

THE PIGEONHOLE PRINCIPLE

If you put 6 pigeons in 5 holes then at least one hole will have more than one pigeon

THE PIGEONHOLE PRINCIPLE

If you put 6 pigeons in 5 holes then at least one hole will have more than one pigeon

THE PIGEONHOLE PRINCIPLE

If you put more than n pigeons in n holes then at least one hole will have more than one pigeon

Every X mapped to its own X.

All 2^n values of X All 2^{n-2} values of X

PIGEONHOLE PRINCIPLE:

We can't map all 2^n Xs to X's of length $< n$. Earth will need to send n bits for some X.

Earth has huge file X that she transferred to Moon. Moon gets Y.

Earth: X Moon: Y

Optimal thing to do

Are X and Y the same n-bit numbers?
Crazy prime number scheme

Earth: X Moon: Y

Picking A Random Prime

Many modern cryptosystems (e.g., RSA) include the instructions:

"Pick a random n-bit prime."

How can this be done efficiently?



Legendre



Gauss

Let $\pi(n)$ be the number of primes between 1 and n. I wonder how fast $\pi(n)$ grows?

Their estimates

| x | $\pi(x)$ | Gauss' Li | Legendre | $x/(\log x - 1)$ |
|-------------|-----------|-----------|-----------|------------------|
| 1000 | 168 | 178 | 172 | 169 |
| 10000 | 1229 | 1246 | 1231 | 1218 |
| 100000 | 9592 | 9630 | 9588 | 9512 |
| 1000000 | 78498 | 78628 | 78534 | 78030 |
| 10000000 | 664579 | 664918 | 665138 | 661459 |
| 100000000 | 5761455 | 5762209 | 5769341 | 5740304 |
| 1000000000 | 50847534 | 50849235 | 50917519 | 50701542 |
| 10000000000 | 455052511 | 455055614 | 455743004 | 454011971 |



Legendre



Gauss

Let $\pi(n)$ be the number of primes between 1 and n. I wonder how fast $\pi(n)$ grows?

Conjecture [1790s]:

$$\lim_{n \rightarrow \infty} \frac{\pi(n)}{n / \ln n} = 1$$

For large n $\pi(n)$ is roughly $n / \ln n$

Two independent proofs of the Prime Density Theorem [1896]:



De la Vallée Poussin

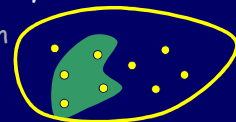


J-S Hadamard

$$\lim_{n \rightarrow \infty} \frac{\pi(n)}{n / \ln n} = 1$$

The Prime Density Theorem

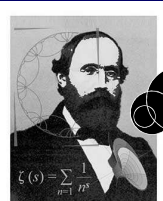
$$\text{Primes } < n \sim n / \ln n$$



Numbers 1..n

This theorem remains one of the celebrated achievements of number theory.

In fact, an even sharper conjecture remains one of the great open problems of mathematics!




Riemann

The Riemann Hypothesis [1859]

$$\lim_{n \rightarrow \infty} \frac{\pi(n) - n / \ln n}{\sqrt{n}} = 0$$

$\pi(n) - n / \ln n \ll \sqrt{n}$



Primes $< n$
 $\sim n / 2 \log n$

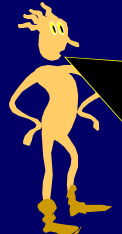
Numbers 1..n

For Computer Science, a much easier bound suffices:
 $\pi(n) \geq n / (2 \log n)$.

If we pick a random number between 1 and n, the probability it is prime is $\pi(n)/n \geq 1 / (2 \log n)$.

A random k bit number has at least a $1 / (2k)$ chance of being prime.

$\pi(n)/n \geq 1 / (2 \log n)$.



A random k bit number is a random number from 1.. 2^k

$\pi(2^k) / 2^k \geq 1 / (2k)$ means that a random k-bit number has at least a $1 / (2k)$ chance of being prime.

Really useful fact

A random k-bit number has at least a $1 / (2k)$ chance of being prime.

So if we pick $2k$ random k-bit numbers the expected number of primes on the list is at least 1

Picking A Random Prime

Many modern cryptosystems (e.g., RSA) include the instructions:

"Pick a random n-bit prime."

How can this be done efficiently?

Picking A Random Prime

"Pick a random n-bit prime."

Strategy:

- 1) Generate random n-bit numbers
- 2) Test each one for primality
[this can be done efficiently!]

Picking A Random Prime

"Pick a random n-bit prime."

1) Generate kn random n-bit numbers

Each trial has a $\geq 1/(2n)$ chance of being prime, and $< (1-1/(2n))$ chance of being composite.

Pr[all kn trials yield composites] Exponentially small
 $\leq (1-1/(2n))^{kn} \leq 1/e^{k/2}$

Picking A Random Prime

"Pick a random n-bit prime."

