Instructions: For pre-classes, make sure you know how to do each part, but I will not collect work from all of you.

Instead, we will prepare a solution jointly. Each person does the problem(s) on one panel of this whiteboard. If there's a problem that everyone struggles with, or is extra challenging, I'll do that.

You can directly write on the whiteboard (by hand, by digital pen/stylus, with a mouse, etc.) or type in LaTeX and import a picture version of the typed solution. Whatever you choose is good for me as long as it's readable.

Make sure to write your name (first name is enough) on your solution page.

In addition to your final work/answer, make sure to put a note about how you thought about the problem, especially if it's not clear from the solution.

For example, if you did something by brute force (say, Sieve method), you can say "I coded Sieve method in Python, here's a snippet/pseudocode of my code."

Or you can note another approach you tried but why it didn't work and how you then later improved that approach. Because a method which didn't work in one situation could actually be just the right thing in another.

Finally, we'll all go check out what others have said and maybe ask them questions or put comments/thoughts on their work.

Divisibility and primes; Problems 1&2.

Sieve method: 30 = 30 * 1, 30 = 15 * 2, 30 = 10 * 3, 30 = 6 * 5. So 1, 2, 3, 5, 6, 10, 15, and 30 divide 30.

We also see that 0 does not divide 30 as for any integer k, 0 * k = 0 =/= 30.

We know that the prime factorization of 30 to be 2 * 3 * 5. For each divisor of 30 we see that they are a combination of these factors (except for 1). Note here that such combinations do not use repeats.

4, 8, 12, 16, 20, 24, and 28 do not divide 30 as each of these require (at least) two 2s in the prime factorization. Similarly, 7, 9, 11, 13, 14, 17, 18, 19, 21, 22, 23, 25, 26, 27, and 29 do not divide 30 as they require primes outside the

It looks like there are two methods to find all divisors of a number I'm seeing here. One, starts with testing numbers 1, 2, 3, 4, 5, ... to see if they divide 30, and then pairs each divisor r with 30/r.

The second is where we use the prime factorization, 30=2*3*5. I will modify the approach after that a bit. Any factor of 30 then will be of the form 2^r1*3^r2*5^r3 (what should r1, r2, r3 be?).

Let a be some non-zero integer. We know a|0 as 0 = 0 * a. We know a|a as a = 1 * a. We also know 1|a as a = 1 * a. Finally we know that 0|a as for any integer k we know that a =/= 0 * k

For "not equal", "not divides", etc. LaTeX notation could work: e.g. a not= 0*k. LaTeX does this as \neq and in some cases you need \not (symbol), like \not | for "does not divide" (although there's better looking alternatives for DND).

This reminds me of a really neat problem I saw once. It's going into the homework 1 in a few minutes.

- Miah :)

Divisibility and primes; Problems 3&4

3

Zero does divide zero by our definition of divides because for example 0 * 3 = 0 and 3 is an integer

But are there other conditions in the definition that might not work?

Ah it says a must be non zero opps.

4

23 is prime because none of the numbers 2, 3, ... , 21, 22 divide 23

91 is a composite number because 91 = 7 * 13

101 is a prime because none of the numbers 2, 3, 4, 5, 6, ... , 49, 50, 51 divide 101. Its enough to just check the first half of the numbers.

Ah, I see. You shortened the list to the first half. Can we do better?

I mean you could use the ceiling of the square root (OR: floor? It's only one number, so it doesn't make too much of a difference, but it still cuts down the work.)

Ah, this is a good exercise. To check that n is prime, do we have to test that all 2, 3, ..., n-1 divide n? Could we shorten the checking?

Or you can realistically do this by checking a primes list

Recursively defined sequences; problem 1

The first 16 Fibonacci numbers are: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610.

Just used brute force to find these values.

Or, we can steal them from a webpage ;) https://www.mathsisfun.com/numbers/fibonacci-sequence.html

Michael

Recursively defined sequences; problem 2

The sums of the first Fibonacci numbers are: 1, 2, 4, 7, 12, 20, 33. Found by brute force.

