DTTF/NB479: Dszquphsbqiz Day 22

- Announcements:
  1. Pass in Homework 5 now.
  2. Term project groups and topics due by Friday
     1. Can use discussion forum to find teammates
  3. HW6 posted

- Questions?

- This week:
  - Primality testing, factoring
  - Discrete Logs

The Square Root Compositeness Theorem gives a way to factor certain composite numbers

Given integers n, x, and y:

$$If\ \ x^2 \equiv y^2 (\bmod n),\ but\ x \neq \pm y (\bmod n)$$

Then n is composite, and gcd(x-y, n) is a non-trivial factor

The Miller-Rabin Compositeness Test just reorders the Fermat test's powermod to catch pseudoprimes

$$a^{n-1} \stackrel{?}{\equiv} 1 \pmod{n}$$

Observe: n is odd and n>1

Trick: write $n-1=2^k m$, where k >=1

$$a^{n-1} = \left(\left(\left(a^m \right)^2 \right)^{\cdots} \right)^2 \stackrel{?}{\equiv} 1 \pmod{n}$$

We'll compute powers from inside out, checking if the result is +1 or -1 at each step

It uses the Square Root Compositeness Theorem to catch most pseudoprimes

Given odd n>1, write $n-1=2^k m$, where k >=1.

Choose a base a randomly (or just pick a=2)

Let $b_0=a^m \pmod n$
If $b_0=+/-1$, stop. n is probably prime by Fermat
For i = 1..k-1
Compute $b_i=b_{i-1}^2$.
If $b_i=1 \pmod n$, stop. n is composite by SRCT, and $\gcd(b_{i-1}-1,n)$ is a factor.
If $b_i=-1 \pmod n$, stop. n is probably prime by Fermat.

If $b_k=1 \pmod n$, stop. n is composite by SRCT
Else n is composite by Fermat.

$$a^{n-1} = \underbrace{\left(\underbrace{\left(\left(\overbrace{a^m}^{b_0}\right)^2\right)}_{b_1}^{\cdots}\right)^2}_{b_k}$$

Examples of Miller-Rabin

Given odd n>1, write $n-1=2^k m$, where k >=1.

Choose a base a randomly (or just pick a=2)

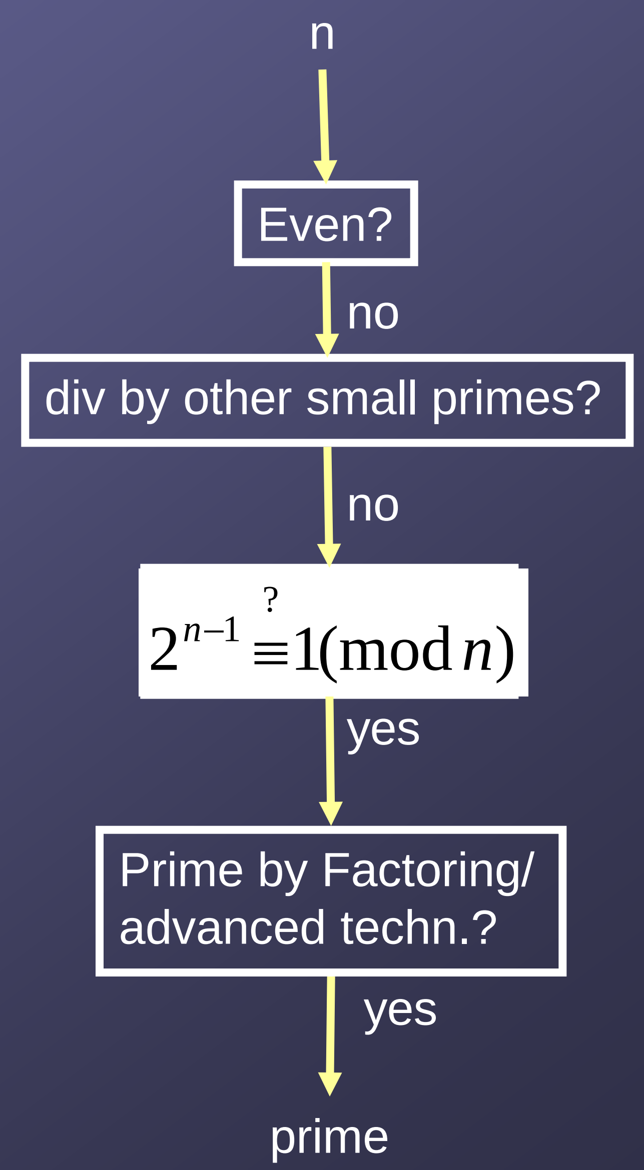
Let $b_0=a^m \pmod n$
If $b_0=+/-1$, stop. n is probably prime by Fermat
For i = 1..k-1
Compute $b_i=b_{i-1}^2$.
If $b_i=1 \pmod n$, stop. n is composite by SRCT, and

$\gcd(b_{i-1}-1,n)$ is a factor.
If $b_i=-1 \pmod n$, stop. n is probably prime by Fermat.

