Assignment 2 - Program Structures and Algorithms Spring 2023(SEC - 1)

NAME: Paurush Batish **NUID: 002755631**

Task:

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

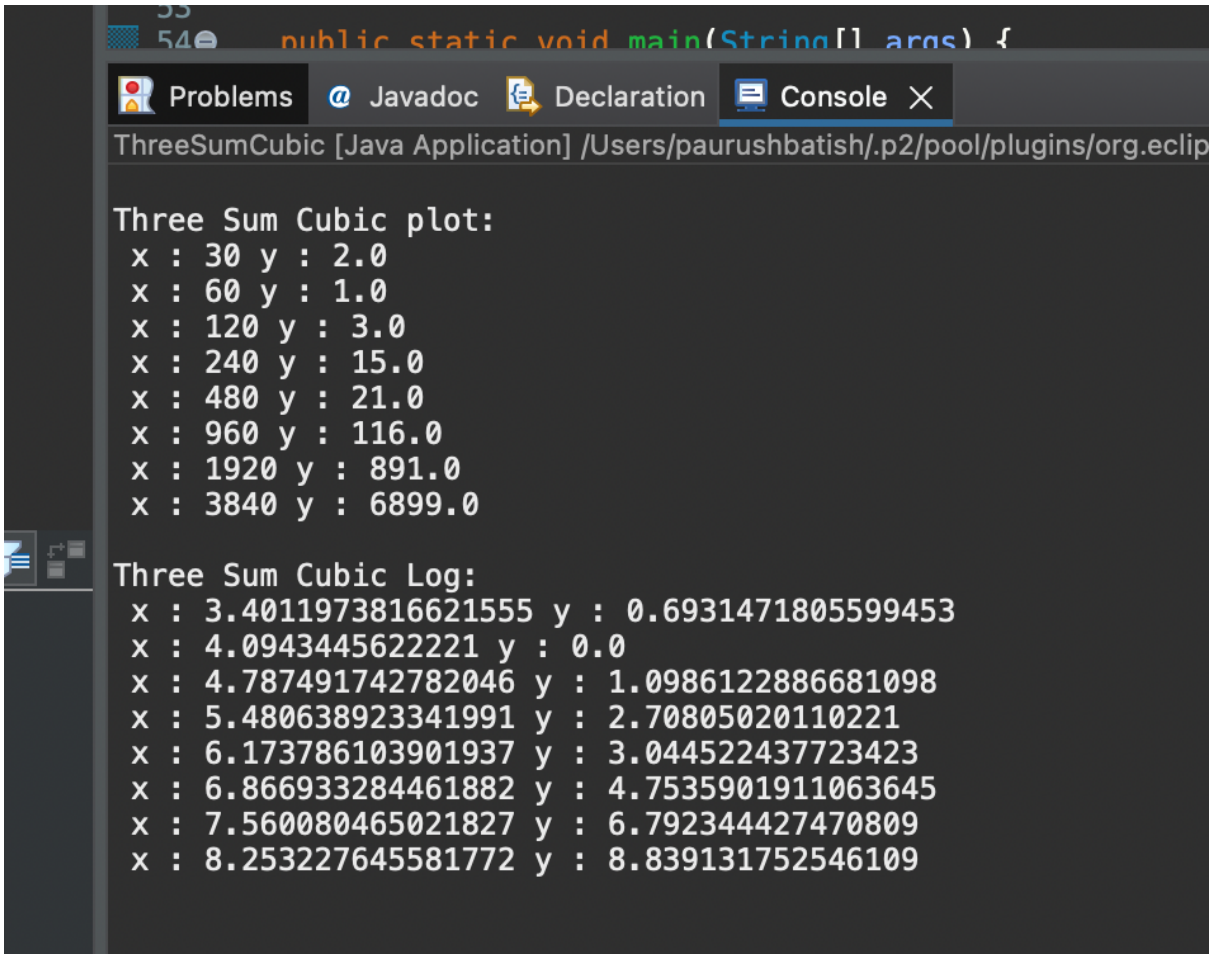
Three Sum Cubic

Relationship conclusion:

