MTH 225: Discrete Structures for Computer Science 1

Daily Preparation, Module 2B: Exponentiation mod n and more applications to cryptography

Due by: 11:59pm ET, Tuesday, September 15

Estimated time requirement: About 45-60 minutes for the whole assignment. *If you have worked on this assignment for 30 minutes and you're not at least halfway done, DON'T work any further* — instead, *stop and ask for help* on the `#dailyprep` channel on CampusWire. Remember these are graded just on completeness and effort — try to be right and understand everything, but don't get bogged down if you get stuck. Just give a good effort and move on, and ask a question.

Overview

In Part A of this module, we learned about the *modulus* operator and how to do addition "mod n" along with subtraction and multiplication mod n, and representing negative numbers mod n. Part B of this module explores one other arithmetic operation done using the modulus operator: Raising integers to powers, or "exponentiation". Exponentiation turns out to be a fundamental building block of many cryptographic systems. In this lesson, we'll learn an algorithm for computing large powers (for example, 12^{100}) mod n, in a very efficient way — for example, although 12^{100} has 108 digits, computing `12^{100} % 10` never uses integers larger than 2 digits. We'll apply this to a cryptosystem that encrypts messages by converting letters to integers like the shift cipher, but then uses exponentiation rather than shifting.

What you will learn

Learning Targets addressed in this module:

- A.3: I can compute `a % b` given integers `a` and `b` and perform modular arithmetic.

BEFORE your class meeting, use the Resources for Learning (below) to learn how to do the following:

- Use the repeated squaring algorithm to evaluate powers of integers mod n.

DURING AND AFTER your class meeting, you will learn how to do the following:

- Apply the repeated squaring algorithm to cryptography techniques for encryption (RSA) and key generation (Diffie-Hellman).

Resources for Learning

Text: There's no assigned text to read this time; the videos do it all. However, as always, Google searches will turn up a *lot* of material on the web in case you are needing more help. (However I found most of the search results were highly technical, so I made the video below to break it down to the beginner level.)

Video: Just one video to watch – "The repeated squaring algorithm" (15:25) which is posted in the Module 2 folder.

You are free to search for and use other resources in addition to, or instead of the above, as long as you can work the exercises below.

Exercises

The exercises for this assignment are found at Classkick this time. Go to http://app.classkick.com and sign in (as a "Portfolio" student). If prompted, use the code `SQV ONX`.

Submission, grading, and getting help

Submitting your work: Your work is to be done on Classkick using the link/code above. Classkick saves your work as you go, so there's nothing to submit – just do the work and you're good.

How this is graded: Daily Prep assignments are graded on the basis of *completeness and effort*: If your submission has **all parts completed** (no blank entries, even if left blank accidentally) and **a good-faith effort to provide a correct solution or explanation is given** (no responses of "I don't know" or "I didn't understand") and **the work is submitted on time**, it gets a "check". Otherwise it gets an "x". If you are stuck on an item, you're expected to ask questions and give your best effort.

Getting help on this assignment: *You may work with others on this assignment, but you may not copy each others' answers.* Evidence of copying will be treated as academic dishonesty. You may also ask questions on the #dailyprep channel on CampusWire, but you may not ask simply to be given the answers; giving and receiving answers on CampusWire will be treated as academic dishonesty.