1. Definition of Divisibility
2. Division Algorithm
3. Base Conversion

Definition of mod:

$a \equiv b \ (mod\ n)$ ←→ n | (a – b)

$96 \equiv 0 \ (mod\ 12)$

$37 \equiv 2 \ (mod\ 7)$

$37 \equiv 23 \ (mod\ 7)$

In programming mod is a function

37%10 = 7 (in programming)

6%19 = 6 (in programming)

But in math, mod is an equivalence relation. Either two things are equivalent under mod n or they are not. There are an infinite number of items we can substitute for r in the following equation:

$37 \equiv r \ (mod\ 7)$

Valid values for r: 2, 9, 16, 23, 30, 37, 44, -5, -12, etc.

One random note about Programming mod: In C and Java it works one way, but in Python, it works differently…

If $a \equiv b \ (mod\ n)$, then

$a + c \equiv b + c \ (mod\ n)$

$a - c \equiv b - c \ (mod\ n)$

$ac \equiv bc \ (mod\ n)$

Note – can't divide…more on this later.

$f(a) \equiv f(b) \ (mod\ n)$, provided that f is a polynomial function with integer coefficients.

Prove rigorously: if $a \equiv b \ (mod\ n)$, then $a^2 \equiv b^2 \ (mod\ n)$

Proof #1:

Equivalently, by definition of mod, we prove that $n|(a^2 - b^2)$.

$(a^2 - b^2) = (a - b)(a + b)$

Since we know that $a \equiv b \ (mod\ n)$, this means that $n|(a - b)$, thus there exists an integer c such that a – b = nc

$$(a^2 - b^2) = (a - b)(a + b)$$
$$= nc(a + b)$$

Since c, a and b are integers, it follows that $n|(a^2 - b^2)$.

By definition of mod n | (a – b) → There exists an integer c such that a – b = cn → a = b + cn.

$$(a^2 - b^2) = (b + cn)^2 - b^2$$
$$= b^2 + 2bcn + c^2n^2 - b^2$$
$$= 2bcn + c^2n^2$$

$$= n(2bc + c^2n)$$

Since b, c and n are integers, $2bc + c^2n$ is also an integer. It follows that the original expression, $(a^2 - b^2)$, is divisible by n.

Calculating modular exponents

Goal: Find the remainder when a^b is divided by n.

This is a common calculation in Public Key Cryptography, specifically both RSA encryption and El Gamal. (If you want to learn more, take CIS 3362, which is offered every Fall – shameless plug =))

Question: What is the remainder when 97^{203} is divided by 32?

We want the unique value of r such that:

$$97^{203} \equiv r\ (mod\ 32)$$

Note that $97 \equiv 1(mod\ 32)$…

$$97^{203} \equiv 1^{203} \equiv 1(mod 32)$$

Answer: 1

Question: What is the remainder when 68^{25} mod 33.

Note that $68 \equiv 2(mod\ 33)$…

$$68^{25} \equiv 2^{25}(mod 33)$$

How can we calculate this efficiently (by hand or by computer…)

Bottom Up Fast Modular Exponentiation

Key item to note: $(b^4)^2 = b^8$. So basically, if we know b^4 mod n, in one multiplication and mod, we can obtain b^8 mod n. Repeating idea, we can quickly generate values for b, b^2, b^4, b^8, b^{16}, etc. our base raised to different powers of 2.

exp	1	2	4	8	16
2^{exp} %33	2	4	16	25	31

$2^8 = 16 \times 16 = 256 \equiv 15 \pmod{33}$

$2^{16} \equiv 2^8 \times 2^8 \equiv 25 \times 25 \equiv (-8)\times(-8) \equiv 64 \equiv 31 \pmod{33}$

Note: convert 25 to base 2 (binary)

2 | 25 R 1

2 | 12 R 0

2 | 6 R 0

2 | 3 R 1

2 | 1 R 1

25 in binary is 11001

$2^{25} = 2^{16}2^8 2^1 \equiv 31 \times 25 \times 2 \equiv (-2) \times (-8) \times 2 \equiv 32 \pmod{33}$

Question: What is the remainder when 12^{28} is divided by 23?

