Program Structures and Algorithms
Spring 2023(SEC – 1)

NAME: Pawan Kumar Krishnan
NUID: 002743773

Task:
Solve 3-SUM using the Quadrithmic, Quadratic, and quadraticWithCalipers approaches, as shown in skeleton code in the repository and compare the computational times.

Relationship Conclusion:
Quadrithmic computation raw time is lower than Quadratic computation raw time and Quadratic computation raw time is lower than Cubic computation raw time for N = 250,500.

For N = 1000 and 2000 Quadrithmic computation raw time is pretty much the same as Quadratic computation raw time but the cubic is significantly higher.

For N beyond 8000 , the time gap between quadratic , quadrithmic and cubic is expanding in an accelerated manner.

To conclude as N keeps increasing beyond ~4000, Quadratic is the most time efficient,the gap between quadratic quadrithmic and cubic times widens in an accelerated manner. For N < ~4000 the time difference between quadratic and quadrithmic is insignificant but the difference between cubic and the other two are significant and evidental.

Quadratic functions generally require fewer parameters to fit the same data, which can make them computationally more efficient. However, the usage of cubic/quadratic/quadrithmic depends on the problem and dataset.

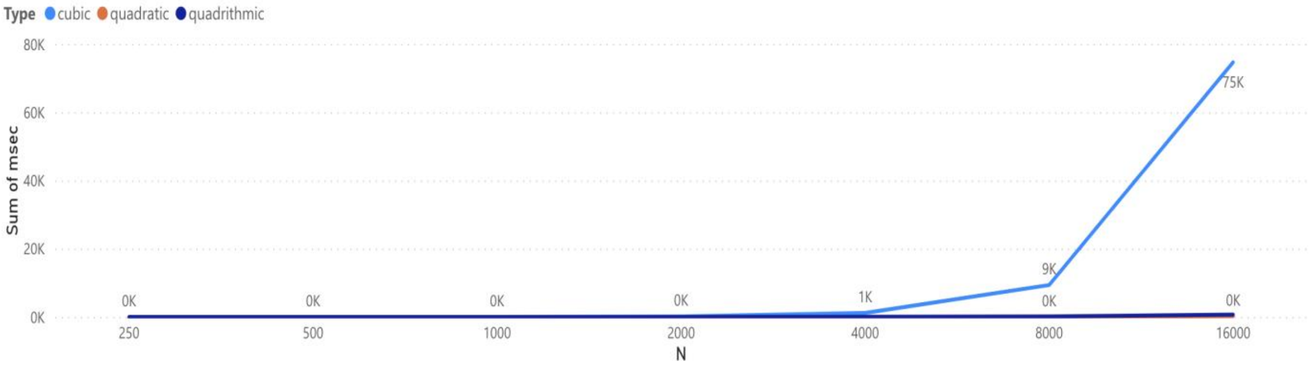
Evidence to support that conclusion:
Below are the raw computational times in miliseconds.

Type	N	msec	Normalized time
quadratic	250	1.9	30.4
quadrithmic	250	1.1	2.21
cubic	250	2	0.13
quadratic	500	1.1	4.4
quadrithmic	500	0.7	0.31
cubic	500	3.5	0.03
quadratic	1000	2.8	2.8
quadrithmic	1000	2.9	0.29
cubic	1000	21.5	0.02
quadratic	2000	8.6	2.15
quadrithmic	2000	8.4	0.19
cubic	2000	151	0.02
quadratic	4000	15.1	0.94
quadrithmic	4000	37.3	0.19

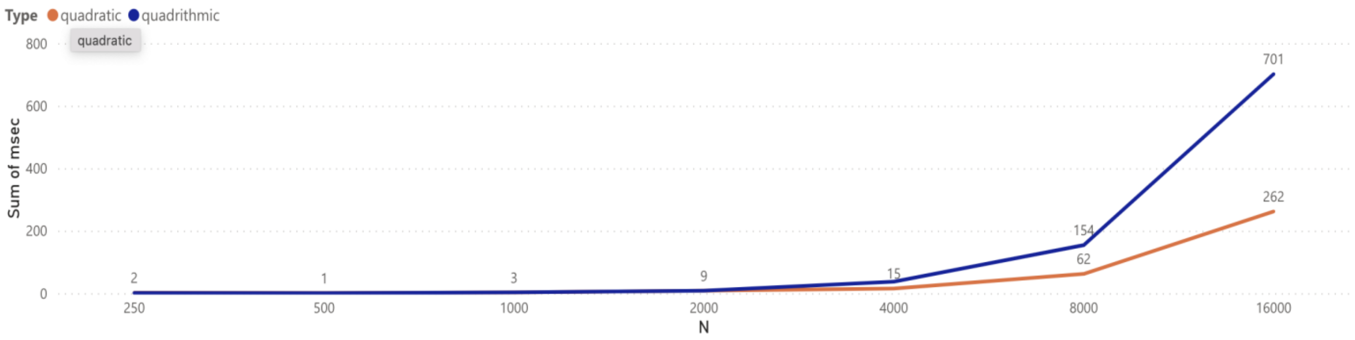
cubic	4000	1184.4	0.02
quadratic	8000	62	0.97
quadrithmic	8000	154	0.19
cubic	8000	9371	0.02
quadratic	16000	261.9	1.02
quadrithmic	16000	701.4	0.2
cubic	16000	74717.8	0.02

