ECS 20 – Winter 2022 – P. Rogaway

Integers and the Pigeonhole Principle

Today:

- **The pigeonhole principle**
- **A bit more number theory**
- **A touch of cryptography**

PHP: statements

Pigeonhole Principle (v1): If N pigeons roost in n holes, with $n< N$, then some two pigeons roost in the same hole.

Pigeonhole Principle (v2): If $f: A \rightarrow B$ where A and B are finite sets and $|A| > |B|$, then f is **not** injective: there are points $a, a' \in A$ such that $a \neq a'$ and $f(a) = f(a')$.

Ex 1. Any room with 3 or more people has some two of the same gender. Assume for this statement that each person identifies as either "male" or "female" but not both.

Ex 2. 20 people at a party, some two have the same number of friends.

Proof idea: two cases: no person knows everyone; some person knows everyone. Then there will be 0..18 possible # of friends in the first case, and 1..19 number of friends in the second case. Apply PHP in each case.

Pigeonhole Principle (v3): If $f: A \rightarrow B$ where A and B are finite sets, then some point $y \in B$ must have at least $\lceil\ |A| / |B|\ \rceil$ preimages under f.

Eg 3: if 100 pigeons roost in 30 holes, some hole has at least 4 pigeons roosting therein.

More examples:

Ex 4: Given five points inside the square whose side is of length 2 feet, prove that two are less than 1.5 feet apart.

Proof idea: divide square into four 1 x 1 cells. The diameter of each cell is $2^{0.5}$ which is less than 1.5.

Ex 5. Prove that for any five points on a sphere, some four must lie on the same hemisphere. Assume that the boundary of the hemisphere is on both hemispheres.

Proof: choose any two of them and draw the great circle route that connects them (take a plane cutting through those two points and the center of the sphere, and see where it intersects the sphere). Three points remain. Two must be on one side of the sphere; one will be on the other. The two points on one side of the sphere, together with the two equatorial points on the great circle, are four points within the same hemisphere.

Ex 6: In any list of 10 integers $a_1, ..., a_{10}$ there's a subsequence of consecutive numbers whose sum is divisible by 10.

Consider the ten sums

$s_1 = a_1$

$s_2 = a_1 + a_2$

...

$s_{10} = a_1 + a_2 + \ldots + a_{10}$

numbers in the list. If any of these divisible by 10, then we are done. Otherwise, each is congruent to a number between 1 and 9 mod 10. So two of these values are congruent to the same number (mod 10): $a_i = a_j$ (mod 10) with $i < j$. Eg, maybe

$s_3 = a_1 + a_2 + a_3$

and

$s_5 = a_1 + a_2 + a_3 + a_4 + a_5$

are both congruent to 7 (mod 10). But then $s_5 - s_3 = 0$ (mod 10), which would mean that $a_4 + a_5 = 0$ (mod 10) . Generalizing,

$a_{i+1} + \ldots + a_j = 0 \quad (\text{mod } 10)$

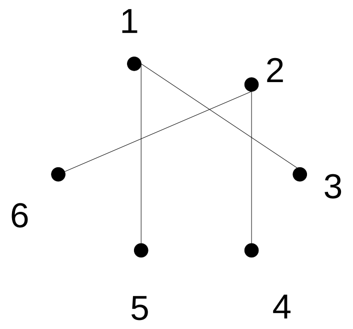
Ex 7: Devon picks 7 different numbers from {1, 2, 3, . . . , 10, 11}. Prove some pair adds up to 12.

Proof. Consider the following partition of Devon's available numbers:

{1, 11}, {2, 10}, {3, 9}, {4, 8}, {5, 7}, {6}.

Since there are only 6 sets in this partition, two of Devon's 7 numbers must be in the same set, hence that set is not {6}. But all the sets with two numbers have elements summing to 12.

Ex 8. (repeated from beginning of term) In any room of 6 people, there are 3 mutual friends or 3 mutual strangers (Ramsey theorem, $R(3,3)=6$)



Consider person #1. Five people are either friends or non-friends with person #1. At least one of those sets has 3 people.

- 3 people are non-friends with 1. If two of them don't know one another, we are done. If all three know one another, we are done.
- 3 people are friends with 1. If two of them know one another, we are done. If all three don't know one another, we are done.

