MATH 240 Module 7: Number Theory

due Friday, 7 April 2023

Learning Goals

- Understand and apply the definition of the greatest common divisor of natural numbers
- Use the Euclidean Algorithm to compute the greatest common divisor
- Use the Extended Euclidean Algorithm to compute modular inverses
- Solve modular equivalences using modular inverses
- Apply Fermat's Little Theorem to test for primality

Submission

You should submit:

- A PDF with your answers to the exercises (you may either type your answers and export as a PDF, or write your answers by hand and scan them using an app such as GeniusScan or CamScanner).
- `module7.disco`. Some of the exercises on this module require you to write Disco code, but unlike previous modules, I have not given you a starting `.disco` file to fill in. You should create your own `.disco` file (feel free to use `.disco` files from previous modules as examples/templates). You are not required to write any documentation or tests for your functions, although you are encouraged to do so since you may find it helpful.

Exercises

Exercise 1 Fill in the blank in the following analogy, and explain your reasoning:

$a \mid b$ is to gcd as $a \leq b$ is to _____.

Exercise 2 This exercise concerns the Euclidean algorithm for finding greatest common divisors.

(a) Use the Euclidan algorithm to find $\gcd(1, 5)$, $\gcd(123, 277)$, and $\gcd(78, 104)$. Be sure to show the steps of the process, not just the final result.

(b) Write a Disco function to find the GCD of two natural numbers using the Euclidean algorithm. Use it to check your answers from part (a).

(c) Use your Disco function to find $\gcd(518303142726377580, 169429189188136020)$.

(d) (**Optional challenge, +1/2 token**) Write a Disco function implementing the *extended* Euclidean algorithm, which finds not only the GCD of a and b, but also integers s and t such that $sa + tb = \gcd(a, b)$. That is, define a function $egcd : \mathbb{N} \times \mathbb{N} \to \mathbb{Z} \times \mathbb{Z} \times \mathbb{N}$ such that if $egcd(a, b) = (s, t, g)$, then $sa + tb = g$ and g is the GCD of a and b. Some hints:

- Start by writing a recursive helper function

  $$egcdH : (\mathbb{Z} \times \mathbb{Z} \times \mathbb{N}) \times (\mathbb{Z} \times \mathbb{Z} \times \mathbb{N}) \to \mathbb{Z} \times \mathbb{Z} \times \mathbb{N}$$

  which takes the previous and current rows of the table and returns the last row of the table (the row containing the GCD of a and b).
- Then implement $egcd : \mathbb{N} \times \mathbb{N} \to \mathbb{Z} \times \mathbb{Z} \times \mathbb{N}$ simply by calling *egcdH* with the right values for the starting rows.

Exercise 3 Solve each of the following modular equivalences for x, giving your answer in the form $x \equiv_m k$, where $0 \leq k < m$.

(a) $34x \equiv_{89} 77$

(b) $5x + 17 \equiv_{23} 2x - 10$

(c) $200x - 13 \equiv_{1001} 0$

Note: if you have completed Exercise 1(d), you may simply use your *egcd* function to find modular inverses (you may even want to define a specialized *modinv* function, using *egcd*, such that $modinv(a, m)$ finds the modular inverse of a modulo m). Otherwise, you must show the work you do to find modular inverses, either by guess-and-check, or using the Extended Euclidean Algorithm.

Exercise 4 For each number n below, *either*:

You should be able to copy-paste the numbers from this PDF!

- find one value of a for which $a^{n-1} \not\equiv_n 1$, *or*
- list at least five values of a such that $a^{n-1} \equiv_n 1$.

In either case, explain what you can conclude about n.

(a) 1399499

(b) 340561

(c) 706113762068412435747683199935230839398684490635512212296530712933315635896349355029272628861810919

(d) 562829045905787729180918245038123892769731482213392342116937806292214008149873442413311203285481229 3

Exercise 5 (**Optional challenge, +1 token**)

You are tracking an international marshmallow smuggling network and have intercepted the following messages, both encrypted using RSA. Below are listed the messages as well as the public key values that were used to encrypt them.

- Message 1:
  - $C_1 = 312115978447989584283633188854752057647191756617524290624257761141431129414132055344386662286347170$
  - $e_1 = 5$
  - $n_1 = 954545170617160542923898352893685694075863345718644782466048593569308115804495403873438952349696261$
- Message 2:
  - $C_2 = 343522738085272859925174198387338839798820425207705001175435596220515506999711321990562235356688401$
  - $e_2 = 3$
  - $n_2 = 238085372419925968880235276059166080445988352134804577897127160405499646738919403547126468565128865 3$

You suspect that the smugglers were lazy/incompetent and reused one of the primes they used to generate the two keys. Take advantage of their incompetence to break their encryption and find out the time of their next big meeting, when you will be able to catch them all at once. What is significant about their meeting time?