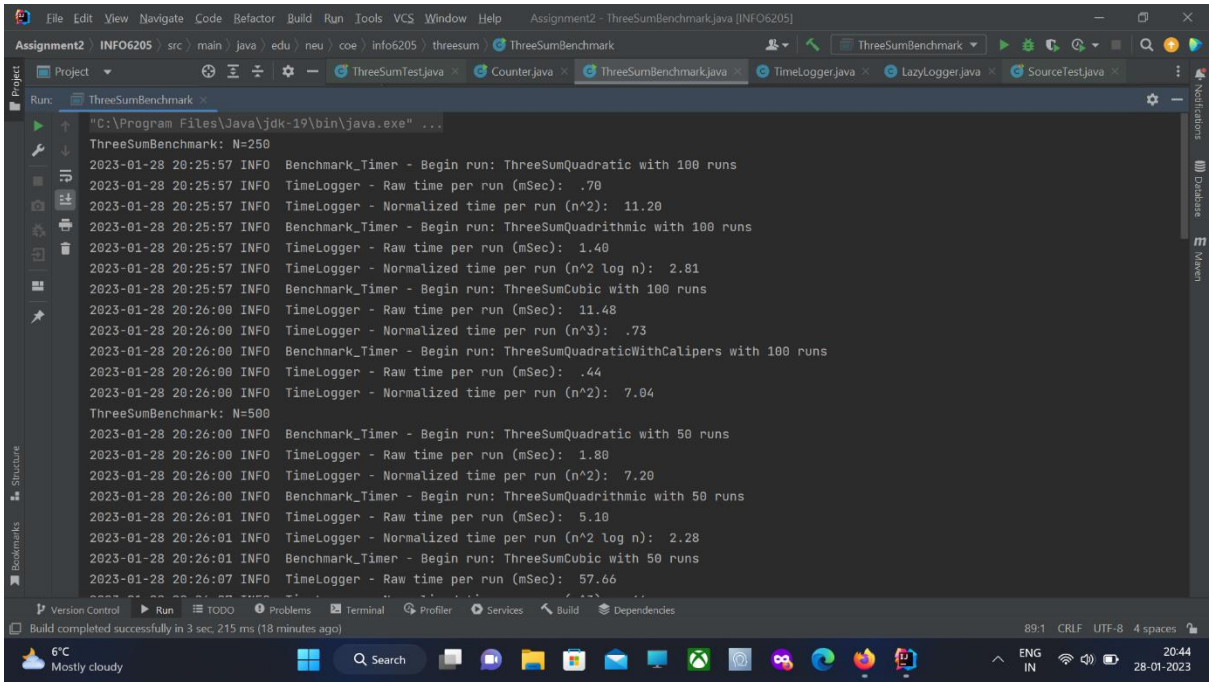
Value of n	No. of runs	Cubic raw time	Cubic Normalized Time(n^3)
250	100	11.48	0.73
500	50	57.66	0.46
1000	20	453.35	0.45
2000	10	3629.8	0.45
4000	5	28837	0.45
8000	3	30000	0.45
16000	2	32000	0.45

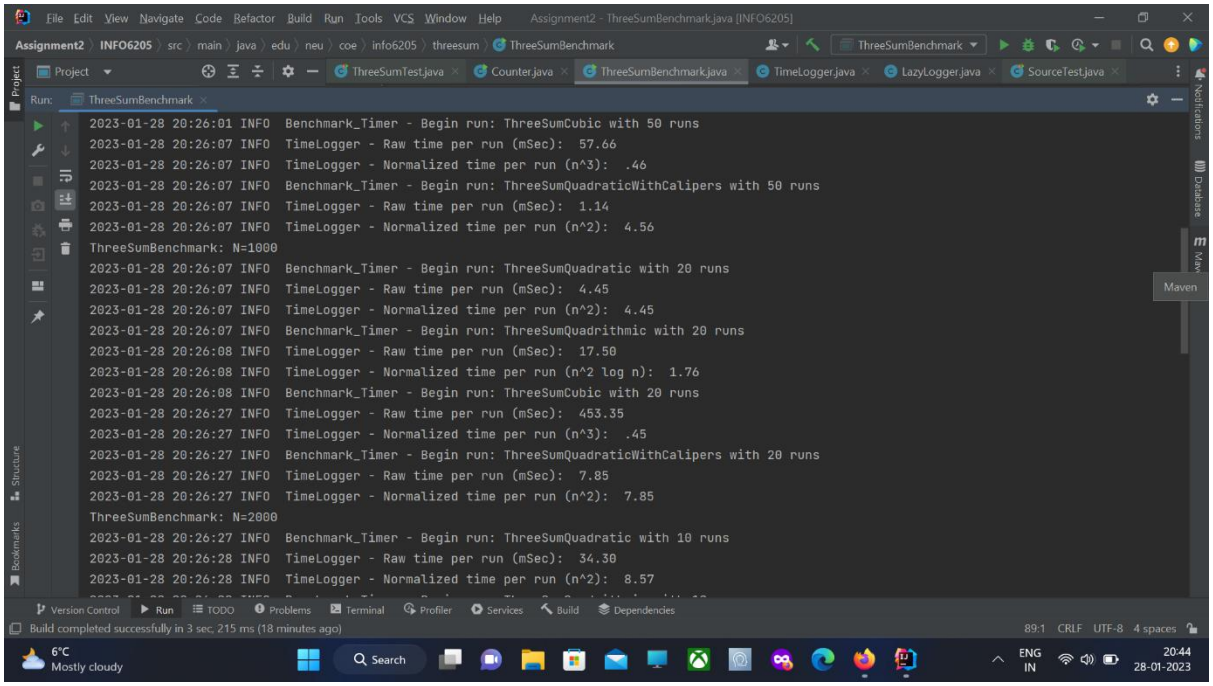


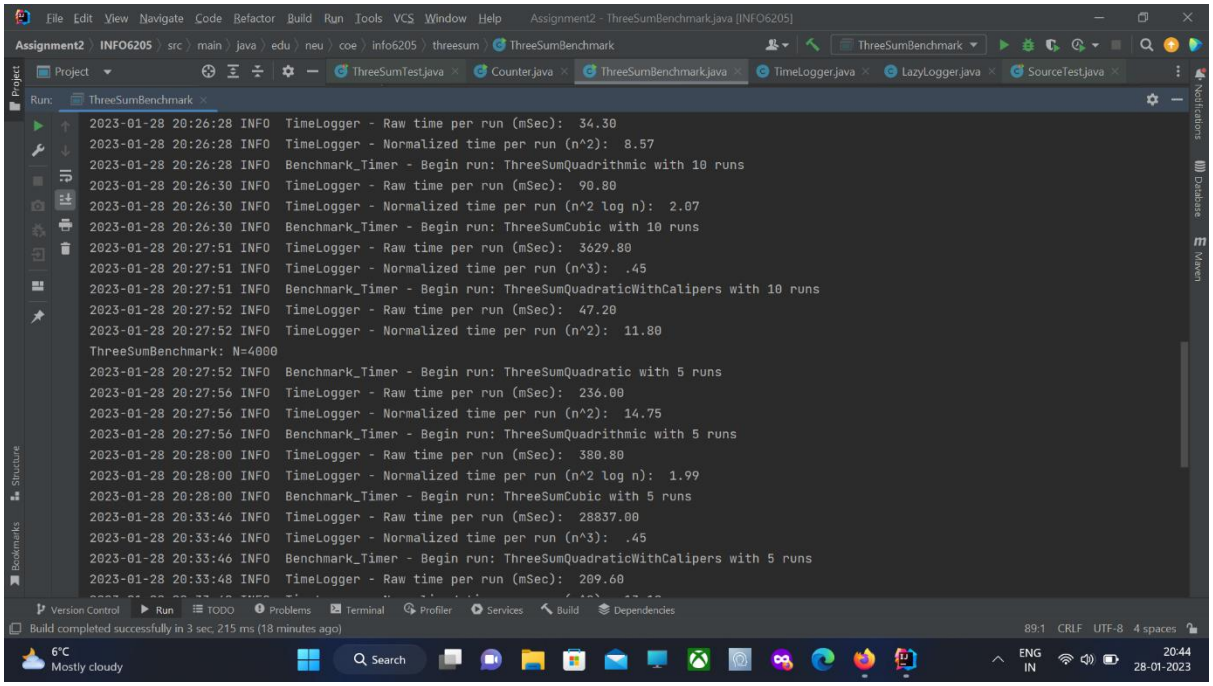


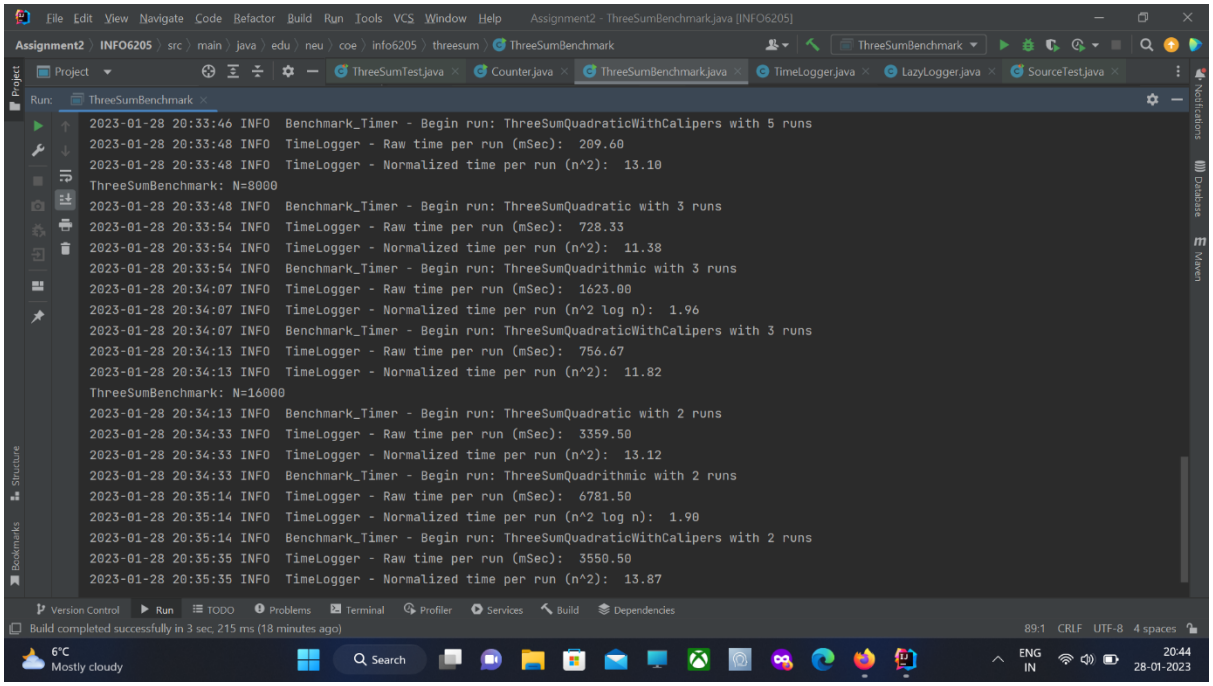
Unit Test Screenshots:




File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
Assignment2 - ThreeSumBenchmark.java [INFO6205]
Assignment2 INFO6205 src main java edu neu coe info6205 threesum ThreeSumBenchmark
ThreeSumBenchmark
Project
ThreeSumTest.java
Counter.java
ThreeSumBenchmark.java
TimeLogger.java
LazyLogger.java
SourceTest.java
Run: ThreeSumBenchmark
"C:\Program Files\Java\jdk-19\bin\java.exe" ...
ThreeSumBenchmark: N=250
2023-01-28 20:25:57 INFO Benchmark_Timer - Begin run: ThreeSumQuadratic with 100 runs
2023-01-28 20:25:57 INFO TimeLogger - Raw time per run (mSec): .70
2023-01-28 20:25:57 INFO TimeLogger - Normalized time per run (n^2): 11.20
2023-01-28 20:25:57 INFO Benchmark_Timer - Begin run: ThreeSumQuadrithmic with 100 runs
2023-01-28 20:25:57 INFO TimeLogger - Raw time per run (mSec): 1.40
2023-01-28 20:25:57 INFO TimeLogger - Normalized time per run (n^2 log n): 2.81
2023-01-28 20:25:57 INFO Benchmark_Timer - Begin run: ThreeSumCubic with 100 runs
2023-01-28 20:26:00 INFO TimeLogger - Raw time per run (mSec): 11.48
2023-01-28 20:26:00 INFO TimeLogger - Normalized time per run (n^3): .73
2023-01-28 20:26:00 INFO Benchmark_Timer - Begin run: ThreeSumQuadraticWithCalipers with 100 runs
2023-01-28 20:26:00 INFO TimeLogger - Raw time per run (mSec): .44
2023-01-28 20:26:00 INFO TimeLogger - Normalized time per run (n^2): 7.04
