

Program Structures and Algorithms  
Spring 2023(SEC –03)

NAME: Patel Dhruv Rajeshkumar  
NUID: 002928881

**Task:** Implement and compare 3 SUM with 4 different algorithms. The algorithms contains different time complexities, which are

- Quadratic ( $O(n^2)$ )
- Qaudratic with calipers method ( $O(n^2)$ )
- Quadrithmic ( $O(n^2 \lg(n))$ )
- Cubic ( $O(n^3)$ )

**Relationship Conclusion:** After comparing time taken by each method using stopwatch shows that Cubic takes the most time out of all whereas Quadratic take the least time of all. Time taken by all the methods is as follows:

Cubic > Quadrithmic > Qaudratic with calipers method > Quadratic

**Evidence to support that conclusion:**

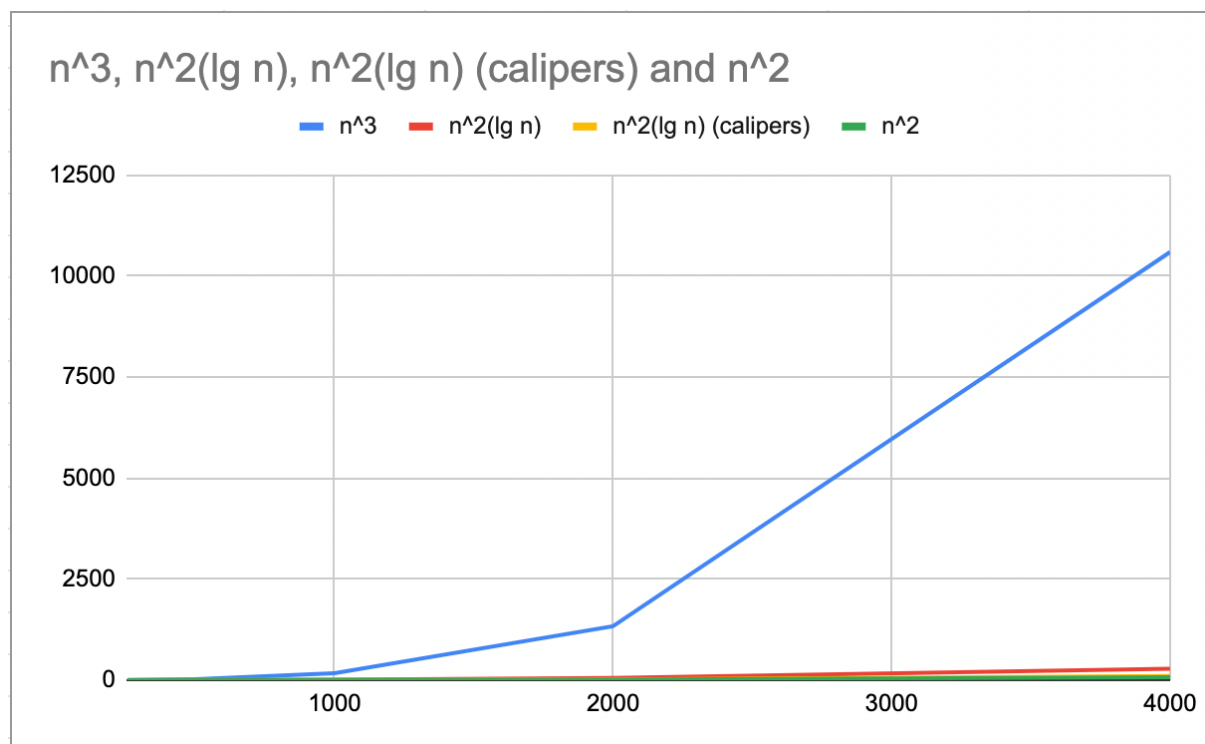
ThreeSumBenchmark: N=250	Time (ms)
Raw time per run (mSec)	0.07
Normalized time per run ( $n^2$ )	1.12
Raw time per run (mSec)	0.05
Normalized time per run ( $n^2$ )(calipers)	0.8
Raw time per run (mSec)	0.05
Normalized time per run ( $n^2 \log n$ )	0.1
Raw time per run (mSec)	2.26
Normalized time per run ( $n^3$ )	0.14
ThreeSumBenchmark: N=500	
Raw time per run (mSec)	0.02
Normalized time per run ( $n^2$ )	0.08
Raw time per run (mSec)	0.84
Normalized time per run ( $n^2$ )(calipers)	3.36

Raw time per run (mSec)	1.04
Normalized time per run ( $n^2 \log n$ )	0.46
Raw time per run (mSec)	21.3
Normalized time per run ( $n^3$ )	0.17
<b>ThreeSumBenchmark: N=1000</b>	
Raw time per run (mSec)	3.1
Normalized time per run ( $n^2$ )	3.1
Raw time per run (mSec)	3.25
Normalized time per run ( $n^2$ )(calipers)	3.25
Raw time per run (mSec)	10.1
Normalized time per run ( $n^2 \log n$ )	1.01
Raw time per run (mSec)	169.1
Normalized time per run ( $n^3$ )	0.17
ThreeSumBenchmark: N=2000	
Raw time per run (mSec)	11.2
Normalized time per run ( $n^2$ )	2.8
Raw time per run (mSec)	17.4
Normalized time per run ( $n^2$ )(calipers)	4.35
Raw time per run (mSec)	56
Normalized time per run ( $n^2 \log n$ )	1.28
Raw time per run (mSec)	1332.7
Normalized time per run ( $n^3$ )	0.17
<b>ThreeSumBenchmark: N=4000</b>	
Raw time per run (mSec)	69.2
Normalized time per run ( $n^2$ )	4.33

