Modular Exponentiation via the cycle method

If $a \equiv b \pmod{n}$, then $ca \equiv cb \pmod{n}$
If $a \equiv b \pmod{n}$, then $a + c \equiv b + c \pmod{n}$
If $a \equiv b \pmod{n}$, then $a^k \equiv b^k \pmod{n}$

Find the remainder when 2^{1003} is divided by 11

Pow	0	1	2	3	4	5	6	7	8	9	10
Val	1	2	4	8	5	10	9	7	3	6	1

When calculating 16 x 2 = x mod 11 the answer is the same as calculating 5 x 2 = x mod 11

This repeats every 10. For $2^{1003} \equiv 2^3 \equiv 8 \pmod{11}$

Since it repeats every 10, adding multiples of 10 to the exponent don't change the mod value.

Imagine the chart going to 1003…it just repeats, so 0 says 1, 10 says 1, 20 says 1…1000 will say 1, 1001 will say 2, 1002 will say 4 and 1003 will say 8

$2^{10} \equiv 1 \pmod{11}$

$2^{1003} \equiv 2^{1000}2^3 \equiv (2^{10})^{100}2^3 \equiv (1)^{100}2^3 \equiv 8 \pmod{11}$

This is good if the cycle is small (which will be the case if the mod value is small)

Fast Modular Exponentiation (bottom up iterative way)
3^{55} mod 19

$3^{55} = 3^{32}3^{16}3^{4}3^{2}3^{1} \equiv 4(-2)(5)(9)(3) \equiv (-8)(45)(3) \equiv (-24)7 \equiv (-5)7 \equiv -35 \equiv 3 \pmod{19}$

Pow	1	2	4	8	16	32
Value	3	9	81→5	25 → 6	36 → 17	4

$3^4 = (3^2)^2 = 9 \times 9 \equiv 5 \pmod{19}$

$3^8 = (3^4)^2 = 81^2 \equiv 5^2 \equiv 25 \equiv 6 \pmod{19}$
$3^{16} = (3^8)^2 \equiv 6^2 \equiv 26 \equiv 17 \pmod{19}$
$3^{32} = (3^{16})^2 \equiv 17^2 \equiv (-2)^2 \equiv 4 \pmod{19}$, because $17 \equiv -2 \pmod{19}$

Fast Modular Exponentiation (top down recursive way)
$2^{55} = (2^{27})^2 \times 2 = ((2^{13})^2 \times 2)^2 \times 2$

By Hand…
$2^{55} = (2^{10})^5 2^5$ → fundamentally, by hand, you can break this down any which way you want, such that the exponents add up to the proper value.

Division

Input two positive integers a and b.
Output two integers q and r such that
a = bq + r, and 0 <= r < b

There is precisely one ordered pair (q, r) which satisfies these requirements for any positive integer (a, b).
q is defined as the quotient.
r is defined as the remainder.
You probably learned this in 4th grade, but without the fancy names or the formal stuff above.

Our goal: to prove that there is exactly one answer to division!!!

81 divided by 11

81 = 7 x 11 + 4, q = 7, r = 4

81 = 6 x 11 + 15, problem→ my supposed remainder is greater than 10.

We can think about division like subtracting out 11s until we can't do it any more. And if we can't do it, what we're left with is less than 11.

Proof by contradiction to prove it!

Assume to the contrary, for some positive integers a and b, that there are two distinct ordered pairs (q, r) and (q', r') (either q != q' or r != r') such that

a = bq + r, 0 <= r < b AND
a = bq' + r', 0 <= r' < b

bq + r = bq' + r'
bq - bq' = r' - r
b(q - q') = r' - r

Two cases:

q-q' = 0	OR	q-q' != 0
r' - r = 0		| b(q - q') | >= b
Contradicts		But | r' - r| < b
(q,r) and (q',r') are Distinct.		This is impossible!

To prove there is at least one answer. We first get a base answer:

A = 0b + a, so set q = 0, r = a, this proves there is one solution without a restriction on r.

Now, assume to the contrary that there is no solution with 0 <= r < b. Then that means that the solution with the minimum positive value for r has a value for r >= b.

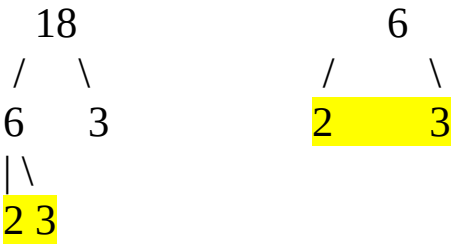
A = bq + r, where r >= b
A = (q+1)b + (r - b), so new solution is q' = q+1, r' = r-b >= 0, this contradicts the assumption that the minimal value of a positive remainder was r.

Euclid's Algorithm and the greatest common divisor

Greatest Common Divisor of two integers is the largest integer that divides evenly into both integers:

gcd(18, 6) = 6
gcd(35, 14) = 7
gcd(123, 77) = 1

In middle school, you learned to calculate these using a factor tree.



It's hard to make factor trees for very large integers…

It's easier to solve this gcd problem (faster computationally), if we utilize division.