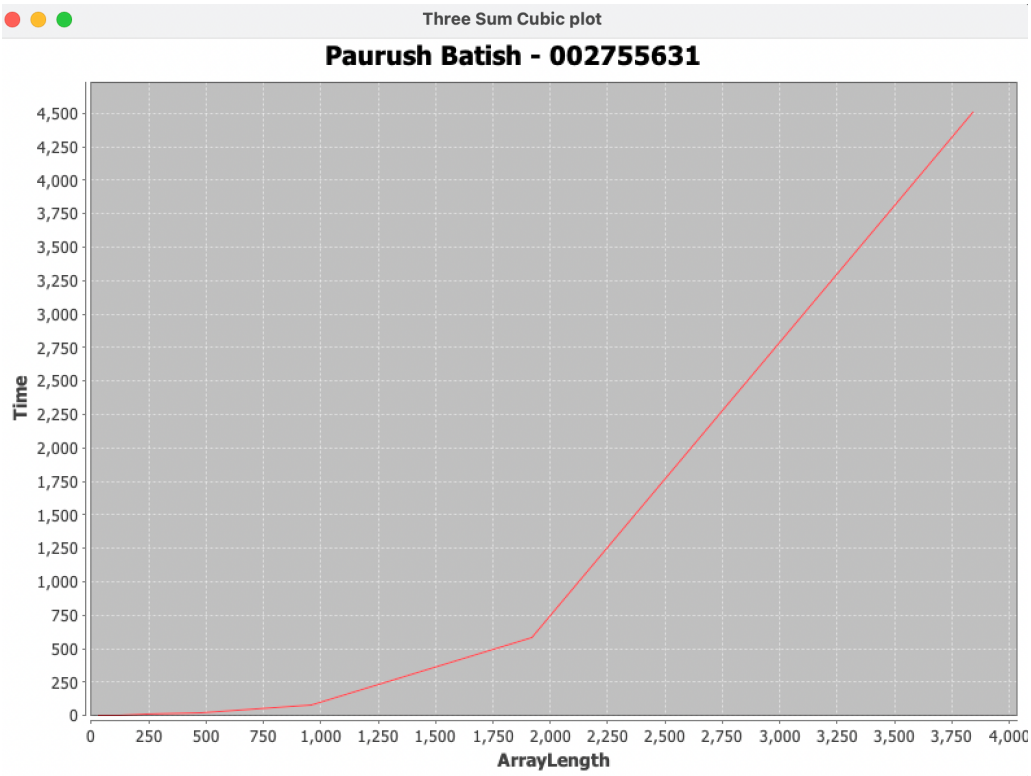
For 3 sum cubic,
The order of growth for is N^3.

Evidence :

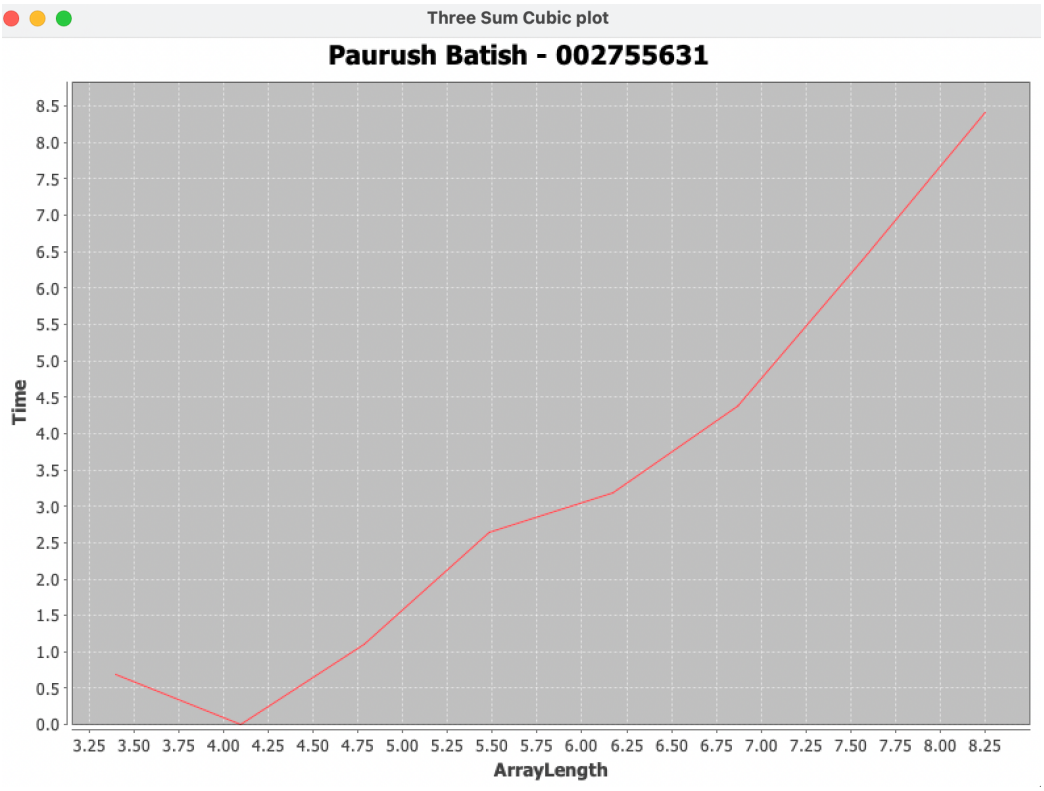
Values:



Graph of Length of array vs time:



ln(array length) vs ln(time):



Slope Calculation:

$$m = \frac{y2 - y1}{x2 - x1}$$

M= (8.413608875159667 - 6.363028103540465)/(8.253227645581772 - 7.560080465021827)