Raw time per run (mSec)	102.6
Normalized time per run ( $n^2$ )(calipers)	6.41
Raw time per run (mSec)	280.6
Normalized time per run ( $n^2 \log n$ )	1.47
Raw time per run (mSec)	10593.8
Normalized time per run ( $n^3$ )	0.17

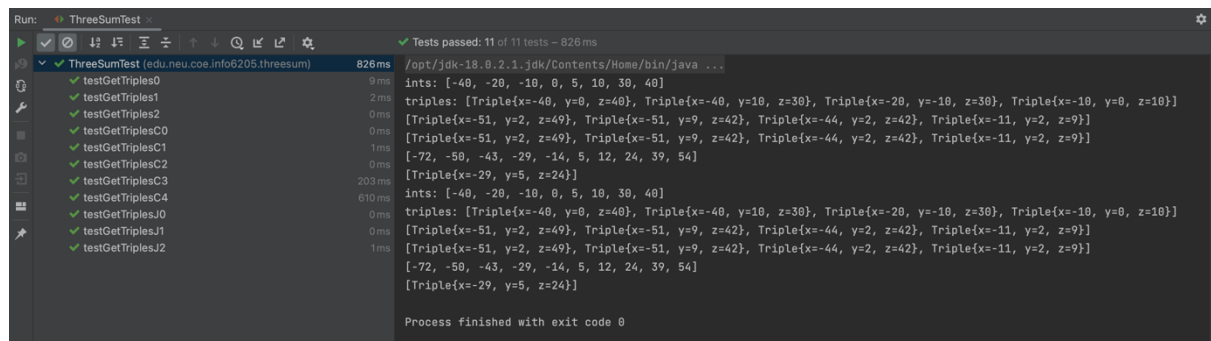
### Graphical Representation:

3 SUM	256	500	1000	2000	4000
$n^3$	2.26	21.3	169.1	1332.7	10593.8
$n^2(\lg n)$	0.05	1.04	10.1	56	280.6
$n^2(\lg n)$ (calipers)	0.05	0.84	3.25	17.4	102.6
$n^2$	0.07	0.02	3.1	11.2	69.2



Here, X Axis represents number of elements in the array. And the Y Axis represents time taken(in milliseconds) by each algorithm.

## Unit Test Screenshots:



```
Run: ThreeSumTest
Tests passed: 11 of 11 tests - 826 ms

ThreeSumTest (edu.neu.coe.info6205.threesum) 826ms /opt/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
  testGetTriples0 9ms ints: [-40, -20, -10, 0, 5, 10, 30, 40]
  testGetTriples1 2ms triples: [Triple{x=-40, y=0, z=40}, Triple{x=-40, y=10, z=30}, Triple{x=-20, y=-10, z=30}, Triple{x=-10, y=0, z=10}]
  testGetTriples2 0ms [Triple{x=-51, y=2, z=49}, Triple{x=-51, y=9, z=42}, Triple{x=-44, y=2, z=42}, Triple{x=-11, y=2, z=9}]
  testGetTriplesC0 0ms [Triple{x=-51, y=2, z=49}, Triple{x=-51, y=9, z=42}, Triple{x=-44, y=2, z=42}, Triple{x=-11, y=2, z=9}]
  testGetTriplesC1 1ms [-72, -50, -43, -29, -14, 5, 12, 24, 39, 54]
  testGetTriplesC2 0ms [Triple{x=-29, y=5, z=24}]
  testGetTriplesC3 203ms ints: [-40, -20, -10, 0, 5, 10, 30, 40]
  testGetTriplesC4 610ms triples: [Triple{x=-40, y=0, z=40}, Triple{x=-40, y=10, z=30}, Triple{x=-20, y=-10, z=30}, Triple{x=-10, y=0, z=10}]
  testGetTriplesJ0 0ms [Triple{x=-51, y=2, z=49}, Triple{x=-51, y=9, z=42}, Triple{x=-44, y=2, z=42}, Triple{x=-11, y=2, z=9}]
  testGetTriplesJ1 0ms [Triple{x=-51, y=2, z=49}, Triple{x=-51, y=9, z=42}, Triple{x=-44, y=2, z=42}, Triple{x=-11, y=2, z=9}]
  testGetTriplesJ2 1ms [-72, -50, -43, -29, -14, 5, 12, 24, 39, 54]
  [Triple{x=-29, y=5, z=24}]

Process finished with exit code 0
```

Please note a screen recorded video has been attached to prove that no test cases were changed while executing these results in the repository in Assignment 2 (3 SUM) folder.