

CS556 Introduction to Cryptography - Prof. Ming-Deh Huang  
Scribe: Iftikhar A Burhanuddin  
burhanud@usc.edu  
Class #2 - August 29, 2002

**Announcements:**

1. Lecture notes for first class are online.
2. We'll start the course with Koblitz's book § 1.1 – 1.3, chapter 3 and § 4.1 – 4.2.
3. Chapter 31, "Introduction to Algorithms" by Cormen et al. is recommended as supplemental reading on elementary number theory.

**Topics for today:**

1. Euclid's gcd algorithm
2. Congruences
3. *Baby* Group Theory

**Lecture:**

We'll study numbers in a computational context, so let's start with some definitions. The *size* of a number defined by bit length is equal to the number of bits to represent the number. Conventional notation for the length of  $x$  is  $\|x\|$  and is  $\sim \log x$ .

Adding two  $n$  bit numbers takes  $n$  or  $n+1$  ( $= O(n)$ ) bit operations. On the other hand multiplying and dividing these numbers by naive methods take  $O(n^2)$  bit operations and the best known algorithms take  $O(n \log n \log \log n)$ . The existence/non-existence of an  $O(n \log n)$  for multiplication is an open problem.

**Euclid's gcd algorithm:**

Suppose  $a, b \in \mathbf{Z}$  we write  $a|b$  if  $b = a.d$  for some  $d \in \mathbf{Z}$ . A *prime* is a positive integer if it's only divisors are 1 and  $p$ .

We are all familiar with the division algorithm for integers, underlying which is the *Division Theorem* which is as follows

$$\forall a \in \mathbf{Z} \forall b \in \mathbf{Z}_{>0} \exists! q, r \in \mathbf{Z} \\ \ni a = qb + r, 0 \leq r < b$$

Ex.  $a=19, b=4. 19 = 4*4+3.$

Greatest common divisor of two positive integers  $a, b$  is denoted by  $\gcd(a, b)$  and is the greatest integer  $d$  such that  $d|a$  and  $d|b$ . Integers decompose *uniquely* into primes and so if we knew the factorization of  $a$  and  $b$ , we could easily compute their gcd by collecting all the common factors. Ex.  $\gcd(21, 18) = \gcd(3 * 7, 3^2 * 2) = 3$ . Unfortunately there are no known *efficient* integer factoring algorithms. Hence we'll seek Euclid's help!

Without loss of generality say  $a > b$ . The Division Thm says  $\exists!q, r$  such that  $a = qb + r$  (\*). Say  $d$  is a common divisor (not necessarily the greatest) of  $a$  and  $b$ . We see that  $d|a, d|b \Rightarrow d|r$ . Conversely  $d|b, d|r \Rightarrow d|a$  (\*\*). And from (\*) and (\*\*) we see that we've just proved  $d|a, d|b$  if and only if  $d|b, d|r$  and hence  $\gcd(a, b) = \gcd(b, r)$ .

Ex.  $\gcd(21, 18) = \gcd(18, 3) = \gcd(3, 0) = 3$ . And this is the *Euclidean gcd* algorithm. Observe that the reason why the algorithm terminates in a finite number of steps is because the numbers involved in each iteration are decreasing and positive integers cannot decrease forever as they are lower bounded by 0. Next we'll compute the number of iterations required to compute  $\gcd(a, b)$ .

We know that  $r > b$  and  $q \geq 1$ . If  $b > \frac{a}{2}$  then  $q = 1$ . So  $r = a - b < \frac{a}{2}$ . On the other hand if  $b \leq \frac{a}{2}$  then  $r < b \leq \frac{a}{2}$ . Therefore  $r$  is always less than  $\frac{a}{2}$ . Hence the number of division requires is  $2 \log a = O(\log a)$ . So the number of divisions is linear in the input length  $\log a$  and the total time complexity of computing the gcd is cubic in  $\log a$  assuming naive divisions are used.

So if  $N$  is the size of the bigger of the two number then addition takes  $O(N)$  bit ops, multiplication consumes  $O(N^2)$  and gcd takes  $O(N^3)$ , the last two being noticeably more expensive than the first. (Can you with a more careful analysis show that gcd takes  $O(N^2)$ ?)

Tracing back through our example...

$3 = 1.3 + 1.0 = 1.3 + 1.(18 - 6.3) = 1.18 - 5.3 = 1.18 - 5(21 - 18) = 6.18 - 5.21$  as  $0 = 18 - 6.3$  and  $3 = 21 - 18$ .

