Scribe Notes for *Algorithmic Number Theory*
Class 8—May 28, 1998

Scribes: Wen Wang, Yizhong Wang, and Jeremy Rotter

Abstract

This class continues the worst-case analysis of Euclidean algorithm. Then we cover the extended Euclidean algorithm.

1 Review

In the refined analysis of the Euclidean algorithm, we developed the following.

- Assume $u > v > 0$. Let $u_0 = u, u_1 = v$, and

$$\begin{aligned} u_0 &= a_0 u_1 + u_2 \\ u_1 &= a_1 u_2 + u_3 \\ u_2 &= a_2 u_3 + u_4 \\ &\vdots \\ u_{n-2} &= a_{n-2} u_{n-1} + u_n \\ u_{n-1} &= a_{n-1} u_n. \end{aligned}$$

  In this case, there are n division steps. We have that $u_0 > u_1 > u_2 \cdots > u_n$ and $u_n = \gcd(u, v)$.

- The Fibonacci numbers are defined recursively, where $F_0 = 0$, $F_1 = 1$, and $F_n = F_{n-1} + F_{n-2}$. We can also find a closed form expression,

$$F_n = \frac{\alpha^n - \beta^n}{\sqrt{5}}$$

  where $\alpha = \frac{1+\sqrt{5}}{2}$ and $\beta = \frac{1-\sqrt{5}}{2}$.
  If we note that $|\beta| < 1$, we can rewrite the expression as

$$F_n = \frac{\alpha^n - \beta^n}{\sqrt{5}} = \frac{\alpha^n}{\sqrt{5}} - \frac{\beta^n}{\sqrt{5}} \sim \frac{\alpha^n}{\sqrt{5}} = \theta(\alpha^n).$$

2 Euclidean Algorithm: Worst-Case Analysis

First we bound n as a function of u.

Lemma 2.1 (4.2.1 in text). *Let there be integers $u > v > 0$ such that the Euclidean algorithm on input (u, v) performs n division steps. Then $u \geq F_{n+2}$ and $v \geq F_{n+1}$.*

Proof. We shall use induction to prove this lemma:

- *Base Case:* Let $n = 1$. Then $u = a_0 v$. Since $u_0 > u_1 > 0$, we must have $u_1 \geq 1$ and $u_0 \geq 2$. Therefore, $u = u_0 \geq F_3 = F_{n+2}$ and $v = u_1 \geq F_2 = F_{n+1}$, as required.

- *Inductive Step:* Suppose the lemma holds true for $1 \leq n < N$,

$$N-1 \left\{ \begin{array}{l} u_0 = a_0 u_1 + u_2 \\ u_1 = a_1 u_2 + u_3 \\ . \\ . \\ . \\ u_{N-1} = a_{N-1} u_N. \end{array} \right.$$

  By our inductive hypothesis,

$$\begin{array}{l} u_1 \geq F_{N-1+2} = F_{N+1} \\ u_2 \geq F_{N-1+1} = F_N \\ u_0 \geq u_1 + u_2 \geq F_{N+1} + F_N = F_{N+2}. \end{array}$$

  Hence, $u = u_0 \geq F_{N+2}$ and $v = u_1 \geq F_{N+1}$. The lemma follows by induction.

$\square$

Corollary 2.2. *In the Euclidean algorithm, the number of division steps is $n = O(lg\ u)$.*

Proof. According to the lemma, $u = u_0 \geq F_{n+2}$, hence

$$u \geq \frac{\alpha^{n+2} - \beta^{n+2}}{\sqrt{5}} = \Omega(\alpha^n)$$

so, taking $\log_\alpha$ of both sides, we get

$$log_\alpha u = \Omega(n)$$

and hence,

$$n = O(\lg u).$$

$\square$

Observation 2.3 (Exercise 4.5 in text). *For every i satisfying $1 \leq i \leq n-1$, we have that*

$$a_i a_{i+1} \cdots a_{n-1} \leq u_i.$$

Proof. We can prove this by using induction from $i = n-1$ down to 0.

