Overview of Quadratic Sieve

1) Choose factor base $p_1, p_2, \ldots, p_m < B$

2) Let $k = \lceil \sqrt{n} \rceil$

3) Try to factor (in the factor base, ie. trial division)

$$k^2 - n$$
$$(k+1)^2 - n$$
$$(k+2)^2 - n$$
etc.

4) Interpret exponents as vectors & find a linear combination that is all even

$$\Rightarrow \quad x^2 \equiv y^2 \pmod{n}$$

5) If $x \not\equiv \pm y$, then $\gcd(x \pm y, n)$ should give a proper factor of n.

Sieve. instead of trial division by factor base, can predict the pattern of divisibility.

$$p \mid k^2 - n \iff n \equiv k^2 \pmod{p} \iff k \text{ is one of the two square roots of } n \bmod p.$$

Example. $n = 2201$

$p = 5$ $\quad 5 \mid k^2 - n \iff n \equiv k^2 \pmod 5 \iff 1 \equiv k^2 \pmod 5$

$$\iff k \equiv \pm 1 \pmod 5$$

k:	47	48	49	50	51	52	53	54	55	56	57	...
$k \bmod 5$:	2	3	4	0	1	2	3	4	0	1	2	...
$5 \mid k^2 - n$?	N	N	Y	N	Y	N	N	Y	N	Y	N	...

loop through $k = 5l \pm 1$ instead of all k for trial div. by 5.

Runtime comments:

We tune # of primes in factor base (B).
# of k's to test

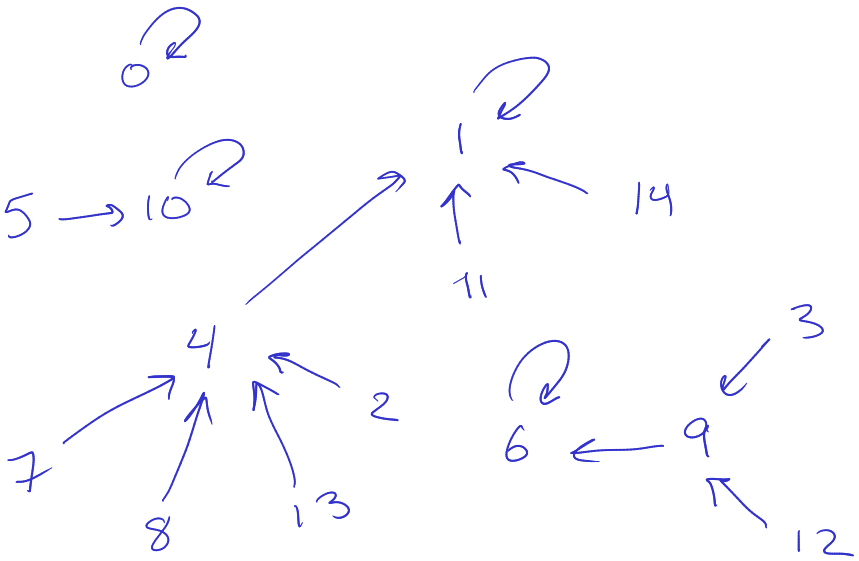
optimizing depends on # of smooth #s in our range,
just like index calculus.

Result: subexponential again.

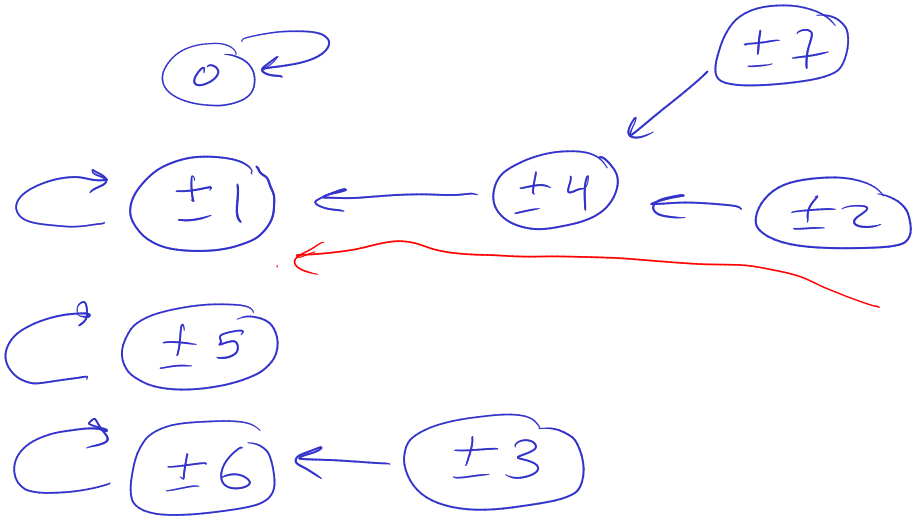
Back to primality testing

eg. $n = 15$ mod n

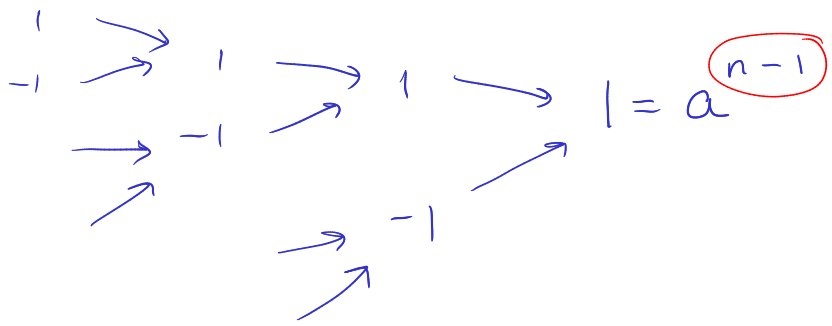
squaring



another way to draw:



If n is prime squaring looks like



Miller-Rabin Factoring

Let $1 < a < n$

Write $n-1 = 2^k m$, m odd

Consider chain

$$a^m \to a^{2m} \to a^{4m} \to \cdots \to a^{2^k m} = a^{n-1} \stackrel{?}{\equiv} 1$$

Fermat test

If n is composite there are 2 ways you might notice:

(a) chain never gets to 1 at all (Fermat)

(b) we see $x \to 1$, $x \not\equiv \pm 1 \pmod n$ (Unusual square root)

Ex. $n = 252601$

$n-1 = 2^3 \cdot 31575$ (odd)

$a = 85132$

If we see @ or (b), composite.

If we see neither, probably prime.

Chain:

$a^{31575} \equiv 191102$ —sq→ $a^{2 \cdot 31575} \equiv 184829$ —sq→ $a^{2^2 \cdot 31575} \equiv 1$ → Composite!

Birthday Attack on RSA ?

Public key (n, e) c = ciphertext.

List 1: $cx^{-e} \pmod n$ for random x

List 2: $y^e \pmod n$ for random y

If collision: $cx^{-e} \equiv y^e \pmod n$

$$c \equiv (\underbrace{xy}_{\text{message}})^e \pmod n$$

(A) Random x, y → lists should be size $\sqrt{n}$ → $O(\sqrt{n})$ attack.

(B) What if we know message $m < B$. Then can try $0 < x, y < \sqrt{B}$.
⇒ likely (?) find $xy = m$.

This would be runtime $O(\sqrt{B})$

Lesson: pad your messages.