Lecture 7. Outline.

1. Quickly finish isoperimetric inequality for hypercube.
2. Modular Arithmetic.
   Clock Math!!!
3. Inverses for Modular Arithmetic: Greatest Common Divisor.
   Division!!!
4. Euclid's GCD Algorithm.
   A little tricky here!

Isoperimetry.

For 3-space:

The sphere minimizes surface area to volume.

Surface Area: $4\pi r^2$, Volume: $\frac{4}{3}\pi r^3$.

Ratio: $1/3r = \Theta(V^{-1/3})$.

Graphical Analog: Cut into two pieces and find ratio of edges/vertices on small side.

Tree: $\Theta(1/|V|)$.

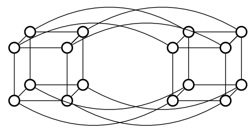
Hypercube: $\Theta(1)$.

Surface Area is roughly at least the volume!

Recursive Definition.

A 0-dimensional hypercube is a node labelled with the empty string of bits.

An n-dimensional hypercube consists of a 0-subcube (1-subcube) which is a $n-1$-dimensional hypercube with nodes labelled $0x$ ($1x$) with the additional edges $(0x, 1x)$.



Hypercube: Can't cut me!

Thm: Any subset S of the hypercube where $|S| \leq |V|/2$ has $\geq |S|$ edges connecting it to $V - S$; $|E \cap S \times (V - S)| \geq |S|$

Terminology:
$(S, V - S)$ is cut.
$(E \cap S \times (V - S))$ - cut edges.

Restatement: for any cut in the hypercube, the number of cut edges is at least the size of the small side.

Proof of Large Cuts.

Thm: For any cut $(S, V - S)$ in the hypercube, the number of cut edges is at least the size of the small side.
Proof:
Base Case: $n = 1$ V= $\{0,1\}$.
$S = \{0\}$ has one edge leaving. $|S| = \phi$ has 0.

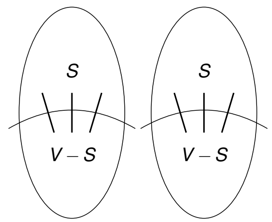
Induction Step Idea

Thm: For any cut $(S, V - S)$ in the hypercube, the number of cut edges is at least the size of the small side.

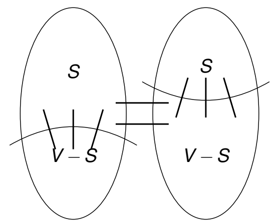
Use recursive definition into two subcubes.

Two cubes connected by edges.

Case 1: Count edges inside subcube inductively.



Case 2: Count inside and across.



Induction Step

Thm: For any cut $(S, V-S)$ in the hypercube, the number of cut edges is at least the size of the small side, $|S|$.

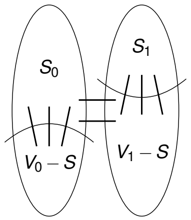
Proof: Induction Step.
Recursive definition:
$H_0 = (V_0, E_0), H_1 = (V_1, E_1)$, edges E_x that connect them.
$H = (V_0 \cup V_1, E_0 \cup E_1 \cup E_x)$
$S = S_0 \cup S_1$ where S_0 in first, and S_1 in other.

Case 1: $|S_0| \le |V_0|/2, |S_1| \le |V_1|/2$
Both S_0 and S_1 are small sides. So by induction.
Edges cut in $H_0 \ge |S_0|$.
Edges cut in $H_1 \ge |S_1|$.

Total cut edges $\ge |S_0| + |S_1| = |S|$. $\square$

Induction Step. Case 2.

Thm: For any cut $(S, V-S)$ in the hypercube, the number of cut edges is at least the size of the small side, $|S|$.
Proof: Induction Step. Case 2.



$|S_0| \ge |V_0|/2$.
Recall Case 1: $|S_0|, |S_1| \le |V|/2$
$|S_1| \le |V_1|/2$ since $|S| \le |V|/2$.
$\implies \ge |S_1|$ edges cut in E_1.
$|S_0| \ge |V_0|/2 \implies |V_0 - S| \le |V_0|/2$
$\implies \ge |V_0| - |S_0|$ edges cut in E_0.

Edges in E_x connect corresponding nodes.
$\implies = |S_0| - |S_1|$ edges cut in E_x.

