The Euclidean algorithm

Saad Mneimneh

1 The greatest common divisor

Consider two positive integers $a_0 > a_1$. The greatest common divisor of a_0 and a_1, denoted $\gcd(a_0, a_1)$ is the largest positive integer g such that $g|a_0$ and $g|a_1$, i.e. g divides both a_0 and a_1.

Observation 1: The $\gcd(a_0, a_1)$ always exists.

Observation 2 (Euclid): Let $a_0 = q_1 a_1 + r$ where $0 \leq r < a_1$ (note that this representation is always possible and unique), then $\gcd(a_0, a_1) = \gcd(a_1, r)$. The proof of this fact consists of showing that $d|a_0$ and $d|a_1 \Leftrightarrow d|a_1$ and $d|r$.

2 The Euclidean algorithm

The Euclidean algorithm finds the gcd recursively by computing the sequence

$$a_0 \ a_1 \ \ldots \ a_k \ a_{k+1}$$

where

$$a_i = a_{i-2} - \lfloor \frac{a_{i-2}}{a_{i-1}} \rfloor a_{i-1} = a_{i-2} - q_{i-1} a_{i-1}$$

$$a_{k+1} = 0$$

The sequence $\{a_i\}$ is strictly decreasing and, therefore, $a_{k+1} = 0$ is guaranteed. We can easily show that $a_k = \gcd(a_0, a_1)$.

$$\gcd(a_0, a_1) = \gcd(a_1, a_2) = \ldots = \gcd(a_{k-1}, a_k)$$

Since $a_{k+1} = 0$, a_{k-1} is a multiple of a_k and hence $\gcd(a_{k-1}, a_k) = a_k$.

Example: The gcd(300,18) is 6.

300 18 12 6 0

3 Running time

We have the following recurrence:

$$a_i = a_{i-2} - q_{i-1}a_{i-1}$$

which we can rewrite as

$$a_{i-2} = a_i + q_{i-1}a_{i-1}$$

and since $q_{i-1} = \left\lfloor \frac{a_{i-2}}{a_{i-1}} \right\rfloor \geq 1$,

$$a_{i-2} \geq a_{i-1} + a_i$$
$$a_{k-1} \geq 2$$
$$a_k \geq 1$$

Compare this to the famous Fibonacci sequence:

$$F_n = F_{n-1} + F_{n-2}$$
$$F_3 = 2$$
$$F_2 = 1$$

F_0	F_1	F_2	F_3	$\ldots$	F_{k+i-2}	$\ldots$	F_{k+2}
0	1	1	2				
	0	a_k	a_{k-1}	$\ldots$	a_i	$\ldots$	a_0

It can be easily seen that $F_{k+2-i} \leq a_i$ (for $i \leq k$). Therefore, $F_{k+2} \leq a_0$. This means k cannot be very large. In deed, we can show by induction that $c\phi^{n-1} \leq F_n$ where ϕ is the golden ratio and c is some positive constant.

$$c\phi^{k+1} \leq a_0$$
$$k \leq \log_\phi \frac{a_0}{c} - 1$$

We conclude that k is logarithmic in a_0 and thus linear in the length of a_0 (in bits for instance). We also conclude that the worst case occurs when a_0 and a_1 are consecutive Fibonacci numbers. Here's an example of the sequence when $a_0 = 13$ and $a_1 = 8$.

13 8 5 3 2 1 0

4 Extended Euclidean algorithm

Instead of simply computing the sequence $\{a_i\}$, we can compute $\{x_i\}$ and $\{y_i\}$ such that:

$$a_i = a_0 x_i + a_1 y_i$$

This can be done inductively by starting with $x_0 = 1$, $y_0 = 0$, and $x_1 = 0$, $y_1 = 1$. Then

$$a_i = a_{i-2} - q_{i-1} a_{i-1} = a_0 x_{i-2} + a_1 x_{i-2} - q_{i-1}(a_0 x_{i-1} + a_1 y_{i-1})$$

By regrouping terms we get,

$$a_i = a_0(x_{i-2} - q_{i-1} x_{i-1}) + a_1(y_{i-2} - q_{i-1} y_{i-1})$$

which leads to the following recurrences:

$$x_i = x_{i-2} - q_{i-1} x_{i-1}$$

$$y_i = y_{i-2} - q_{i-1} y_{i-1}$$

Since $\gcd(a_0, a_1) = a_k$, we now have that $\gcd(a_0, a_1)$ is a linear combination of a_0 and a_1.

