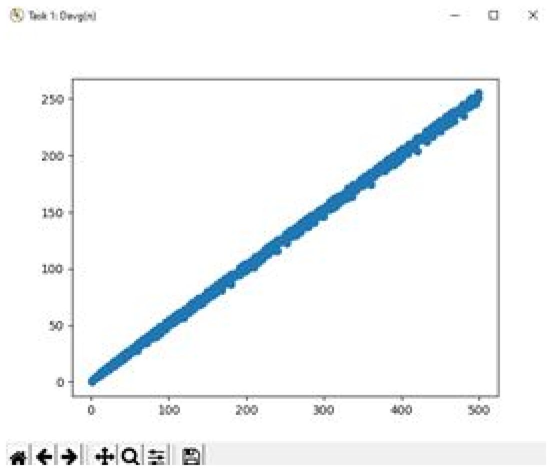
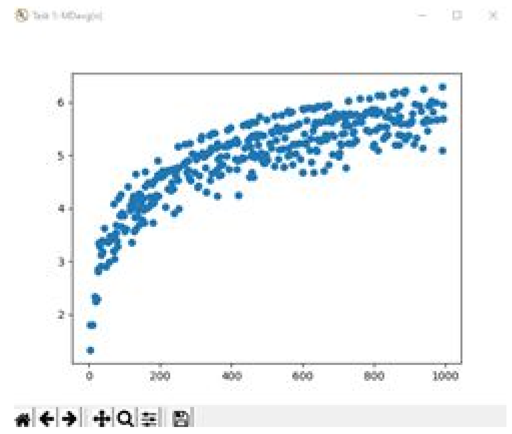
Authors: Spencer Greco, Nicky Valdecanas
Task 1:

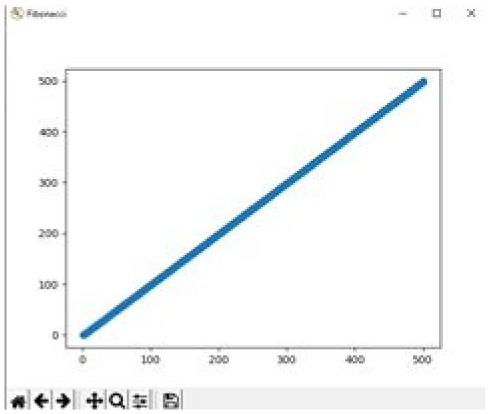


Graph of Average Divisions using Consecutive Integer Checking
Algorithms average-case efficiency: Θ(n)
Number Range(**n**): 1-500 where the x-axis is n and y-axis is the average number of divisions.
The average number of divisions seems to be increasing in a linear fashion.



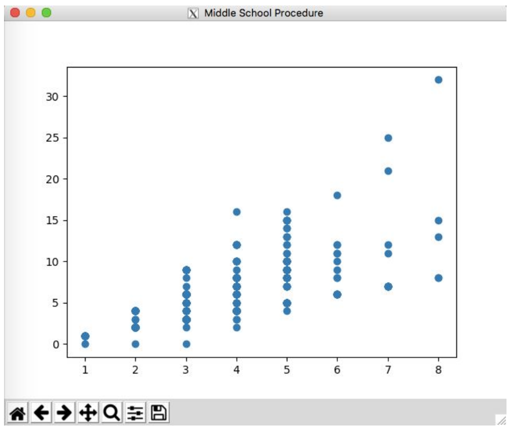
Graph of Average Modulus Divisions using Euclid’s Algorithm
Number Range(**n**): 500 random numbers where the x-axis is a random number n between 1 and 1000 and the y-axis is the number of modulo divisions
Algorithms average-case efficiency class using Euclid's algorithm: Θ(logn)
The number of average modulus divisions slows down as the numbers used increase. When we compare these algorithms side by side, we can see that using modulus divisions is more efficient. The reason for this is because the number of basic operations used is much lower as the input grows larger.

Task2:



Number of Modulus Division taken to compute the GCD in the Fibonacci sequence(m) where m is F(k+1). The x-axis is also m and the y-axis is the number of comparisons.
Worst case efficiency class: n
How does this compare to the average case efficiency of Euclid’s algorithm from task 1:
The amount of divisions in the Fibonacci sequence is roughly twice as many when compared to the amount of divisions used in Euclid’s algorithm from Task 1.
How does the analysis change when measuring the actual time taken instead: It doesn’t change because it seems like they both take longer as the number gets bigger. Even when measuring the processing time it always took long.

Task3:



Basic Operations compared to max list length. The x-axis the max size between the two lists containing common prime factors and the y-axis is the number of total comparisons to find the common prime factors
Number range(**m**): Random input
Number range(**n**): Random input
The bigger the input the more basic operations required: Θ(g) time, where the x-axis is the maximum size of the two lists of common prime factors and y is the number of comparisons between two lists using the middle school procedure to generate the common prime factors for m and n