Total edges cut:
$\ge |S_1| + |V_0| - |S_0| + |S_0| - |S_1| = |V_0|$
$|V_0| = |V|/2 \ge |S|$. $\square$
Also, case 3 where $|S_1| \ge |V|/2$ is symmetric.

Hypercubes and Boolean Functions.

The cuts in the hypercubes are exactly the transitions from 0 sets to 1 set on boolean functions on $\{0,1\}^n$.

Central area of study in computer science!

Yes/No Computer Programs $\equiv$ Boolean function on $\{0,1\}^n$

Central object of study.

Next Up.

Modular Arithmetic.

Clock Math

If it is 1:00 now.
What time is it in 2 hours? 3:00!
What time is it in 5 hours? 6:00!
What time is it in 15 hours? 16:00!
Actually 4:00.

16 is the "same as 4" with respect to a 12 hour clock system.
Clock time equivalent up to to addition/subtraction of 12.

What time is it in 100 hours? 101:00! or 5:00.
$101 = 12 \times 8 + 5$.
5 is the same as 101 for a 12 hour clock system.
Clock time equivalent up to addition of any integer multiple of 12.

Custom is only to use the representative in $\{12, 1, \ldots, 11\}$
(Almost remainder, except for 12 and 0 are equivalent.)

Day of the week.

Today is Monday.
What day is it a year from now? on February 6, 2018?
Number days.
0 for Sunday, 1 for Monday, . . . , 6 for Saturday.

Today: day 2.
5 days from now. day 7 or day 0 or Sunday.
25 days from now. day 27 or day 6. $27 = (7)3 + 6$
two days are equivalent up to addition/subtraction of multiple of 7.
11 days from now is day 6 which is Saturday!

What day is it a year from now?
This year is not a leap year. So 365 days from now.
Day 2+365 or day 367.
Smallest representation:
subtract 7 until smaller than 7.
divide and get remainder.
367/7 leaves quotient of 52 and remainder 3. $365 = 7(52) + 3$
or February 6, 2018 is a Wednesday.

Years and years...

80 years from now? 20 leap years. 366×20 days
60 regular years. 365×60 days
Today is day 2.
It is day $2 + 366 \times 20 + 365 \times 60$. Equivalent to?

Hmm.
What is remainder of 366 when dividing by 7? $52 \times 7 + 2$.
What is remainder of 365 when dividing by 7? 1
Today is day 2.
Get Day: $2 + 2 \times 20 + 1 \times 60 = 102$
Remainder when dividing by 7? $102 = 14 \times 7 + 4$.
Or February 7, 2096 is Thursday!

Further Simplify Calculation:
20 has remainder 6 when divided by 7.
60 has remainder 4 when divided by 7.
Get Day: $2 + 2 \times 6 + 1 \times 4 = 18$.
Or Day 4. February 6, 2095 is Thursday.

"Reduce" at any time in calculation!

Modular Arithmetic: refresher.

x **is congruent to** y **modulo** m or "$x \equiv y \pmod{m}$"
if and only if $(x - y)$ is divisible by m.
...or x and y have the same remainder w.r.t. m.
...or $x = y + km$ for some integer k.

Mod 7 equivalence classes:
$\{\dots, -7, 0, 7, 14, \dots\}$ $\{\dots, -6, 1, 8, 15, \dots\}$ ...

Useful Fact: Addition, subtraction, multiplication can be done with any equivalent x and y.

or " $a \equiv c \pmod{m}$ and $b \equiv d \pmod{m}$
$\implies a + b \equiv c + d \pmod{m}$ and $a \cdot b = c \cdot d \pmod{m}$"

Proof: If $a \equiv c \pmod{m}$, then $a = c + km$ for some integer k.
If $b \equiv d \pmod{m}$, then $b = d + jm$ for some integer j.
Therefore, $a + b = c + d + (k + j)m$ and since $k + j$ is integer.
$\implies a + b \equiv c + d \pmod{m}$. $\square$

Can calculate with representative in $\{0, \dots, m-1\}$.

Notation

$x \pmod{m}$ or $\bmod(x, m)$
- remainder of x divided by m in $\{0, \dots, m-1\}$.

$\bmod(x, m) = x - \lfloor \frac{x}{m} \rfloor m$

$\lfloor \frac{x}{m} \rfloor$ is quotient.

