Modular Arithmetic Essentials

John York

Blue Ridge Community College

Fall 2019

Modular addition “wraps”

- In a 12-hour clock, what is 6 hours after 10 o’clock?
  - 10 + 6 = 16, but that doesn’t fit inside the 0 - 11 hours on the clock
  - 16 - 12 = 4, so the answer is 4 o’clock -- “wraps”
  - 16 / 12 = 1 remainder 4 (division returns number of wraps; 1 this time)
  - 16 mod 12 = 4 -- the remainder (how far you go after wraps are done)
- With 26 letters, indexed 0 – 25 (starts at 0, not 1)
  - 15 + 20 = 35, but that isn’t in 0 - 25
  - 35 - 26 = 9, so 15 + 20 “wraps” to 9
  - 35/26 = 1 remainder 9
  - 35 mod 26 = 9 -- the remainder

Modular subtraction also "wraps"

- 12-hour clock, what is 10 hours before 2?
  - 2 - 10 = -8 -- that's not between 0 and 11
  - -8 + 12 = 4 -- 4 o'clock is 10 hours before 2
  - -8 mod 12 = 4
- 26 letters, indexed 0 - 25
  - 2 - 10 = -8
  - -8 + 26 = 18
  - -8 mod 26 = 18

Modular multiplication (1)

- Multiplication jumbles things a little--handy for encryption

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
	A	B	C	D	E	F	G	H	I	J	K	L	M	M	O	P	Q	R	S	T	U	V	W	X	Y	Z
3*index	0	3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51	54	57	60	63	66	69	72	75
(3*index)mod 26	0	3	6	9	12	15	18	21	24	1	4	7	10	13	16	19	22	25	2	5	8	11	14	17	20	23
	A	D	G	J	M	P	T	V	Y	B	E	H	K	M	Q	Y	W	Z	C	F	I	L	P	R	U	W

- But what happened here?

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
	A	B	C	D	E	F	G	H	I	J	K	L	M	M	O	P	Q	R	S	T	U	V	W	X	Y	Z
4*index	0	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72	76	80	84	88	92	96	100
(4*index)mod 26	0	4	8	12	16	20	24	2	6	10	14	18	22	0	4	8	12	16	20	24	2	6	10	14	18	22
	A	E	I	M	Q	U	Y	B	G	K	O	S	W	A	E	I	M	Q	U	Y	B	G	K	O	S	W
13*index	0	13	26	39	52	65	78	91	104	117	130	143	156	169	182	195	208	221	234	247	260	273	286	299	312	325
(13*index)mod 26	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13
	A	N	A	N	A	N	A	N	A	N	A	N	A	N	A	N	A	N	A	N	A	N	A	N	A	N

Modular Multiplication (2)

- If multiplier and the modulus share a common divisor,
  - Multiplication “wraps” onto same space, over and over
  - Example: (13 * index) mod 26
  - 13 and 26 are both divisible by 2, so 2 is common divisor
  - 0 and 13 are the only answers we get
- Not good for encryption
- But if the modulus were 29 instead of 26…
  - 29 is a prime number, it is only divisible by 1
  - We could multiply by any number 0 - 28 without problems

Greatest Common Divisor (GCD)

- Take two numbers
- GCD is the largest number that can divide both
  - gcd(2, 26) = 2
  - gcd(18, 26) = 2
  - gcd(13, 26) = 2
  - gcd(3, 26) = 1 -- no common divisor, relatively prime
- GCD = 1 means the two numbers
  - have no common divisor
  - are relatively prime
- Euclid developed a method for finding GCD over 2,000 years ago

Modular Inverse

- Division doesn’t work in modular arithmetic
  - 3 / 5 is a fraction, modular arithmetic only has integers
- Instead use modular inverse
  - In real numbers 3 * 1 / 3 = 1 so 1 / 3 is the inverse of 3
  - 3 * (mod inverse of 3) = 1
  - use wrapping--there must be some number that wraps to 1
  - 3 * 9 = 27,   27 mod 26 = 1
  - So 9 is the mod inverse of 3 <u>when you are using mod 26</u>

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
	A	B	C	D	E	F	G	H	I	J	K	L	M	M	O	P	Q	R	S	T	U	V	W	X	Y	Z
3*index	0	3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51	54	57	60	63	66	69	72	75
(3*index)mod 26	0	3	6	9	12	15	18	21	24	1	4	7	10	13	16	19	22	25	2	5	8	11	14	17	20	23
	A	D	G	J	M	P	T	V	Y	B	E	H	K	M	Q	Y	W	Z	C	F	I	L	P	R	U	W

Mod Inverse requires gcd(number, mod) = 1

- If the number and the modulus have common factors, no inverse exists.

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
	A	B	C	D	E	F	G	H	I	J	K	L	M	M	O	P	Q	R	S	T	U	V	W	X	Y	Z
4*index	0	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72	76	80	84	88	92	96	100
(4*index)mod 26	0	4	8	12	16	20	24	2	6	10	14	18	22	0	4	8	12	16	20	24	2	6	10	14	18	22
	A	E	I	M	Q	U	Y	C	G	K	O	S	W	A	E	I	M	Q	U	Y	C	G	K	O	S	W
13*index	0	13	26	39	52	65	78	91	104	117	130	143	156	169	182	195	208	221	234	247	260	273	286	299	312	325
(13*index)mod 26	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13
	A	N	A	N	A	N	A	N	A	N	A	N	A	N	A	N	A	N	A	N	A	N	A	N	A	N

- gcd(4, 26) = 2
  - There is no “1” in the (4*index)mod26 line
- gcd(13, 26) = 13   No “1” in that line either

Computing Modular Multiplicative Inverses

- Brute force
  - Try every number until a*i mod(n) = 1

  (Should also check gcd(a, n) == 1 before start to ensure inverse exists)
- Extended Euclidean Algorithm
  - Efficient algorithm
  - Available in cryptomath.py module
    - findModInverse(a, b)

```
>>> a = 11
>>> n = 26
>>> for i in range(n):
        if a * i % n == 1:
                break
>>> i
19
>>> 19*11 % 26
1
```

```
>>> from cryptomath import findModInverse
>>> findModInverse(11, 26)
19
>>>
```

Python

- Python operators
  - % is the modulo (mod) operator, 97 % 6 will return 1
    - 1 is the remainder when 97 is divided by 6
  - // is the integer division operator, 97 // 6 will return 16
  - 16 * 6 + 1 = 97 (quotient * modulus + remainder gives us the initial number)
- cryptomath function
  - gcd(97, 6) = 1
    - 97 and 6 have no common divisors, relatively prime

Why this is important

- Most encryption uses modular arithmetic
- Multiplication happens *a lot* in encryption
- Modular Inverse happens *a lot* in encryption
- Modular Inverse does not exist unless gcd = 1
  - No common divisors, or relatively prime
- Therefore prime numbers are important in encryption
- Whether or not the inverse exists is important in encryption