M = 2.95836271016
M ~ 3

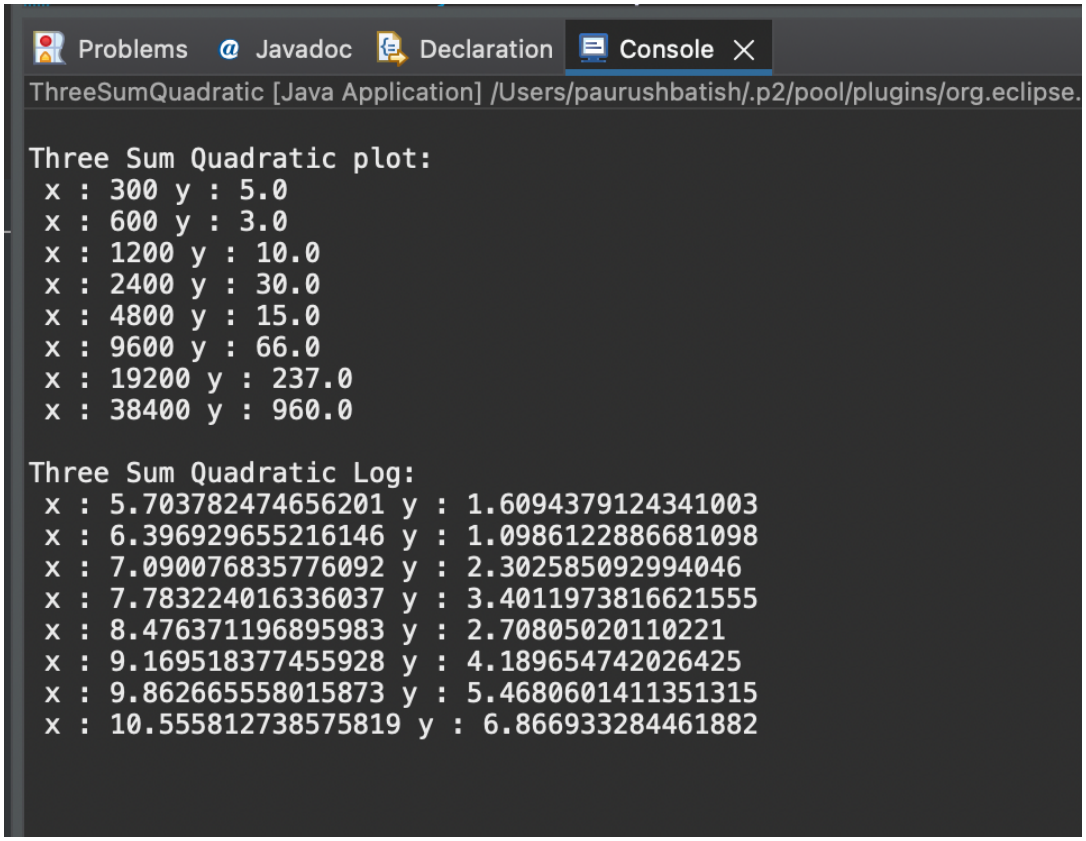
Hence we get the equation of the graph as

$$y = ax^3$$

Hence the order of growth of the code is N^3

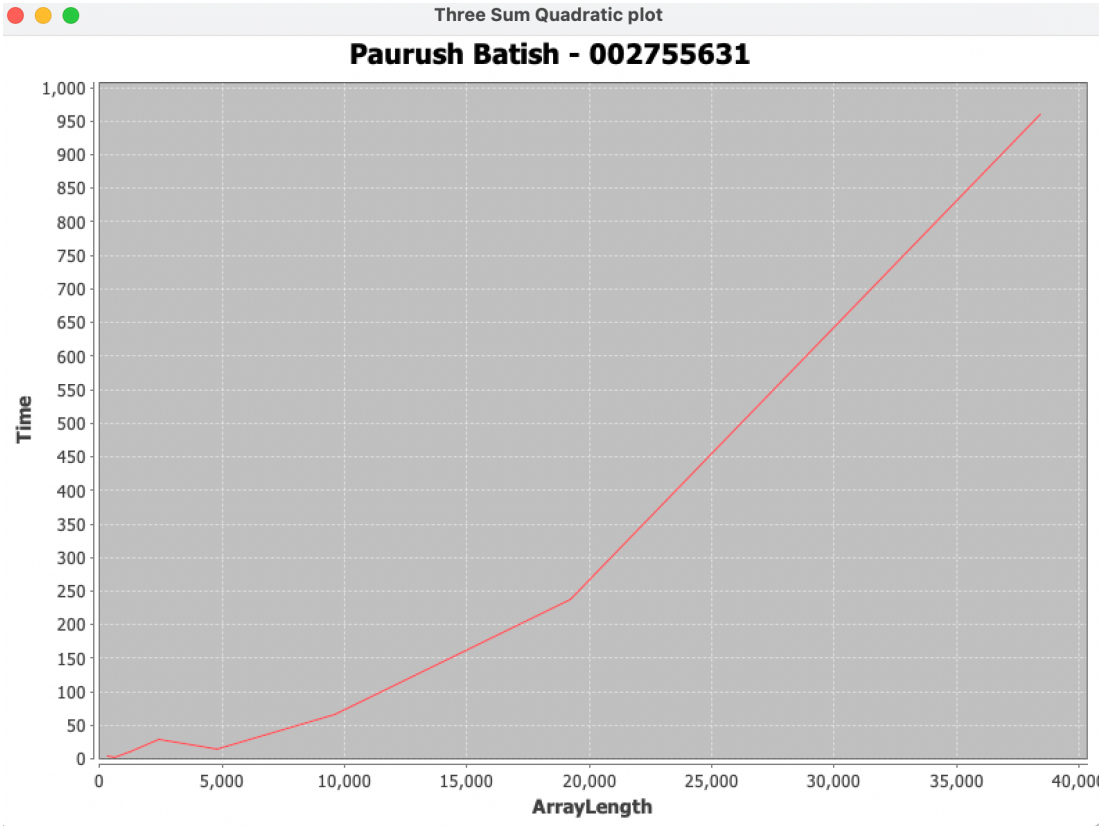
Three Sum Quadratic

Values:

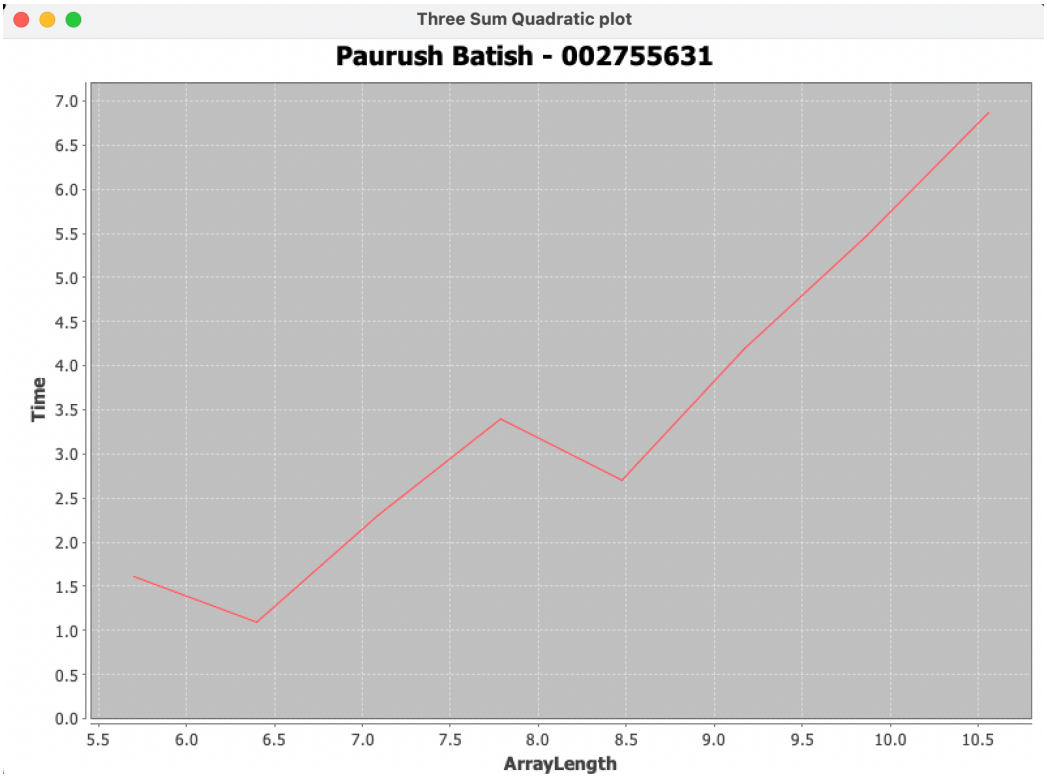


Evidence :

Plotting the graph of array length vs time:



Plotting the graph of ln(array length) vs ln(time):



Slope Calculation:

$$m = \frac{y2 - y1}{x2 - x1}$$

M= (6.866933284461882- 5.4680601411351315)/(10.555812738575819- 9.862665558015873)