$\bmod(29, 12) = 29 - (\lfloor \frac{29}{12} \rfloor) \times 12 = 29 - (2) \times 12 = \cancel{4} = 5$

Work in this system.
$a \equiv b \pmod{m}$.
Says two integers a and b are equivalent modulo m.

Modulus is m

$6 \equiv 3 + 3 \equiv 3 + 10 \pmod{7}$.

$6 = 3 + 3 = 3 + 10 \pmod{7}$.

Generally, not $6 \pmod{7} = 13 \pmod{7}$.
But probably won't take off points, still hard for us to read.

Inverses and Factors.

Division: multiply by multiplicative inverse.

$$2x = 3 \implies (\frac{1}{2}) \cdot 2x = (\frac{1}{2}) \cdot 3 \implies x = \frac{3}{2}.$$

Multiplicative inverse of x is y where $xy = 1$;
1 **is multiplicative identity element.**

In modular arithmetic, 1 is the multiplicative identity element.

Multiplicative inverse of x **mod** m is y with $xy = 1 \pmod{m}$.

For 4 modulo 7 inverse is 2: $2 \cdot 4 \equiv 8 \equiv 1 \pmod{7}$.

Can solve $4x = 5 \pmod{7}$.
$2 \cdot 4x = 2 \cdot 5 \pmod{7}$
$8x = 10 \pmod{7}$
$x = 3 \pmod{7}$
Check! $4(3) = 12 = 5 \pmod{7}$.

$x = 3 \pmod{7}$ :: Check! $4(3) = 12 = 5 \pmod{7}$.

For 8 modulo 12: no multiplicative inverse!

"Common factor of 4" $\implies$
$8k - 12\ell$ is a multiple of four for any ℓ and k $\implies$
$8k \not\equiv 1 \pmod{12}$ for any k.

Greatest Common Divisor and Inverses.

Thm:
If greatest common divisor of x and m, $\gcd(x, m)$, is 1, then x has a multiplicative inverse modulo m.

Proof $\implies$**:**
Claim: The set $S = \{0x, 1x, \dots, (m-1)x\}$ contains $y \equiv 1 \bmod m$ if all distinct modulo m.

Each of m numbers in S correspond to different one of m equivalence classes modulo m.
$\implies$ One must correspond to 1 modulo m. Inverse Exists!

Proof of Claim: If not distinct, then $\exists a, b \in \{0, \dots, m-1\}$, $a \neq b$, where
$(ax \equiv bx \pmod{m}) \implies (a - b)x \equiv 0 \pmod{m}$
Or $(a - b)x = km$ for some integer k.

$gcd(x, m) = 1$
$\implies$ Prime factorization of m and x do not contain common primes.
$\implies$ $(a - b)$ factorization contains all primes in m's factorization.
So $(a - b)$ has to be multiple of m.
$\implies (a - b) \geq m$. But $a, b \in \{0, \dots m-1\}$. Contradiction. $\square$

Proof review. Consequence.

Thm: If $\gcd(x, m) = 1$, then x has a multiplicative inverse modulo m.

Proof Sketch: The set $S = \{0x, 1x, \dots, (m-1)x\}$ contains $y \equiv 1 \bmod m$ if all distinct modulo m.
... $\square$

For $x = 4$ and $m = 6$. All products of 4...
$S = \{0(4), 1(4), 2(4), 3(4), 4(4), 5(4)\} = \{0, 4, 8, 12, 16, 20\}$
reducing $\pmod{6}$
$S = \{0, 4, 2, 0, 4, 2\}$
Not distinct. Common factor 2. Can't be 1. No inverse.

For $x = 5$ and $m = 6$.
$S = \{0(5), 1(5), 2(5), 3(5), 4(5), 5(5)\} = \{0, 5, 4, 3, 2, 1\}$
All distinct, contains 1! 5 is multiplicative inverse of $5 \pmod{6}$.
(Hmm. What normal number is it own multiplicative inverse?) 1 -1.

$5x = 3 \pmod{6}$ What is x? Multiply both sides by 5.
$x = 15 = 3 \pmod{6}$

$4x = 3 \pmod{6}$ No solutions. Can't get an odd.
$4x = 2 \pmod{6}$ Two solutions! $x = 2, 5 \pmod{6}$

Very different for elements with inverses.

