CS265, Winter 2025

Class 3: Agenda, Questions, and Links

1 Warm-Up

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

Group Work: Solutions

Yes, this is a group! Since it's a subset of $\mathbb{Z}_n^*$, it automatically satisfies most of the axioms. The only one that we still need to establish that it is closed under multiplication and taking inverses.

For closure under multiplication: for any $x, y \in S_r$, $x \cdot y \in S_r$ also. This is true because for any such x, y,

$$(x \cdot y)^r = x^r y^r = (\pm 1) \cdot (\pm 1) = \pm 1 \mod n.$$

For inverses, suppose that $x \in S_r$, and let $y = x^{-1}$. (We know that y exists in $\mathbb{Z}_n^*$; our goal is to show that $y \in S_r$). Then $xy = 1$, so $(xy)^r = 1$, so $x^r y^r = 1$. Since $x^r = \pm 1$, we have $(\pm 1)y^r = 1$ and hence $y^r = \pm 1$, using the fact that $1^{-1} = 1$ and $(-1)(-1) = 1 \mod n$, and hence $(-1)^{-1} = -1$. This shows that $y \in S_r$.

2 Announcements

- HW1 is due Friday!

3 Questions?

Any questions from the minilectures or the quiz? (Group Theory 101; Primality Testing)

4 Miller-Rabin Algorithm

[A bit of lecture to introduce the Miller-Rabin Algorithm. Summary below.]

The main idea behind the Miller-Rabin algorithm is the fact (which is not obvious) that:

- If n is prime, then there are exactly two square roots of 1 in $\mathbb{Z}_n^*$, $+1$ and -1.
- If n is an odd composite number, not the power of a prime, then there are *more* than two square roots of 1 in $\mathbb{Z}_n^*$.

The following Proposition is true, check out the lecture notes (or think about it yourself if you finish the group work early!) for more details.

Proposition 1. *If n is prime, then there are exactly two square roots of 1 in $\mathbb{Z}_n^*$, $+1$ and -1.*

Consider the following way to generate a list of numbers, using n. (It should not be obvious at this point why we are doing this).

Procedure for generating some numbers

- Write $n-1 = 2^k m$ where m is odd.
- Choose $x \in \{1, \ldots, n-1\}$ uniformly at random.
- Consider the list of numbers $x^m, x^{2m}, x^{2^2 m}, \ldots, x^{2^k m} \mod n$.

In the following group work, we'll play around with some examples.

4.1 Group work: Exploring this list of numbers

Group Work

1. Figure out how to generate these lists automatically. Here are ways you can do that.
   - Go to
     https://web.stanford.edu/~marykw/CS265Class3.html
     This just runs a python script that takes x and n and generates this list.
   - If you want to do it yourself, there's some python code at the end of this document that you can copy and paste.

   Make sure that you can replicate the example we saw with $x = 3$, $n = 21$. You should get $12, 18, 9$.
2. Suppose that n is prime. What should you get as the *last* number generated in this list? (Hint, Fermat's little theorem). Verify your answer by trying it out.
3. Suppose that n is prime. What are the possible answers you could get as the *second-to-last* number generated in this list? (Hint, Proposition 1). Verify your answer by trying it out.

   (Note: If you see the value "$n-1$" show up in the widget, remember that this is the same as "-1" mod n.)
4. Suppose that $n = 561$. (Recall that this is the first Carmichael number—so it's not prime, but for any x with $gcd(x, n) = 1$, we have $x^{n-1} = 1 \mod n$.)

- Choose $x = 23$. What do you get? Why does this answer prove to you that n is not prime? (Hint, previous question).
- Choose $x = 13$. What do you get? Why does this answer prove to you that n is not prime? (Hint, Prop. 1).
- Choose $x = 63$. What do you get? Why does this answer prove to you that n is not prime? (Hint, Fermat's little theorem).
- Choose $x = 458$. Does this answer prove to you that n is not prime? Why or why not?

5. Come up with a candidate randomized algorithm to test primality based on these observations. You can try out some other examples to test it.
6. If you have time, try to prove that your candidate randomized algorithm works!