ThreeSumBenchmark: N=500
2023-01-28 20:26:00 INFO Benchmark_Timer - Begin run: ThreeSumQuadratic with 50 runs
2023-01-28 20:26:00 INFO TimeLogger - Raw time per run (mSec): 1.80
2023-01-28 20:26:00 INFO TimeLogger - Normalized time per run (n^2): 7.20
2023-01-28 20:26:00 INFO Benchmark_Timer - Begin run: ThreeSumQuadrithmic with 50 runs
2023-01-28 20:26:01 INFO TimeLogger - Raw time per run (mSec): 5.10
2023-01-28 20:26:01 INFO TimeLogger - Normalized time per run (n^2 log n): 2.28
2023-01-28 20:26:01 INFO Benchmark_Timer - Begin run: ThreeSumCubic with 50 runs
2023-01-28 20:26:07 INFO TimeLogger - Raw time per run (mSec): 57.66
Version Control Run TODO Problems Terminal Profiler Services Build Dependencies
Build completed successfully in 3 sec, 215 ms (18 minutes ago)
89:1 CRLF UTF-8 4 spaces
6°C Mostly cloudy
Search
ENG IN
20:44 28-01-2023


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
Assignment2 - ThreeSumBenchmark.java [INFO6205]
Assignment2 INFO6205 src main java edu neu coe info6205 threesum ThreeSumBenchmark
ThreeSumBenchmark
Project
ThreeSumTest.java
Counter.java
ThreeSumBenchmark.java
TimeLogger.java
LazyLogger.java
SourceTest.java
Run: ThreeSumBenchmark
2023-01-28 20:26:01 INFO Benchmark_Timer - Begin run: ThreeSumCubic with 50 runs
2023-01-28 20:26:07 INFO TimeLogger - Raw time per run (mSec): 57.66
2023-01-28 20:26:07 INFO TimeLogger - Normalized time per run (n^3): .46
2023-01-28 20:26:07 INFO Benchmark_Timer - Begin run: ThreeSumQuadraticWithCalipers with 50 runs
2023-01-28 20:26:07 INFO TimeLogger - Raw time per run (mSec): 1.14
2023-01-28 20:26:07 INFO TimeLogger - Normalized time per run (n^2): 4.56
ThreeSumBenchmark: N=1000
