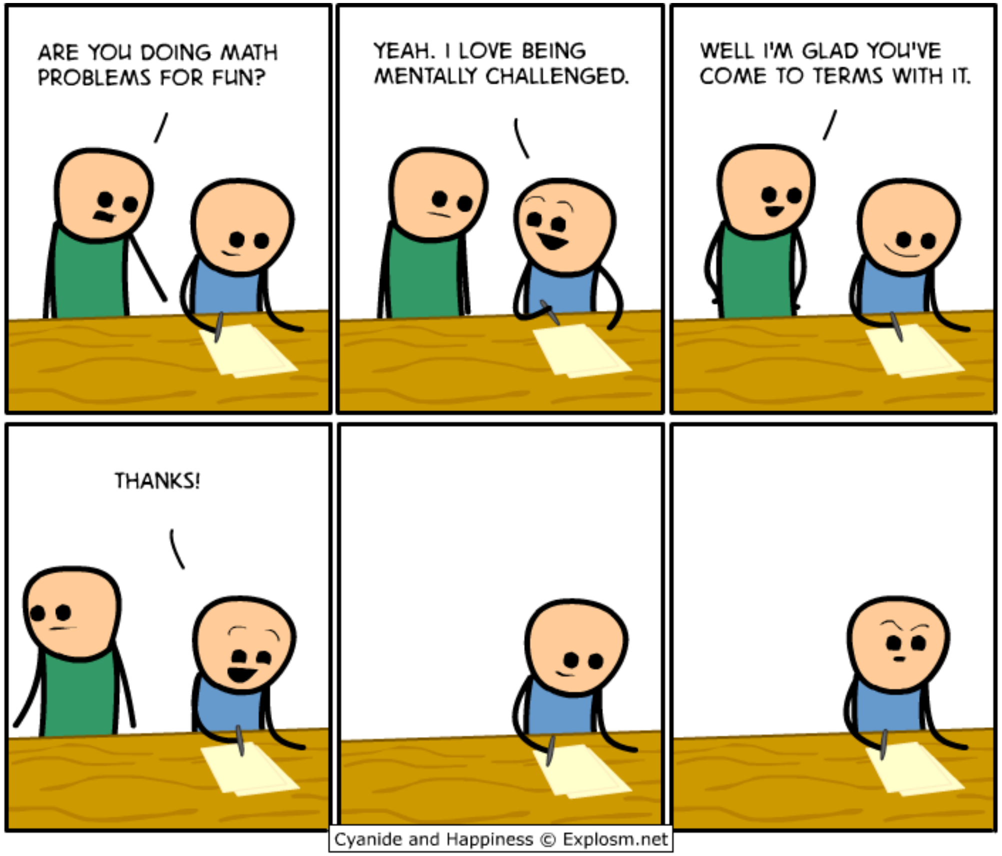
CS 13

Mathematical Foundations of Computing

CSE 311: Foundations of Computing

Lecture 13: Modular Inverses



Division

Let's get existential.
What, really, IS division?

Division

In normal arithmetic, if I multiply x * (1/x), I get back 1.

In MODULAR arithmetic, if I multiply x * ?, I get back 1.

"1/x" is the unique number that, when multiplied by x gives 1.

$7 \cdot x^{-1} \bmod 10$

$x^{-1} = 3$

Greatest Common Divisor

GCD(a, b):

Largest integer d such that $d \mid a$ and $d \mid b$

- GCD(100, 125) =
- GCD(17, 49) =
- GCD(11, 66) =
- GCD(13, 0) =
- GCD(180, 252) =

GCD and Factoring

$a = 2^3 \cdot 3 \cdot 5^2 \cdot 7 \cdot 11 = 46{,}200$

$b = 2 \cdot 3^2 \cdot 5^3 \cdot 7 \cdot 13 = 204{,}750$

$GCD(a, b) = 2^{\min(3,1)} \cdot 3^{\min(1,2)} \cdot 5^{\min(2,3)} \cdot 7^{\min(1,1)} \cdot 11^{\min(1,0)} \cdot 13^{\min(0,1)}$

Factoring is expensive!

Can we compute GCD(a,b) without factoring?

Useful GCD Fact

If a and b are positive integers, then

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

Useful GCD Fact

> If *a* and *b* are positive integers, then
> gcd(a,b) = gcd$(b, a \bmod b)$

Proof:

By definition of mod, $a = qb + (a \bmod b)$ for some integer $q = a$ div b.

Let $d = \gcd(a,b)$. Then $d|a$ and $d|b$ so $a = kd$ and $b = jd$ for some integers k and j.
Therefore $(a \bmod b) = a - qb = kd - qjd = d(k - qj)$.
So, $d \mid (a \bmod b)$ and since $d \mid b$ we must have $d \le \gcd(b, a \bmod b)$.