5 Applications

Consider a positive integer n and let $a \in \{1, \ldots, n-1\}$ be such that $\gcd(n, a) = 1$ (n and a are relatively prime or coprimes). The extended Euclidean algorithm can be used to find the multiplicative inverse of a, denoted a^{-1}, i.e. a positive integer $a^{-1} \in \{1, \ldots, n-1\}$ such that

$$aa^{-1} \equiv 1 \bmod n$$

Example: Let $n = 18$ and consider the set integers less than 18 that are relatively prime to 18, $\{1, 5, 7, 11, 13, 17\}$. The following represent multiplications modulo 18.

$$1 \cdot 1 = 1$$

$$5 \cdot 11 = 1$$

$$7 \cdot 13 = 1$$

$$17 \cdot 17 = 1$$

Here's how to find the multiplicative inverse. Since $\gcd(n, a) = 1$ then,

$$1 = nx + ay$$

where y is not necessarily in $\{1, \ldots, n-1\}$.

$$ay \equiv 1 \bmod n$$

$$a(y \bmod n) \equiv 1 \bmod n$$

Therefore, $a^{-1} = y \bmod n$ is the multiplicative inverse of a.

The concept of a multiplicative inverse is used in cryptography.

RSA

1. generate two large primes p and q
2. compute $n = pq$
3. find $e \in \{1, \ldots, (p-1)(q-1)-1\}$ such that $\gcd((p-1)(q-1), e) = 1$
4. publish (e, n)
5. compute the secret d such that $ed \equiv 1 \bmod (p-1)(q-1)$ (multiplicative inverse)

Given a message x ($x < n$), compute $y = x^e \bmod n$. This is the encryption of x. Only the one who has secret d can decrypt the message, by computing $x = y^d \bmod n$ (in principle, one could compute the e^{th} root of y modulo n, but we don't know of an easy way to do this without the knowledge of d).

Now we prove that $x = y^d \mod n$.

$$y^d = x^{ed} = x^{k(p-1)(q-1)+1} = [x^{k(q-1)}]^{p-1}x$$

We now use the following celebrated result:

Fermat's Theorem
if p is prime and p does not divide a, then $a^{p-1} \equiv 1 \bmod p$.

Therefore, if p does not divide $x^{k(p-1)}$, then $[x^{k(q-1)}]^{p-1} \equiv 1 \bmod p$, which means $[x^{k(q-1)}]^{p-1}x \equiv x \bmod p$. If p divides $x^{k(p-1)}$, then p must divide x, which means $x \equiv 0 \bmod p$ and hence $[x^{k(q-1)}]^{p-1}x \equiv 0 \bmod p$. In both cases, we conclude that

$$y^d \equiv x \bmod p$$

and by switching the roles of p and q, we also get:

$$y^d \equiv x \bmod q$$

Both p and q are primes with $n = pq$; therefore,

$$y^d \equiv x \bmod n$$
$$y^d - x \equiv 0 \bmod n$$
$$(y^d \bmod n) - x \equiv 0 \bmod n$$

Since $y^d \bmod n$ and x are both less than n, they must be equal.

The extended Euclidean algorithm can also be used to obtain a constructive proof for the Chinese Remainder Theorem.

Chinese Remainder Theorem
Let $x \equiv a_i \bmod n_i$ for $i = 1 \ldots k$, and let $n_1, n_2, \ldots, n_k$ be pairwise coprimes. Then x has a solution, and all solutions are congruent modulo $n = \prod_{i=1}^{k} n_i$.

Note that n_i and n/n_i are coprimes and hence must satisfy:

$$1 = n_i r_i + (n/n_i) s_i$$

Let $e_i = (n/n_i)s_i$ (which can be found using the extended Euclidean algorithm). Then,

$$e_i \equiv 1 \bmod n_i$$
$$e_i \equiv 0 \bmod n_j, j \neq i$$

Now set $x = \sum_{i=1}^{k} e_i a_i$. It is easy to see that x satisfies $x \equiv a_i \bmod n_i$ for all $i = 1 \ldots k$. In fact, any integer congruent to x modulo n does. Furthermore, if x and y are both solutions, then $x - y \equiv 0 \bmod n_i$ for all $i = 1 \ldots k$, which implies that $x - y \equiv 0 \bmod n$ (because the n_i's are pairwise coprimes).