2023-01-28 20:26:07 INFO Benchmark_Timer - Begin run: ThreeSumQuadratic with 20 runs
2023-01-28 20:26:07 INFO TimeLogger - Raw time per run (mSec): 4.45
2023-01-28 20:26:07 INFO TimeLogger - Normalized time per run (n^2): 4.45
2023-01-28 20:26:07 INFO Benchmark_Timer - Begin run: ThreeSumQuadrithmic with 20 runs
2023-01-28 20:26:08 INFO TimeLogger - Raw time per run (mSec): 17.50
2023-01-28 20:26:08 INFO TimeLogger - Normalized time per run (n^2 log n): 1.76
2023-01-28 20:26:08 INFO Benchmark_Timer - Begin run: ThreeSumCubic with 20 runs
2023-01-28 20:26:27 INFO TimeLogger - Raw time per run (mSec): 453.35
2023-01-28 20:26:27 INFO TimeLogger - Normalized time per run (n^3): .45
2023-01-28 20:26:27 INFO Benchmark_Timer - Begin run: ThreeSumQuadraticWithCalipers with 20 runs
2023-01-28 20:26:27 INFO TimeLogger - Raw time per run (mSec): 7.85
2023-01-28 20:26:27 INFO TimeLogger - Normalized time per run (n^2): 7.85
ThreeSumBenchmark: N=2000
2023-01-28 20:26:27 INFO Benchmark_Timer - Begin run: ThreeSumQuadratic with 10 runs
2023-01-28 20:26:28 INFO TimeLogger - Raw time per run (mSec): 34.30
2023-01-28 20:26:28 INFO TimeLogger - Normalized time per run (n^2): 8.57
Maven
Version Control Run TODO Problems Terminal Profiler Services Build Dependencies
Build completed successfully in 3 sec, 215 ms (18 minutes ago)
89:1 CRLF UTF-8 4 spaces
6°C Mostly cloudy
Search
ENG IN
20:44 28-01-2023


2023-01-28 20:26:28 INFO TimeLogger - Raw time per run (mSec): 34.30
2023-01-28 20:26:28 INFO TimeLogger - Normalized time per run (n^2): 8.57
2023-01-28 20:26:28 INFO Benchmark_Timer - Begin run: ThreeSumQuadrithmic with 10 runs
2023-01-28 20:26:30 INFO TimeLogger - Raw time per run (mSec): 90.80
