**(2a) partial execution output**

Enter n: 15

i = 8; gcd (5,8) = 1 took 4 module operations

i = 9; gcd (5,8) = 1 took 4 module operations

i = 10; gcd (5,8) = 1 took 4 module operations

i = 11; gcd (5,8) = 1 took 4 module operations

i = 12; gcd (5,8) = 1 took 4 module operations

i = 13; gcd (8,13) = 1 took 5 module operations

i = 14; gcd (8,13) = 1 took 5 module operations

i = 15; gcd (8,13) = 1 took 5 module operations

seconds elapsed 2.05187

**(2b) a graph plotting the maximum number of modules operations where *n = 8 ~ 3000, and log3/22n/3***

**(2c) your mathematical Big-O estimation F(n) and T(n)/F(n) from part 2, and**

Based on the graph I feel that the program is n^2 because my loops would give you and n^2

|  |  |  |  |
| --- | --- | --- | --- |
| **n** | **T(n)** | **F(n)/O(n^2)** | **T(n)/F(n)** |
| 15 | 128.031 | 225 | 0.569026667 |
| 30 | 415.087 | 900 | 0.461207778 |
| 45 | 778.913 | 2025 | 0.384648395 |
| 60 | 2135.04 | 3600 | 0.593066667 |
| 75 | 4036.9 | 5625 | 0.717671111 |
| 90 | 6492.85 | 8100 | 0.80158642 |

**(2d) a table and plot that verifies the Big-O estimation of your program. The table should include n, T(n), F(n), and T(n)/F(n).**