THE EUCLIDEAN ALGORITHM

I have isolated proofs at the end. *Fancy not, even for a moment, that this means the proofs are unimportant!* They are *essential* to understanding the algorithm. Rather, I thought it easier to use this as a reference if you could see the algorithms with the examples first, and the proofs later.

The Euclidean Algorithm. *If $b \neq 0$, the following algorithm terminates with* $\gcd(a, b)$.

```
inputs
   $a, b \in \mathbb{Z}$ where $b \neq 0$
outputs
   $\gcd(a, b)$
do
   Let $n = \max(a, b)$, $d = \min(a, b)$
   while $d \neq 0$ do
      Compute $q, r \in \mathbb{Z}$ such that $n = dq + r$ and $0 \leq r < d$
      Let $n = d$, $d = r$
   return $n$
```

Example. To compute $\gcd(30, 286)$, the algorithm generates

$$\begin{aligned} 286 &= 9 \times 30 + 16 \\ 30 &= 1 \times 16 + 14 \\ 16 &= 14 \times 1 + 2 \\ 14 &= 7 \times 2 + 0. \end{aligned}$$

The Extended Euclidean Algorithm, *or*, Bezout's Identity. *For any integers a, b we can find integers m, n such that*

$$\gcd(a, b) = ma + nb.$$

In addition, we can find m, n by reversing the equations generated during the Euclidean Algorithm.

Example (continued)**.** Isolate the remainder in the generated equations:

$$2 = 16 + (-1) \times 14 \tag{0.1}$$

$$14 = 30 + (-1) \times 16 \tag{0.2}$$

$$16 = 286 + (-9) \times 30. \tag{0.3}$$

Equation (0.1) one has the form

$$\gcd(30, 286) = 16 + (-1) \times 14.$$

Into this, we substitute the value of 14 from equation (0.2), obtaining

$$\begin{aligned} \gcd(30, 286) &= 16 + (-1) \times [30 + (-1) \times 16] \\ &= 2 \times 16 + (-1) \times 30. \end{aligned}$$

We now substitute the value of 16 from equation (0.3), obtaining

$$\begin{aligned}\gcd(30,286) &= 2\times[286+(-9)30]+(-1)\times 30\\ &= 2\times 286+(-19)\times 30.\end{aligned}$$

The gcd property of $\mathbf{ax+by=d}$. *For fixed $a,b\in\mathbb{Z}$, the smallest positive number d that can be written in the form $d=ax+by$, where $x,y\in\mathbb{Z}$, is $d=\gcd(a,b)$.*

The following corollary depends on the gcd property of $ax+by=d$, and will not be explained further.

Corollary. *For fixed $a,b\in\mathbb{Z}$, if we can find $x,y\in\mathbb{Z}$ such that $1=ax+by$, then $\gcd(a,b)=1$.*

So why does this work?

Inspired by the Extended Euclidean Algorithm, we'll explain everything in reverse.

Proof of the gcd property of $ax+by=d$. Inasmuch as $\gcd(a,b)$ divides both a and b, it divides the left hand side of $ax+by=d$. It must divide the right hand side, as well. The smallest positive multiple of $\gcd(a,b)$ is itself. □

The Euclidean Algorithm depends on the following Lemma, which should help answer the question of why we do it this way.

Lemma 1. *If $a=bq+r$ then $\gcd(a,b)=\gcd(b,r)$.*

Proof of Lemma 1. Let $d=\gcd(a,b)$ and $d'=\gcd(b,r)$. Since $a=bq+r$, we see that d' divides the left hand side; thus, it divides a; hence d' is a common divisor of a and b. Since d is the greatest common divisor of a and b, $d'\le d$.

Rewrite the equation as $a-bq=r$. We see that d divides the right hand side; thus, it divides r; hence d is a common divisor of b and r. Since d' is the greatest common divisor of b and r, $d\le d'$.

We have shown that $d'\le d\le d'$. This forces $d=d'$, and the lemma is proved. □

Proof of the Euclidean Algorithm. We know from Lemma 1 that the gcd is preserved in each remainder. The algorithm concludes when $r=0$. Hence $\gcd(a,b)=\gcd(n,r)=\gcd(n,0)=n$.

Moreover, we know the algorithm terminates because r always satisfies $0\le r<n$, r decreases on each pass through the loop, and no sequence of strictly decreasing positive numbers is infinite (by the well ordering of the natural numbers).

Each equation generated by the Euclidean Algorithm has the form

$$n_i=q_id_i+r_i$$

where $n_1=\max(a,b)$, $d_1=\min(a,b)$, $n_i=d_{i-1}$, and $d_i=r_{i-1}$. Suppose k is index of the last equation generated, so that $n_k=q_kd_k+0$.

Since $\gcd(a,b)=d_k$, the *next-to-last* equation of the Euclidean Algorithm has the form

$$n_{k-1}=q_{k-1}d_{k-1}+\gcd(a,b).$$

Rewrite this to isolate $\gcd(a,b)$:

$$\gcd(a,b)=n_{k-1}+(-q_{k-1})\,d_{k-1}. \tag{0.4}$$

The equation before this in the Euclidean Algorithm has the form

$$n_{k-2}=q_{k-2}d_{k-2}+r_{k-2},$$

but $r_{k-2} = d_{k-1}$, so we can rewrite it as

$$d_{k-1} = n_{k-2} + \left(-q_{k-2}\right) d_{k-2},$$

and substitute into equation 0.4 to obtain

$$\begin{aligned}\gcd(a, b) &= n_{k-1} + \left(-q_{k-1}\right)\left[n_{k-2} + \left(-q_{k-2}\right) d_{k-2}\right]\\ &= n_{k-1} + \left(-q_{k-1}\right) n_{k-2} + \left(q_{k-1} q_{k-2}\right) d_{k-2}.\end{aligned}$$

Recall that $n_{k-1} = d_{k-2}$, so we can rewrite this as

$$\gcd(a, b) = \left(-q_{k-1}\right) n_{k-2} + \left(1 + q_{k-1} q_{k-2}\right) d_{k-2}. \tag{0.5}$$

Again, $r_{k-3} = d_{k-2}$, so we can repeat this until we run out of equations with n_1 and d_1, which we assigned to be a and b. □