Now, let $e = \gcd(b, a \bmod b)$. Then $e \mid b$ and $e \mid (a \bmod b)$. It follows that $b = me$ and $(a \bmod b) = ne$ for some integers m and n. Therefore

$$a = qb + (a \bmod b) = qme + ne = e(qm+n)$$

So, $e \mid a$ and since $e \mid b$ we must have $e \le \gcd(a, b)$.

Therefore $\gcd(a, b) = \gcd(b, a \bmod b)$.

Euclid's Algorithm

If a and b are positive integers, then

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

GCD Algorithm

```
gcd(a, 0) = a
gcd(a, b) = gcd(b, a mod b)
```

```
gcd(126, 660) =
```

Euclid’s Algorithm

If *a* and *b* are positive integers, then
$$\gcd(a,b) = \gcd(b, a \bmod b)$$

GCD Algorithm

```
gcd(a, 0) = a
gcd(a, b) = gcd(b, a mod b)
```

```
gcd(126, 660) = gcd(660, 126 mod 660)
              = gcd(660, 126)
              = gcd(126, 660 mod 126)
              = gcd(126, 30)
              = gcd(30, 126 mod 60)
              = gcd(30, 6)
              = gcd(6, 30 mod 6)
              = gcd(6, 0)
              = 6
```

Euclid’s Algorithm

GCD Algorithm

```
gcd(a, 0) = a
gcd(a, b) = gcd(b, a mod b)
```

```
gcd(a, b) {
    if (b == 0) {
        return a;
    }
    else {
        return gcd(b, a mod b);
    }
}
```

Finding x & y

Bézout's Theorem

If *a* and *b* are positive integers, then there exist integers $\mathbf{x}_{(a,b)}$ and $\mathbf{y}_{(a,b)}$ such that

$$\gcd(a, b) = a\mathbf{x}_{(a,b)} + b\mathbf{y}_{(a,b)}$$

GCD Algorithm

```
gcd(a, 0) = a
gcd(a, b) = gcd(b, a mod b)
```

7 mod 10

$\rightarrow 1 = 10x + 7y$

$10(x+7) + 7(y-10)$

$\rightarrow 1 \equiv_{10} 7y$

Finding x & y

Bézout's Theorem

If *a* and *b* are positive integers, then there exist integers $\mathbf{x}_{(a,b)}$ and $\mathbf{y}_{(a,b)}$ such that gcd(a, b) = a$\mathbf{x}_{(a,b)}$ + b$\mathbf{y}_{(a,b)}$

GCD Algorithm

```
gcd(a, 0) = a
gcd(a, b) = gcd(b, a mod b)
```

Case 1: gcd(a, 0) = a

gcd(a, 0) = a*$\mathbf{X}_{a,0}$ + 0*$\mathbf{Y}_{a,0}$

Finding x & y

Bézout's Theorem

If *a* and *b* are positive integers, then there exist integers $\mathbf{x}_{(a,b)}$ and $\mathbf{y}_{(a,b)}$ such that gcd(a, b) = a$\mathbf{x}_{(a,b)}$ + b$\mathbf{y}_{(a,b)}$

Case 2: `gcd(a, b) = gcd(b, a mod b)`

gcd(a, b) = a$X_{a,b}$ + b$Y_{a,b}$

GCD Algorithm

```
gcd(a, 0) = a*1 + 0*0
gcd(a, b) = gcd(b, a mod b)
```

We've figured out the answer for the "base case".

Finding x & y

Bézout's Theorem

If *a* and *b* are positive integers, then there exist integers $\mathbf{x}_{(a,b)}$ and $\mathbf{y}_{(a,b)}$ such that gcd(a, b) = a$\mathbf{x}_{(a,b)}$ + b$\mathbf{y}_{(a,b)}$

GCD Algorithm

```
gcd(a, 0) = a*1 + 0*0
gcd(a, b) = gcd(b, a mod b)
```

Case 2: `gcd(a, b) = gcd(b, a mod b)`

$$\text{gcd(a, b)} = a\mathbf{X}_{a,b} + b\mathbf{Y}_{a,b}$$
$$= \text{gcd(b, a mod b)} = ?????????$$

We're stuck. We need to find $X_{a,b}$ and $Y_{a,b}$.
We're looking for an equation with a*x + b*y. The "a mod b" doesn't belong.

$$\text{gcd(b, a mod b)} = b\mathbf{X}_{b,\text{a mod b}} + (\text{a mod b})\mathbf{Y}_{b,\text{a mod b}}$$

Division Theorem

a = b(a div b) + (a mod b)

Finding x & y

Bézout's Theorem

If *a* and *b* are positive integers, then there exist integers $\mathbf{x}_{(a,b)}$ and $\mathbf{y}_{(a,b)}$ such that gcd(a, b) = a$\mathbf{x}_{(a,b)}$ + b$\mathbf{y}_{(a,b)}$

