Number Theory

Assignment 1 — MTH3251

Greatest Common Divisor

Binary GCD algorithm is an algorithm that computes the greatest common divisor of two non-negative integers. This algorithm replaces division with arithmetic shifts, comparisons, and subtraction. It provides greater efficiency by using bit-wise shift operators which can be optimized in hardware and less expensive than division operations.

1. Prove the following properties of GCD, and use them to justify the intuition behind the binary GCD algorithm.

   i $\gcd(ac, bc) = c \,.\, \gcd(a, b)$

   ii If $\gcd(a, c) = 1$, then $\gcd(a, bc) = \gcd(a, b)$

   iii $\gcd(a, b) = \gcd(a - b, b)$ assuming $a > b$.

2. Write a program that compares the run time of Euclid GCD and Binary GCD for randomly generated integers. Draw a graph like Fig. 1 showing the increase of run time as the number of digits vary between 8 and 64 in the binary representation system. You can use any programming language for implementing the algorithms. You can increase the number of binary digits by using other data types such as *long* rather than *uint64*.

   Note: If you use Python, please note that it is interpreted language, not a programming language, i.e., it is not compiled to optimized machine code. To have a fair comparison, you will need to use a Just-in-Time compiler like *numba* that compiles your python script to hardware machine code. Before measuring the run time, please make sure that you run the two functions at least one time so that they are JIT compiled to avoid including the compilation time in your run time measurement.

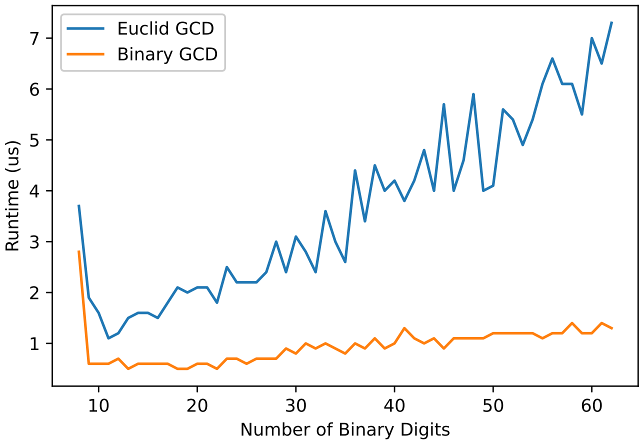

Figure 1: Performance Evaluation of Euclid's GCD vs. Binary GCD