ALGORITHMS FOR NUMBER-THEORY PROBLEMS

We will cover:

- Divisibility
  - Euclid's algorithm for the greatest common divisor (gcd)
- Congruent structures (modular arithmetic)
  - $\mathbb{Z}_n$, $\mathbb{Z}_n^*$, inverse computation
  - The Chinese remainder theorem
- APPLICATIONS
  - RSA Cryptosystem
  - Miller & Rabin's randomized PRIMALITY testing

(INTEGER) DIVISIBILITY

DEF Let $d \in \mathbb{Z} - \{0\}$, $a \in \mathbb{Z}$. We write $d \mid a$ (read d divides a) if $\exists k \in \mathbb{Z} : a = k \cdot d$ (a multiple of d)

FACT $d \mid a \Rightarrow -d \mid a$ $(k \to -k)$

DEF If $d \mid a$ and $d > 0$, d is called a divisor of a

PROPERTIES (prove as an exercise)

a. $\forall d \in \mathbb{Z} - \{0\} : d \mid 0$ $(k = 0)$

b. $(a \neq 0) \wedge (d \mid a) \Rightarrow |d| \leq |a|$

c. $(a \mid b) \wedge (b \mid a) \Rightarrow a = \pm b$ (or $|a| = |b|$)

DEF A positive integer $p > 1$ is prime if:

$$\forall d > 0 : (d \mid p) \Rightarrow (d = 1) \vee (d = p)$$

(p admits only 1 and p as divisors)

DIVISION THEOREM (no proof :-)

$$\forall a \in \mathbb{Z},\ \forall m \in \mathbb{Z}^+ :$$
$$\exists! \ q, r : a = q \cdot m + r, \text{ with}$$
$$q, r \in \mathbb{Z} \text{ and } 0 \le r < m$$

(uniqueness)

- q is called quotient (of the integer division of a by m)
- r is called reminder (of the integer division of a by m)

NOTATION : $q = \lfloor a/m \rfloor$, $r = a \bmod m$

e.g.) $-1 = (-1) \cdot m + (m-1)$ ($a = -1$)

$$\Rightarrow -1 \bmod m = m - 1$$
$$\lfloor -1/m \rfloor = -1$$

DEF

$d > 0$ is common divisor of a and b, with $a, b \in \mathbb{Z}$, if $(d \mid a) \wedge (d \mid b)$

PROPERTIES OF COMMON DIVISORS (proof: exercise)

- $(d \mid a) \wedge (d \mid b) \Rightarrow d \mid (a \pm b)$
- More generally, $\forall x, y \in \mathbb{Z}$:
  $(d \mid a) \wedge (d \mid b) \Rightarrow d \mid (ax + by)$

DEF Let $a, b \in \mathbb{Z}$, with $|a| + |b| > 0$ (at least one $\neq 0$). We set

$$\gcd(a,b) = \max\{d > 0 : (d \mid a) \wedge (d \mid b)\}$$

NOTE Since either a or b is > 0, the set $\{d > 0 : (d \mid a) \wedge (d \mid b)\}$ is finite, since $d \leq |a|$ if $a \neq 0$ or $d \leq |b|$ if $b \neq 0$

CONVENTION ($|\{d > 0 : d \mid 0\}| \to \infty$)

We set $\gcd(0,0) = 0$

PROPERTIES OF gcd (prove as exercise)

1. If $a, b \neq 0$: $1 \leq \gcd(a,b) \leq \min\{|a|, |b|\}$
2. $\gcd(a,0) = |a|$
3. $\gcd(a,b) = \gcd(b,a) = \gcd(-a,b) = \gcd(|a|,|b|)$ (symmetric, sign insensitive)
4. $\gcd(a, ka) = |a| \quad \forall a, k \in \mathbb{Z}$

HOME

Important characterization of $\gcd(a,b)$

THEOREM (Bezout identity) (1779)

Let $|a| + |b| > 0$. Then

$$\gcd(a,b) = \min\{d > 0 \mid \exists x, y \in \mathbb{Z} \mid d = ax + by\}$$

In words: $\gcd(a,b)$ is the minimum positive linear combination of a and b with integer coefficients

important characterization of gcd with many applications

CASA

PROOF Let $S = \min\{d>0 \mid \exists x,y\in\mathbb{Z} \mid d=ax+by\}$

We will prove $\gcd(a,b) \leq S$ and $\gcd(a,b) \geq S$

$\leq$ By the property of common divisors we know that $\gcd(a,b) \mid ax+by \quad \forall x,y \in \mathbb{Z}$

Since $S = \min\{d>0 \mid \exists x,y\in\mathbb{Z} \mid d=ax+by\} = a\bar{x}+b\bar{y}$ we have $\gcd(a,b) \mid S$. Also $S>0$, thus $0 < \gcd(a,b) \leq |S| = S$

$\geq$ We'll prove that $S \mid ax+by \quad \forall x,y\in\mathbb{Z}$

If this is the case: $S \mid a$ $(x=1, y=0)$ and $S \mid b$ $(x=0, y=1)$. Thus S is a common divisor of a and b, therefore $\gcd(a,b) \geq S$.