Euclid's algorithm to determine gcd(123, 77):

123 = 1 x 77 + 46
77 = 1 x 46 + 31
46 = 1 x 31 + 15
31 = 2 x 15 + 1 , the last non-zero remainder is the gcd.
15 = 15 x 1 + 0

If gcd(a,b) = 1, we say that a and b are "relatively prime", or that a and b are "co-prime"

Gcd(198, 78) = 6

198 = 2 x 78 + 42, here we factor 6 out from 42, 78, so 6 | 198
78 = 1 x 42 + 36, here we can factor 6 out of 36 and 42, so 6 | 78.
42 = 1 x 36 + 6, we can factor 6 out of both terms so 6 | 42
36 = 6 x 6 + 0

Prove that the value Euclid's Algorithm produces is the GCD. We need to prove two things:

1) Euclid's generates a value d such that d | a and d | b. Thus, it generates a common divisor.

2) For all common divisors d', d' | d, where d is the value produced by Euclid's Algorithm.

$a = bq_1 + r_1$
$b = r_1q_2 + r_2$
$r_1 = r_2q_3 + r_3$
$r_2 = r_3q_4 + r_4$
…
$r_{k-2} = r_{k-1}q_k + r_k$
$r_{k-1} = r_kq_{k+1}$, and no remainder! Euclid's answer is r_k.

Goal for step 1, prove that $r_k \mid a$ and $r_k \mid b$.

$a = bq_1 + r_1$
$b = r_1q_2 + r_2$
$r_1 = r_2q_3 + r_3$
$r_2 = r_3q_4 + r_4$
…
$r_{k-2} = r_{k-1}q_k + r_k$
$r_{k-1} = r_kq_{k+1}$, thus $r_k \mid r_{k-1}$, so I can rewrite $r_{k-1} = r_kc$, for some int c

$r_{k-2} = r_{k-1}q_k + r_k$
$r_{k-2} = cr_kq_k + r_k$
$r_{k-2} = r_k(cq_k + 1)$, so this means that $r_k \mid r_{k-2}$

At each step of the proof, we show that $r_k \mid r_i$, for each value of i starting at k-1, and continuing to r_1, and finally ending with b and a.

Now, we want to prove that for any arbitrary common divisor d' of a and b, that d' | d.

$a = bq_1 + r_1$, so d' | a and d' | b, so $a - bq_1 = r_1$, so we can factor out d' from both a and bq_1, so this shows that d' | r_1

$b = r_1q_2 + r_2$, d' | b and d' | r_1, $b - r_1q_2 = r_2$, thus d' | r_2

and so on…

198 = 2 x 78 + 42, 3 divides into 198 and 3 divides 78, so 3 must Divide into 42, since 198 - 2 x 78 = 42 3(66 - 2x26) = 42, so 3 | 42.
78 = 1 x 42 + 36, 78 - 42 = 3(26 - 14) = 36, so 3 | 36
42 = 1 x 36 + 6, since 42 - 36 = 3(14 - 12) = 6, 3 | 6.

```
int gcd(int a, int b) {
     return b == 0 ? a : gcd(b, a%b);
}
```

Two more examples of the Euclidean algorithm.

Gcd(75, 49)

75 = 1 x 49 + 26
49 = 1 x 26 + 23
26 = 1 x 23 + 3
23 = 7 x 3 + 2
3 = 1 x 2 + **1**

Gcd(728, 206)

728 = 3 x 206 + 110
206 = 1 x 110 + 96
110 = 1 x 96 + 14
96 = 6 x 14 + 12
14 = 1 x 12 + **2**
12 = 6 x 2

What is the run-time of Euclid's algorithm?

How many steps in the worst case (how many divisions) will it take for the algorithm to terminate?

$a = bq_1 + r_1$
$b = r_1q_2 + r_2$
$r_1 = r_2q_3 + r_3$

We will prove that $2r_1 < a$. We assume $a > b$, since this is the natural way to start the algorithm.

Since $a > b$, $q_1 >= 1$

$a = bq_1 + r_1 >= b + r_1$
$> r_1 + r_1 = 2r_1$,because by def $b > r_1$.

$A > 2r_1$

In every two steps of Euclid's Algorithm, the number on the LHS at least gets divided by 2.

Thus, the max number of steps = 2k, where k is the number of times we have to divide a by 2, to get it down to 1.

$$\frac{a}{2^k} = 1$$

$$a = 2^k$$

$$k = log_2 a$$

Max # of steps is $2\log_2 a$. (Upper bound not actually perfectly achievable…)

Worst case for Euclid's is the Fibonacci numbers…

Gcd(144, 89)

144 = 1 x 89 + 55
89 = 1 x 55 + 34
55 = 1 x 34 + 21
34 = 1 x 21 + 13
21 = 1 x 13 + 8
13 = 1 x 8 + 5
8 = 1 x 5 + 3
5 = 1 x 3 + 2
3 = 1 x 2 + 1

# of steps for gcd(F_n, F_{n-1}) = n-3, depending not counting the 0 remainder step. (If you count that step it's n-2.)