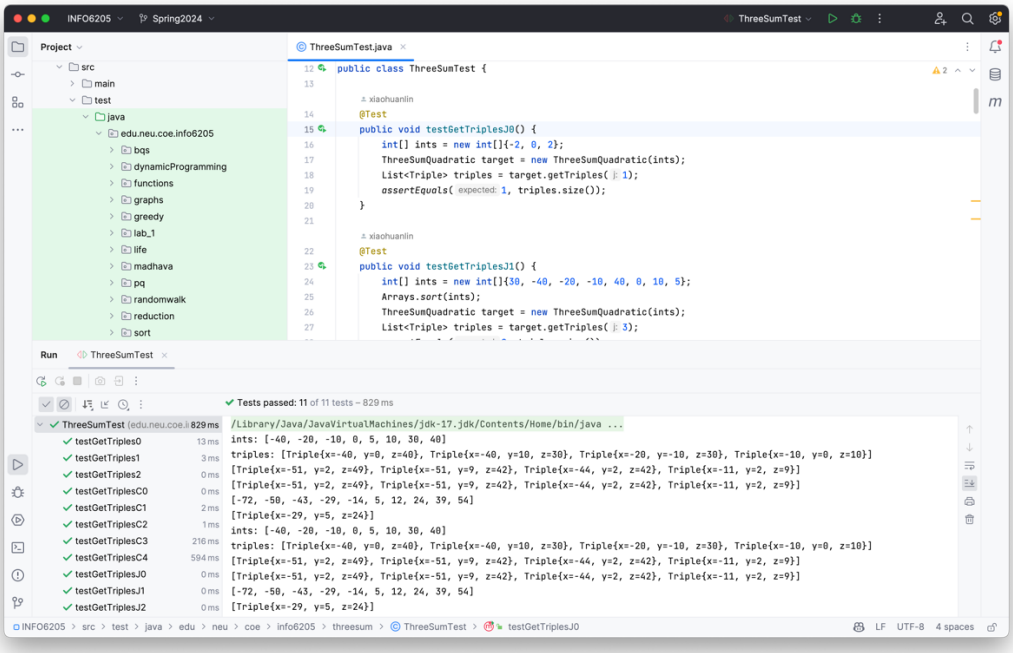
Program Structures and Algorithms
Spring 2024

NAME: Zenan Fan
NUID: 002854067
GITHUB LINK: https://github.com/LearningMach1ne/INFO6205

Task: Assignment 2 (3 Sum)

Unit Test Screenshots:



Spreadsheet showing timing observations:

	A	B	C	D	E
1	**N**	**Time**	**Quadratic**	**Quadrithmic**	**Cubic**
2	250	Raw time per run (mSec)	1.01	1.70	6.31
3		Normalized time per run	16.20	3.42	0.40
4	500	Raw time per run (mSec)	2.28	4.72	46.02
5		Normalized time per run	9.13	2.11	0.37
6	1000	Raw time per run (mSec)	7.25	23.39	351.96
7		Normalized time per run	7.25	2.35	0.35
8	2000	Raw time per run (mSec)	26.01	117.11	2750.31
9		Normalized time per run	6.50	2.67	0.34
10	4000	Raw time per run (mSec)	136.47	558.26	21785.97
11		Normalized time per run	8.53	2.92	0.34
12	8000	Raw time per run (mSec)	723.26	2659.12	N/A
13		Normalized time per run	11.30	3.20	N/A
14	16000	Raw time per run (mSec)	37.55.59	11266.88	N/A
15		Normalized time per run	14.67	3.15	N/A

Why the quadratic method(s) work:
Since the array is sorted, we can use two pointers to point the head and the tail of the array, we can just initialize the first number to be the first element in the array, and second number to be the second element in the array, the third number should be the last element in the array. We can compare the sum with 0, if sum is greater than 0, we can move the tail pointer, which makes sum less, if sum is less than 0, we can move head pointer, which makes sum greater. When the sum is equal to 0, these three numbers is answer.