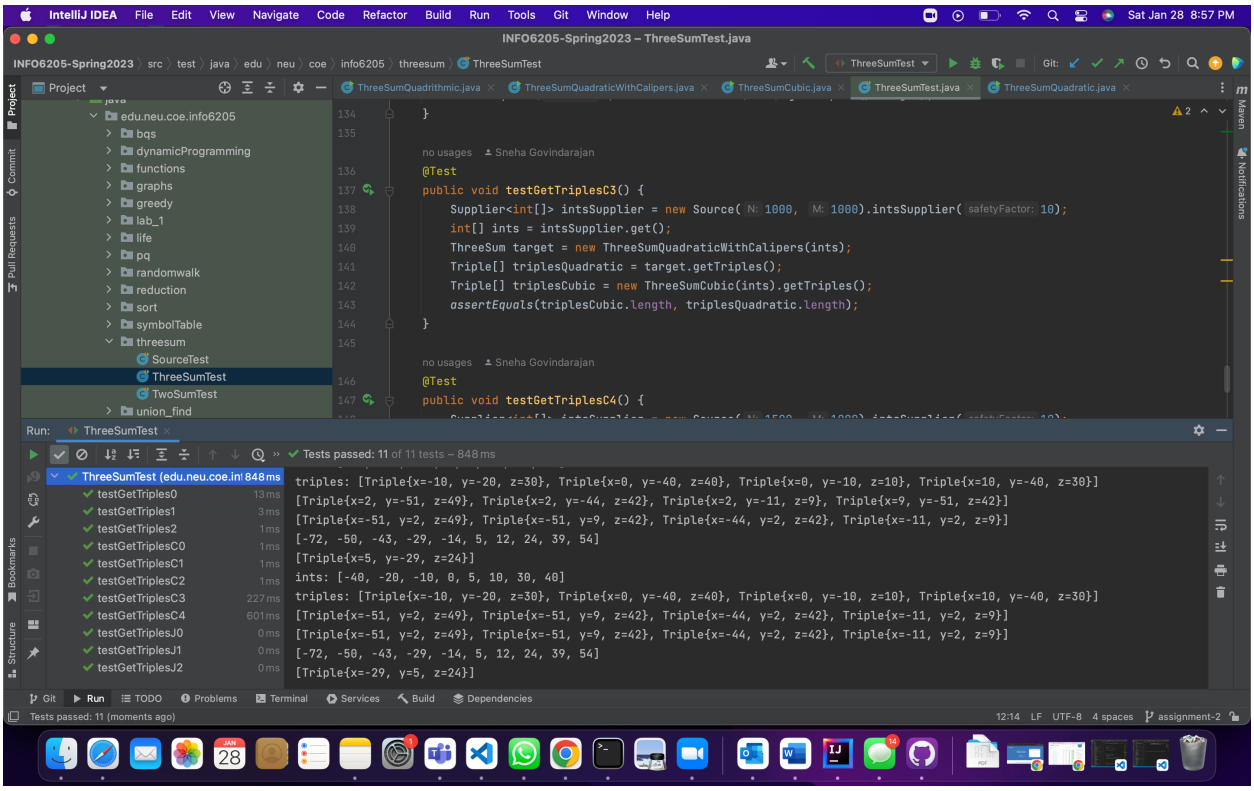
Assignment - 2

I. Screenshot Of Unit test cases



II. Spreadsheet showing your timing observations – using Stopwatch class

ThreeSum Observation

Quaratic - ThreeSum

Size	Triplets	Elapsed Time (ms)
70	77	2ms
140	271	1ms
280	1008	1ms
560	4043	4ms
1120	16595	21ms
2240	65500	43ms

QuadraticWithCaliper - ThreeSum

Size	Triplets	lapsed Time (ms)
70	77	1ms
140	271	1ms
280	1008	1ms
560	4043	6ms
1120	16595	41ms
2240	65500	61ms

Quadrithmic - ThreeSum

Size	Triplets	Elapsed Time (ms)
70	77	1ms
140	271	1ms
280	1008	1ms
560	4043	5ms
1120	16595	27ms
2240	65500	47ms

Cubic - ThreeSum

Size	Triplets	lapsed Time (ms)
70	77	1ms
140	271	1ms
280	1008	1ms
560	4043	6ms
1120	16595	49ms
2240	65500	130ms

Assignment - 2

III. Explanation of why the quadratic method(s) work

The quadratic method for solving the three sum problem is better than cubic and quardrithmic methods because it has a lower time complexity. Specifically, the quadratic method has a time complexity of O(n^2), which means that the time it takes to solve the problem increases at a rate of n^2 as the size of the input array increases. On the other hand, a cubic method would have a time complexity of O(n^3) and a quadrithmic method would have a time complexity of O(n^2 log n).

As the input size increase, the time complexity of cubic method increases at a rate of n^3 and the time complexity of quadrithmic method increases at a rate of n^2 log n which are significantly greater than O(n^2). In practice, this means that for large input sizes, the quadratic method will be much faster than cubic or quadrithmic methods. This is the reason why quadratic method is preferred over other higher complexity method for solving three sum problem.

An example of the "three sum" problem would be finding all sets of three numbers in an array that add up to a specific target sum.
For example, given an array of integers [-1, 0, 1, 2, -1, -4], and a target sum of 0, the possible three sums would be: [-1, 0, 1], [-1, -1, 2]

This can be done with a O(n^2) algorithm, by first sorting the array, then iterating through the array and for each element, using the two pointer technique to find the two other numbers that would sum to the target.