

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 $N < \sim 4000$ the time difference between quadratic and quadrithmic is insignificant but the difference between cubic and the other two are significant and evidential.

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:

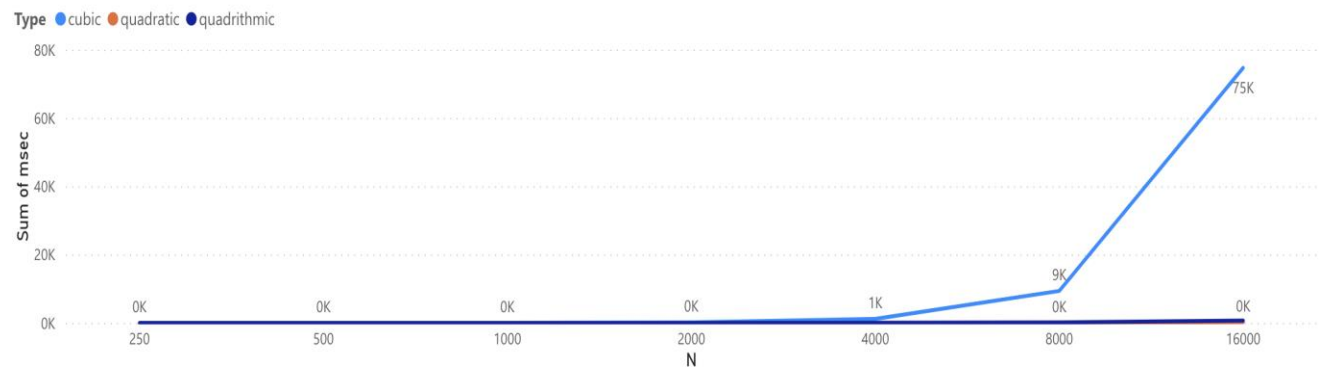
Below are the raw computational times in milliseconds.

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

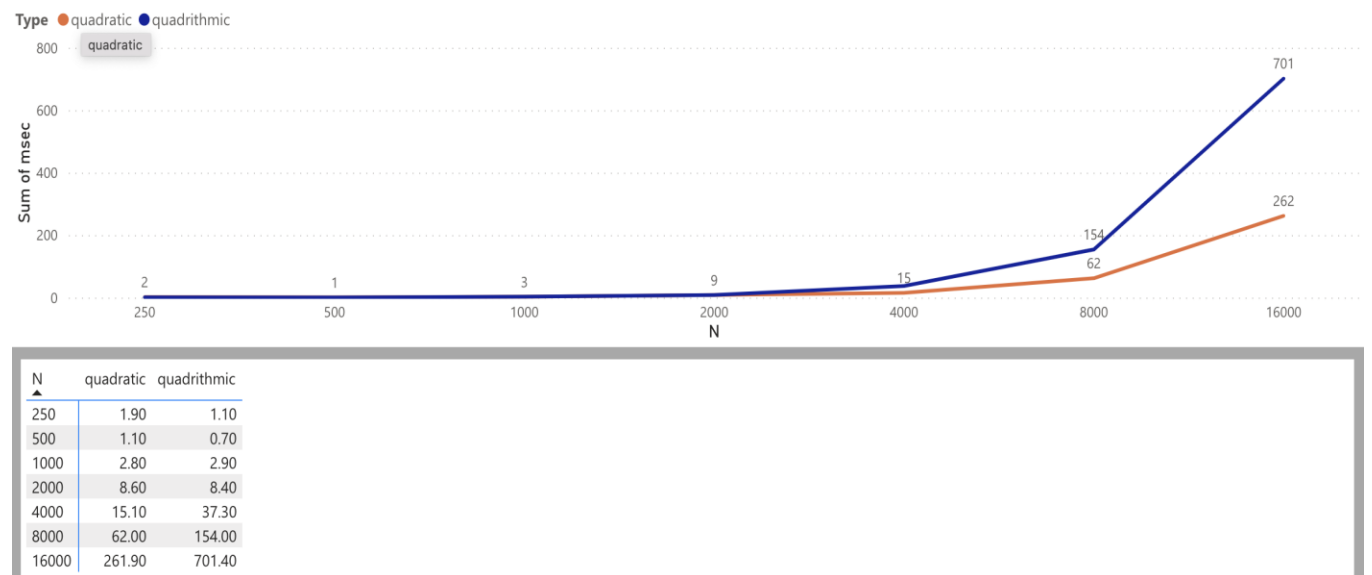
cubic	4000	1184.4	0.02
quadratic	8000	62	0.97
quadrithmic	8000	154	0.19
cubic	8000	9371	0.02
quadratic	16000	261.9	1.02
quadrithmic	16000	701.4	0.2
cubic	16000	74717.8	0.02

