Program Structures and Algorithms
Spring 2023(SEC – 3)

NAME:KEERTHANA SATHEESH
NUID: 002747795

Task:

(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.

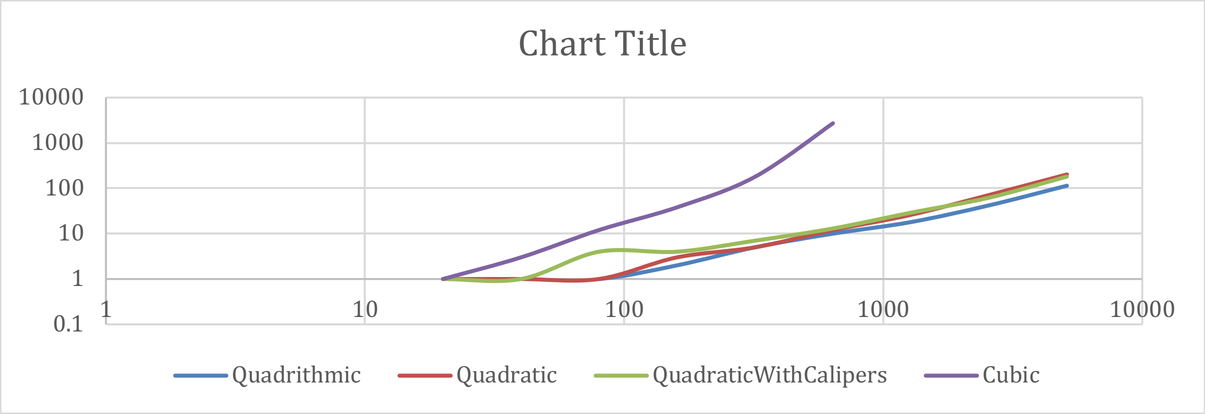
Relation Conclusion:

- Cubic is a O(N^3)
- QuadraticWithCalipers is O(N^2)
- Quadratic is O(N^2)
- Quadrithmic is O(N^2 log N)

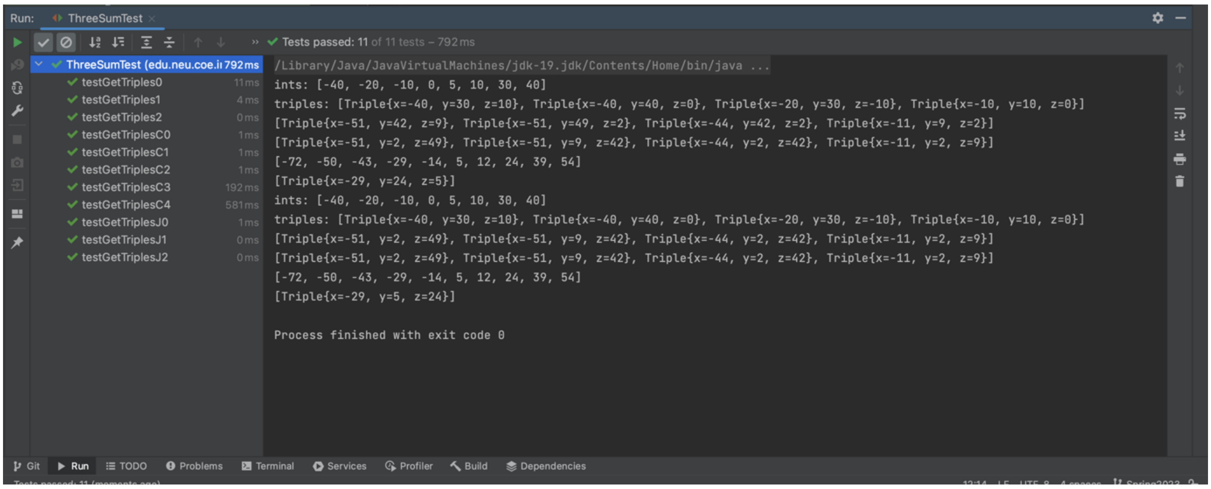
Time taken to run of each N

N	Quadrithmic	Quadratic	QuadraticWithCalipers	Cubic
20	0	1	1	1
40	1	1	1	3
80	1	1	4	12
160	2	3	4	38
320	5	5	7	179
640	10	12	13	2714
1280	18	26	29	
2560	42	71	62	
5120	114	202	181	

Graphical Representation:
LOG T vs LOG N



Unit Test Screenshots:



✓ Tests passed: 11 of 11 tests – 792 ms

```
✓ ThreeSumTest (edu.neu.coe.i 792 ms
    ✓ testGetTriples0      11 ms
    ✓ testGetTriples1       4 ms
    ✓ testGetTriples2       0 ms
    ✓ testGetTriplesC0      1 ms
    ✓ testGetTriplesC1      1 ms
    ✓ testGetTriplesC2      1 ms
    ✓ testGetTriplesC3    192 ms
    ✓ testGetTriplesC4    581 ms
    ✓ testGetTriplesJ0      1 ms
    ✓ testGetTriplesJ1      0 ms
    ✓ testGetTriplesJ2      0 ms
```

```
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home/bin/java ...
ints: [-40, -20, -10, 0, 5, 10, 30, 40]
triples: [Triple{x=-40, y=30, z=10}, Triple{x=-40, y=40, z=0}, Triple{x=-20, y=30, z=-10}, Triple{x=-10, y=10, z=0}]
[Triple{x=-51, y=42, z=9}, Triple{x=-51, y=49, z=2}, Triple{x=-44, y=42, z=2}, Triple{x=-11, y=9, z=2}]
[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=24, z=5}]
ints: [-40, -20, -10, 0, 5, 10, 30, 40]
triples: [Triple{x=-40, y=30, z=10}, Triple{x=-40, y=40, z=0}, Triple{x=-20, y=30, z=-10}, Triple{x=-10, y=10, z=0}]
[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}]

Process finished with exit code 0
```

Quadratic method : -

There are two pointers :-

1) left => middle index -1
2) right => middle index + 1

The sum of left, right, and middle index is compared to target value.
If sum > target, right pointer moves right.
If sum < target, left pointer moves left.
If sum = 0, left--,right++
This process is then carried in a loop for different left and right indexes.
The complexity is O(N^2).

Quadratic method with Calipers : -

We start off with two index's : -

1) left => from the given index i +1
2) right => last index position of array -1

The sum of left, right, and given index is calculated.
If sum > target, left pointer is moved right
If sum < target, right pointer moves left.
If sum = 0, left++, right --
This process is then carried in a loop for different left and right indexes.
The complexity is O(N^2).