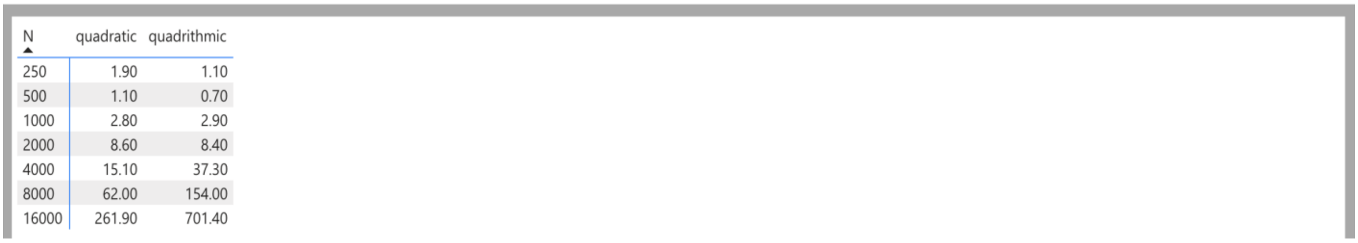
Graphical Representation:

Cubic vs Quadratic vs Quadrithmic raw times



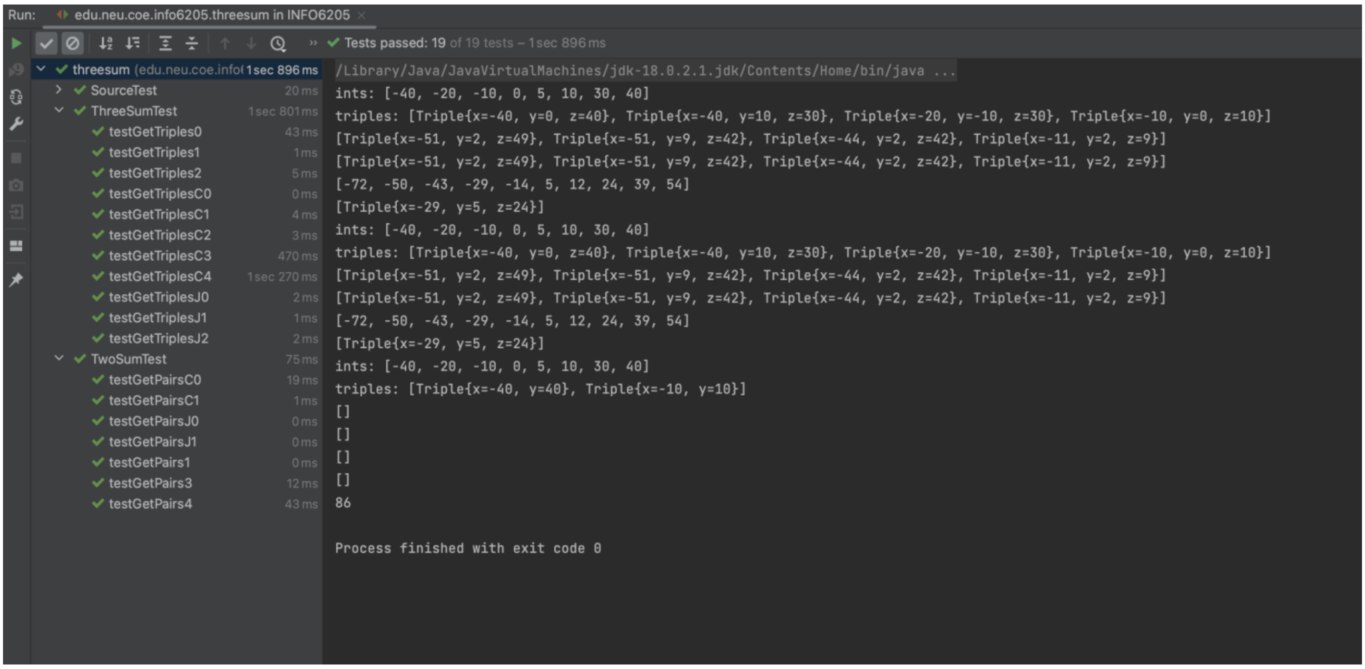
Quadratic vs Quadrithmic raw times





N ▲	quadratic	quadrithmic
250	1.90	1.10
500	1.10	0.70
1000	2.80	2.90
2000	8.60	8.40
4000	15.10	37.30
8000	62.00	154.00
16000	261.90	701.40

Unit Test Screenshots:



```
Tests passed: 19 of 19 tests – 1 sec 896 ms

threesum (edu.neu.coe.info6   1 sec 896 ms
  SourceTest                  20 ms
  ThreeSumTest                1 sec 801 ms
    testGetTriples0           43 ms
    testGetTriples1           1 ms
    testGetTriples2           5 ms
    testGetTriplesC0          0 ms
    testGetTriplesC1          4 ms
    testGetTriplesC2          3 ms
    testGetTriplesC3          470 ms
    testGetTriplesC4          1 sec 270 ms
    testGetTriplesJ0          2 ms
    testGetTriplesJ1          1 ms
    testGetTriplesJ2          2 ms
  TwoSumTest                  75 ms
    testGetPairsC0            19 ms
    testGetPairsC1            1 ms
    testGetPairsJ0            0 ms
    testGetPairsJ1            0 ms
    testGetPairs1             0 ms
    testGetPairs3             12 ms
    testGetPairs4             43 ms
```

```
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
ints: [-40, -20, -10, 0, 5, 10, 30, 40]
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}]
[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}]
[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}]
[-72, -50, -43, -29, -14, 5, 12, 24, 39, 54]
[Triple{x=-29, y=5, z=24}]
ints: [-40, -20, -10, 0, 5, 10, 30, 40]
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}]
[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}]
[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}]
[-72, -50, -43, -29, -14, 5, 12, 24, 39, 54]
[Triple{x=-29, y=5, z=24}]
ints: [-40, -20, -10, 0, 5, 10, 30, 40]
triples: [Triple{x=-40, y=40}, Triple{x=-10, y=10}]
[]
[]
[]
[]
86

Process finished with exit code 0
```