**Program Structure and Algorithm**

**Spring 2023(SEC-1)**

**Name: Swarag Sanjay Gutte**

**NUID: 002728422**

**TASK:**

Solve 3-SUM using the *Quadrithmic*, *Quadratic*, and (bonus point) *quadraticWithCalipers* approaches, as shown in skeleton code in the repository

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 and Conclusion :**

ThreeSumCubic

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.

This algorithm runs in O(N^3) time.

ThreeSumQuadratic

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).

ThreeSumQuadrithmic

Implementation of ThreeSum which follows the simple optimization of

requiring a sorted array, then using binary search to find an element x where

-x the sum of a pair of elements.

The array provided in the constructor MUST be ordered.

*This algorithm runs in O(N^2 log N) time.*

ThreeSumQuadraticWithCalipers

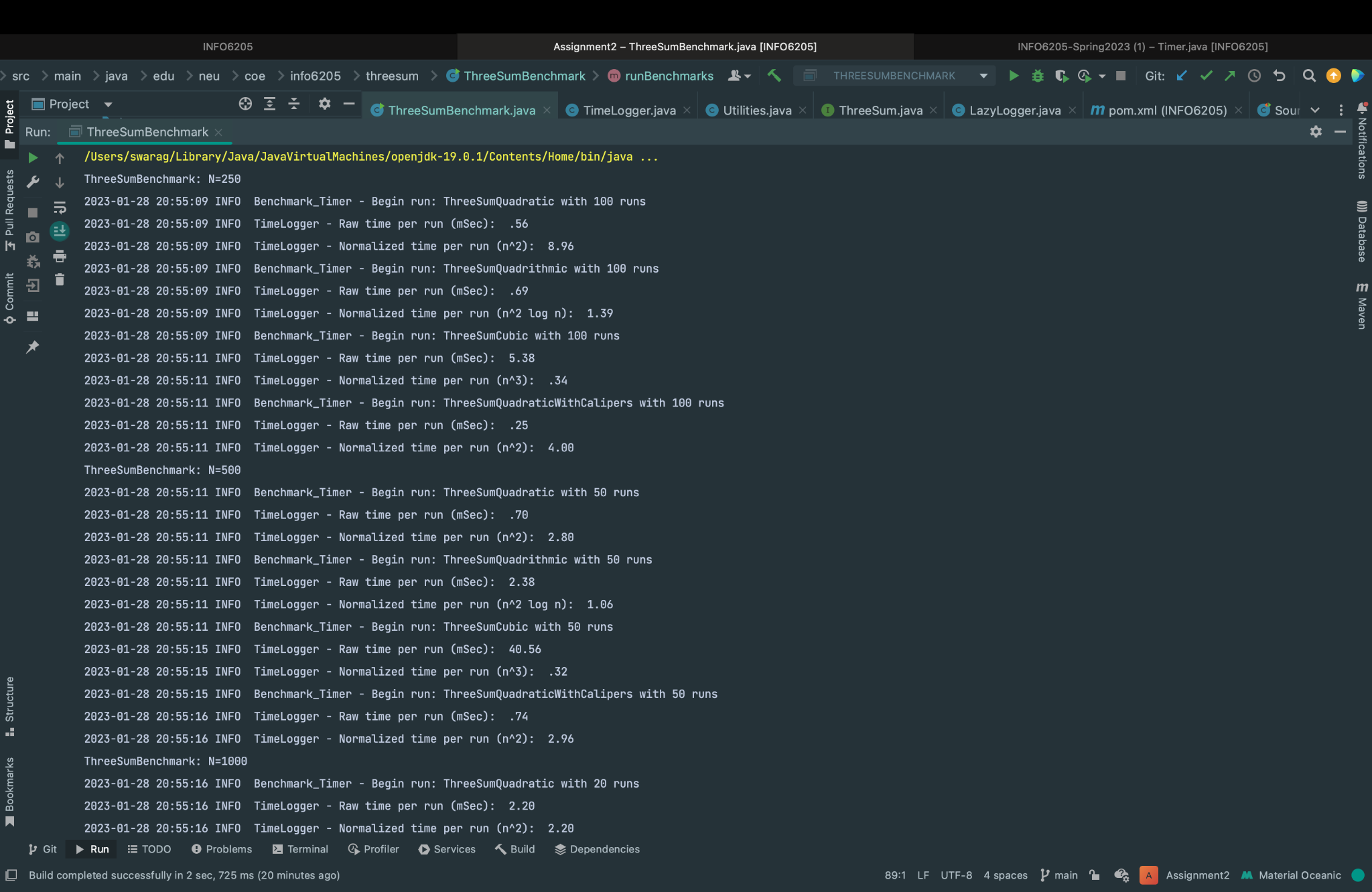
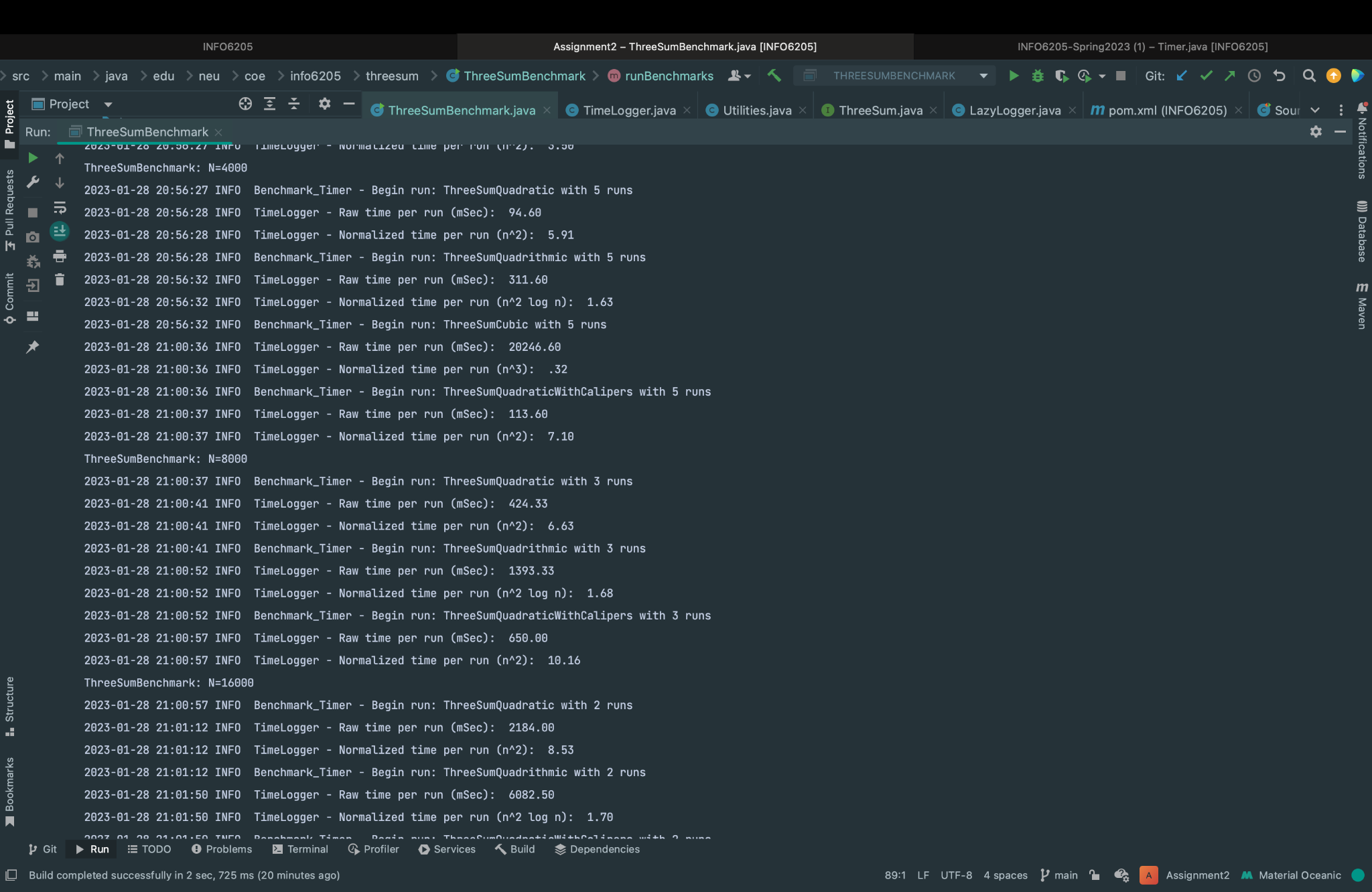
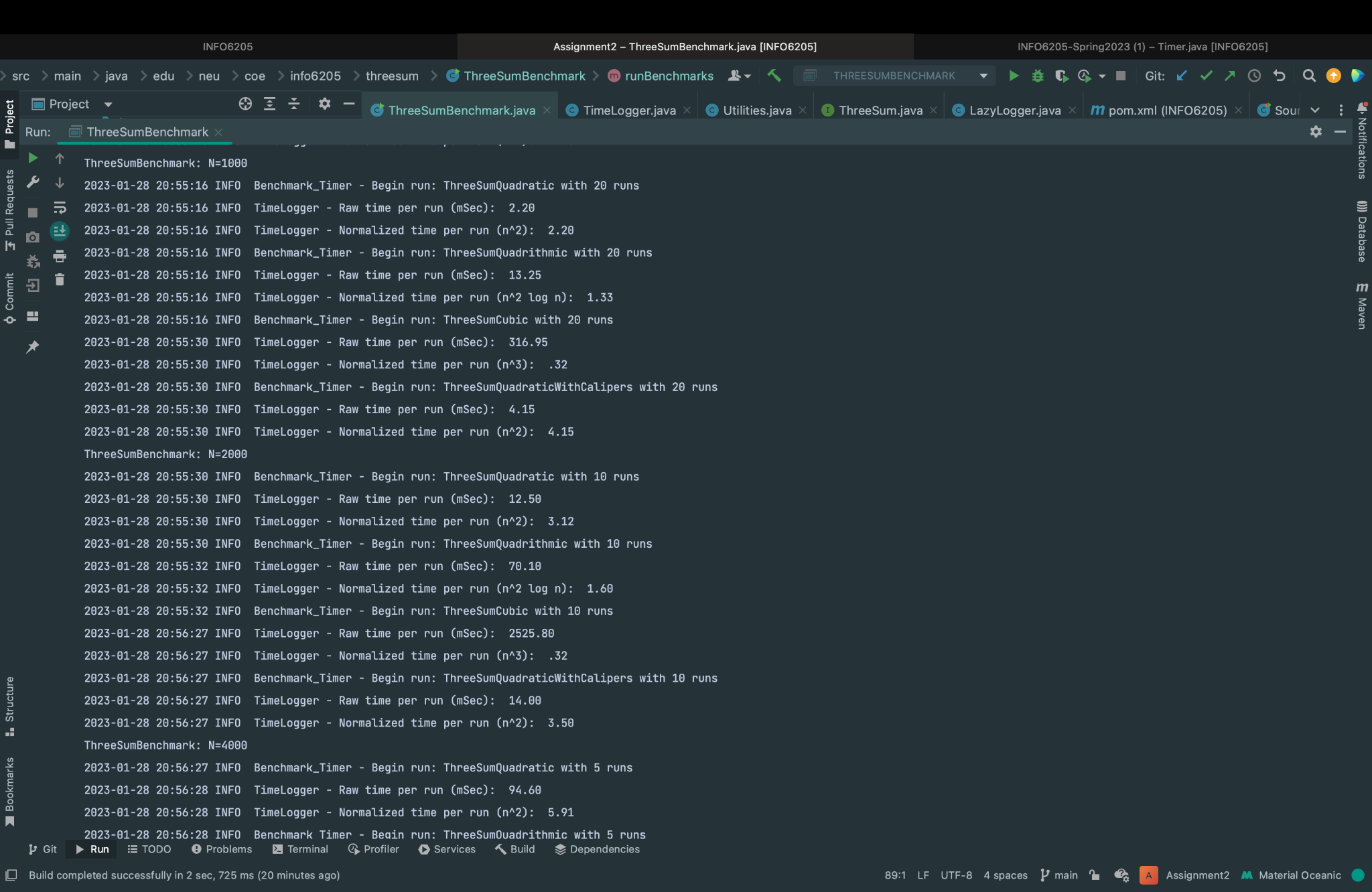
Implementation of ThreeSum which follows the approach of dividing the solution-space into

