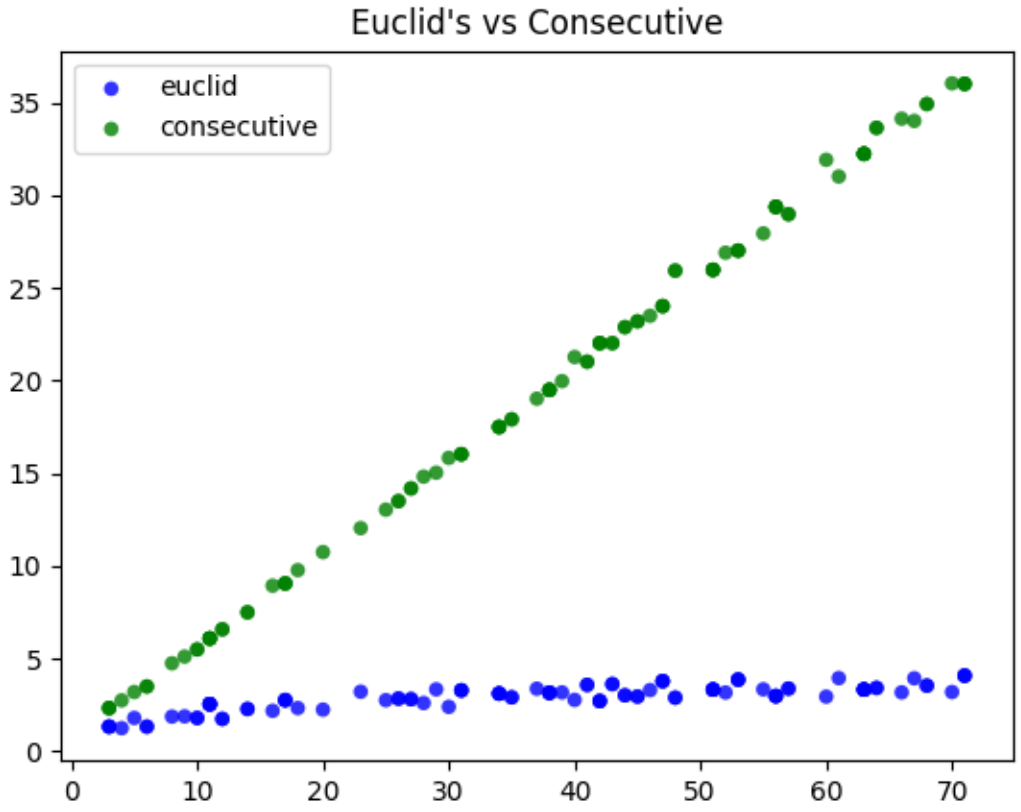
CS 415 Project 1

Keegan Donley, Jeff Hultman

0.1 Task 1

Average-case efficiency of Euclid's algorithm and consecutive integer checking algorithm To test the average-case efficiency of these algorithms, we generate 100 values of n from 1 to 70, then count the number of operations needed to calulate the average GCD for n using Euclid's algorithm and consecutive integer checking.



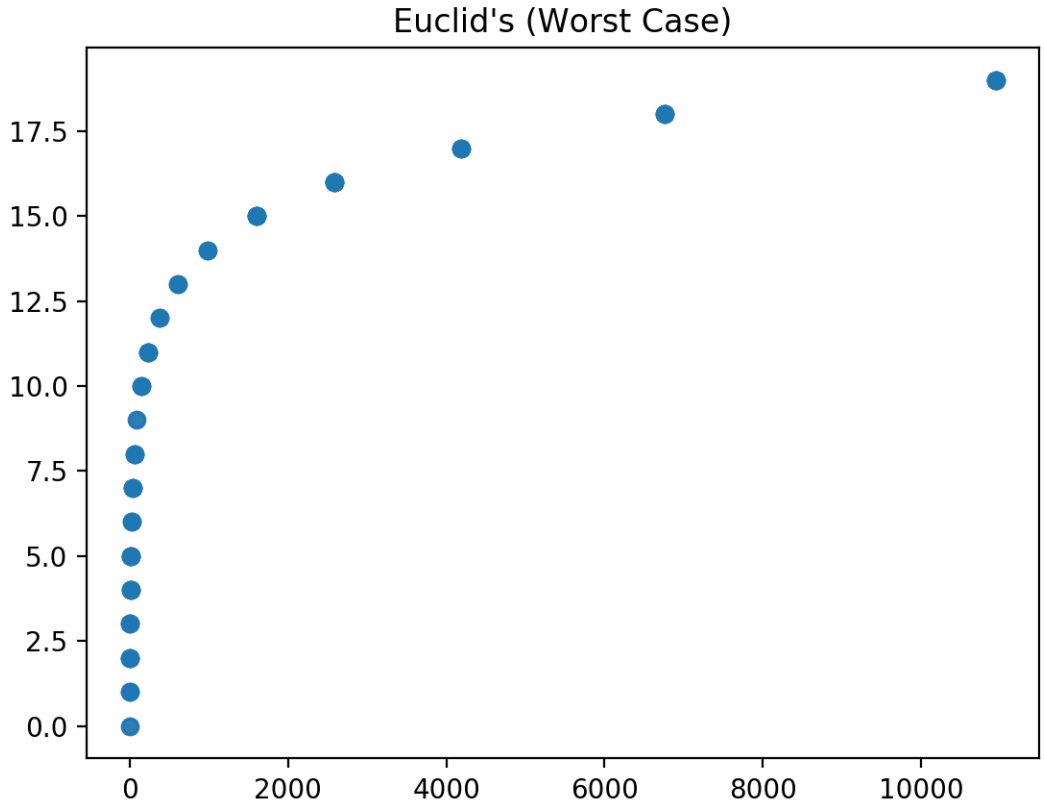
Euclid $\theta(\log n)$

Consecutive Integer $\theta(n)$

0.2 Task 2

Worst-case efficiency of Euclid's algorithm The worst case for Euclid's algorithm occurs when two consecutive integers from the Fibonacci sequence are used as m and n. To test the efficiency, we generate

200 values for k between 1 and 200. k is an index in the Fibonacci sequence, where m = k + 1 and n = k.



0.3 Task 3

The "middle-school procedure" Information