DTTF/NB479: Dszquphsbqiz Day 21

- Announcements:
  1. *Congrats* on reaching the halfway point once again!
  2. Reminders:
     1. HW5 due tomorrow
     2. Term project groups and topics due by Friday.

- Questions?

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

Term projects

- Use Ch 10 – 19 as inspiration.
  - Elliptic curves?
  - Quantum crypto?
  - Security protocols?

- Deliverables:
  - A paper demonstrating your understanding of the topic
  - A 20-min in-class presentation $9^{th}/10^{th}$ week
    - Groups of 4 to bound presentation time.
  - Preliminary details posted

Plus-delta

- Please give me 5 minutes of your time for feedback on the course so far

Where were we?

- RSA: public-key system: *n*, *e* known
  - Easy to encrypt
  - But need factorization of n (pq) to find d to decrypt.
  - Factorization is a "one-way" function
  - Builds on lots of ch 3 number theory, like Euclid, Fermat, and Euler.
  - Slow, but can be used to send AES "session" keys
- You used Maple to send messages
- You looked at some "implementation mistakes" (for example, using small values for e)

Compositeness testing

Oops, did I say primality testing?

Today, we discuss three techniques that can guarantee a number is composite, and guess when one is prime.

1. Square Root Compositeness Theorem

   +

2. Fermat's Theorem

   =

3. Miller-Rabin Compositeness Test

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

Given integers n, x, and y:

$$\boxed{If\ \ x^2 \equiv y^2 (\operatorname{mod} n),\, but\ x \neq \pm y (\operatorname{mod} n)}$$

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

Proof: on board

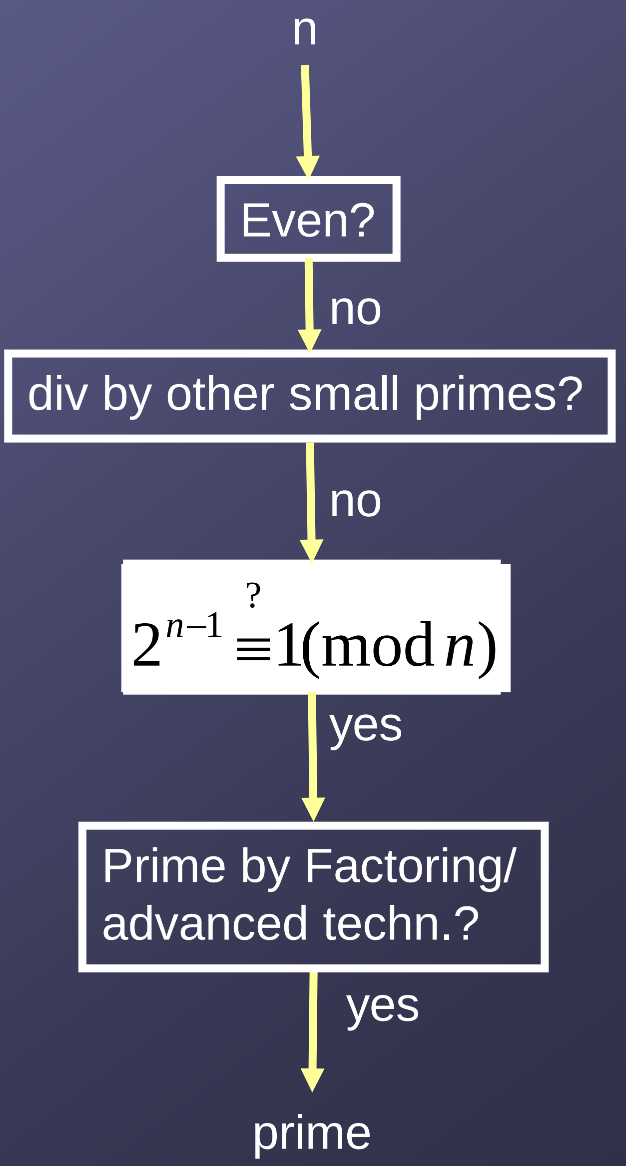
Toy example showing 21 is composite using x=2 and y=16.

Review: Fermat can be used to test for compositeness, but doesn't give factors

- Fermat's little theorem:
  - If n is prime and doesn't divide a, then $a^{n-1} \equiv 1 \pmod{n}$
- Contrapositive:
  - If $a^{n-1} \not\equiv 1 \pmod{n}$ then n is composite
- In practice,
  - If $a^{n-1} \equiv 1 \pmod{n}$ then n is probably prime
  - Rare counterexamples (15k of first 10B pos integers) called *pseudoprimes*
- Notes
  - Never gives factors
  - Compute using powermod

A is… \ a^{n-1}	=1	≠1
Prime	Usually true	None
Composite	Rare pseudoprime	All

Review: Primality testing schemes typically use the contrapositive of Fermat



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

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

Observe: n is odd and n>1

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

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

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

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

Let $b_0=a^m$(mod 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(mod n), stop. n is composite by SRCT, and gcd(b_{i-1}-1,n) is a factor.
- If b_i=-1(mod n), stop. n is probably prime by Fermat.

If b_k=1 (mod n), stop. n is composite by SRCT

Else n is composite by Fermat.