If $b_k=1 \pmod n$, stop. n is composite by SRCT
Else n is composite by Fermat.

1. n=189
2. n=561 (recall Fermat says prob prime)
3. Complete the table on your quiz

Fermat's contrapositive is OK, but Miller-Rabin is better!



Fermat's contrapositive is OK, but Miller-Rabin is better!

- Finding large probable primes
  - #primes < x = $\pi(x) \to \dfrac{x}{\ln(x)}$

Density of primes: ~1/ln(x)

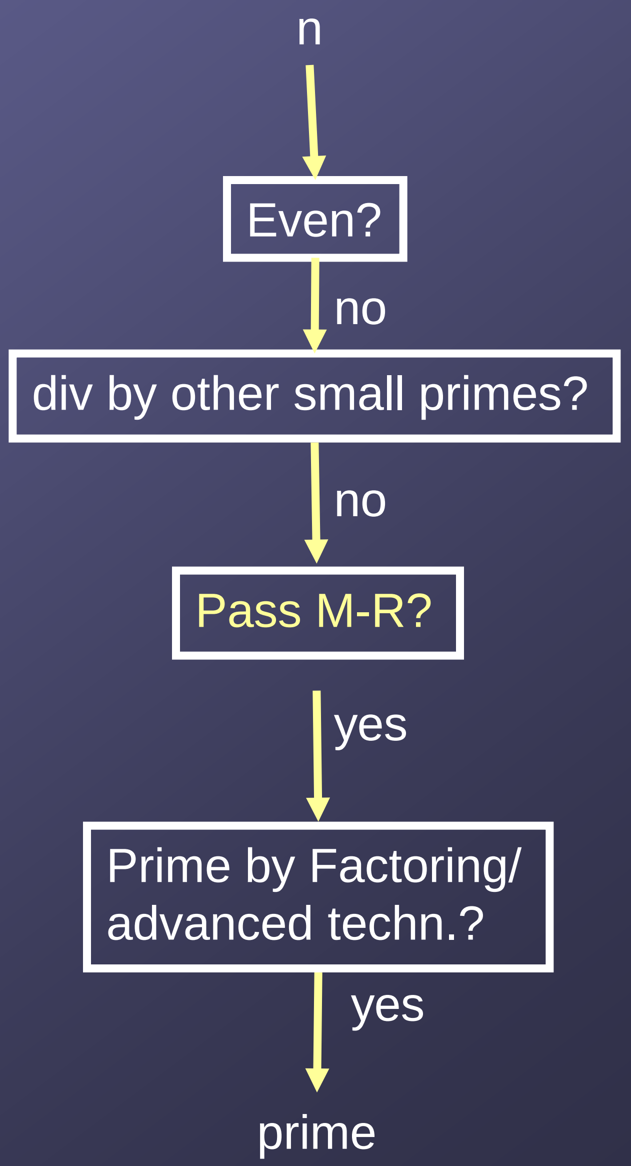
For 100-digit numbers, ~1/230.

So ~1/115 of odd 100-digit numbers are prime

Can start with a random large odd number and iterate, applying M-R to remove composites. We'll soon find one that is a likely prime.

Can repeat with different bases to improve probability that it's prime.

Maple's **nextprime()** appears to do this, but also runs the *Lucas test*: http://www.mathpages.com/home/kmath473.htm



Factoring

- If you are trying to factor $n=pq$ and know that p and q are close, use *Fermat factoring*:
  - Compute $n + 1^2$, $n + 2^2$, $n + 3^2$, until you reach a perfect square, say $r^2 = n + k^2$
  - Then $n = r^2 - k^2 = (r+k)(r-k)$
- Example: factor 2405597
- The moral of the story?
  - Choose p and q such that _____

(p-1) Algorithm

- Useful if p|n and (p-1) has only small factors
- Choose any a>1 (like a=2) and bound B
- Compute $b=a^{B!}$(mod n) (How?)
- Then compute d=gcd(b-1, n)
  - If 1<d<n, then d is a non-trivial factor

- Matlab example: n=5183. We'll use a=2, B=6.
- Why does it work?

Moral of this story?

- To get a 100-digit number n=pq resistant to this attack:
  - Make sure (p-1) has at least 1 large prime factor:
  - Pick $\mathbf{p_0 = nextprime(10^{40})}$
  - Choose $k \sim 10^{60}$ such that $\mathbf{p=(kp_0+1)}$ is prime
    - How to test?
  - Repeat for q.

Example

Factor $n = 3837523$

- Concepts we will learn also apply to factoring really big numbers. They are the basis of the best current methods
- All you had to do to win $30,000 was factor a 212 digit number.
- This is the RSA Challenge: http://www.rsa.com/rsalabs/node.asp?id=2093#RSA704