- *Base Case:* If $i = n-1$, then $u_{n-1} = a_{n-1} u_n$, so, clearly $a_{n-1} \leq u_{n-1}$.

- *Inductive Step:* Suppose $a_{i+1} a_{i+2} \cdots a_{n-1} \leq u_{i+1}$. We know that

$$u_i = a_i u_{i+1} + u_{i+2} \geq a_i u_{i+1}$$

  and, using our inductive hypothesis,

$$u_i \geq a_i u_{i+1} \geq a_i a_{i+1} a_{i+2} \cdots a_{n-1},$$

  as required.

$\square$

Corollary 2.4. *The bit complexity of the Euclidean algorithm is* $O((\lg u)(\lg v))$.

Proof. The bit complexity can be written

$$\begin{aligned} O\left(\sum_{0\le i\le n-1}(\lg a_i)(\lg u_{i+1})\right) &= O\left((\lg v)\sum_{0\le i\le n-1}\lg a_i\right) \\ &= O\left((\lg v)\left(n+\sum_{0\le i\le n-1} log_2 a_i\right)\right) \\ &= O\left((\lg v)\left(\lg u + log_2 \prod_{0\le i\le n-1} a_i\right)\right) \\ &= O\left((\lg v)(\lg u + log_2 u)\right) \\ &= O((\lg v)(\lg u)). \end{aligned}$$

$\square$

3 Extended Euclidean Algorithm

Theorem 3.1. *Suppose that* $u, v, a, b, c, d \in \mathbb{Z}$, $M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, *and* $M \cdot \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}$.
If $det(M) = \pm 1$, *then* $\gcd(u, v) = \gcd(x, y)$.

Proof. We shall consider the two values for $\det m$ separately:

- First, suppose $det(M) = 1$. Then $ad - bc = 1$. Also, $x = au + bv$ and $y = cu + dv$. Clearly, $\gcd(u, v) \mid \gcd(x, y)$.

  Now, $M^{-1} = \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$, so $M^{-1} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix}$.

  Since $\det M^{-1} = 1$, the same argument yields $\gcd(x, y) \mid \gcd(u, v)$. Hence, $\gcd(x, y) = \gcd(u, v)$.

- Now consider the case where $det(M) = -1$ and let $\widetilde{M} = \begin{bmatrix} -a & -b \\ c & d \end{bmatrix}$.

  Then $\widetilde{M} \cdot \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -x \\ y \end{bmatrix}$ and $det(\widetilde{M}) = 1$, so $\gcd(u, v) = \gcd(-x, y) = \gcd(x, y)$.

$\square$

The Euclidean algorithm maps $(u_i, u_{i+1}) \longrightarrow (u_{i+1}, u_{i+2})$ by computing

$$u_i = a_i u_{i+1} + u_{i+2}.$$

If we consider the matrix multiplication

$$\begin{bmatrix} a_i & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} u_{i+1} \\ u_{i+2} \end{bmatrix} = \begin{bmatrix} u_i \\ u_{i+1} \end{bmatrix},$$

or the matrix multiplication

$$\begin{bmatrix} 0 & 1 \\ 1 & -a_i \end{bmatrix} \begin{bmatrix} u_i \\ u_{i+1} \end{bmatrix} = \begin{bmatrix} u_{i+1} \\ u_{i+2} \end{bmatrix},$$

we can see that they both preserve the greatest common divisor, since

$$\det\left(\begin{bmatrix} a_i & 1 \\ 1 & 0 \end{bmatrix}\right) = \det\left(\begin{bmatrix} 0 & 1 \\ 1 & -a_i \end{bmatrix}\right) = -1.$$

Now, if we define

$$M_k = \begin{bmatrix} a_0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a_1 & 1 \\ 1 & 0 \end{bmatrix} \cdots \begin{bmatrix} a_k & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} b_k & c_k \\ d_k & e_k \end{bmatrix},$$

we can see that, since $\det(M_k) = \pm 1$,

$$\begin{bmatrix} u_0 \\ u_1 \end{bmatrix} = M_k \begin{bmatrix} u_{k+1} \\ u_{k+2} \end{bmatrix}$$

and

$$\begin{bmatrix} u_{k+1} \\ u_{k+2} \end{bmatrix} = M_k \begin{bmatrix} u_0 \\ u_1 \end{bmatrix}.$$