Group Work: Solutions

1. Works for me!
2. The last number will be 1, by Fermat's little theorem.
3. The second-to-last number can only be $\pm 1 \mod n$. This is because the second-to-last number, squared, gives us the last number, which is 1 by the first part. But if n is prime, then the only numbers in $\{1, \ldots, n-1\}$ so that $a^2 = 1 \mod n$ are $a = \pm 1$.
4. - For $x = 23$, we get [386, 331, 166, 67, 1]. This proves that n is *not* prime, since the second-to-last element is neither 1 nor $-1 \mod n$.
   - For $x = 13$, we get [208, 67, 1, 1, 1]. This proves that n is not prime, since $67^2 = 1$, but if n were prime then only 1 and 560 could square to 1.
   - For $x = 63$, we get [351, 342, 276, 441, 375]. This proves that n is not prime since $63^{n-1} \neq 1 \mod n$, so it violates Fermat's test.[a]
   - For $x = 458$, we get [560, 1, 1, 1, 1]. This is consistent with n being prime—we can't rule it out.

[a] You might be thinking...but I thought that the point of Carmichael numbers was that they did not fail Fermat's test?? That's true for numbers $x \in \mathbb{Z}_n^*$, but this x is not relatively prime to 561, so it can still fail Fermat's test.

4.2 Group Work: Analyzing the Miller-Rabin Test

[A bit of lecture to state the Miller-Rabin test; see lecture notes for more details, and the relevant parts are summarized below.]

The relevant part of the test[1] is:

[1] See lecture notes for the actual algorithm

Relevant part of Miller-Rabin Test

- Generate the list of k numbers as above.
- If the last number is not 1, output "composite!"
- If there's a number that's *not* equal to ± 1, but the next number is equal to 1, output "composite!"
- Otherwise, output "prime!"

[On slides, we will define the following set S and make the following claims.]

Define

$$S = \left\{ y \in \mathbb{Z}_n^* \, : \, y^{2^b m} = \pm 1 \mod n \right\},$$

where b is the largest value $i < k$ so that there exists an $x^{2^i m} = \pm 1 \mod n$. We made the following 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

Assuming Claims 1,2, and 3, prove the following:

If n is odd, composite, and not a prime power, the algorithm above says "Composite!" with probability at least $1/2$.

(We have already waved our hands about this in class. What you're supposed to be doing now is explaining to each other. Try to write down a formal proof to see how all the pieces fit together.)

If you have time, think about the following:

- Analyze the running time of this algorithm.
- Prove Proposition 1. (Hint: use the fact that $\mathbb{Z}_n^*$ is cyclic).
- Prove Claim 3. (Disclaimer: this is tricky!) (Hint: by the assumptions on n, we can write $n = s \cdot t$ where s and t are relatively prime. The *Chinese Remainder Theorem* implies the following. For any s and t that are relatively prime and for any $x \in \mathbb{Z}_n^*$, there exists a $y \in \mathbb{Z}_n$ so that $y = x \mod s$ and $y = 1 \mod t$. Show that such a y satisfies $y \in \mathbb{Z}_n^*$ but $y \notin S$.)

Group Work: Solutions

Suppose that n is odd, composite, and not a prime power. If the algorithm says "Prime!" then Claim 1 implies that $x \in S$. But Claims 2 and 3 together with Lagrange's theorem imply that $|S| \leq |\mathbb{Z}_n^*|/2$. Thus, if we choose x at random, the probability that the algorithm says "Prime!" is at most $1/2$.

(See lecture notes for the extra questions.)

5 Useful Code

If you don't want to use my web widget thing, here's some Python code to generate the list of numbers, given x and n:

```
def generateList(x,n):
if n <= 1:
return []
        # generate k and m
        k = 0
        m = n-1
        while m % 2 == 0:
                k += 1
                m = m/2
        # now generate x^m, x^(2m), x^(4m), ..., x^(2^k m) mod n
        ret = []
        for i in range(k+1):
                y = x**int(m) % n
                for j in range(i):
                        y = y**2 % n
                ret.append(int(y))
        return ret
```