2023-01-28 20:26:30 INFO TimeLogger - Normalized time per run (n^2 log n): 2.07
2023-01-28 20:26:30 INFO Benchmark_Timer - Begin run: ThreeSumCubic with 10 runs
2023-01-28 20:27:51 INFO TimeLogger - Raw time per run (mSec): 3629.80
2023-01-28 20:27:51 INFO TimeLogger - Normalized time per run (n^3): .45
2023-01-28 20:27:51 INFO Benchmark_Timer - Begin run: ThreeSumQuadraticWithCalipers with 10 runs
2023-01-28 20:27:52 INFO TimeLogger - Raw time per run (mSec): 47.20
2023-01-28 20:27:52 INFO TimeLogger - Normalized time per run (n^2): 11.80
ThreeSumBenchmark: N=4000
2023-01-28 20:27:52 INFO Benchmark_Timer - Begin run: ThreeSumQuadratic with 5 runs
2023-01-28 20:27:56 INFO TimeLogger - Raw time per run (mSec): 236.00
2023-01-28 20:27:56 INFO TimeLogger - Normalized time per run (n^2): 14.75
2023-01-28 20:27:56 INFO Benchmark_Timer - Begin run: ThreeSumQuadrithmic with 5 runs
2023-01-28 20:28:00 INFO TimeLogger - Raw time per run (mSec): 380.80
2023-01-28 20:28:00 INFO TimeLogger - Normalized time per run (n^2 log n): 1.99
2023-01-28 20:28:00 INFO Benchmark_Timer - Begin run: ThreeSumCubic with 5 runs
2023-01-28 20:33:46 INFO TimeLogger - Raw time per run (mSec): 28837.00
2023-01-28 20:33:46 INFO TimeLogger - Normalized time per run (n^3): .45
2023-01-28 20:33:46 INFO Benchmark_Timer - Begin run: ThreeSumQuadraticWithCalipers with 5 runs
2023-01-28 20:33:48 INFO TimeLogger - Raw time per run (mSec): 209.60
Build completed successfully in 3 sec, 215 ms (18 minutes ago)
89:1 CRLF UTF-8 4 spaces
6°C Mostly cloudy
Search
ENG IN
20:44 28-01-2023


2023-01-28 20:33:46 INFO Benchmark_Timer - Begin run: ThreeSumQuadraticWithCalipers with 5 runs
2023-01-28 20:33:48 INFO TimeLogger - Raw time per run (mSec): 209.60