We can define

$$M_k^{-1} = (-1)^{k+1} \begin{bmatrix} e_k & -c_k \\ -d_k & b_k \end{bmatrix}$$

And, applying it to the two numbers we wish to find the greatest common divisor of, we get

$$\begin{bmatrix} u_n \\ 0 \end{bmatrix} = M_{n-1}^{-1} \begin{bmatrix} u_0 \\ u_1 \end{bmatrix}$$

or, more specifically,

$$\gcd(u,v) = u_n = (-1)^n e_{n-1} u_0 + (-1)^{n+1} c_{n-1} u_1.$$

Theorem 3.2 (4.3.1 in text). *Let $u, v, c \in \mathbb{Z}$. Then $au + bv = c$ has a solution $a, b \in \mathbb{Z}$ if and only if $\gcd(u,v) \mid c$.*

Proof. Assume $\gcd(u,v) \mid c$. Then $c = \gcd(u,v)k$ for some $k \in \mathbb{Z}$, so we can just multiply the above equation by k, getting

$$\gcd(u,v)k = c = \left((-1)^n e_{n-1} k\right) u + \left((-1)^{n+1} c_{n-1} k\right) v.$$

Now, assume $au + bv = c$ has a solution $a, b \in \mathbb{Z}$. Clearly $\gcd(u,v) \mid au$ and $\gcd(u,v) \mid bv$, hence $\gcd(u,v)$ must divide their sum, c. $\square$

Extended Euclid(u, v)

$M \leftarrow \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

$u' \leftarrow u$

$v' \leftarrow v$

$i \leftarrow 0$

while $(v' \neq 0)$ **do**

  $q \leftarrow \left\lfloor \frac{u'}{v'} \right\rfloor$

  $r \leftarrow u' - qv'$

  $M \leftarrow M \begin{bmatrix} q & 1 \\ 1 & 0 \end{bmatrix}$

  $(u', v') \leftarrow (v', r)$

  $i \leftarrow i + 1$

return $(u', (-1)^i M_{22}, (-1)^{i+1} M_{12})$

Figure 1: Pseudocode for the Extended Euclidean algorithm

3.1 The Extended Euclidean Algorithm

The pseudocode for the Extended Euclidean algorithm is in Figure 3.1.

Example 3.3. $u_0 = 216$, $u_1 = 183$, $n = 5$, and $(a_0, a_1, a_2, a_3, a_4) = (1, 5, 1, 1, 5)$.

$$u_0 = 216 \quad M_{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$u_1 = 183 \quad M_0 = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

$$u_2 = 33 \quad M_1 = \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 5 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 6 & 1 \\ 5 & 1 \end{bmatrix}$$

$$u_3 = 18 \quad M_2 = \begin{bmatrix} 6 & 1 \\ 5 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 7 & 6 \\ 6 & 5 \end{bmatrix}$$

$$u_4 = 15 \quad M_3 = \begin{bmatrix} 7 & 6 \\ 6 & 5 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 13 & 7 \\ 11 & 6 \end{bmatrix}$$

$$u_5 = 3 \quad M_4 = \begin{bmatrix} 13 & 7 \\ 11 & 6 \end{bmatrix} \begin{bmatrix} 5 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 72 & 13 \\ 61 & 11 \end{bmatrix}$$

$$u_6 = 0 \quad M_4^{-1} = \begin{bmatrix} -11 & 13 \\ 61 & -72 \end{bmatrix}$$

And, we have our answer:

$$(-11) \cdot 216 + (13) \cdot 183 = 3.$$

Corollary 3.4 (4.3.3 in text). *We can find integers a and b such that $au + bv = \gcd(u, v)$ in time $O((\lg u)(\lg v))$.*

4 Next Time

The next class will cover Section 4.4 (Continuants) and Section 4.5 (Continued Fractions) in the text.