GCD Algorithm

```
gcd(a, 0) = a*1 + 0*0
gcd(a, b) = gcd(b, a mod b)
```

Case 2: `gcd(a, b) = gcd(b, a mod b)`

$$\text{gcd(a, b)} = aX_{a,b} + bY_{a,b}$$
$$= \text{gcd(b, a mod b)} = ?????????$$

We're stuck. We need to find $X_{a,b}$ and $Y_{a,b}$.
We're looking for an equation with a*x + b*y. The "a mod b" doesn't belong.

$$\text{gcd(b, a mod b)} = bX_{b,a \bmod b} + (a \bmod b)Y_{b,a \bmod b}$$
$$= bX_{b,a \bmod b} + (a - b(a \text{ div } b))Y_{b,a \bmod b}$$

Division Theorem

$$a = b(a \text{ div } b) + (a \bmod b)$$
$$(a \bmod b) = a - b(a \text{ div } b)$$

Finding x & y

Bézout's Theorem

If *a* and *b* are positive integers, then there exist integers $\mathbf{x}_{(a,b)}$ and $\mathbf{y}_{(a,b)}$ such that gcd(a, b) = a$\mathbf{x}_{(a,b)}$ + b$\mathbf{y}_{(a,b)}$

GCD Algorithm

```
gcd(a, 0) = a*1 + 0*0
gcd(a, b) = gcd(b, a mod b)
```

Case 2: gcd(a, b) = gcd(b, a mod b)

gcd(a, b) = a$X_{a,b}$ + b$Y_{a,b}$
= gcd(b, a mod b) = ?????????

gcd(b, a mod b) = b$X_{b,a \bmod b}$ + (a mod b)$Y_{b,a \bmod b}$
= b$X_{b,a \bmod b}$ + (a – b(a div b))$Y_{b,a \bmod b}$

We're still looking for $X_{a,b}$ and $Y_{a,b}$. The equation has a and b terms. Group them…

Finding x & y

Bézout's Theorem

If *a* and *b* are positive integers, then there exist integers $\mathbf{x}_{(a,b)}$ and $\mathbf{y}_{(a,b)}$ such that $\gcd(a, b) = a\mathbf{x}_{(a,b)} + b\mathbf{y}_{(a,b)}$

GCD Algorithm

```
gcd(a, 0) = a*1 + 0*0
gcd(a, b) = gcd(b, a mod b)
```

Case 2: `gcd(a, b) = gcd(b, a mod b)`

$$\gcd(a, b) = a\mathbf{X}_{a,b} + b\mathbf{Y}_{a,b}$$
$$= \gcd(b, a \bmod b) = ?????????$$

$$\gcd(b, a \bmod b) = b\mathbf{X}_{b,a \bmod b} + (a \bmod b)\mathbf{Y}_{b,a \bmod b}$$
$$= b\mathbf{X}_{b,a \bmod b} + (a - b(a \operatorname{div} b))\mathbf{Y}_{b,a \bmod b}$$

We're still looking for $X_{a,b}$ and $Y_{a,b}$. The equation has a and b terms. Group them…

$$= b\mathbf{X}_{b,a \bmod b} + a\mathbf{Y}_{b,a \bmod b} - b(a \operatorname{div} b)\mathbf{Y}_{b,a \bmod b}$$

Finding x & y

Bézout's Theorem

If *a* and *b* are positive integers, then there exist integers $\mathbf{x}_{(a,b)}$ and $\mathbf{y}_{(a,b)}$ such that gcd(a, b) = a$\mathbf{x}_{(a,b)}$ + b$\mathbf{y}_{(a,b)}$

GCD Algorithm

```
gcd(a, 0) = a*1 + 0*0
gcd(a, b) = gcd(b, a mod b)
```

Case 2: gcd(a, b) = gcd(b, a mod b)

$$\text{gcd}(a, b) = a\mathbf{X}_{a,b} + b\mathbf{Y}_{a,b}$$
$$= \text{gcd}(b, a \bmod b) = ?????????$$

$$\text{gcd}(b, a \bmod b) = b\mathbf{X}_{b,a \bmod b} + (a \bmod b)\mathbf{Y}_{b,a \bmod b}$$
$$= b\mathbf{X}_{b,a \bmod b} + (a - b(a \text{ div } b))\mathbf{Y}_{b,a \bmod b}$$

We're still looking for $X_{a,b}$ and $Y_{a,b}$. The equation has a and b terms. Group them…

$$= b\mathbf{X}_{b,a \bmod b} + a\mathbf{Y}_{b,a \bmod b} - b(a \text{ div } b)\mathbf{Y}_{b,a \bmod b}$$
$$= b(X_{b,a \bmod b} - (a \text{ div } b)\mathbf{Y}_{b,a \bmod b}) + a\mathbf{Y}_{b,a \bmod b}$$