Graphical Representation:

Cubic vs Quadratic vs Quadrithmic raw times



Quadratic vs Quadrithmic raw times



Unit Test Screenshots:

The screenshot displays the test results for a Java project named 'threesum' in the 'edu.neu.coe.info6205.threesum' package. The tests are organized into a tree view on the left, and the output of the tests is shown on the right. The tests are categorized into 'SourceTest', 'ThreeSumTest', and 'TwoSumTest'. The 'ThreeSumTest' category contains several tests, including 'testGetTriples0', 'testGetTriples1', 'testGetTriples2', 'testGetTriplesC0', 'testGetTriplesC1', 'testGetTriplesC2', 'testGetTriplesC3', 'testGetTriplesC4', 'testGetTriplesJ0', 'testGetTriplesJ1', and 'testGetTriplesJ2'. The 'TwoSumTest' category contains 'testGetPairsC0', 'testGetPairsC1', 'testGetPairsJ0', 'testGetPairsJ1', 'testGetPairs1', 'testGetPairs3', and 'testGetPairs4'. The output on the right shows the results of these tests, including the number of tests passed (19 of 19), the total time taken (1sec 896ms), and the specific output of each test. The output includes arrays of integers and Triples, as well as the final exit code (0).

```
Run: edu.neu.coe.info6205.threesum in INFO6205 x
Tests passed: 19 of 19 tests - 1sec 896 ms

SourceTest (edu.neu.coe.info6205.threesum) 1sec 896 ms
  SourceTest 20 ms
    ints: [-40, -20, -10, 0, 5, 10, 30, 40]
  ThreeSumTest 1sec 801 ms
    testGetTriples0 43 ms
      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}]
    testGetTriples1 1 ms
      [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}]
    testGetTriples2 5 ms
      [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 0 ms
      [-72, -50, -43, -29, -14, 5, 12, 24, 39, 54]
    testGetTriplesC1 4 ms
      [Triple{x=-29, y=5, z=24}]
    testGetTriplesC2 3 ms
      ints: [-40, -20, -10, 0, 5, 10, 30, 40]
    testGetTriplesC3 470 ms
      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}]
    testGetTriplesC4 1sec 270 ms
      [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}]
    testGetTriplesJ0 2 ms
      [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 1 ms
      [-72, -50, -43, -29, -14, 5, 12, 24, 39, 54]
    testGetTriplesJ2 2 ms
      [Triple{x=-29, y=5, z=24}]
  TwoSumTest 75 ms
    testGetPairsC0 19 ms
      ints: [-40, -20, -10, 0, 5, 10, 30, 40]
    testGetPairsC1 1 ms
      triples: [Triple{x=-40, y=40}, Triple{x=-10, y=10}]
    testGetPairsJ0 0 ms
      []
    testGetPairsJ1 0 ms
      []
    testGetPairs1 0 ms
      []
    testGetPairs3 12 ms
      []
    testGetPairs4 43 ms
      86

Process finished with exit code 0
```