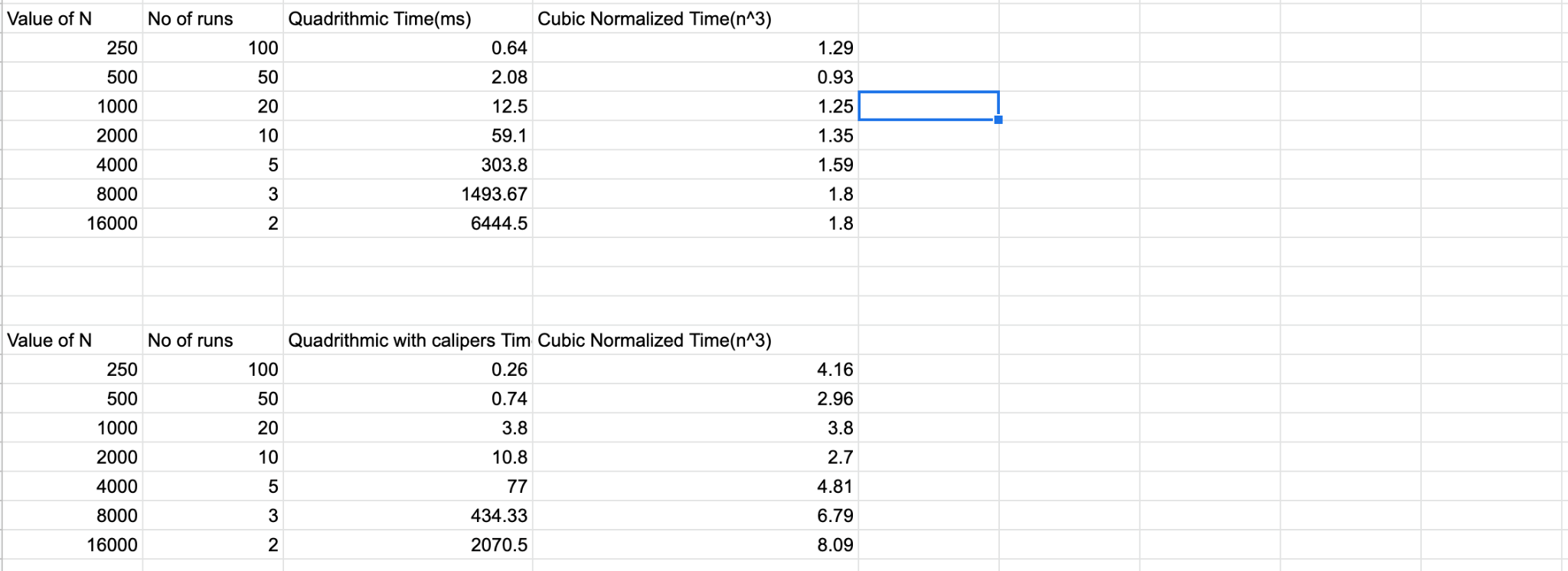
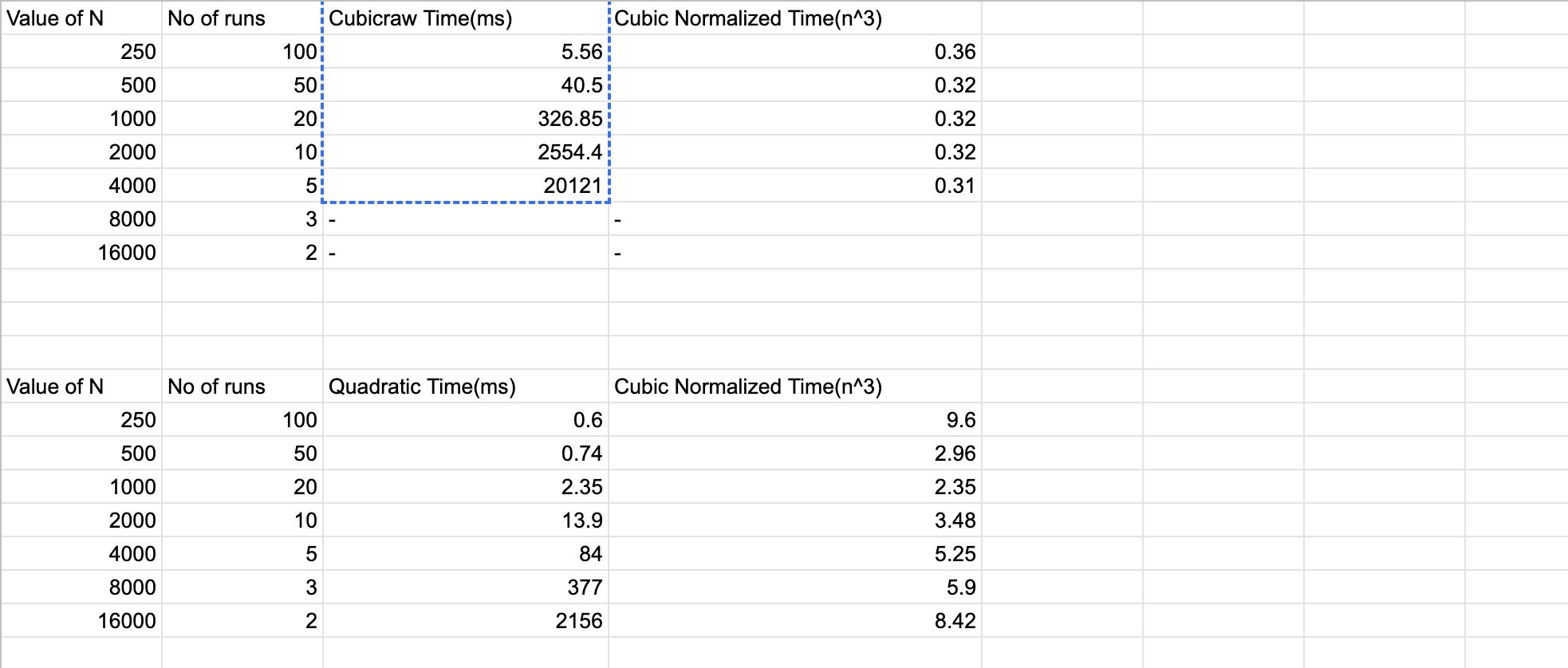
N sub-spaces where each sub-space corresponds to a fixed value for the middle index of the three values.

