Euclidean gcd

- Given $a, b \in \mathbb{N}$ in bit representation, compute $\gcd(a,b)$.

largest $c \in \mathbb{N}$ s.t. $c|a$ & $c|b$.

Eg. $(100, 1001) = (100, 100\times 10+1) = (100, 1) = 1$.

- Euclid gives an algorithm to compute this in his book Elements (300 BC).

- The key step is based on the

quotient, remainder

Fact: If $a > b \in \mathbb{N}_{>0}$ & $a = qb + r$, with $r \in [-\frac{b}{2}, \frac{b}{2}]$, then $(a,b) = (b,r)$.

Algorithm: Use this repeatedly to compute (a,b).

It will stop as we are reducing a & b.

Analysis: We write the first step as a matrix product: $\begin{pmatrix} 0 & 1 \\ 1 & -q_1 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} b \\ r_1 \end{pmatrix}$ ($a = r_{-1}$, $b = r_0$).

- The next step gives a similar expression:

$$\begin{pmatrix} 0 & 1 \\ 1 & -q_2 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 \\ 1 & -q_1 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}.$$

- We have $b < a$, $|r_1| \le b/2$, $|r_2| \le \left|\frac{r_1}{2}\right| \le \frac{b}{2^2}$.

▷ Euclid's algorithm has $\lg b$ rounds. ↖ optimal

- It will stop at $r_i = 0$, yielding

$$\begin{pmatrix} 0 & 1 \\ 1 & -q_i \end{pmatrix} \cdots \begin{pmatrix} 0 & 1 \\ 1 & -q_1 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \gcd(a,b) \\ 0 \end{pmatrix}.$$

- The overall time complexity is:

$$\sum_{1 \le j < i} O(\lg|q_j| \cdot \lg|r_{j-1}|)$$

▷ $r_{j-2} - q_j r_{j-1} = r_j$ & $r_{-1} = a$, $r_0 = b$.

$$= O(\lg b) \cdot \sum_{1 \le j < i} \lg|q_j|$$

$$= O(\lg b) \cdot \sum_{j < i} (\lg|r_{j-2}| - \lg|r_{j-1}|)$$

$$= O(\lg b) \cdot (\lg a)$$

- This proves

Theorem: Gcd of integers a, b is computable in time $O(\lg|a| \cdot \lg|b|)$.

Moreover, the algorithm yields $u_1, u_2 \in \mathbb{N}_{>0}$ s.t. $u_1 a + u_2 b = (a, b)$, and $|u_1| < b$, $|u_2| < a$.

▷ $\langle a, b \rangle_{\mathbb{Z}} = \langle (a, b) \rangle$.

Corollary: Given coprime integers a, b, we can compute $a^{-1} \pmod b$ in time $O(\lg|a| \cdot \lg|b|)$.

▷ Similarly, $\mathrm{lcm}(a, b)$ is efficiently computable.

- Arithmetic complexity in finite fields, polynomial rings, etc. is similar except that one has to properly measure the input size.

- Polynomial arithmetic in $R[x]$:

▷ $f \pm g$ can be computed in $O(\deg f + \deg g)$ R-additions.

▷ $f \cdot g$ can be computed in $O(\deg f \cdot \deg g)$ R-operations.

▷ $f = q \cdot g + r$, with $\deg r < \deg g$, can be computed in $O(\deg q \cdot \deg g)$ R-operations.

Similarly, $\gcd(f, g)$ & $f^{-1} \bmod g$.

- When we work with rings, it is useful to factor them in some way. The most basic result is:

Chinese Remainder Theorem (~500 AD).

- For rings R_1, R_2 we define $R_1 \times R_2$ to be the ring with coordinatewise operations.

Theorem (CRT): If $a, b \in \mathbb{Z}$ are coprime, then $\mathbb{Z}/(a) \times \mathbb{Z}/(b) \cong \mathbb{Z}/(ab)$.

Moreover, the isomorphism is computable in $O(\lg|a| \cdot \lg|b|)$ time.

Proof sketch:

- Compute $u := b^{-1} \bmod a$ & $v := a^{-1} \bmod b$.
- Consider the map

$$\varphi: \mathbb{Z}/(a) \times \mathbb{Z}/(b) \to \mathbb{Z}/(ab),$$
$$(x_1, x_2) \mapsto x_1 bu + x_2 av.$$

- Note that $\varphi(x_1, x_2) \equiv x_1 \pmod a$
  & $\equiv x_2 \pmod b$.

Thus, (1) φ is a ring homomorphism
(2) φ is injective
$\Rightarrow$ (3) φ is surjective. ← Compare ring sizes (or ranks)
$\Rightarrow$ φ is a ring isomorphism.

□

CRT for polynomials: If $f, g \in \mathbb{F}[x]$ are coprime polynomials, then
$\mathbb{F}[x]/(f) \times \mathbb{F}[x]/(g) \cong \mathbb{F}[x]/(fg)$.
Moreover, the isomorphism is computable in $O(\deg f \cdot \deg g)$ $\mathbb{F}$-operations.

- The coprimality condition is required.
  eg. $\mathbb{Z}/\langle 4\rangle \ncong \mathbb{Z}/\langle 2\rangle \times \mathbb{Z}/\langle 2\rangle$.