$R(4,4) = 18$ (1955)
$R(5,5)$ open!

The Division Theorem. If n is any integer and d is a positive integer, there exist <u>unique</u> integers q and r such that $n = dq + r$ and $0 \leq r < d$.

One can prove the Division Theorem from the well-ordering principle. The trick is to define $R = \{n - iq: i \in \mathrm{N} \text{ and } n - iq > 0\}$. Then R is a nonempty subset of the natural, so it has a least element. That least element is the r, and the i that goes with it is the d, and you can establish that they're unique.

The integral quotient q when you divide n by d has a nice notation in Python 3 – it's $q = n$ // d (the "**s**"). The remainder r has a nice notation in lots of languages – it's $r = n \% d$ or $r = n \bmod d$.

3 // 2 = 1   -3//2 = -2   3%2 = 1   -3%2 = 1
2 // 3 = 0   -2//3 = -1   2%3 = 0   -2%3 = 1

The name "floor division" provides an easy way to understand (and to compute) stuff like -3/2. Think of computing the quotient in the reals, and then go to the left—take the floor—until you hit an integer.

Notation: $d \mid n$ if there exists i such that $di = n$.
Alternative, if as $n \bmod d = 0$

$\mathbb{Z}_n = \{0,1,\ldots n\text{-}1\}$ with an operation of addition modulo n.
This is called "the group of integers modulo n"

There are variant notations for mod that actually correspond to **different ways of thinking about it.** Recall the circle of n points, moving right for increment and left for decrement, as a way of thinking of arithmetic in the world mod n. Conceptualizations:

$23 \bmod 5 = 3$	as a **binary operator**
$23 \equiv 3 \pmod 5$	23 is in the **same equivalence class** as 3 with respect to $a \equiv b$ true when $5 \mid a - b$
$1+4 = 0$ in $\mathbb{Z}_5$	Forget the integers, forget thinking of mod as an operator: we are working in **the group** $\mathbb{Z}_5$. I like this way.

Greatest common divisors

For a and b positive numbers, let's define
$\gcd(a, b)$ = the **largest** integer d such that $d \mid a$ and $d \mid b$.

Well defined because d is **at most** $\min(a, b)$

Define $\gcd(a,0) = \gcd(0, a) = a$ for any **nonzero** a.

Can you figure out the gcd of a and b when they're in factored form? Sure.

It's the product of p^i values where p occurs in the factorization of both a and b and i is the smaller of the two exponents. It is 1 if there are no common primes.

Of course gcd(a, b) = gcd(b, a)

Claim: gcd(a, b) = gcd($a – b$, b) = gcd(a mod b, b)
This gives rise to an efficient algorithm to find the gcd.
It's called **Euclid's algorithm.**

Let's prove Euclid's algorithm.
If $d \mid a$ (so $a = i\, d$) **and** $d \mid b$ (so $b = j\, d$) then
$a – b = i\, d – j$ d $= d\, (i – j)$ whence $d \mid a – b$.
So all the divisors of a and b are divisors of $a – b$ and b.
Similarly, all the divisors of $a – b$ and b are divisors of a and b.
Since they have the exact same set of divisors, their gcd's are the same.

Example: Compute gcd(360, 1000).
gcd(1000,360) = gcd(360,280) = gcd(280,80) = gcd(80,40) = gcd(40,0)=40.

If you do some extra bookkeeping when you compute the gcd you can find, when you compute gcd(a, b), two numbers x and y such that
$ax + by$ = gcd(a, b)

That the algorithm finds them tell you that they always exist ;-)

Theorem: Given numbers a and b, not both 0, we can find integers
x and y such that $ax + by$ = gcd(a ,b).

This is useful!

Numbers a and b are said to be **relatively prime** if gcd(a, b) = 1. This is the same as saying that they have no common prime factor.

Note that if p is prime and a is in [1 .. $\underline{p}$-1] then gcd(a, p) =1. That is, a prime is relatively prime to any positive number less than then itself.

Suppose gcd(a, n) = 1.
Then the theorem above promises an x, y such that $ax + ny = 1$. Suppose we're doing this is $\mathbb{Z}_n$ (that is, take "mod n" of both sides). In that group,

n=0, so ny=0, so ax = 1. In other words, x is the inverse to a. That is, a has a multiplicative inverse mod n.

