

CS265 Class 3

Primality-Testing and the Miller-Rabin Test

1 Warm-Up

Let $S_r = \{x \in \mathbb{Z}_n^* : x^r = \pm 1 \pmod n\}$. Is S_r a group?

Announcements

- HW1 Due Friday!

Recall: Primality testing

- **Goal:** Given n , is it prime?
- Today we'll see a pretty simple randomized algorithm.
- It was open for a long time to give a deterministic algorithm.
 - Finally solved in 2002 by Agrawal, Kayal, and Saxena.
 - We won't talk about the deterministic algorithm in class: see links on website if you want to learn about it on your own.

When we last left our heroes...

- We saw **Fermat's test**.
 - Check if $x^{n-1} = 1$ for a random x
- We proved that it mostly worked:
 - For lots of numbers n (all those that are not Carmichael numbers):
 - If n is prime, Fermat's test will always say so.
 - If n is not prime, Fermat's test will say "not prime" with probability at least $\frac{1}{2}$.

Goal for today:

- Can we come up with a test that gives this guarantee for *all* n ?

But first, questions?

- Warm-up, quiz, mini-lectures?

1 Warm-Up

Let $S_r = \{x \in \mathbb{Z}_n^* : x^r = \pm 1 \pmod n\}$. Is S_r a group?

1 Warm-Up

Warm-up

Let $S_r = \{x \in \mathbb{Z}_n^* : x^r = \pm 1 \pmod n\}$. Is S_r a group?

- Yes, this is a group!
 - Since \mathbb{Z}_n^* is a group, we just need to establish that it's closed under multiplication and inverses.
- Closure:
 - if $x^r = \pm 1$ and $y^r = \pm 1$, then $(xy)^r = x^r y^r = (\pm 1)(\pm 1) = \pm 1$
- Inverses:
 - Let $x \in S_r$ and let $y = x^{-1} \in \mathbb{Z}_n^*$. Want to show $y \in S_r$.
 - $xy = 1$
 - $x^r y^r = 1$
 - $(\pm 1)y^r = 1$
 - $y^r = \pm 1$

Plan for today

- We will develop the Miller-Rabin algorithm for primality testing.
- We will **mostly*** prove that it works.

Compare to Fermat's test, where we **proved that it mostly works.**

*We will skip over stuff that's interesting from an algebra perspective but not so interesting from a "randomized algorithms and probabilistic analysis" perspective. Details in lecture notes if you are curious.



How many square roots of 1 in \mathbb{Z}_n^* ?

- That is, how many $x \in \mathbb{Z}_n^*$ have $x^2 = 1$?

- Example: $n = 7$

$$\begin{array}{l} 1^2 = 1 \\ 2^2 = 4 \\ 3^2 = 2 \\ 4^2 = 2 \\ 5^2 = 4 \\ 6^2 = 1 \end{array}$$

± 1 are the only square roots of 1

- Example: $n = 15$

$$\begin{array}{l} 1^2 = 1 \\ 2^2 = 4 \\ 4^2 = 1 \\ 7^2 = 4 \\ 8^2 = 4 \\ 11^2 = 1 \\ 13^2 = 4 \\ 14^2 = 1 \end{array}$$

There are more than two square roots of 1

Why are we only looking at the numbers 1,2,4,7,8,11,13,14?



Idea of the Miller-Rabin Algorithm

- If n is prime, then there are exactly two square roots of 1 in \mathbb{Z}_n^* , $+1$ and -1 .
 - Assume that this is true for now!
 - Check out the lecture notes, or prove it yourself!
 - Uses the fact that \mathbb{Z}_n^* is cyclic for prime n .
- If n is an odd composite number that is not the power of a prime, then there are **more** than two square roots of 1 in \mathbb{Z}_n^* .
- The Miller-Rabin Algorithm gives a clever way to find some $x \neq \pm 1$ so that $x^2 = 1$ when one exists.

Here is a way to generate a list of numbers

Why? We'll see soon...

- Suppose that n is odd, and not a power of a prime.
- Find k and m so that $n - 1 = 2^k m$ where m is odd.
- Choose $x \in \{1, \dots, n - 1\}$ uniformly at random, and compute the list:

$$x^m, x^{2m}, x^{2^2 m}, \dots, x^{2^k m} \pmod n$$

Here is a way to generate a list of numbers

Why? We'll see soon...

