Divisors and the Euclidean Algorithm

Jonathan Lam

February 1, 2020

DEF Let $a, b \in \mathbb{Z}, a \neq 0$. Then a **divides** b (den. $a|b$) if $\exists q \in \mathbb{Z} : aq = b$

(Note that $\forall a \neq 0, a|0$).

THM (Division algorithm) $\forall a, d \in \mathbb{Z}, d \neq 0 \ \exists!\ q, r \in \mathbb{Z} : a = dq + r,\ 0 \leq r < d$.

THM (Well-ordering principle) For any nonempty set $S \subseteq \mathbb{Z}_+$, S has a (unique) minimum element.

DEF (gcd) $\forall a, b \in \mathbb{Z}_+, d = \gcd(a, b)$ if $d|a$, $d|b$, and $d'|a \wedge d'|b \Rightarrow d'|d\ \forall d' \in \mathbb{Z}_+$.

(I.e., d divides both a and b, and all divisors of both a and b also divide d. Clearly, d is the maximum of all divisors of both a and b.)

THM (Alternate definition of the gcd) Define

$$S \coloneqq \left\{ ax + by\ (\in \mathbb{Z}_+) : \begin{array}{r} a, b \in \mathbb{Z}_+ \\ x, y \in \mathbb{Z} \\ ax + by > 0 \end{array} \right\}$$

Then, for some $a, b \in \mathbb{Z}_+$, define

$$d \coloneqq \min S$$

Then d exists, and

$$\gcd(a, b) = d$$

PF Three things need to be proved: (1): existence of d; (2): $d|a$ and $d|b$; (3): if $d'|a$ and $d'|b$, then $d'|d$. (2) and (3) are the hypotheses for the definition of gcd.

1. **Existence of d**

   S is a nonempty subset of $\mathbb{Z}$; by the well-ordering principle, it has a minimum element. Thus d exists (and is unique).

2. d **divides** a **and** b

   By the division algorithm, $a = dq + r,\ q, r \in \mathbb{Z},\ r < d$. To show that $d|a$, we have to show that $r = 0$. We prove this by contradiction: assume $r > 0$.

$$d \in S \Rightarrow \exists x, y \in \mathbb{Z} : ax + by = d$$

$$d > r = a - dq = a - (ax + by)q = a(q - xq) + b(yq) \in S \geq d \Rightarrow \perp$$

   By the contradiction, $r = 0$. The same logic applies to show that $d|b$.

3. **Any divisor** d' **of both** a **and** b **also divides** d

   By hypothesis, $\exists h, k \in \mathbb{Z} : d'h = a, d'k = b$.

$$d \in S \Rightarrow \exists x, y \in \mathbb{Z} : ax + by = d$$

$$d = (d'h)x + (d'h)y = d'(hx + ky) \Rightarrow d'|d \ \blacksquare$$

THM Let $a, b \in \mathbb{Z}_+,\ b \neq 0$. Then $\gcd(a, b) = \gcd(b, r)$, where r is obtained by the division algorithm applied on a and b.

PF By division algorithm, $a = bq + r,\ q, r \in \mathbb{Z}$. Define D to be the set of integers that divide both a and b, and define D' to be the set of integers that divide both b and r. Suppose $c \in D$; i.e., c divides both a and b Then

$$c|bq \Rightarrow c|a - bq \Rightarrow c|r$$

Thus c is divides both b and r, and thus

$$c \in D' \Rightarrow D' \subseteq D$$

Conversely, suppose $c \in D'$, i.e., c divides both b and r. Then

$$c|bq \Rightarrow c|bq + r \Rightarrow c|a$$

Thus c divides both a and r, and thus

$$c \in D \Rightarrow D \subseteq D'$$

Thus $D = D'$. In particular, $\gcd(D) = \max D = \max D' = \gcd(D')$. $\blacksquare$

(Note that this theorem doesn't guarantee termination of the Euclidean algorithm; for this to be true, $a \geq b$ is a necessary condition).

ALG (Euclidean Algorithm) Define the following algorithm:

```
gcd(a, b) {
    if(a < b)
        swap(a, b)
    while(b != 0) {
        r = a mod b
        a = b
        b = r
    }
    return a
}
```

Alternatively, recursively:

```
// the driver assures that a<=b
gcd(a, b) {
    if(b = 0)
        return a
    return gcd(b, a mod b)
}
gcd_drv(a, b) {
    return (a < b) ? gcd(b, a) : (a, b)
}
```

Intuitive PF This is applying the above theorem to a and b, recursively. Since $b \leq a$ (in the algorithm after the appropriate swapping, not necessarily for the initial invocation), $r < \min\{a, b\}$, the size of the inputs (from (a, b) to (b, r)) are (strictly) decreasing with subsequent invocations of the function. This means it will eventually reduce to the base case $(c, 0)$; since $gcd(c, 0) = c = gcd(a, b)$ (by the above theorem), this algorithm is correct.

THM The equation $ax + by = m\, a, b, m \in \mathbb{Z}$, a and b not both 0, has a solution $(x, y) \in \mathbb{Z}^2$ iff $\gcd(a, b) | m$.

PF ($\Leftarrow$) Since $d = \gcd(a, b) \in S$ (S defined in earlier theorem), $\exists x', y' \in \mathbb{Z} : d = ax' + by'$. Since $d|m$, $kd = m \Rightarrow k(ax' + by') = m \Rightarrow a(kx') + b(ky') = m \Rightarrow (kx', ky')$ is a solution to the equation.

PF ($\Rightarrow$) Let $ax + by = m$, $dh = a$, $dk = b$. Then $(dh)x + (dk)y = d(hx + ky)m \Rightarrow d|m$. ■

ALG Algorithm to find a solution to $sa + tb = \gcd(a, b)$. (When the right side of the equation is $m \gcd(a, b)$, then multiply the determined coefficients by m).

```
gcd_st(a, b) {
    if(a < b)
        swap(a, b)
    if(b = 0)
        return (gcd=a, s=1, t=0)
    (gcd, s', t') = gcd_st(b, a mod b)
    return (gcd=gcd, s=t', t=s'-t'*(a/b))
}
```

(Here, the division a/b represents the integer division quotient.) This algorithm works due to the recurrence relation where $\gcd(a, b) = sa + tb = (t')a + (s' - t'(a/b))b$, where s' and t' are the integers such that $\gcd(b, r) = s'b + t'r$. The base case for this recursive relation is the case $\gcd(d, 0)$, which can be expressed as $(1)d + (0)0 = d$.

Primes stuff Fundamental Theorem of Arithmetic, Infinite Primes, GCD/LCM in terms of Prime Factorization, LCM in terms of GCD