→ Say that $S = a\bar{x}+b\bar{y}$. Let $C = ax+by$, for arbitrary $x,y \in \mathbb{Z}$.

Apply the division theorem to (C,S):

$$\exists q,r : C = ax+by = q\cdot S + r \qquad 0 \leq r < S$$

We have that $r = C - qS = ax+by - q(a\bar{x}+b\bar{y})$

$= a(x-q\bar{x}) + b(y-q\bar{y}) = ax'' + by''$

Then $r=0$, or otherwise we would have found $0<r=ax''+by''<s$, which contradicts $s=\min\{d>0 \mid \exists x,y\in\mathbb{Z} \mid d=ax+by\}$.

Since $r=0$ we have that $s \mid ax+by$!

Efficient computation of $\gcd(a,b)$

From now on, we restrict to the case $a,b\geq 0$ ($\gcd(a,b)=\gcd(|a|,|b|)$)

Euclid's algorithm (300 BC):

```
EUCLID (a, b)
if (b=0) then return a
return EUCLID(b, a mod b)
```

Simple recursive algorithm.

CORRECTNESS

(1) EUCLID (a,b) terminates.

In each call the second parameter decreases (from b to $a \bmod b < b$) $\Rightarrow$ No more than $b+1$ calls.

NOTE. If $a<b$, EUCLID (a,b) calls EUCLID (b,a). In subsequent calls the first parameter is always $\geq$ than the second.

② EUCLID(a, b) returns gcd(a, b)

This is true for $b = 0$, since $\gcd(a, 0) = a$. It remains to prove that when $b > 0$,

$$\gcd(a, b) = \gcd(b, a \bmod b)$$
$$\leq \quad \geq$$

$\leq$ By Bezout's identity:

① $\gcd(a, b) = \min\{d > 0 \mid \exists x, y \in \mathbb{Z} : d = ax + by\}$

② $\exists x', y' : \gcd(b, a \bmod b) = bx' + (a \bmod b)y'$

Let $a = qb + a \bmod b$ (by division theorem)

Then

$a \bmod b = a - qb$, hence from ②:

$$\exists x', y' : 0 < \gcd(b, a \bmod b) = bx' + (a - qb)y'$$
$$= ay' + (x' - qy')b$$

We have written $\gcd(b, a \bmod b)$ as a positive integer linear combination of a and b. Therefore, from ①

$$\gcd(a, b) \leq \gcd(b, a \bmod b)$$

// Let $d' = \gcd(b, a \bmod b)$. We have $d' \mid b$ and $d' \mid a \bmod b$

hence $d' \mid x \cdot b + y \cdot a \bmod b \quad \forall x, y \in \mathbb{Z}$

Recall that $a = q b + a \bmod b$

and set $x = q$ and $y = 1$:

$$d' \mid q b + a \bmod b = a$$

But then

$$(d' \mid b) \wedge (d' \mid a) \Rightarrow \gcd(a, b) \geq d' = \gcd(b, a \bmod b)$$

RUNNING TIME

Let us evaluate the number of recursive calls performed by EUCLID(a,b) in the worst case.

We can assume that $a > b > 0$.

Otherwise:

1. $b = 0 \Rightarrow$ one call (base case)
2. $a = b \Rightarrow$ two calls:

   EUCLID(a,b) = EUCLID(b,b) → EUCLID(b, 0)
3. $a < b \Rightarrow$ EUCLID(a,b) → EUCLID(b,a)

   and the condition is established by swapping a and b (+ 1 extra call)

The analysis relies on Fibonacci numbers

$F_1 = F_2 = 1$, $F_k = F_{k-1} + F_{k-2}$ $k > 2$

1, 1, 2, 3, 5, 8 ...

