Summary on Lecture 3, April 4, 2018

We continue with Recurrence Relations

Fibonacci numbers again: nontrivial application. Now we denote by F_n the Fibonnaci numbers defined above, i.e. $F_0 = 0$, $F_1 = 1$ and $F_n = F_{n-1} + F_{n-2}$ for $n \geq 2$. Let $\alpha = \frac{1+\sqrt{5}}{2}$. Recall the following property we proved last time:

Lemma 1. $F_n > \alpha^{n-2}$ for $n \geq 3$.

Let m, k be positive integers, $k \geq 2$, and we look at the division:

$$m = q \cdot k + r, \quad 0 \leq r < k.$$

Recall that a key to compute $\gcd(m, k)$ is the identity $\gcd(m, k) = \gcd(k, r)$. We organize the Euclidian Algorithm as follows to match the notations from the book.

Let $r_0 = m$, $r_1 = k$. Then we have the divisions:

$$\begin{array}{lcll} r_0 & = & q_1 r_1 + r_2 & 0 \leq r_2 < r_1 \\ r_1 & = & q_2 r_2 + r_3 & 0 \leq r_3 < r_2 \\ r_2 & = & q_3 r_3 + r_4 & 0 \leq r_4 < r_3 \\ \cdots & & \cdots & \cdots \\ r_{n-2} & = & q_{n-1} r_{n-1} + r_n & 0 \leq r_n < r_{n-1} \\ r_{n-1} & = & q_n r_n & \end{array} \tag{1}$$

Then we have the sequence of identities:

$$\gcd(m, k) = \gcd(r_0, r_1) = \gcd(r_1, r_2) = \gcd(r_2, r_3) = \cdots = \gcd(r_{n-1}, r_n) = \gcd(r_n, 0) = r_n.$$

We notice that we have performed n divisions, and every quotient $q_i \geq 1$ for all $i = 1, 2, \ldots, n-1$. Then the $r_{n-1} = q_n r_n$ and $r_n < r_{n-1}$ imply that $q_n \geq 2$.

Now we examine the remainders $r_n, r_{n-1}, \ldots, r_2, r_1$ (here $r_1 = k$). We have:

$$\begin{array}{lllcl} r_n > 0, \text{ i.e. } r_n \geq 1 \text{ thus } r_n \geq F_2 = 1 & \text{i.e.} & r_n & \geq & F_2 \\ q_n \geq 2 \text{ and } r_n \geq 1 \text{ thus } r_{n-1} = q_n r_n \geq 2 \cdot 1 = 2 = F_3 & \text{i.e.} & r_{n-1} & \geq & F_3 \\ r_{n-2} = q_{n-1} r_{n-1} + r_n \geq 1 \cdot r_{n-1} + r_n \geq F_2 + F_3 = F_4 & \text{i.e.} & r_{n-2} & \geq & F_4 \\ r_{n-3} = q_{n-2} r_{n-2} + r_{n-1} \geq 1 \cdot r_{n-2} + r_{n-1} \geq F_3 + F_4 = F_5 & \text{i.e.} & r_{n-3} & \geq & F_5 \\ \cdots\cdots\cdots\cdots\cdots & \cdots\cdots & & & \\ r_2 = q_3 r_3 + r_4 \geq 1 \cdot r_3 + r_4 \geq F_{n-1} + F_{n-2} = F_n & \text{i.e.} & r_2 & \geq & F_n \\ r_1 = q_2 r_2 + r_3 \geq 1 \cdot r_2 + r_3 \geq F_n + F_{n-1} = F_{n+1} & \text{i.e.} & r_1 & \geq & F_{n+1} \end{array}$$

Since $k = r_1$, we obtain $k \geq F_{n+1}$, $m \geq k \geq 2$. Lemma 1 then implies that

$$k \geq F_{n+1} \geq \alpha^{n+1-2} = \alpha^{n-1}, \quad \text{or} \quad \log_{10} k \geq (n-1) \log_{10} \alpha$$

Then we have that $\log_{10} \alpha = \log_{10}(\frac{1+\sqrt{5}}{2}) = 0.208... > 0.2 = \frac{1}{5}$, i.e., $\log_{10} k \geq \frac{n-1}{5}$. This means that if k is such that $10^{s-1} \leq k < 10^s$, then

$$s = \log_{10} 10^s > \log_{10} k \geq \frac{n-1}{5}, \quad \text{or} \quad n < 5s + 1.$$

We proved the following result.