Each sub-space is then solved by expanding the scope of the other two indices outwards from the starting point.

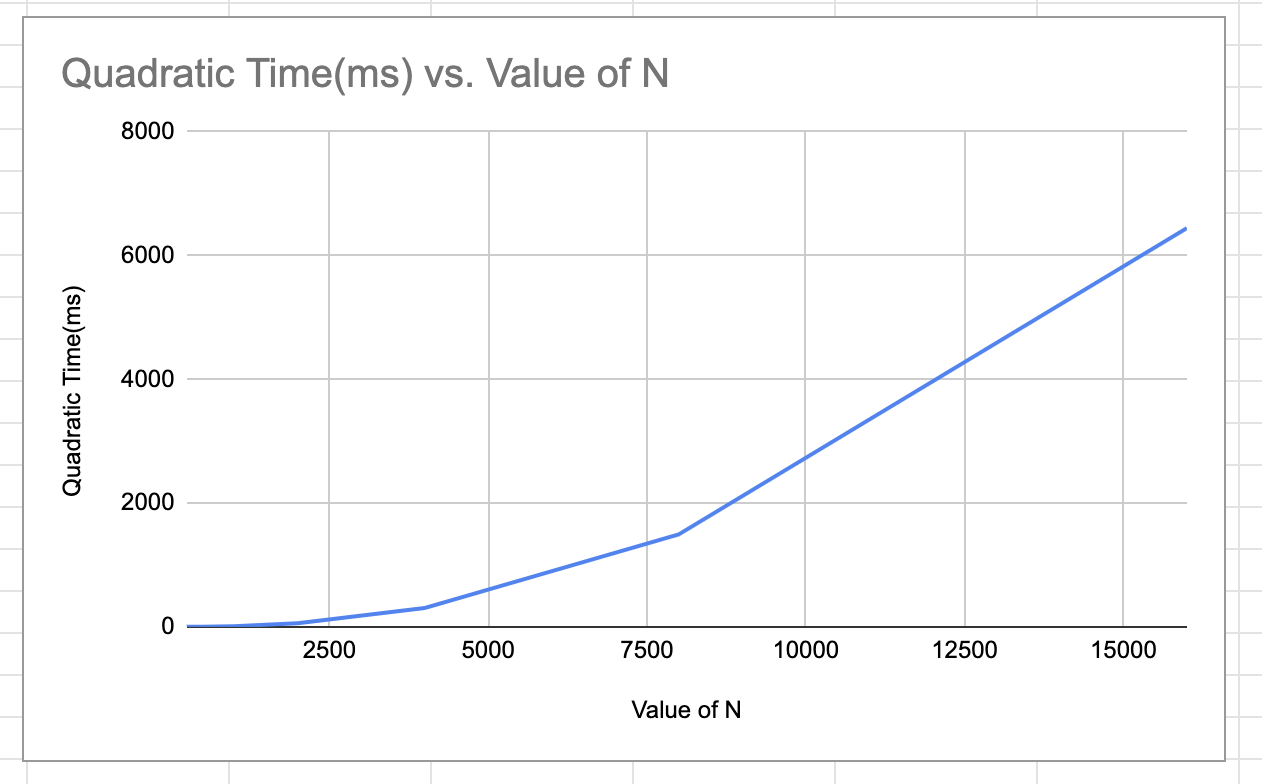
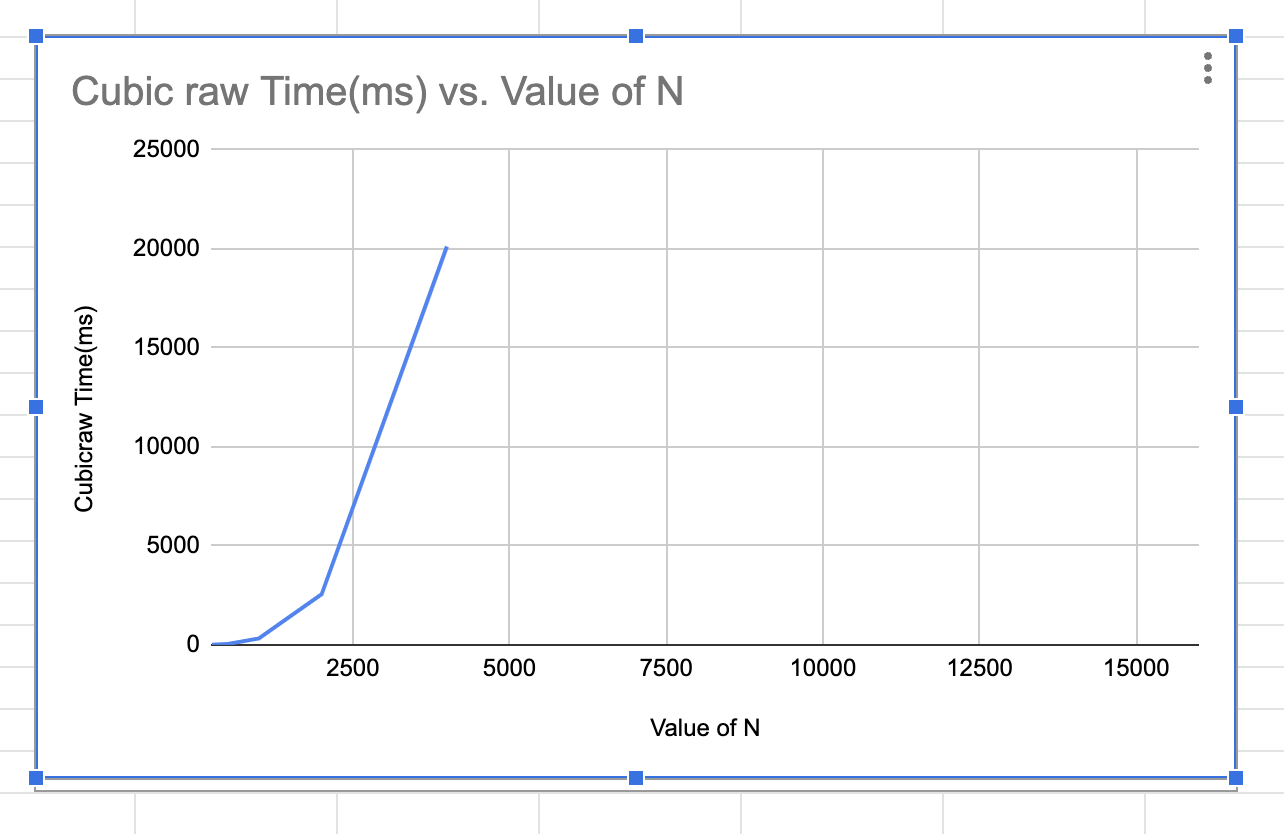
Since each sub-space can be solved in O(N) time, the overall complexity is O(N^2).

