Encryption via Exponentiation

Reading

- Section 9.4

Practice Problems

9.4 1, 2

Notes

Encryption via Exponentiation

Exponentiation and Euler's and Fermat's theorems offer us a new way to encrypt and decrypt messages.

The key to this process is looking at two numbers that are inverses modulo $\phi(n)$, using one of them for encryption and the other for decryption. Let's see how this might work:

We will consider $n = 29$, which is prime. Then $\phi(n) = 28$. We need two numbers that are inverses modulo 28. They must be relatively prime to $28 = 2^2 7$, so let us use $e = 5$ for the one number. The Euclidean algorithm tells us that $5 \cdot 17 \equiv 1 \bmod 28$, so $d = 17$ is its inverse.

To encode a message, we will convert each letter to a number, then raise it to the 5th power. Suppose we take the word MATH, which in numbers is "13 1 20 8". We now raise each of these to the 5th, modulo 29 to get: "6 1 24 27". We then transmit those 4 numbers, converted to letters and with a convention on what to use for "27" (maybe some punctuation).

To decode, we raise to the 17th power and compute modulo 29. You will see we get our numbers back! Use our techniques of fast exponentiation to do that.

> Use this method to encode the same word, MATH, in a system where $n = 40 = 2^3 5$. In this case $\phi(n) = 16$. You will need to choose two numbers d and e such that $de = 1 \bmod 16$.

Theoretical Description

> Suppose we want to encrypt modulo n, and d and e are such that $de = 1 \bmod \phi(n)$. Then we encrypt via:
>
> $$c = x^e \bmod n$$
>
> and decrypt via:

$$x = c^d \bmod n$$

The fact that these are inverse processes to each other follows from Euler's Theorem:

- Since $de = 1 \bmod \phi(n)$, there is a k such that $de = 1 + k\phi(n)$.
- Then $c^d = (x^e)^d = x^{de} = x^{1+k\phi(n)} = x \cdot (x^{\phi(n)})^k = x \bmod n$.