2023-01-28 20:33:48 INFO TimeLogger - Normalized time per run (n^2): 13.10
ThreeSumBenchmark: N=8000
2023-01-28 20:33:48 INFO Benchmark_Timer - Begin run: ThreeSumQuadratic with 3 runs
2023-01-28 20:33:54 INFO TimeLogger - Raw time per run (mSec): 728.33
2023-01-28 20:33:54 INFO TimeLogger - Normalized time per run (n^2): 11.38
2023-01-28 20:33:54 INFO Benchmark_Timer - Begin run: ThreeSumQuadrithmic with 3 runs
2023-01-28 20:34:07 INFO TimeLogger - Raw time per run (mSec): 1623.00
2023-01-28 20:34:07 INFO TimeLogger - Normalized time per run (n^2 log n): 1.96
2023-01-28 20:34:07 INFO Benchmark_Timer - Begin run: ThreeSumQuadraticWithCalipers with 3 runs
2023-01-28 20:34:13 INFO TimeLogger - Raw time per run (mSec): 756.67
2023-01-28 20:34:13 INFO TimeLogger - Normalized time per run (n^2): 11.82
ThreeSumBenchmark: N=16000
2023-01-28 20:34:13 INFO Benchmark_Timer - Begin run: ThreeSumQuadratic with 2 runs
2023-01-28 20:34:33 INFO TimeLogger - Raw time per run (mSec): 3359.50
2023-01-28 20:34:33 INFO TimeLogger - Normalized time per run (n^2): 13.12
2023-01-28 20:34:33 INFO Benchmark_Timer - Begin run: ThreeSumQuadrithmic with 2 runs
2023-01-28 20:35:14 INFO TimeLogger - Raw time per run (mSec): 6781.50
2023-01-28 20:35:14 INFO TimeLogger - Normalized time per run (n^2 log n): 1.90
2023-01-28 20:35:14 INFO Benchmark_Timer - Begin run: ThreeSumQuadraticWithCalipers with 2 runs
2023-01-28 20:35:35 INFO TimeLogger - Raw time per run (mSec): 3550.50
2023-01-28 20:35:35 INFO TimeLogger - Normalized time per run (n^2): 13.87
Build completed successfully in 3 sec, 215 ms (18 minutes ago)
89:1 CRLF UTF-8 4 spaces
6°C Mostly cloudy
Search
ENG IN
20:44 28-01-2023