Strategy:

- 1) Generate random n-bit numbers
- 2) Test each one for primality

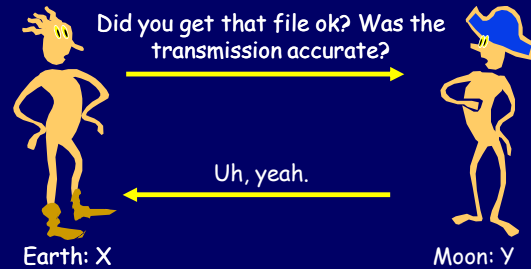
For 1000-bit primes, if we try out 10000 random 1000-bit numbers, chance of failing $\leq e^{-5} < 0.007$

Moral of the story

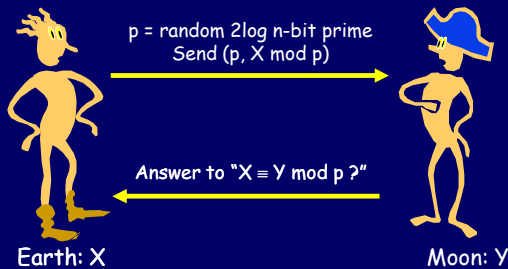
Picking a random prime is "almost as easy as" picking a random number.

(Provided we can check for primality...)

Earth has huge file X that she transferred to Moon. Moon gets Y.



Are X and Y the same n-bit numbers?



Answer to "X ≡ Y mod p ?"

Why is this any good?

Easy case:

If $X = Y$, then $X \equiv Y \pmod{p}$,

so if $X \not\equiv Y \pmod{p}$, then $X \neq Y$.

Answer to "X ≡ Y mod p?"

Why is this any good?

Harder case:
 What if X ≠ Y but we got X ≡ Y mod p?
 Recall: X ≡ Y mod p only when p | (X-Y).
 We mess up only if p | (X-Y).

We mess up only if p | (X-Y).

Define Z = (X-Y). To mess up, p must divide Z.

Z is an n-bit number.
 ⇒ Z is at most 2ⁿ.

Let Z have t prime factors. Any prime ≥ 2.
 2ⁿ > Z = p₁*p₂*...*p_t ≥ 2*2*...*2 = 2^t

Hence t < n and Z has at most n prime factors.

Almost there...

Z has at most n prime divisors.

How many 2logn-bit primes are there?

$\pi(N) \geq N/(2 \log N)$.

⇒ at least 2^{2logn} / (2*2logn) = n² / (4logn) ≫ 2n
 2log n-bit primes.


At most half of them can be divisors of Z = X-Y.

Crazy protocol messes up only if p | (X-Y).

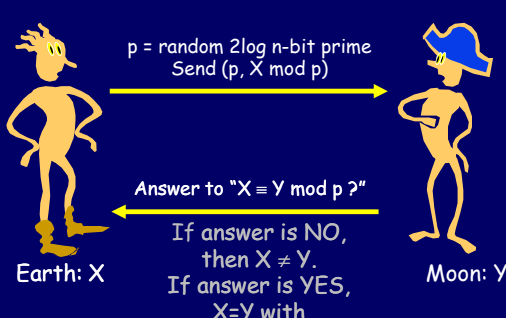
Theorem: Let X and Y be distinct n-bit numbers. Let p be a random 2logn-bit prime.

Then
 Prob [X = Y mod p] < 1/2

Earth-Moon protocol makes mistake with probability at most 1/2!



Are X and Y the same n-bit numbers?



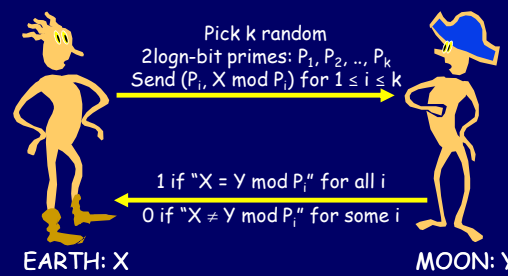
p = random 2log n-bit prime
 Send (p, X mod p)

Answer to "X ≡ Y mod p?"

If answer is NO, then X ≠ Y.
 If answer is YES, X=Y with probability > 1/2.

Earth: X Moon: Y

Are X and Y the same n-bit numbers?



Pick k random 2logn-bit primes: P₁, P₂, ..., P_k
 Send (P_i, X mod P_i) for 1 ≤ i ≤ k

1 if "X = Y mod P_i" for all i
 0 if "X ≠ Y mod P_i" for some i

EARTH: X MOON: Y

Exponentially smaller error probability

If $X=Y$,
 then $X = Y \pmod{P_i}$ for all i , so
 Moon always says YES.

When $k=10$
 $(1/2)^k < 0.001$,
 $1 - (1/2)^k > 0.999$

If $X \neq Y$,
 Prob [$X = Y \pmod{P_i}$ for all i] $\leq (1/2)^k$
 so Moon says YES with tiny probability $(1/2)^k$
 and Moon says NO with probability $> 1 - (1/2)^k$

Are X and Y the same n-bit numbers?