Theorem 3. Let $m \geq k \geq 2$. Assume k has at most s digits, (i.e., $10^{s-1} \leq k < 10^s$). Then the Euclidian Algorithm requires at most $5s$ divisions to compute $\gcd(m, k)$.

Example: legal arithmetic expressions without parenthesis. In most computing languages, it important to use "legal arithmetic expressions without parenthesis". These expressions are made up out of the digits 0,1,. . . , 9 and binary symbols $+, *, /$. For example, the expressions $7+8$, $5+7*3$, $33*7+4+6*4$ are legal expressions, and the expressions $/7+8$, $5+7*3+$, $33*7+/4+6*4$ are not.

We denote by a_n the number of legal expressions of length n. Then $a_1 = 10$ since the only legal expressions of length 1 are the digits $0, 1, \ldots, 9$. Then $a_2 = 100$ which accounts for the expressions 00, 01, $\ldots$, 99.

Let $n \geq 3$. We observe:

(1) Let x be an arithmetic legal expression of $(n-1)$ symbols. Then the last symbol must be a digit. We add one more digit to the right of x and obtain $10x$ more legal expressions of the length n.

(2) Let y be an arithmetic legal expression of $(n-2)$ symbols. Then we can add to the right of y one of the following 29 2-symbol expressions: $+0$, $+1$, $\ldots$, $+9$, $*0$, $*1$, $\ldots$, $*9$, $/1$, $\ldots$, $/9$ (no division by 0 is allowed).

We obtain the recurrence relation: $a_1 = 10$, $a_2 = 100$, $a_n = 10a_{n-1} + 29a_{n-2}$ for $n \geq 3$. We notice that $a_0 = 0$, indeed, $100 = a_2 = 10 \cdot a_1 + 29 \cdot a_0 = 10 \cdot 10 + 29 \cdot a_0$. i.e., $a_0 = 0$.

Exercise: Find a closed formula for the recurrence relation: $a_0 = 0$, $a_1 = 10$, $a_n = 10a_{n-1} + 29a_{n-2}$, $n \geq 2$.

Example. We would like to find a number of binary sequences of the length n without any consecutive 0's.

Let a_n denote the number of such sequences of length $n \geq 1$. Clearly, if $n = 1$, we have 0, 1, i.e., $a_1 = 2$, if $n = 2$, we have the sequences 01, 10, 11, i.e., $a_2 = 3$.

Let $n \geq 3$. Let $x_1 \cdots x_{n-2}x_{n-1}x_n$ be a sequence like that. There are two cases:

(1) The last symbol $x_n = 1$. Then the sequence $x_1 \cdots x_{n-2}x_{n-1}$ has no consecutive 0's.

(2) The last symbol $x_n = 0$. Then $x_{n-1} = 1$, and the sequence $x_1 \cdots x_{n-2}$ has no consecutive 0's.

Thus we conclude that $a_n = a_{n-1} + a_{n-2}$. Also we notice that the initial conditions $a_1 = 2$, $a_2 = 3$ could be replaced by $a_0 = 1$, $a_1 = 2$. Then $a_2 = a_1 + a_0 = 3$.

Exercise: Find a closed formula for the recurrence relation: $a_0 = 1$, $a_1 = 2$, $a_n = a_{n-1} + a_{n-2}$ for $n \geq 2$.