Proof Review 2: Bijections.

If gcd(x,m) = 1.
Then the function $f(a) = xa \mod m$ is a bijection.
One to one: there is a unique pre-image.
Onto: the sizes of the domain and co-domain are the same.
$x = 3, m = 4$.
$f(1) = 3(1) = 3 \pmod 4, f(2) = 6 = 2 \pmod 4, f(3) = 1 \pmod 3$.
Oh yeah. $f(0) = 0$.

Bijection $\equiv$ unique pre-image and same size.
All the images are distinct. $\implies$ unique pre-image for any image.

$x = 2, m = 4$.
$f(1) = 2, f(2) = 0, f(3) = 2$
Oh yeah. $f(0) = 0$.

Not a bijection.

Finding inverses.

How to find the inverse?

How to find **if** x has an inverse modulo m?

Find gcd (x, m).
Greater than 1? No multiplicative inverse.
Equal to 1? Mutliplicative inverse.

Algorithm: Try all numbers up to x to see if it divides both x and m.

Very slow.

Inverses

Next up.

Euclid's Algorithm.
Runtime.
Euclid's Extended Algorithm.

Refresh

Does 2 have an inverse mod 8? No.
Any multiple of 2 is 2 away from $0 + 8k$ for any $k \in \mathbb{N}$.

Does 2 have an inverse mod 9? Yes. 5
$2(5) = 10 = 1 \mod 9$.

Does 6 have an inverse mod 9? No.
Any multiple of 6 is 3 away from $0 + 9k$ for any $k \in \mathbb{N}$.
$3 = gcd(6, 9)$!

x has an inverse modulo m if and only if
$gcd(x, m) > 1$? No.
$gcd(x, m) = 1$? Yes.

Now what?:
Compute gcd!
Compute Inverse modulo m.

Divisibility...

Notation: $d|x$ means "d divides x" or
$x = kd$ for some integer k.

Fact: If $d|x$ and $d|y$ then $d|(x + y)$ and $d|(x - y)$.

Is it a fact? Yes? No?

Proof: $d|x$ and $d|y$ or
$x = \ell d$ and $y = kd$

$\implies x - y = kd - \ell d = (k - \ell)d \implies d|(x - y)$ $\square$

More divisibility

Notation: $d|x$ means "d divides x" or
$x = kd$ for some integer k.

Lemma 1: If $d|x$ and $d|y$ then $d|y$ and $d| \mod (x, y)$.

Proof:

$$\begin{aligned} \mod (x, y) &= x - \lfloor x/y \rfloor \cdot y \\ &= x - \lfloor s \rfloor \cdot y \quad \text{for integer } s \\ &= kd - s\ell d \quad \text{for integers } k, \ell \text{ where } x = kd \text{ and } y = \ell d \\ &= (k - s\ell)d \end{aligned}$$

Therefore $d| \mod (x, y)$. And $d|y$ since it is in condition. $\square$

Lemma 2: If $d|y$ and $d| \mod (x, y)$ then $d|y$ and $d|x$.
Proof...: Similar. Try this at home. $\square$ish.

GCD Mod Corollary: $\gcd(x, y) = \gcd(y, \mod (x, y))$.
Proof: x and y have **same** set of common divisors as x and $\mod (x, y)$ by Lemma 1 and 2.
Same common divisors $\implies$ largest is the same. $\square$

Euclid's algorithm.

GCD Mod Corollary: $\gcd(x,y) = \gcd(y, \mod(x,y))$.

Hey, what's $\gcd(7,0)$? 7 since 7 divides 7 and 7 divides 0
What's $\gcd(x,0)$? x

```
(define (euclid x y)
  (if (= y 0)
    x
    (euclid  y (mod x y))))  ***
```

Theorem: (euclid x y) $= \gcd(x,y)$ if $x \geq y$.

Proof: Use Strong Induction.
Base Case: $y = 0$, "x divides y and x"
$\implies$ "x is common divisor and clearly largest."
Induction Step: $\mod(x,y) < y \leq x$ when $x \geq y$

call in line (***) meets conditions plus arguments "smaller"
and by strong induction hypothesis
computes $\gcd(y, \mod(x,y))$
which is $\gcd(x,y)$ by GCD Mod Corollary. □

Excursion: Value and Size.