Finding x & y

Bézout's Theorem

If *a* and *b* are positive integers, then there exist integers $\mathbf{x}_{(a,b)}$ and $\mathbf{y}_{(a,b)}$ such that gcd(a, b) = a$\mathbf{x}_{(a,b)}$ + b$\mathbf{y}_{(a,b)}$

GCD Algorithm

```
gcd(a, 0) = a*1 + 0*0
gcd(a, b) = gcd(b, a mod b)
```

Case 2: gcd(a, b) = gcd(b, a mod b)

gcd(a, b) = a$X_{a,b}$ + b$Y_{a,b}$
= gcd(b, a mod b) = ?????????

gcd(b, a mod b) = b$X_{b,a \bmod b}$ + (a mod b)$Y_{b,a \bmod b}$
= b$X_{b,a \bmod b}$ + (a – b(a div b))$Y_{b,a \bmod b}$

We're still looking for $X_{a,b}$ and $Y_{a,b}$. The equation has a and b terms. Group them…

= b$X_{b,a \bmod b}$ + a$Y_{b,a \bmod b}$ – b(a div b)$Y_{b,a \bmod b}$
= b($X_{b,a \bmod b}$ – (a div b)$Y_{b,a \bmod b}$) + a$Y_{b,a \bmod b}$
= a$Y_{b,a \bmod b}$ + b($X_{b,a \bmod b}$ – (a div b)$Y_{b,a \bmod b}$)

gcd(b, a mod b) = a$Y_{b,a \bmod b}$ + b($X_{b,a \bmod b}$ – (a div b)$Y_{b,a \bmod b}$)

Finding x & y

Bézout's Theorem

If *a* and *b* are positive integers, then there exist integers $\mathbf{x}_{(a,b)}$ and $\mathbf{y}_{(a,b)}$ such that gcd(a, b) = a$\mathbf{x}_{(a,b)}$ + b$\mathbf{y}_{(a,b)}$

GCD Algorithm

```
gcd(a, 0) = a*1 + 0*0
gcd(a, b) = gcd(b, a mod b)
```

Case 2: `gcd(a, b) = gcd(b, a mod b)`

$$
\begin{aligned}
\mathrm{gcd}(a, b) &= a X_{a,b} + b Y_{a,b} \\
&= \mathrm{gcd}(b, a \bmod b) \\
&= a Y_{b, a \bmod b} + b(X_{b, a \bmod b} - (a \operatorname{div} b) Y_{b, a \bmod b})
\end{aligned}
$$

Finding x & y

Bézout's Theorem

If *a* and *b* are positive integers, then there exist integers $\mathbf{x}_{(a,b)}$ and $\mathbf{y}_{(a,b)}$ such that gcd(a, b) = a$\mathbf{x}_{(a,b)}$ + b$\mathbf{y}_{(a,b)}$

GCD Algorithm

```
gcd(a, 0) = a
gcd(a, b) = gcd(b, a mod b)
```

Case 2: `gcd(a, b) = gcd(b, a mod b)`

$$\begin{aligned} \texttt{gcd(a, b)} &= \texttt{a}\mathbf{X}_{\texttt{a,b}} + \texttt{b}\mathbf{Y}_{\texttt{a,b}} \\ &= \texttt{gcd(b, a mod b)} \\ &= \texttt{a}\mathbf{Y}_{\texttt{b,a mod b}} + \texttt{b}(\mathbf{X}_{\texttt{b,a mod b}} - (\texttt{a div b})\mathbf{Y}_{\texttt{b,a mod b}}) \end{aligned}$$

EGCD Algorithm

$$\texttt{egcd(a, 0)} = \texttt{a*1 + 0*0}$$

$$\texttt{egcd(a, b)} = \texttt{a*}\mathbf{Y}_{\texttt{b,a mod b}} + \texttt{b*}(\mathbf{X}_{\texttt{b,a mod b}} - (\texttt{a div b})\mathbf{Y}_{\texttt{b,a mod b}})$$

Finding x & y

GCD Algorithm

```
gcd(a, 0) = a
gcd(a, b) = gcd(b, a mod b)
```

EGCD Algorithm

egcd(a, 0) = a*1 + 0*0

egcd(a, b) = $a*Y_{b,a \bmod b} + b*(X_{b,a \bmod b} - (a \text{ div } b)Y_{b,a \bmod b})$

EGCD Algorithm

egcd(a, 0) = (a, 1, 0)

egcd(a, b) = $(gcd(b, a \bmod b), Y_{b,a \bmod b}, X_{b,a \bmod b} - (a \text{ div } b)*Y_{b,a \bmod b})$