- Suppose that n is odd, and not a power of a prime.
- Find k and m so that $n - 1 = 2^k m$ where m is odd.
- Choose $x \in \{1, \dots, n - 1\}$ uniformly at random, and compute the list:

$$x^m, x^{2m}, x^{2^2 m}, \dots, x^{2^k m} \pmod n$$

Example

- Say $n = 21$
- Then $n - 1 = 2^2 \cdot 5$, so $k = 2, m = 5$
- Choose $x = 3$.
- The list is:
 - $3^5 = 12 \pmod{21}$,
 - $3^{10} = 18 \pmod{21}$,
 - $3^{20} = 9 \pmod{21}$.

Group work



GenerateSomeNumbers:

- Write $n - 1 = 2^k \cdot m$ where m is odd.
- Choose $x \in \{1, \dots, n - 1\}$ uniformly at random.
- Consider $x^m, x^{2m}, x^{2^2m}, \dots, x^{2^k m} \pmod n$.

1. Mess around with web.stanford.edu/~marykw/CS265Class3.html
2. Say n is prime. What is the last number generated by this list?
3. Say n is prime. What are the options for the second-to-last number?
4. Say $n=561$ (the first Carmichael number). Consider $x=23, 13, 63, 458$. What list of numbers do you get for each? Does this list prove to you that n is not prime?
5. If time, come up with (and analyze) an algorithm for primality testing.

Solutions to group work

- If n is prime, the last number generated should be 1.
- If n is prime, the second-to-last number should be ± 1 .
- If $n=561$,
 - $x=23 \rightarrow [386, 331, 166, 67, 1]$. Not prime! $(67)^2 = 1$
 - $x=13 \rightarrow [208, 67, 1, 1, 1]$. Not prime! $(67)^2 = 1$
 - $x=63 \rightarrow [351, 342, 276, 441, 375]$. Not prime! $(63)^{n-1} = 375 \neq 1$
 - $x=458 \rightarrow [560, 1, 1, 1, 1]$. Could be prime...

Miller-Rabin Algorithm


- Given an input n :
 - If n is even, or the power of a prime, output **Composite!**
 - Choose a random x and find the special list of numbers.
 - Make sure it looks like $[*, *, *, -1, 1, 1, 1, 1]$ or $[1, 1, 1, 1, 1, 1, 1, 1]$:
 - If the last thing isn't 1, output **Composite!** n fails Fermat's test!
 - If there's ever (not ± 1) followed by 1, output **Composite!**
 - Output **Prime!**

How do we
check for these
efficiently?