Before discussing running time of gcd procedure...

What is the value of 1,000,000?

one million or 1,000,000!

What is the "size" of 1,000,000?

Number of digits in base 10: 7.

Number of bits (a digit in base 2): 21.

For a number x, what is its size in bits?

$$n = b(x) \approx \log_2 x$$

Euclid procedure is fast.

Theorem: (euclid x y) uses $2n$ "divisions" where $n = b(x) \approx \log_2 x$.

Is this good? Better than trying all numbers in $\{2, \ldots y/2\}$?

Check 2, check 3, check 4, check 5 . . . , check $y/2$.

If $y \approx x$ roughly y uses n bits ...
2^{n-1} divisions! Exponential dependence on size!

101 bit number. $2^{100} \approx 10^{30} =$ "million, trillion, trillion" divisions!

$2n$ is much faster! .. roughly 200 divisions.

Algorithms at work.

Trying everything

Check 2, check 3, check 4, check 5 . . . , check $y/2$.

"(gcd x y)" at work.

```
euclid(700,568)
  euclid(568, 132)
    euclid(132, 40)
      euclid(40, 12)
        euclid(12, 4)
          euclid(4, 0)
            4
```

Notice: The first argument decreases rapidly.
At least a factor of 2 in two recursive calls.

(The second is less than the first.)

Maybe Break.

Runtime Proof.

```
(define (euclid x y)
 (if (= y 0)
     x
    (euclid  y (mod x y))))
```

Theorem: (euclid x y) uses $O(n)$ "divisions" where $n = b(x)$.

Proof:

Fact:
First arg decreases by at least factor of two in two recursive calls.

After $2\log_2 x = O(n)$ recursive calls, argument x is 1 bit number.
One more recursive call to finish.
1 division per recursive call.
$O(n)$ divisions. □

Runtime Proof (continued.)

```
(define (euclid x y)
 (if (= y 0)
     x
    (euclid  y (mod x y))))
```

Fact:
First arg decreases by at least factor of two in two recursive calls.

Proof of Fact: Recall that first argument decreases every call.

Case 1: $y < x/2$, first argument is y
$\implies$ true in one recursive call;

Case 2: Will show "$y \geq x/2$" $\implies$ "$mod(x,y) \leq x/2$."

mod (x,y) is second argument in next recursive call,
and becomes the first argument in the next one.
When $y \geq x/2$, then

$\lfloor \frac{x}{y} \rfloor = 1$,

$\text{mod}\,(x,y) = x - y\lfloor \frac{x}{y} \rfloor = x - y \leq x - x/2 = x/2$

$\square$

Finding an inverse?

We showed how to efficiently tell if there is an inverse.

Extend euclid to find inverse.

Euclid's GCD algorithm.

```
(define (euclid x y)
 (if (= y 0)
     x
    (euclid  y (mod x y))))
```

Computes the gcd(x,y) in $O(n)$ divisions.

For x and m, if gcd$(x,m) = 1$ then x has an inverse modulo m.

Multiplicative Inverse.

GCD algorithm used to tell **if** there is a multiplicative inverse.

How do we **find** a multiplicative inverse?

Extended GCD

Euclid's Extended GCD Theorem:
For any x,y there are integers a,b where

$ax + by = d \quad$ where $d = gcd(x,y)$.

"Make d out of sum of multiples of x and y."

What is multiplicative inverse of x modulo m?

By extended GCD theorem, when gcd$(x,m) = 1$.

$ax + bm = 1$
$ax \equiv 1 - bm \equiv 1 \pmod{m}$.

So a multiplicative inverse of $x \pmod{m}$!!
Example: For $x = 12$ and $y = 35$, gcd$(12,35) = 1$.

$(3)12 + (-1)35 = 1$.

$a = 3$ and $b = -1$.
The multiplicative inverse of $12 \pmod{35}$ is 3.

Make d out of x and y..?

```
gcd(35,12)
  gcd(12, 11)  ;;  gcd(12, 35%12)
    gcd(11, 1)  ;;  gcd(11, 12%11)
      gcd(1,0)
        1
```

How did gcd get 11 from 35 and 12?
$35 - \lfloor \frac{35}{12} \rfloor 12 = 35 - (2)12 = 11$

How does gcd get 1 from 12 and 11?
$12 - \lfloor \frac{12}{11} \rfloor 11 = 12 - (1)11 = 1$

Algorithm finally returns 1.