So we can write the greatest common divisor as a linear combination of  $a$  and  $b$ ,  $\gcd(a, b) = x.a + y.b$  for some  $x, y \in \mathbf{Z}$  (can be proved inductively). Using similar analysis we can also show that  $\gcd(a, b) = \gcd(b, w)$  where  $w$  is not necessarily smaller than  $b$ . This seems to indicate that something deeper is going on in Euclid's algorithm and we'll explore this by introducing congruences.

**Congruences:**

Let  $m \in \mathbf{Z}_{>0}$  and for  $a, b \in \mathbf{Z}$ , we write  $a \equiv b \pmod{m}$  if  $m|a - b$ .  $m$  is called the modulus. Please note the notational difference between “ $x \equiv y \pmod{z}$ ” which means  $z|x - y$  and “ $x = y \pmod{z}$ ” which means that  $z$  is the remainder when dividing  $y$  by  $z$ .

We observe that as a relation on integers mod  $m$  displays the following properties

1. Reflexive:  $a \equiv a \pmod{m}$  for all  $a \in \mathbf{Z}$
2. Symmetric:  $a \equiv b \pmod{m} \Rightarrow b \equiv a \pmod{m}$  for all  $a, b \in \mathbf{Z}$  (follows from definition)
3. Transitive:  $a \equiv b \pmod{m}$  and  $b \equiv c \pmod{m} \Rightarrow a \equiv c \pmod{m}$  for all  $a, b, c \in \mathbf{Z}$ . This is true as  $m|a - b$ ,  $m|b - c \Rightarrow m|(a - b) + (b - c) \Rightarrow m|a - c$

Hence the above congruence relation defines an equivalence relation on the integers. When we look at the first non-trivial value of  $m = 2$ , all the even integers fall into  $[0]$  (read as class of 0) and all the odd integers into  $[1]$  congruence class. More generally the mod  $m$  relation divides the integers into  $m$  classes,  $[0], [1], \dots, [m - 1]$ , where  $[0]$  are integers which are multiples of  $m$ ,  $[1]$  has integers which are of the form  $mk + 1$ , that is integers which leave a remainder of 1 when divided by  $m$  and so on.

When we consider these classes as a set we see that addition and multiplication respect the congruences.  $a \equiv u \pmod{m}$  and  $b \equiv v \pmod{m}$  implies  $a + b \equiv u + v \pmod{m}$  and also  $a \cdot b \equiv u \cdot v \pmod{m}$ . This set  $\{[0], [1], \dots, [m - 1]\}$  is denoted by  $\mathbf{Z}/m\mathbf{Z}$  (and by  $\mathbf{Z}_m$  which has the risk of being mistaken for the ring of  $p$ -adic integers.)

$\mathbf{Z}/m\mathbf{Z}$  is closed under  $+$ ,  $*$  as it inherits it from the set of integers. Exploring further we see that the  $\mathbf{Z}/m\mathbf{Z}$

1. is closed under the modular addition operation
2. has a natural identity  $[0]$  as  $[a] + [0] = [0] + [a] = [a]$ , that is  $a + 0 \equiv a \pmod{m}$
3. has an inverse for each element in the set  $[a] + [-a] = [0]$
4. the modular addition operation is associative as  $(a + b) + c = a + (b + c)$  in  $\mathbf{Z}$  implies  $(a + b) + c \equiv a + (b + c) \pmod{m}$ , that is  $([a] + [b]) + [c] = [a] + ([b] + [c])$

5. also the operator  $+ \text{ mod } m$  is commutative  $[a] + [b] = [b] + [a]$

The first four make the set  $\mathbf{Z}/m\mathbf{Z}$  along with  $+$  a group and commutativity makes it an abelian group. Also as the set is finite to begin with, the group  $(\mathbf{Z}/m\mathbf{Z}, +)$  is a finite abelian group. Next we'll look at a more formal definition of a group.

A *Group* is a set  $(G, \odot)$  where  $\odot$  is a binary operator such that

$$G * G \xrightarrow{\odot} G$$

and behaves as follows:

1. Closed under  $\odot$ , which is true due to the definition
2. Existence of identity  $e \in G$ .  $\forall a \in G \ a \odot e = e \odot a = a$
3. Existence of inverse.  $\exists b \in G$  such that  $a \odot b = e$
4. Associativity.  $\forall a, b, c \in G \ (a \odot b) \odot c = a \odot (b \odot c)$

The group  $G, \odot$  is abelian (commutative) if  $\forall a, b \in G \ a \odot b = b \odot a$ .

**Exercise:** Suppose  $G, \odot$  is a group show that  $a \odot b = e \Rightarrow b \odot a = e$ .

So do the natural numbers form a group under multiplication? No, as an inverse doesn't exist for almost all elements. In the next class we'll see whether  $(\mathbf{Z}/m\mathbf{Z}, *)$  for a group and if so for which values of  $m$ .