There's some number other than ± 1
that squares to 1 mod n

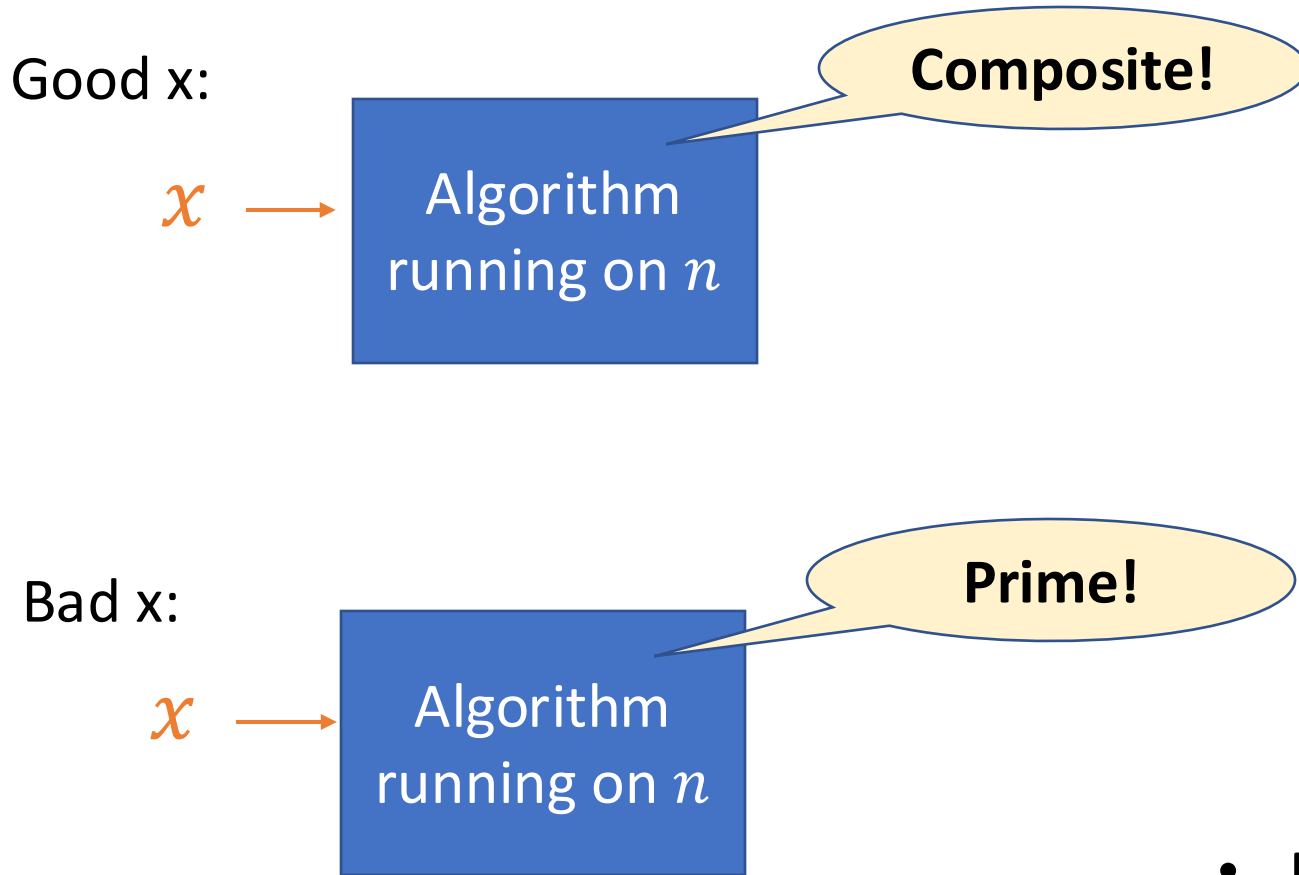
Theorem

- If n is prime, the Miller-Rabin Test will always output “Prime!” 
- If n is composite, the Miller-Rabin Test will output “Composite!” with probability at least $\frac{1}{2}$.
- The running time is polynomial in $\log n$

When n is composite

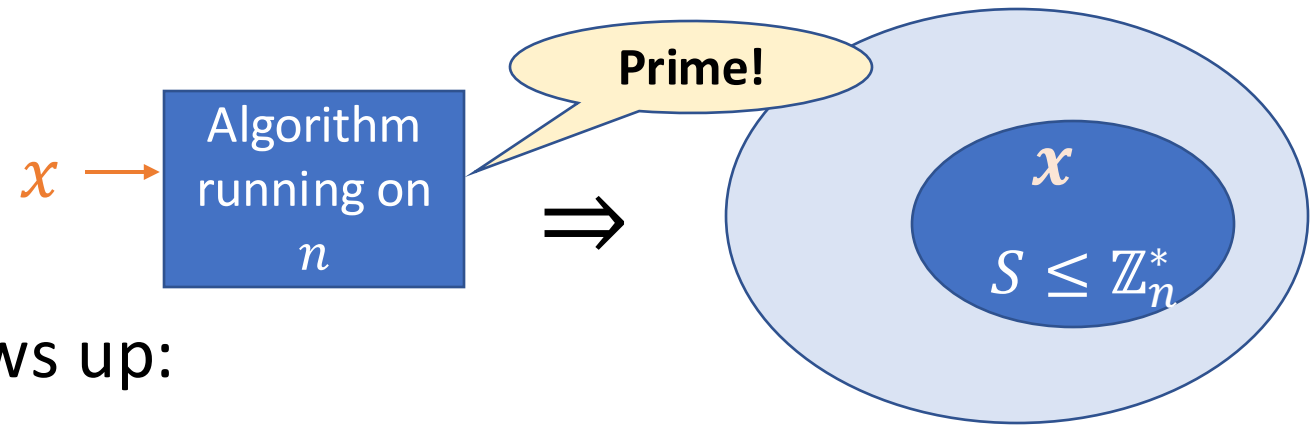
and odd and not a prime power

- We will define a proper subgroup $S \leq \mathbb{Z}_n^*$ that contains all of the bad x 's.