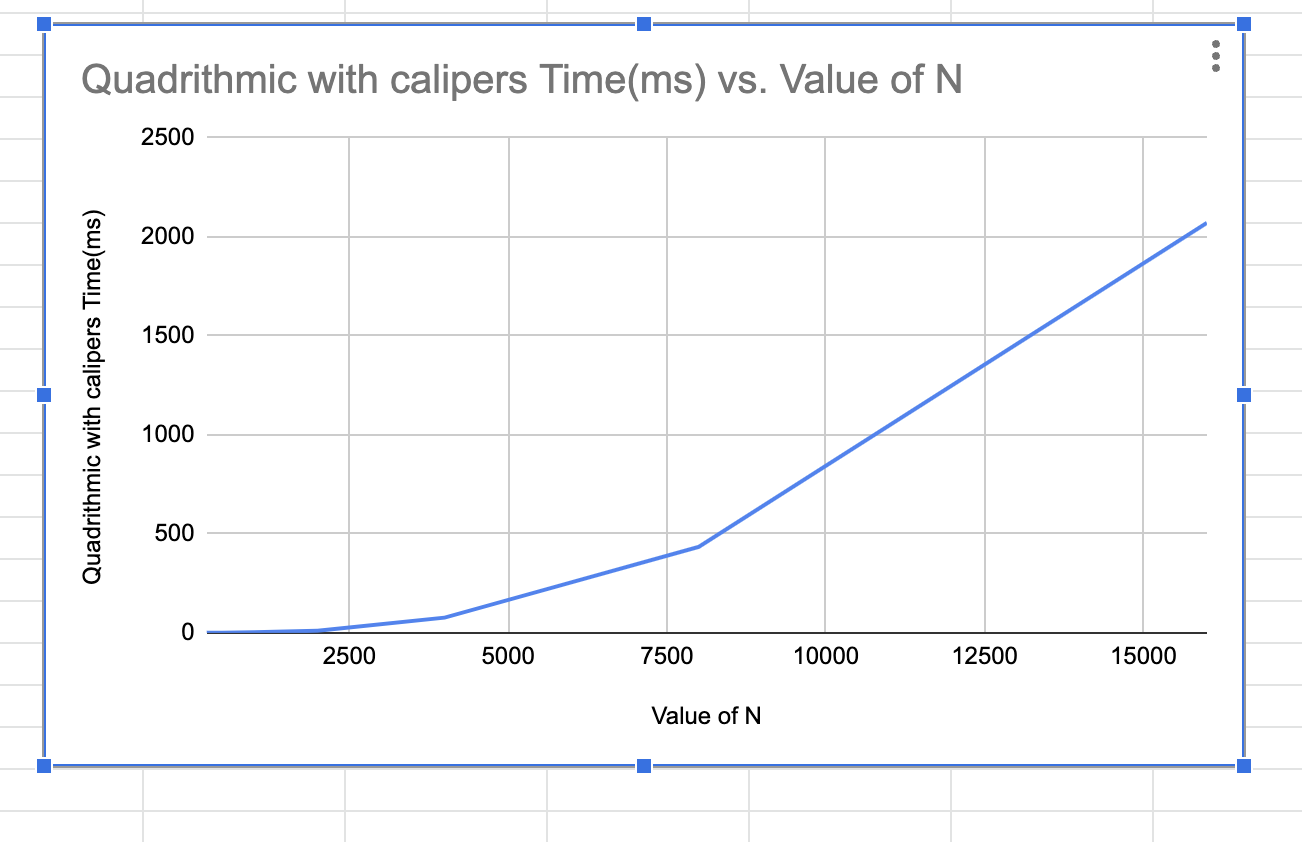
The array provided in the constructor MUST be ordered.