Note: If you happen to know Fermat's Theorem can make this calculation easier, but for our purposes, I am going to use regular fast modular exponentiation.

exp	1	2	4	8	16
12^{exp} %23	12	6	13	8	18

$12 \times 12 = 144 \equiv 6 \pmod{23}$

$6 \times 6 = 36 \equiv 13 \pmod{23}$

$13 \times 13 = 169 \equiv 8 \pmod{23}$

$8 \times 8 = 64 \equiv 18 \pmod{23}$

28 in binary is 11100.

$12^{28} = 12^{16}12^{8}12^{4} \equiv 18 \times 8 \times 13 \equiv (-5) \times 8 \times (-10) \equiv 50 \times 8 \equiv 4 \times 8 \equiv 32 \equiv 9$

Answer: 9

Cycle Method for Fast Modular Exponentiation

What's the remainder when 74^{2023} is divided by 11?

$$74^{2023} \equiv (-3)^{2023} \ (mod\ 11)$$

Exp	0	1	2	3	4	5	6	7	8	9	10
$(-3)^{exp}$	1	-3	9	6	4	-1	3	2	-6	7	1

$3^3 = 3^2 \times (-3) = 9 \times 3 \equiv -27 \equiv 6 \pmod{11}$

$3^4 = 3^3 \times 3 = 6 \times (-3) \equiv -18 \equiv 4 \pmod{11}$

$3^5 = 3^3 \times 3 = 4 \times (-3) \equiv -12 \equiv -1 \pmod{11}$

Since $3^{10} \equiv 1 \pmod{11}$, then $3^{10c} \equiv 1 \pmod{11}$, for any integer c.

$$74^{2023} \equiv (-3)^{2023} \equiv (-3)^{10\times 202}(-3)^3 \equiv 1^{202}(6) \equiv 6$$

Greatest Common Divisor(a, b) is the largest integer that divides evenly into both a and b:

gcd(20, 25) = 5

gcd(33, 18) = 3

gcd(35, 98) = 7

gcd(201, 202) = 1

Naïve Brute Force algorithm:

Answer = 1

For (int i=2; i<=min(a,b); i++)

    If (a%i == 0 && b%i == 0)

        Answer = i;

Try dividing each integer from 2 upto the minimum of a and b into both. If both are divisible, update our answer!

Yes, this works…Problem: It's really slow!

Better Algorithm (slightly) from grade school:

Write a factor tree for both numbers, and circle all the common prime factors:

gcd(30, 105)

30 → 2 x 3 x 5

105 → 7 x 3 x 5, gcd(30, 105) = 3 x 5 = 15

This works as fast as prime factorization…The standard way to find a prime factor of an integer is trial division upto the square root of the number! (Faster than the other method, but still kind of slow…)

Euclid, the father of geometry, came up with a way to find the GCD of two integers very efficiently. He just did repeated division!!!

$a = bq_0 + r_1$, For the next step, treat b as the dividend, and r_1 and divider $\gcd(a, b) = \gcd(b, r_1)$

$b = r_1q_1 + r_2$, repeat this idea over until $r_{k+1} = 0$

$r_1 = r_2q_2 + r_3$, …

$r_{k-2} = r_{k-1}q_{k-1} + r_k$, $r_k > 0$ and $r_k < r_{k-1}$

$r_{k-1} = r_kq_k + 0$ (no remainder)

$\gcd(a, b) = r_k$

1. Examples of the algorithm.

2. Prove that it works

Gcd(184, 74)

184 = 2 x 74 + 36

74 = 2 x 36 + 2

36 = 18 x 2

Gcd(317, 103)

317 = 3 x 103 + 8

103 = 12 x 8 + 7

8 = 1 x 7 + 1

7 = 7 x 1

gcd(208, 128)

208 = 1 x 128 + 80

128 = 1 x 80 + 48

80 = 1 x 48 + 32

48 = 1 x 32 + 16

32 = 2 x 16