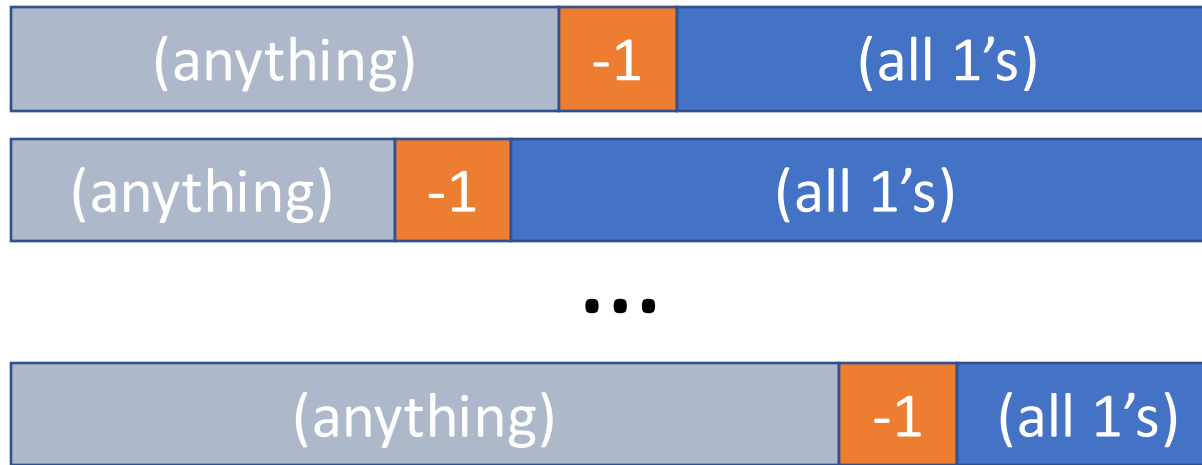


- By Lagrange's theorem, $|S| \leq \frac{1}{2} |\mathbb{Z}_n^*|$
- The algorithm succeeds with probability $\geq \frac{1}{2}$.

Defining S



- Consider all the x 's where -1 shows up:



Say that this is $x^{2^b m}$

- Let b be the **largest** value of i less than k so that there exists an $x \in \mathbb{Z}_n^*$ so that $x^{2^i m} = -1 \pmod n$.
- Define $S = \{y \in \mathbb{Z}_n^* : y^{2^b m} = \pm 1\}$

Claim 1

- For any x so that the algorithm says “Prime!”, $x \in S$

If $x \in \mathbb{Z}_n^*$, then...



Definition of b
and the fact that
all x 's look like this:

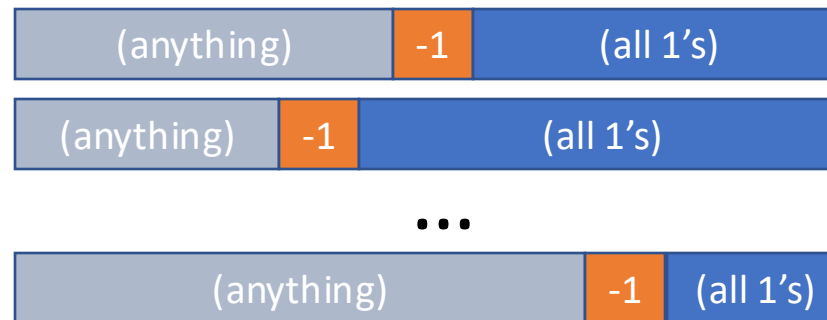
If $x \notin \mathbb{Z}_n^*$, then...