M = 2.01814734671
M ~ 2

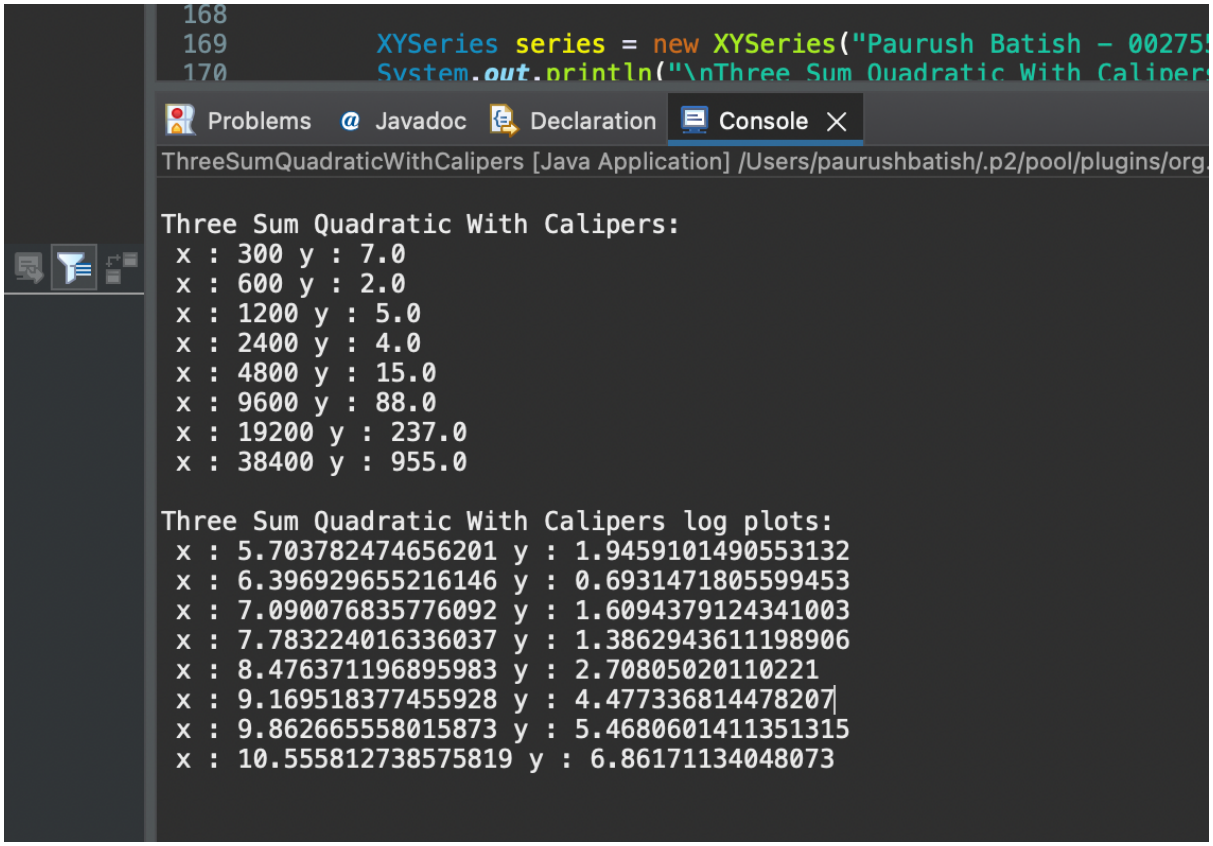
Hence we get the equation of the graph as

$$y = ax^2$$

Hence the order of growth of the code is N^2

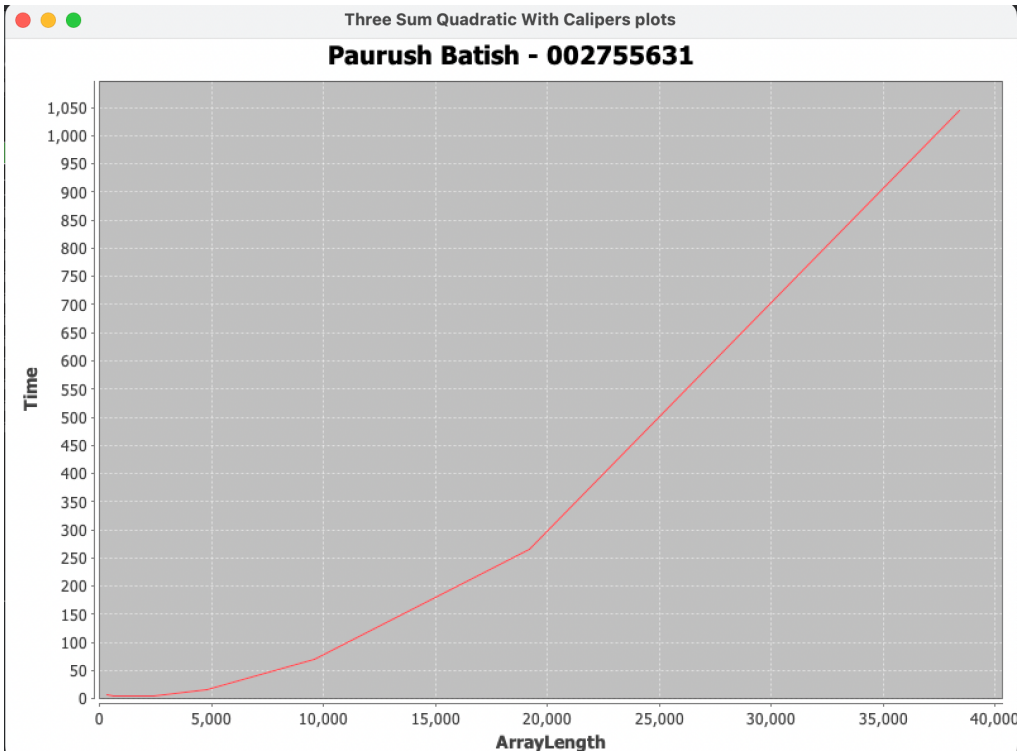
Three Sum Quadratic with calipers

Values:

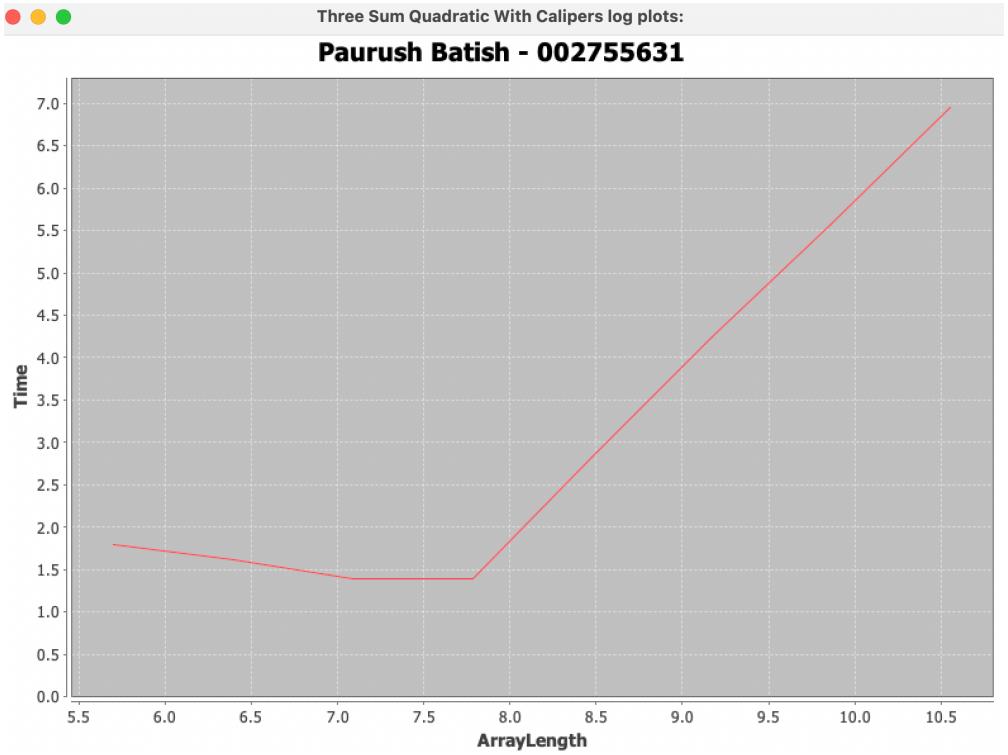


Evidence :

Plotting the graph of array length vs time:



Plotting the graph of ln(array length) vs ln(time):



Slope Calculation:

$$m = \frac{y2 - y1}{x2 - x1}$$

M= (6.951772164398911- 5.579729825986222)/(10.555812738575819- 9.862665558015873)

M = 1.97943867751
M ~ 2

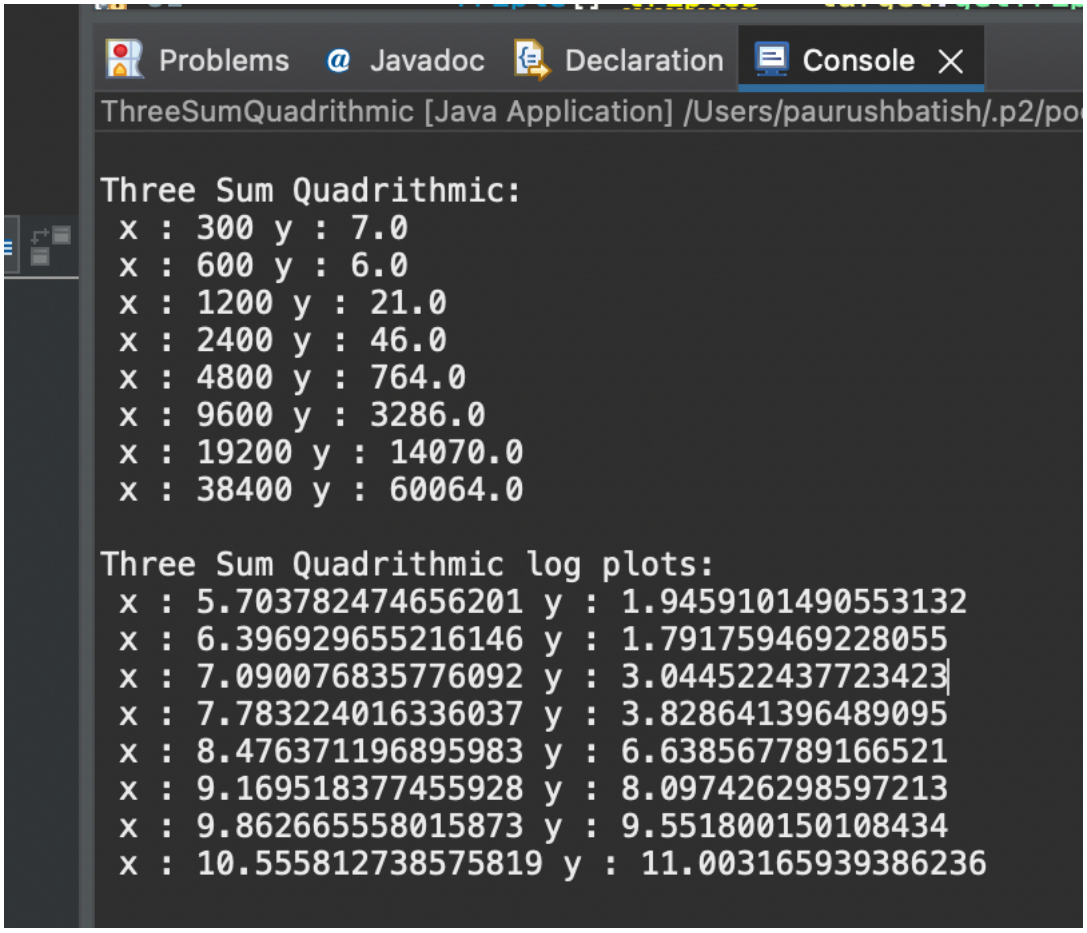
Hence we get the equation of the graph as

