Solve 3-SUM using the *Quadrithmic*, *Quadratic*, and (bonus point) *quadraticWithCalipers* approaches, as shown in skeleton code in the repository. There are hints at the end of Lesson 2.5 Entropy.

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.

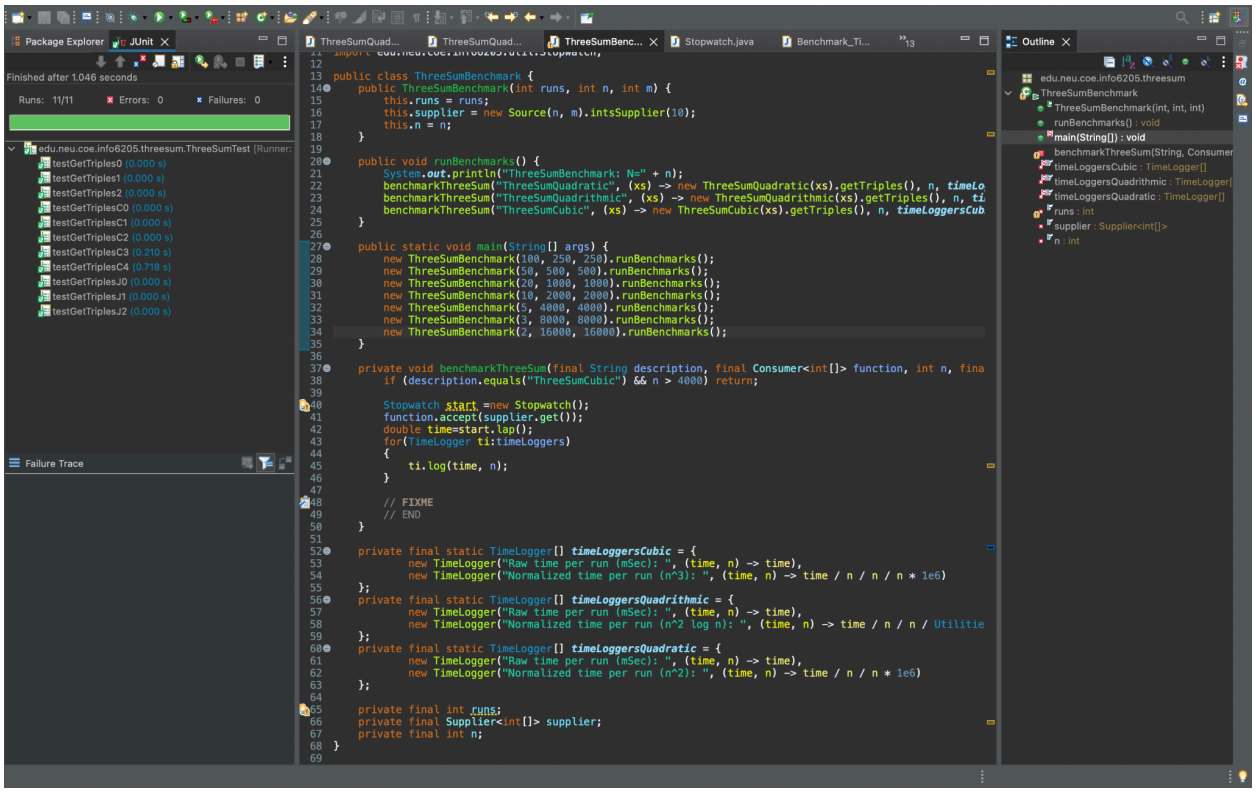
Submit (in your own repository--see instructions elsewhere--include the source code and the unit tests of course):

(a) evidence (screenshot) of your unit tests running (try to show the actual unit test code as well as the green strip);