It holds:

$$F_k = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^k - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^k$$

$$\geq \frac{1}{2\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^k \qquad < 0$$

(exponential in k $\left(\frac{1+\sqrt{5}}{2} \simeq 1.62\right)$)

We prove that for EUCLID to perform many recursive calls, the inputs must be very large (thus, the number of calls is "small" w.r.t. the size of the inputs)

LEMMA For $a > b > 0$, if EUCLID(a,b) makes k calls overall (including the external call), then

$$a \geq F_{k+2} \text{ and } b \geq F_{k+1}$$

We will get a bound on the number of calls for a specific input size by inverting the inequality (solving w.r.t. k). (later)

PROOF By induction on K.

BASE $K=1$: Since $b>0 \Rightarrow b \geq 1 = F_2$

Also, $a > b \Rightarrow a > 1 \Rightarrow a \geq 2 = F_3$

HP True for $K-1$ calls, $K>1$

TH If EUCLID(a,b) makes K calls, then EUCLID$(b, a \bmod b)$ (first recursive call) makes $K-1$ calls.

By HP: $b \geq F_{(K-1)+2} = F_{K+1}$, (TH for b)

$a \bmod b \geq F_{(K-1)+1} = F_K$

However $a = q \cdot b + a \bmod b$, with $q \geq 1$ $(a > b)$. Thus:

$a \geq b + a \bmod b \geq F_{K+1} + F_K = F_{K+2}$ (TH for a)

Let K: $F_K < b \leq F_{K+1}$. Then K is an upper bound to the number of calls performed by EUCLID(a,b)

We have $b > F_K \geq \frac{1}{2\sqrt{5}} \left(\frac{1+\sqrt{5}}{2}\right)^K \Rightarrow$

$\log_{\frac{1+\sqrt{5}}{2}} b > K + c \quad \left(c = \log_{\frac{1+\sqrt{5}}{2}} 1/(2\sqrt{5}) < 0\right)$

$\Rightarrow K < \log_{\frac{1+\sqrt{5}}{2}} b - c \Rightarrow K = O(\log b)$

Since $k = O(\log b) = O(|\langle a, b \rangle|)$

EUCLID (a, b) makes at most a LINEAR number of calls in the size $|\langle a, b \rangle|$.

Each call must compute $a \bmod b$ which takes time $O(|\langle a, b \rangle|^2)$ (exercise: implement integer division) (in class)

$\Rightarrow$ $T_{EUCLID}(|\langle a, b \rangle|) = O(|\langle a, b \rangle|^3)$!

EXTENDED EUCLID ALGORITHM

EUCLID (a, b) can be easily extended to compute the integer coefficients of Bezout's identity:

$$\gcd(a, b) = a x' + b y', \quad x', y' \in \mathbb{Z}$$

These coefficients have important applications (e.g., RSA).

Extension:

BASE CASE $b = 0 \Rightarrow \gcd(a, b) = a$

$\Rightarrow x' = 1, \; y' = 0$ $\left(a = a \cdot 1 + b \cdot 0\right)$

REMARK: any other choice for y' would be fine: coefficients are not unique

SUBSTRUCTURE: Suppose that we are given:

$$\gcd(b, a \bmod b) = b\bar{x} + (a \bmod b)\bar{y}$$

Since $\gcd(a,b) = \gcd(b, a \bmod b)$ we rewrite $b\bar{x} + (a \bmod b)\bar{y}$ as a linear combination of a and b.

We know that $a = \lfloor a/b \rfloor \cdot b + a \bmod b$ (division theorem)

Thus $a \bmod b = (a - \lfloor a/b \rfloor b)$

Substituting:

$$\gcd(a,b) = \gcd(b, a \bmod b) =$$
$$= b\bar{x} + (a - \lfloor a/b \rfloor b)\bar{y} =$$
$$= a\bar{y} + b(\bar{x} - \lfloor a/b \rfloor \cdot \bar{y})$$

($\bar{y}$ = x', $\bar{x} - \lfloor a/b \rfloor \cdot \bar{y}$ = y')

The algorithm immediately follows

EXTENDED_EUCLID (a,b) (EE(a,b) for short) returns the struct $\{\gcd(a,b), (x', y')\}$

```
EE(a,b)
B if (b=0)
    then return {a, (1, 0)}
R {d, (x̄, ȳ)} ← EE(b, a mod b)
C return {d, (ȳ, x̄ − ⌊a/b⌋ ȳ)}
```

EXAMPLE

EE(24, 16) ⟶ {8, (1, −1·1) = (1, −1)}

EE(16, 8) ⟶ {8, (0, 1 − 2·0) = (0, 1)}

EE(8, 0) ⟶ {8, (1, 0)}

Indeed $\gcd(24, 16) = 8 = 24 \cdot 1 + 16 \cdot (-1)$