$$y = ax^2$$

Hence the order of growth of the code is N^2

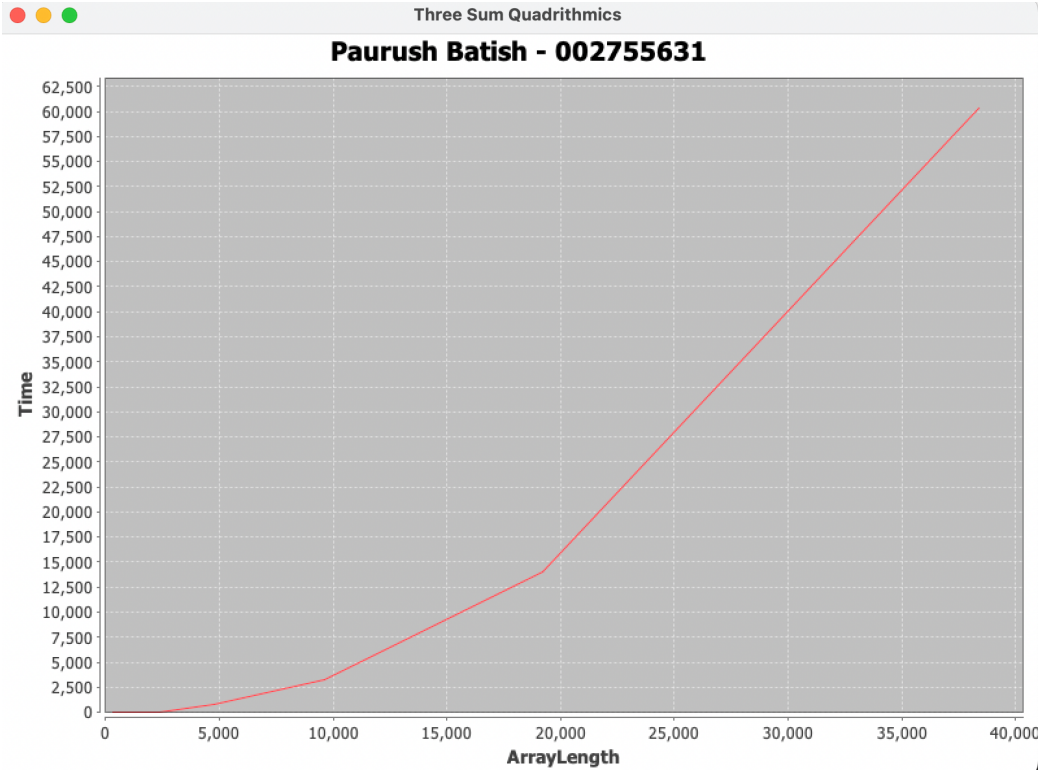
Three Sum Quadrithmic

Values:

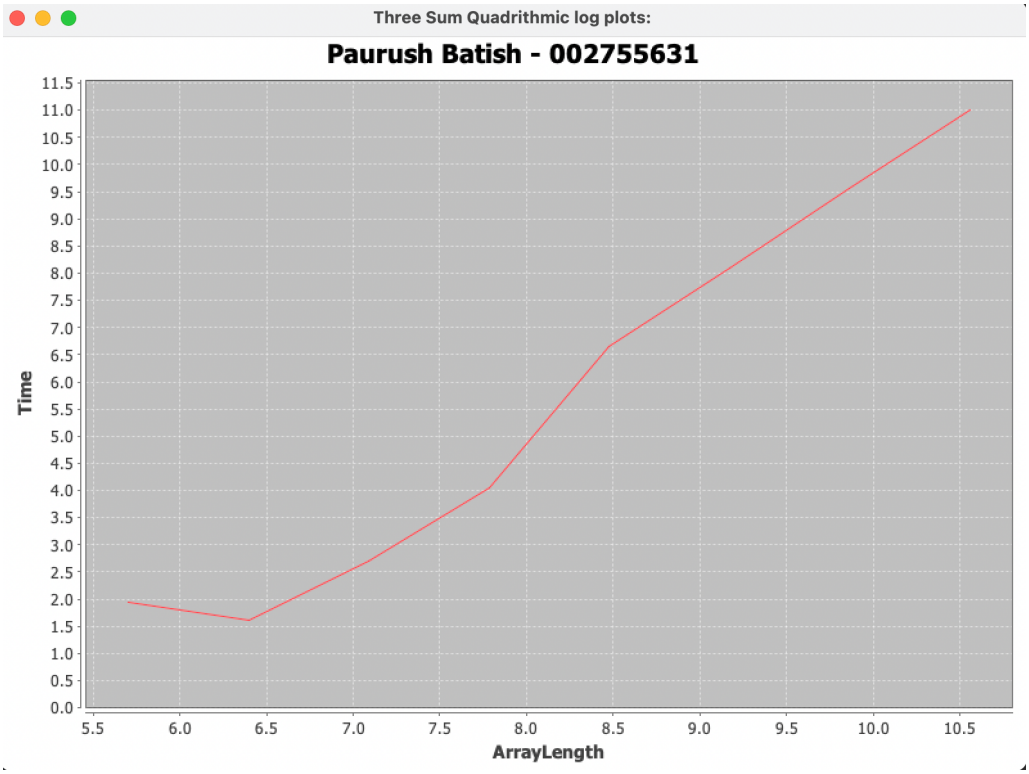


Evidence :

