MATH 314 Fall 2019 – Class Notes

September 16, 2019

Scribe: George Watson

Summary: The One Time Pad and the beginnings of Elementary Number Theory used in modern cryptography, through the extended Euclidean Algorithm.

Notes:

The One Time Pad

Is basically a Vigenère cipher with one main difference – the key has the same length as the plaintext and is a completely random string of letters

Only use the key one time, hence the name

Even an all a's chosen plaintext attack won't help. You will discover the key, but since that key will never be used again, it is not useful.

This is a mathematically unbreakable cipher. So why not always use it?

Downsides:

Need a new key for every message sent

Key size is the same as the size of the message being sent – raises issues of data transmission and storage

Using the key more than once does render the one-time pad vulnerable to attack.

Elementary Number Theory – studies properties of the integers and, in particular, the primes

Frequent problem: find the greatest common divisor of two integers: $\gcd(a,b)$

One idea – **trial division**. Try dividing both a and b by all the numbers up to $\min(a,b)$. Then take the largest number that divides both.

How long does this take?

Let $n = \min(a,b)$ and let x = the number of bits required to write n in base 2.

Then $x = \lceil \log_2(n) \rceil$

If we do n trial divisions, then running time = $O(n) = O(2^x)$, which is an exponential running time → not good! We need to speed things up.

Use **Prime Factorization** – factor *a* and *b* into primes and find the gcd by taking the product of all of the primes that divide both *a* and *b*.

To do this with large numbers is difficult

The fastest known algorithm for factoring an *x* – bit number runs in $O\left(e^{\sqrt{x \ln x}}\right)$ time (~quasi-exponential time) which is sub-exponential time but still much slower than polynomial time. We need something faster.

The Euclidean Algorithm – 3 millenia old and surprisingly fast

Running time is $O(x) = O(\log n)$

Key idea – division with remainder

Fact 1(has been proven): Given any two positive integers *a* and *b*, there exist two other integers *q* and *r*, with $a = bq + r$ AND $0 < r < b$.

Proof:

Given *a, b*, compute $q = \left\lfloor \frac{a}{b} \right\rfloor$ and then compute $r = a - bq$. (Note: $r + bq = a$)

Need to show that $0 \le r < b$.

Observe that $\frac{a}{b} - 1 < \left\lfloor \frac{a}{b} \right\rfloor \le \frac{a}{b}$

And now $\frac{a}{b} - 1 < q \le \frac{a}{b}$

So that $a - b < bq \le a$ (multiplying through by *b*)

Now $bq \le a \Rightarrow 0 \le a - qb = r. \quad \therefore 0 \le r$

And $a - b < qb \Rightarrow r = a - qb < b. \quad \therefore r < b.$

Q.E.D.

Fact 2: If *d* divides *n* and *m*, then *d* divides $n + m$ and $n - m$.

Suppose $d = \gcd(a, b)$. Write $a = bq + r$ and consequently $r = a - bq$. Then *d* has to divide *r*. In fact, $\gcd(a, b) = \gcd(b, r)$. Apply this fact recursively until remainder = 0 appears. Then $\gcd(a, b)$ is the last nonzero remainder obtained in the division process.

Example 1: Find the gcd of 158 and 38.

$$158 = 4 \cdot 38 + 6$$
$$38 = 6 \cdot 6 + 2$$
$$6 = 2 \cdot 3 + 0$$

So that $\gcd(158, 38) = \gcd(38, 6) = \gcd(6, 2) = 2$

So finding **gcd**'s is a lot faster than expected using the Euclidean algorithm, and we will exploit this fact repeatedly in subsequent work.

The Euclidean Algorithm Extended

Fact: If $\gcd(a, b) = d$, then there exist integers x and y such that $ax + by = d$

(*d* is a linear combination of a and b. In general, one of x and y will be positive and the other will be negative.**)**

Euclid's Algorithm tells us how to find x and y by working backwards through the algorithm. Start with the equation that gave the last nonzero remainder and solve for that remainder.

$2 = 38 - 6 \cdot 6$ Solve the next equation for its remainder and substitute in.

$2 = 38 - 6(158 - 4(38))$ Simplify and continue until x and y appear.

$$2 = 38 - 6(158) + 24(38))$$
$$2 = 25(38) - 6(158)$$
$$2 = -6(158) + 25(38)$$

$\therefore x = -6,\ y = 25$

Example Find x and y so that $72x + 25y = 1$

First use the Euclidean Algorithm forward to find $\gcd(72, 25)$

$$72 = 2 \cdot 25 + 22$$
$$25 = 1 \cdot 22 + 3$$
$$22 = 7 \cdot 3 + 1$$
$$3 = 3 \cdot 1 + 0$$

Now work backwards to find x and y

$$1 = 22 - 7 \cdot 3$$
$$1 = 22 - 7(25 - 1 \cdot 22)$$
$$1 = 22 - 7(25) + 7(22)$$

$$1 = -7(25) + 8(22)$$
$$1 = -7(25) + 8\big(72 - 2(25)\big)$$
$$1 = -7(25) + 8(72) - 16(25)$$
$$1 = 8(72) - 23(25)$$
$$\therefore x = 8 \text{ and } y = -23$$

Next, we will use this process to find inverses of numbers in modular arithmetic systems.