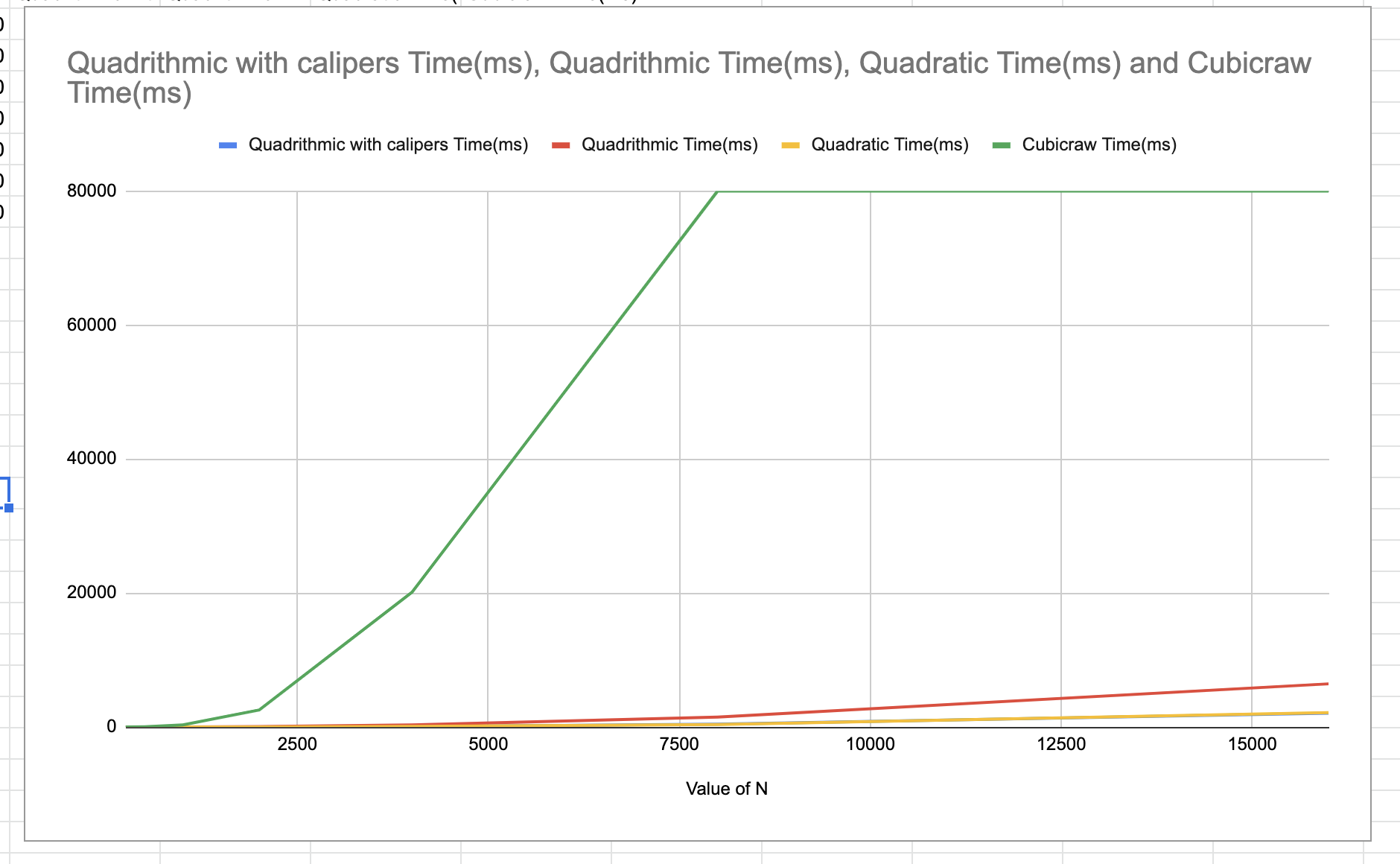
**OUTPUT:** 

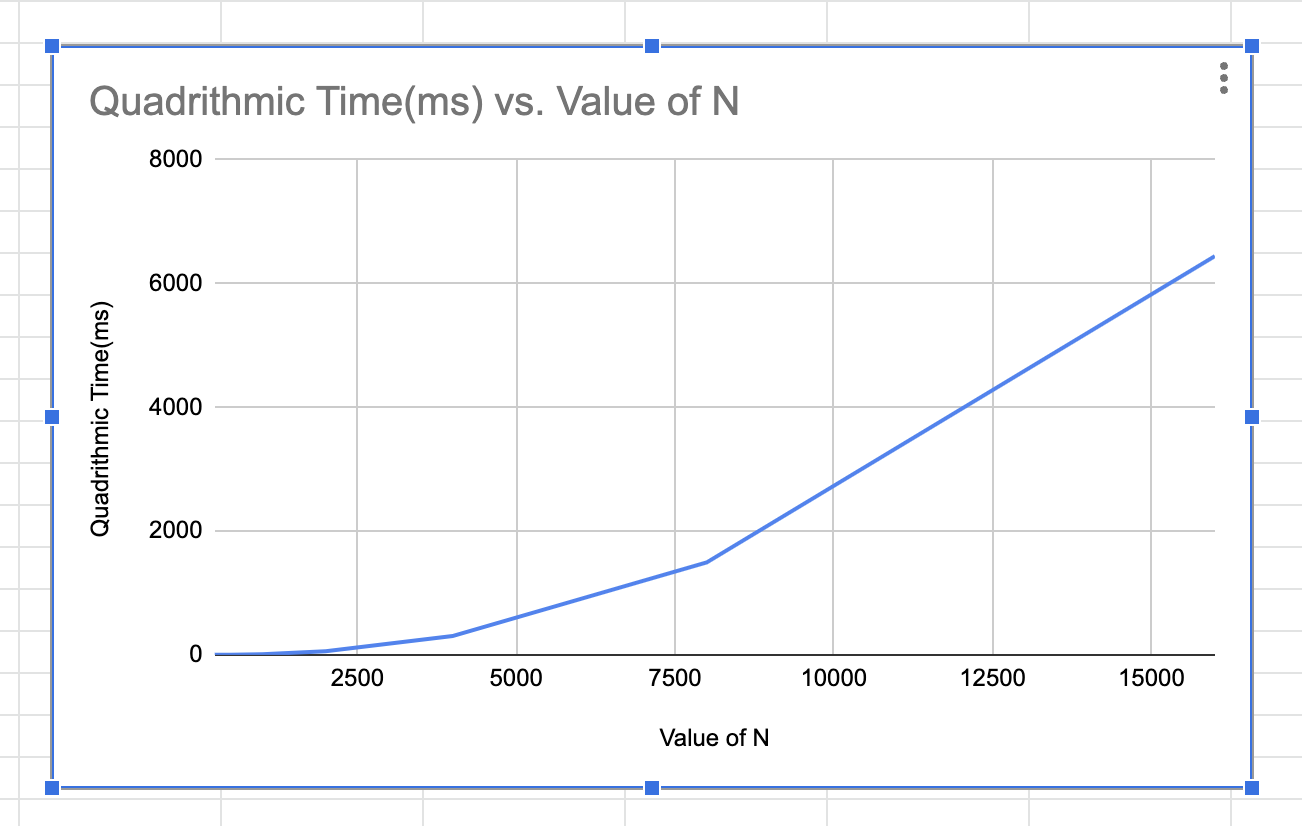
**Evidence to support the conclusion**:  


**Graphical Representatio**n:









**Result:**