These numbers appear to correspond to one less than future fibonnaci numbers.

Example:
f_0 + f_1 + f_2 + f_3 = f_5 - 1 = 4

Conjecture: $$\sum_{i=0}^{n} f_i = f_{n+2} - 1$$

Everyone: Any guesses which method would be good to use for proving this conjecture? (No need to do the proof.)

Probably induction

-Nick

Congruences; Problem 1

i just used the Sieve method and applied the definition of congruence modulo n to solve these

(a) $23 \equiv 2(mod 7)$ since $23 - 2 = 21$ and $7|21$

(b) $2 \equiv 23(mod 7)$ since $2\text{-}23 = \text{-}21$ and $7| - 21$ (and congruence modulo n is an equivalence relation)

(c) $2 \not\equiv -2(mod 5)$ since $2 - (-2) = 4$, and 5 cannot divide 4.

(d) $-1 \equiv -9(mod 5)$ since $-1 - 9 = -10$ and $5 \cdot -2 = -10$ which means $5| - 10$

(e) $-3 \equiv -8(mod 5)$ since $-3 - (-8) = 5$, and every integer divides itself.

Modulo notation hint: I use a\equiv b \pmod{n} for modulo notation in LaTeX. I think it needs amsmath package.

Aha! There's a neat side note here: "And congruence is an equivalence relation." Everyone but Joe: What does this imply/mean in this context?

For all integers a,b,c and ~ means congruent.
1. a ~ a. 2. if a ~ b and b ~ c then a ~ c
3. if a ~ b then b ~ a

A note on screenshots: if you make the image very big on your screen first (full-screen or zoom in on overleaf) and then screenshot, it'll be a less blurry image. -Nick

Joe

Congruences; Problem 2

Find 3 to 4 valid congruences on your own.

Let ¬ denote congruence :) (couldn't find congruence symbol on Mac).

I like stealing symbols too. Google and find a page with the symbol, then copy-paste: ≡. (I couldn't find a page with a list for Mac symbols easily. That might be another option.)

2 ¬ 14 (mod 3). Let a=2, b=14, and n=3. The statement holds so long as n|(a-b), and by substitution, 3|(2-14) = 3|(-12), which is true.

18 ¬ -15 (mod 11). Let a=18, b=-15, and n=11. Using a similar argument as prior, 11 divides [18-(-15)], therefore our congruence statement holds true. :)

5 ¬ 4 (mod 1). Let a=5, b=4, and n=1. Using a similar argument as before, we can substitute our values, giving us 1|(5-4) = 1|1, which is true.

Aha! This sounds like any two integers are congruent mod 1. That is boring! For that reason, some people exclude n=1 as a possible mod n option (see Wikipedia about mod, and the textbook).

11 ¬ 15 (mod 4). Let a=11, b=15, and n=4. By definition of congruence, a is congruent to b modulo n if n|(a-b). By substitution, we know that 4|(11-15) = 4|(-4), which holds true.

Mirza

Congruences; Problem 3

A

First set of congruent numbers: 3K
Second set of congruent numbers: 3K+1
Third set of congruent numbers: 3K+2

First set of congruent numbers: 0, 3, 9, 12, 21, -3.
Second set of congruent numbers: 1, 4, 7, 22, 100, -2, -5
Third set of congruent numbers: 2, 5, 20, -1, -4

This is because all of the numbers in a group differ by 3

B

We see there are 3 residue classes modulo 3 because every integer can be written as 3K + either 0, 1, or 2. There are three possible remainders

These possible remainders (0, 1, 2) are great class representatives

If we generalize this to n, what numbers will form the representatives? This set will be called the least residue system modulo n. For each a mod n, there is a unique least residue that it's equivalent to.

There are n residue classes with 0, 1, 2, ... , n-1 as their representatives

Cian