Joshua Landron

CSS 342

Lab 2

2a) Partial output

At i = 8 gcd( 5, 8) = 1 took 5 modulus operations...time required = 8

At i = 9 gcd( 5, 8) = 1 took 5 modulus operations...time required = 4

At i = 10 gcd( 5, 8) = 1 took 5 modulus operations...time required = 5

At i = 11 gcd( 5, 8) = 1 took 5 modulus operations...time required = 6

...

At i = 74 gcd( 34, 55) = 1 took 9 modulus operations...time required = 115

At i = 75 gcd( 34, 55) = 1 took 9 modulus operations...time required = 99

At i = 76 gcd( 34, 55) = 1 took 9 modulus operations...time required = 101

At i = 77 gcd( 34, 55) = 1 took 9 modulus operations...time required = 173

...

At i = 836 gcd( 377, 610) = 1 took 14 modulus operations...time required = 23852

At i = 837 gcd( 377, 610) = 1 took 14 modulus operations...time required = 24806

At i = 838 gcd( 377, 610) = 1 took 14 modulus operations...time required = 23813

At i = 839 gcd( 377, 610) = 1 took 14 modulus operations...time required = 25131

...

At i = 1993 gcd( 987, 1597) = 1 took 16 modulus operations...time required = 157399

At i = 1994 gcd( 987, 1597) = 1 took 16 modulus operations...time required = 158935

At i = 1995 gcd( 987, 1597) = 1 took 16 modulus operations...time required = 164776

At i = 1996 gcd( 987, 1597) = 1 took 16 modulus operations...time required = 168468

...

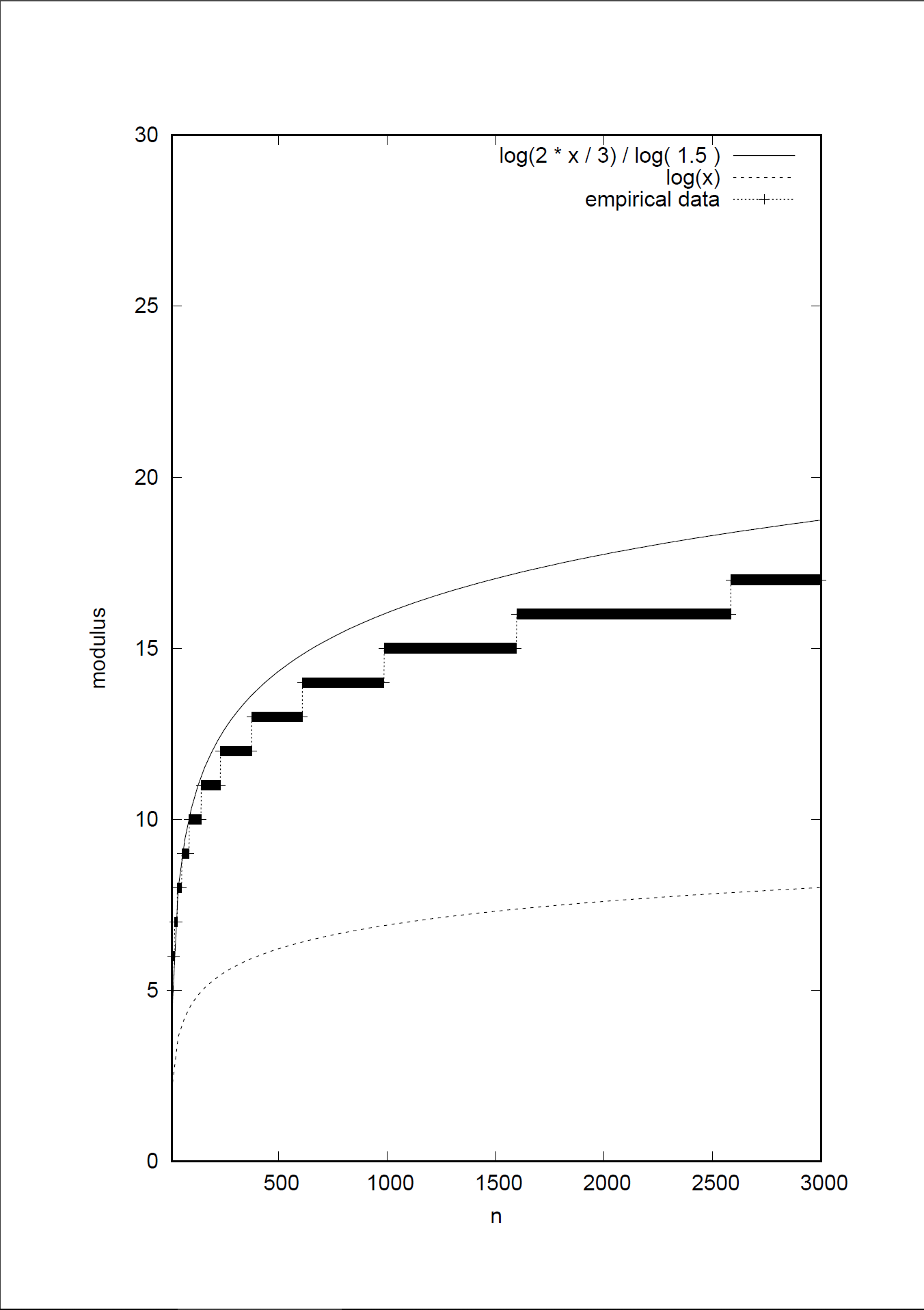
At i = 2997 gcd( 1597, 2584) = 1 took 17 modulus operations...time required = 394224

At i = 2998 gcd( 1597, 2584) = 1 took 17 modulus operations...time required = 389623

At i = 2999 gcd( 1597, 2584) = 1 took 17 modulus operations...time required = 376865

At i = 3000 gcd( 1597, 2584) = 1 took 17 modulus operations...time required = 388518

2b) Graph



2c) Upper bound =

< Average case for single Euclid’s method operation is <

2d) to check that the method of iterating through all possibilities is O(n^2), I have plotted time/n^2 vs n. This shows that the time for each set of calculations is bound to n^2 because the graph converges to a constant value