Thus we have: Every element relatively prime to n has a multiplicative inverse in $\mathbb{Z}_n$.

Eg: In $\mathbb{Z}_{10}$ the numbers 1, 3, 7, 9 have multiplicative inverses. Make a multiplication table for them. Note closure: if a and b are both relatively to n then so is their product.

Eg: In the world of integer operations mod 2^{32}, all the odd numbers have multiplicative inverses.

$\mathbb{Z}_n^* = \{a \in [1..n]: \gcd(n, a)=1\}$ is a group.

Example: Make a multiplication table for $\mathbb{Z}_{10}^*$

A general fact about finite groups is that, for any group G, $a^{|G|} = 1$.
This is called **Lagrange's Theorem.** I'm not going to prove this, but you will certainly prove it if you take a class that spends some time on group theory.

A particularly interesting group is $\mathbb{Z}_p^* = [1..p\text{-}1]$ where p is a prime. By Lagrange's theorem, $a^{p-1} = 1$.
This actually provides a **test for primality**.
If $a^{p-1} \neq 1 \pmod{p}$ then you **know** that p is **not** prime. This is "Fermat's test for primality".

We don't know for sure that p is prime just because $a^{p-1} = 1$. Yet composite numbers for which $2^{p-1} = 1 \pmod{p}$ are rare (although there are infinitely many). There are even nasty number, **Fermat pseudoprimes**, where $a^{p-1} = 1 \pmod{p}$ for all $a \neq 1$.

Another interesting fact about $\mathbb{Z}_p^*$ is that it's **cyclic**: there will be an element g in this group such that $\mathbb{Z}_p^* = \{g^0, g^1, \ldots, g^{p-2}\}$. In words, you can create the entire group from g by just starting with 1 and repeatedly multiplying by g.

Do an example to figure out what are the generators of $\mathbb{Z}_5^*$

If you have g and p and i, it is easy to efficiently compute $g^{\,i-2}$. You don’t have to repeatedly multiply g by itself i-1 times: you can do much better. For an efficient algorithm, try “powering up”, computing g, g^2, g^4, g^8, …

While it is easy to compute g^i mod p from g, i, and p, nobody knows an efficient algorithm for computing i from g, p, and g^i mod p . Computing it is known as the **discrete log problem**. There is a discrete log problem for any finite cyclic group. In the group we’ve been talking about, we think it is **hard**. That is, we think there is no efficient way to solve this problem. It is an example of supposed **one-way-function**. Something that is easy to compute in one direction, but apparently hard to compute in the other direction.

We are now ready to do a little **cryptography**!

Alice wants to communicate with Bob. They first meet and agree on a great big prime number p and a generator g for $\mathbb{Z}_p{}^*$. To generate a big prime number they can generate a random big number and test if it is prime. Testing if $2^{\,p-1} = 1$ (mod p) is actually enough to give high confidence that the random p is prime. It is also easy to test if a given value, say g=2, is a generator. If its not, Alice and Bob could just choose a different prime. The number 2 will often be a generator for a random large prime p.

Alice computes a random a in [0..p-2] and send Bob $A = g^{\,a}$. Bob compute a random b in [0..p-2] and send Bob $B = g^b$. Everything is done mod p.

$A = g^a$.

Alice ---------------------------------> Bob

$B = g^b$

<-------------------------------

At this point Alice can compute $B^a = (g^b)^a = g^{ab}$
At this point Bob can compute $A^b = (g^a)^b = g^{ab}$
So they share the same “secret”.
On the other hand, the adversary E watching all of this knows g, p, A, and B, but, try as it may, there is no clear way for it to compute from this g^{ab} .
It can try computing AB, for example, but that’s g^{a+b}, not g^{ab}.
It could try to compute a and b, but that’s the discrete log problem, which we think to be computationally intractable.

Now that Alice and Bob share a secret $K = g^{ab}$ they can use it to send messages to one another. For example, if Alice and Bob have only a bit m they want to communicate, they could try something like $c = m$ xor lsb(K), where lsb(x) denotes the rightmost (least-significant) bit of K, as a way to communicate m. If the message M is long, Alice could send a message M by transmitting M xor H(K) for some (non-secret) “hash function” H.