...the algorithm won't say “Prime”

$x^{n-1} \neq 1$, so fails Fermat's test.

Otherwise $x^{n-1} \cdot x = 1$, but
supposedly x doesn't have an
invers

- Consider all the x 's where -1 shows up:



Say that this is $x^{2^b m}$

- Let b be the **largest** value of i less than k so that there exists an $x \in \mathbb{Z}_n^*$ so that $x^{2^i m} = -1 \pmod n$.
- Define $S = \{y \in \mathbb{Z}_n^* : y^{2^b m} = \pm 1\}$

Claim 2

- S is a subgroup of \mathbb{Z}_n^* .



This was the warm-up!

Claim 3

- If n is odd, composite, and not a prime power, then $S \neq \mathbb{Z}_n^*$

Fun exercise if you have time!

$$S = \{y \in \mathbb{Z}_n^* : y^{2^b m} = \pm 1\}$$

Three claims:

- Claim 1: For any x so that the algorithm says “Prime!”, $x \in S$
- Claim 2: S is a subgroup of \mathbb{Z}_n^* .
- Claim 3: If n is odd, composite, and not a prime power, $S \neq \mathbb{Z}_n^*$

Group work:

Claims 1, 2 and 3 imply that the algorithm works with probability at least $\frac{1}{2}$.

- Claim 1: For any x so that the algorithm says “Prime!”, $x \in S$
 - Claim 2: S is a subgroup of \mathbb{Z}_n^* .
 - Claim 3: If n is odd, composite, and not a prime power, $S \neq \mathbb{Z}_n^*$
-
- If n is odd, composite, not a prime power, then by Claims 2 and 3, $S < \mathbb{Z}_n^*$ is a proper subgroup.
 - That means that $|S| \leq \frac{|\mathbb{Z}_n^*|}{2}$ by Lagrange’s theorem.
 - $\Pr[\text{test says } n \text{ is prime}] \leq \Pr[\text{test picks } x \in S] \leq \frac{1}{2}$
- By Claim 1
By the above

Theorem: for any n ,

Modulo the missing proposition...

- If n is prime, the Miller-Rabin Test will always output “Prime!”
- If n is composite, the Miller-Rabin Test will output “Composite!” with probability at least $\frac{1}{2}$.
- The running time is polynomial in $\log n$

You check this!

Modulo proof of Claim 3...

$$n - 1 = 2^k m \text{ where } m \text{ is odd}$$

Running time

- If n is even, or the power of a prime, output **Composite!**
- Choose random x and find $x^m, x^{2m}, \dots, x^{2^k m}$
- Make sure it looks like $[*, *, *, -1, 1, 1, 1, 1]$ or $[1, 1, 1, 1, 1, 1, 1, 1]$:
 - If the last thing isn't 1, output **Composite!**
 - If there's ever (not ± 1) followed by 1, output **Composite!**
- Output **Prime!**

Time poly(log(n))

- Compute x^m by repeated squaring
 - $O(\log m) = O(\log n)$ multiplications
- Square x^m $k = O(\log n)$ times
- Need to check $k = O(\log n)$ things

Total time poly(log n)

Recap

The Miller-Rabin Primality Test

- Cleverly generate a list of numbers for a random x .
- If n isn't prime, probably these numbers will give it away.
- The point: There is a fast and not-too-hard randomized algorithm for primality testing!
- It was open for a long time to get a deterministic algorithm
 - (And it isn't that simple).

Next time! Markov and Chebyshev!