But we want 1 from sum of multiples of 35 and 12?

Get 1 from 12 and 11.
$1 = 12 - (1)11 = 12 - (1)(35 - (2)12) = (3)12 + (-1)35$
Get 11 from 35 and 12 and plugin.... Simplify. $a = 3$ and $b = -1$.

Extended GCD Algorithm.

```
ext-gcd(x,y)
  if y = 0 then return(x, 1, 0)
     else
         (d, a, b) := ext-gcd(y, mod(x,y))
         return (d, b, a - floor(x/y)  * b)
```

Claim: Returns (d,a,b): $d = gcd(a,b)$ and $d = ax + by$.
Example: $a - \lfloor x/y \rfloor \cdot b = 1 - \lfloor 35/12 \rfloor \cdot (-1) = 3$

```
    ext-gcd(35,12)
      ext-gcd(12, 11)
        ext-gcd(11, 1)
          ext-gcd(1,0)
          return (1,1,0) ;; 1 = (1)1 + (0) 0
        return (1,0,1)   ;; 1 = (0)11 + (1)1
      return (1,1,-1)    ;; 1 = (1)12 + (-1)11
   return (1,-1, 3)      ;; 1 = (-1)35 +(3)12
```


Extended GCD Algorithm.

```
ext-gcd(x,y)
  if y = 0 then return(x, 1, 0)
     else
         (d, a, b) := ext-gcd(y, mod(x,y))
         return (d, b, a - floor(x/y)  * b)
```

Theorem: Returns (d,a,b), where $d = gcd(a,b)$ and

$$d = ax + by.$$

Correctness.

Proof: Strong Induction.[1]
Base: ext-gcd$(x,0)$ returns $(d = x, 1, 0)$ with $x = (1)x + (0)y$.

Induction Step: Returns (d,A,B) with $d = Ax + By$
Ind hyp: **ext-gcd**$(y, \bmod(x,y))$ returns (d,a,b) with
$d = ay + b(\bmod(x,y))$

ext-gcd(x,y) calls **ext-gcd**$(y, \bmod(x,y))$ so

$$\begin{aligned} d &= ay + b \cdot (\bmod(x,y)) \\ &= ay + b \cdot (x - \lfloor \frac{x}{y} \rfloor y) \\ &= bx + (a - \lfloor \frac{x}{y} \rfloor \cdot b)y \end{aligned}$$

And ext-gcd returns $(d, b, (a - \lfloor \frac{x}{y} \rfloor \cdot b))$ so theorem holds! □

[1] Assume d is $gcd(x,y)$ by previous proof.

Review Proof: step.

Prove: returns (d,A,B) where $d = Ax + By$.

```
ext-gcd(x,y)
  if y = 0 then return(x, 1, 0)
     else
         (d, a, b) := ext-gcd(y, mod(x,y))
         return (d, b, a - floor(x/y)  * b)
```

Recursively: $d = ay + b(x - \lfloor \frac{x}{y} \rfloor \cdot y) \implies d = bx - (a - \lfloor \frac{x}{y} \rfloor b)y$

Returns $(d, b, (a - \lfloor \frac{x}{y} \rfloor \cdot b))$.

Hand Calculation Method for Inverses.

Example: $gcd(7,60) = 1$.
egcd(7,60).

$$\begin{aligned} 7(0) + 60(1) &= 60 \\ 7(1) + 60(0) &= 7 \\ 7(-8) + 60(1) &= 4 \\ 7(9) + 60(-1) &= 3 \\ 7(-17) + 60(2) &= 1 \end{aligned}$$

Confirm: $-119 + 120 = 1$

Wrap-up

Conclusion: Can find multiplicative inverses in $O(n)$ time!

Very different from elementary school: try 1, try 2, try 3...
$2^{n/2}$

Inverse of 500,000,357 modulo 1,000,000,000,000?
≤ 80 divisions.
versus 1,000,000

Internet Security.
Public Key Cryptography: 512 digits.
512 divisions vs.
$(10000000000000000000000000000000000000000000)^5$ divisions.

Internet Security: Thursday.