Plotting the graph of array length vs time:



Plotting the graph of ln(array length) vs ln(time):



Slope Calculation:

$$m = \frac{y2 - y1}{x2 - x1}$$

M= (11.007883085959966 - 9.552439604462528)/(10.555812738575819-9.862665558015873)

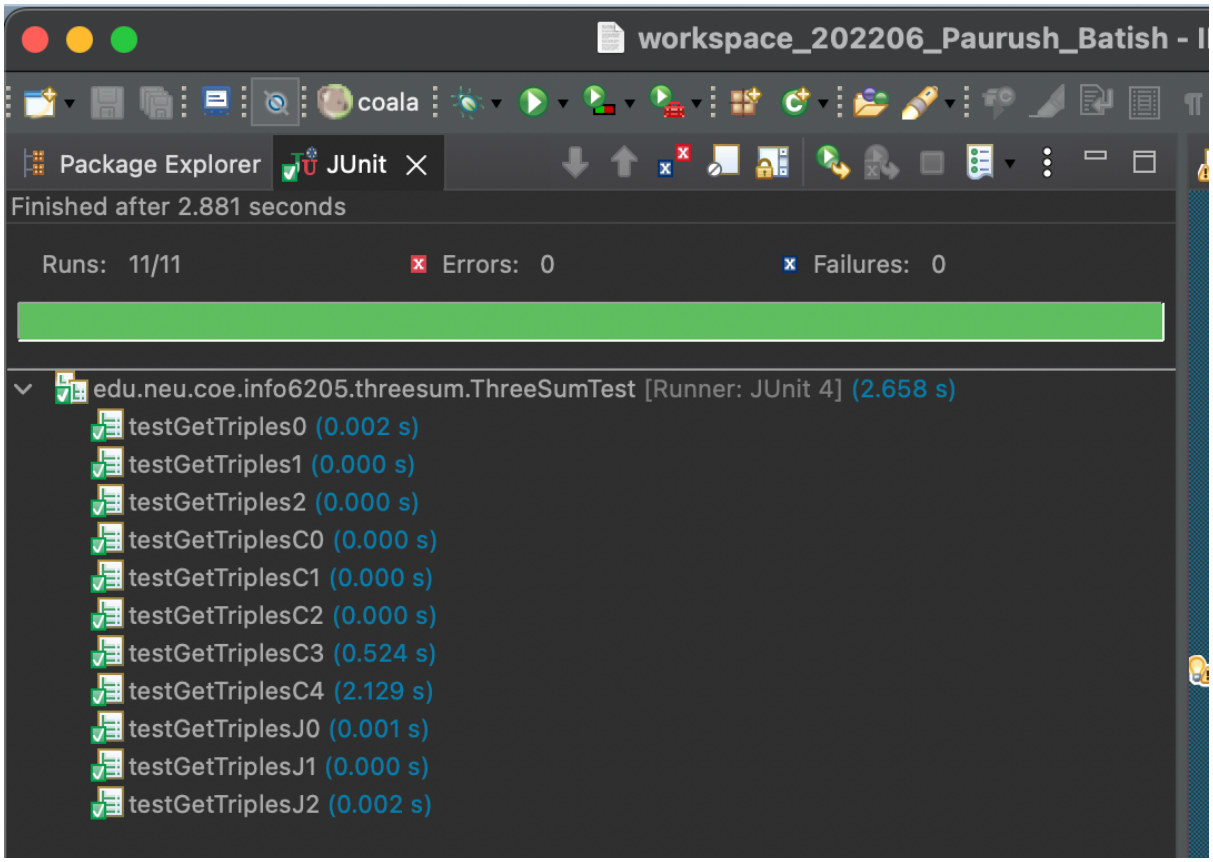
M = 2.19976109305
M ~ 2.2

Hence we get the equation of the graph as

$$y = aN^2 logN$$

Hence the order of growth of the code is $N^2 logN$

Unit Test Cases →



Explanation for code without calipers

The code we are given uses two pointers, one starting at the middle index - 1 and the other at the middle index + 1. The sum of the values at these two pointers and the middle index are compared to a target value. If the sum is greater than the target, the right pointer moves to the right. If the sum is less than the target, the left pointer moves to the left. This process is repeated for each middle index in the loop. The overall complexity of the algorithm is O(N^2) due to the two nested for loops used.

Explanation of code with calipers

The given starting index is used to establish two pointers, one called low starting at the next index and another called high positioned at the last index -1. Depending on the sum generated, the low pointer is moved to the right if it is less than the target and the high pointer is moved to the left if it is greater than the target. This process is repeated for every starting index passed. The overall complexity is O(N^2) as there are two loops involved: one for iterating through the starting indexes and another for adjusting the position of the low and high pointers to find the triplets.