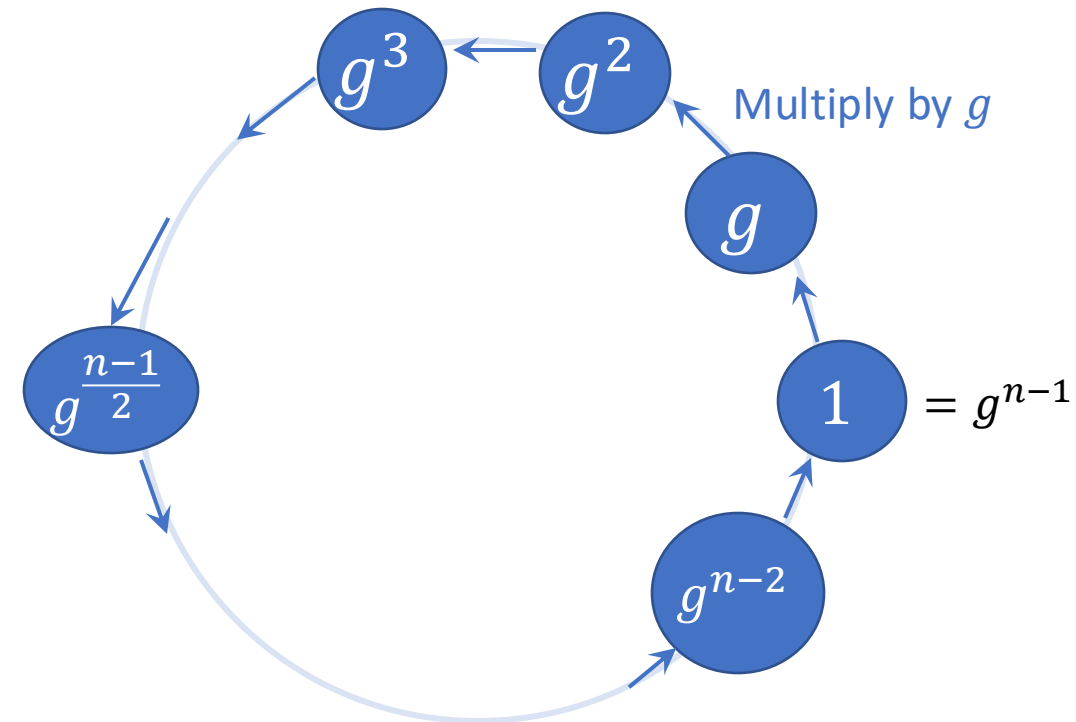
- Before next time:
 - Watch the videos/read the lecture notes
 - Do the quiz

Missing piece 1:

Square Roots of 1 in \mathbb{Z}_n^* when n is prime

- Recall that \mathbb{Z}_n^* is cyclic when n is prime.
- Corollary: For prime n , if $a \in \mathbb{Z}_n^*$ has $a^2 = 1$, then either $a = 1$ or $a = -1$
- Proof Sketch:

- $n = 2$: [You check]
- $n > 2$:
 - See picture.
 - If $(g^j)^2 = g^{2j} = 1$, then $2j = 0 \pmod{n-1}$.
 - The only such numbers in $\{1, \dots, n-1\}$ are $j = n-1$ and $j = \frac{n-1}{2}$
 - We already had two roots, ± 1 , so this must be all of them.



- Let b be the **largest** value of i less than k so that there exists an x with $x^{2^i m} = -1 \pmod n$.
- $S = \{y \in \mathbb{Z}_n^* : y^{2^b m} = \pm 1\}$

Missing piece 2:

Proof of Claim 3

If n is odd, composite, and not a prime power, $S \neq \mathbb{Z}_n^*$

- Let $r = 2^b m$ for convenience.
- Write $n = s \cdot t$ where $s, t > 1$ are relatively prime. We can do this since n is composite, and not a prime power.
- Suppose $x \in \mathbb{Z}_n^*$ is such that $x^r = -1$. This x exists by the definition of r and b
- Claim: There exists a $y \in \mathbb{Z}_n^*$ so that $y = x \pmod s$, and $y = 1 \pmod t$

This follows from the Chinese Remainder Theorem

- Claim: $y \in \mathbb{Z}_n^*$ By def of \mathbb{Z}_n^* , $\gcd(x, n) = 1$
Then $\gcd(x, s) = 1$, since $s|n$.
But then we also have $\gcd(y, s) = 1$, since $y \equiv x \pmod s$.

- Claim: $y \notin S_r$ $y^r = 1 \pmod t$, and $y^r = x^r = -1 \pmod s$.
Case 1: $y^r = -1 \pmod n \Rightarrow y^r = -1 \pmod t$. (Since $t|n$)
Case 2: $y^r = 1 \pmod n \Rightarrow y^r = 1 \pmod s$. (Since $s|n$)
Either way it's a contradiction.