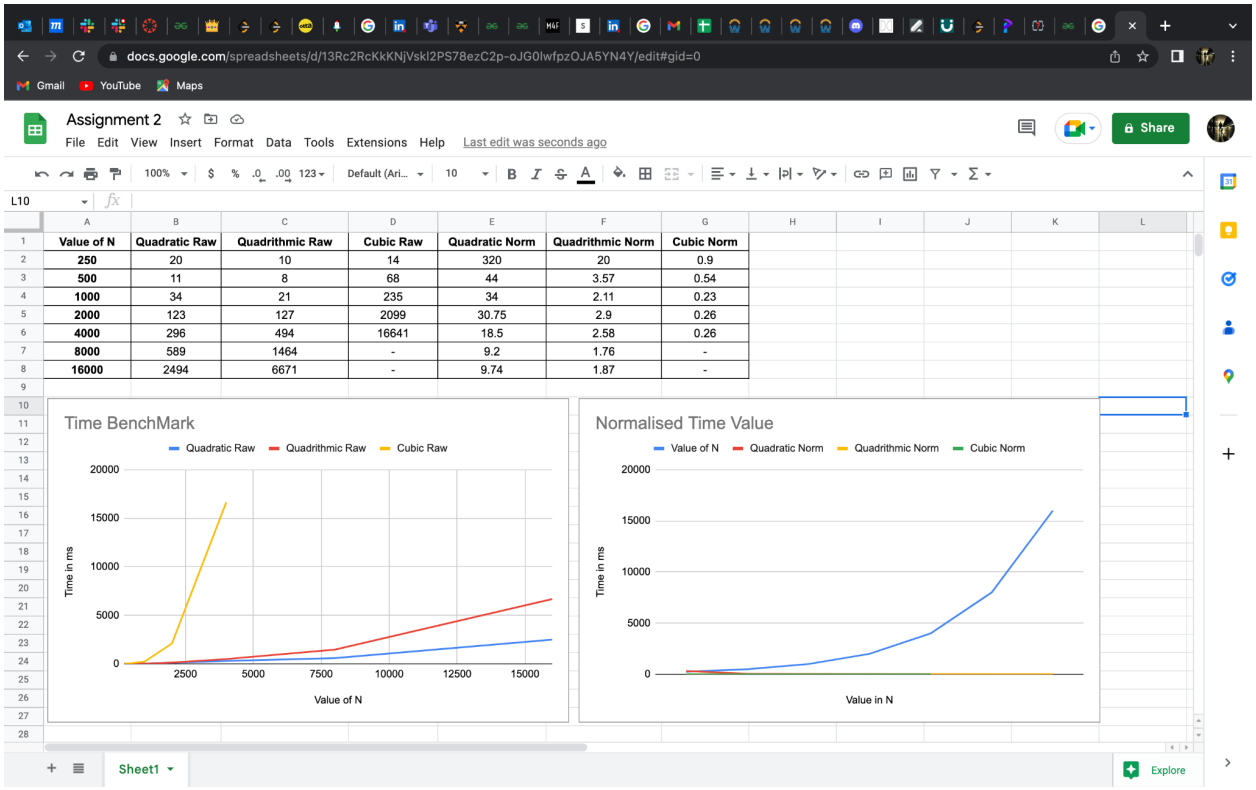
(b) a spreadsheet showing your timing observations--using the doubling method for at least five values of N--for each of the algorithms (include cubic); Timing should be performed either with an actual stopwatch (e.g. your iPhone) or using the Stopwatch class in the repository.

(c) your brief explanation of why the quadratic method(s) work.

Answer
a



b.



Value of N	Quadratic Raw	Quadrithmic Raw	Cubic Raw	Quadratic Norm	Quadrithmic Norm	Cubic Norm
250	20	10	14	320	20	0.9
500	11	8	68	44	3.57	0.54
1000	34	21	235	34	2.11	0.23
2000	123	127	2099	30.75	2.9	0.26
4000	296	494	16641	18.5	2.58	0.26
8000	589	1464	-	9.2	1.76	-
16000	2494	6671	-	9.74	1.87	-

C.

Quadratic Method

The quadratic Method will uses one for loop to iterate the the array value, Before starting the arrays value are initially sorted to get O(n logn) as the time complexity and later on the values are iterated thought and later on using the two pointer approach will find the value and calculate the sum of all the values and if the target value equals zero then it is added to list
The first for loop runs for O(n) loops and the two pointer uses O(n) and therefore it multiple $O(n^2)$. From the sheets we can determine that the taken for the execution for quadratic takes less time than the others.

Quadratic Caliper

The quadratic caliper Method will uses three variables and the arrays initially sorted which takes O(nlogn) time and therefore after that one variables is fixed, using the other two variables the values are iterated around that fixed variable and the one variable will have i+1 values and the other value starts from last value. Using the two pointer approach the values are found out.

For fixing the value and iterating thought the loop it take O(n) and for the two pointer approach uses O (n) and therefore for each fixed values of numbers inside the array there it searches for

two values which equal to the target 0. Therefore for all over all time complexity for quadratic caliper the time complexity is O (n^2).