MAS114: Lecture 16

James Cranch

http://cranch.staff.shef.ac.uk/mas114/

2021–2022

The Chinese Remainder Theorem

The Chinese Remainder Theorem

It would be good to know something that reassures us that there *will* be a solution in some family of cases, and here's a result, named after its discovery by the ancient Chinese:

The Chinese Remainder Theorem

It would be good to know something that reassures us that there *will* be a solution in some family of cases, and here's a result, named after its discovery by the ancient Chinese:

Theorem (Chinese Remainder Theorem)

Let m_1 and m_2 be coprime, and let a_1 and a_2 be any integers. The simultaneous congruences

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2} \end{aligned}$$

have a solution modulo $m_1 m_2$.

Proving the Chinese Remainder Theorem

Proving the Chinese Remainder Theorem

Proof.

Suppose given m_1 and m_2 coprime.

Proving the Chinese Remainder Theorem

Proof.

Suppose given m_1 and m_2 coprime.
We'll solve two of the easiest imaginable pairs of simultaneous congruences first, and then we'll observe that, in fact, that's enough work to do anything.

Proving the Chinese Remainder Theorem

Proof.

Suppose given m_1 and m_2 coprime.

We'll solve two of the easiest imaginable pairs of simultaneous congruences first, and then we'll observe that, in fact, that's enough work to do anything.

The first easy pair of simultaneous congruences is

$$\begin{aligned} y &\equiv 1 \pmod{m_1} \\ y &\equiv 0 \pmod{m_2}. \end{aligned}$$

Proving the Chinese Remainder Theorem

Proof.

Suppose given m_1 and m_2 coprime.
We'll solve two of the easiest imaginable pairs of simultaneous congruences first, and then we'll observe that, in fact, that's enough work to do anything.
The first easy pair of simultaneous congruences is

$$\begin{aligned} y &\equiv 1 \pmod{m_1} \\ y &\equiv 0 \pmod{m_2}. \end{aligned}$$

The first equation says that $y = 1 - km_1$ for some k, and the second says that y is a multiple of m_2. In other words, we have $m_2 \mid 1 - km_1$, so

$$km_1 \equiv 1 \pmod{m_2}.$$

Proving the Chinese Remainder Theorem

Proof.

Suppose given m_1 and m_2 coprime.
We'll solve two of the easiest imaginable pairs of simultaneous congruences first, and then we'll observe that, in fact, that's enough work to do anything.
The first easy pair of simultaneous congruences is

$$\begin{aligned} y &\equiv 1 \pmod{m_1} \\ y &\equiv 0 \pmod{m_2}. \end{aligned}$$

The first equation says that $y = 1 - km_1$ for some k, and the second says that y is a multiple of m_2. In other words, we have $m_2 \mid 1 - km_1$, so

$$km_1 \equiv 1 \pmod{m_2}.$$

But m_1 and m_2 are coprime, so we know we can solve this.

Proving the Chinese Remainder Theorem

Proof.

Proving the Chinese Remainder Theorem

Proof.

Another easy pair of simultaneous congruences are

$$
\begin{aligned}
z &\equiv 0 \pmod{m_1} \\
z &\equiv 1 \pmod{m_2}.
\end{aligned}
$$

Proving the Chinese Remainder Theorem

Proof.

Another easy pair of simultaneous congruences are

$$\begin{aligned} z &\equiv 0 \pmod{m_1} \\ z &\equiv 1 \pmod{m_2}. \end{aligned}$$

This looks exactly the same, but the other way around: the second says that z is of the form $z = 1 - lm_2$ for some l, and the first says that z is a multiple of m_1.

Proving the Chinese Remainder Theorem

Proof.

Another easy pair of simultaneous congruences are

$$\begin{aligned} z &\equiv 0 \pmod{m_1} \\ z &\equiv 1 \pmod{m_2}. \end{aligned}$$

This looks exactly the same, but the other way around: the second says that z is of the form $z = 1 - lm_2$ for some l, and the first says that z is a multiple of m_1. In other words, we need

$$lm_2 \equiv 1 \pmod{m_1}.$$

We know we can do this.

Proving the Chinese Remainder Theorem

Proof.

Proving the Chinese Remainder Theorem

Proof.

In fact, instead of going through the method twice, the same process does *both* these pairs of congruences: if we use Euclid's algorithm to give a solution to

$$rm_1 + sm_2 = 1,$$

in fact taking $z = rm_1$ and $y = sm_2$ gives us what we want:

Proving the Chinese Remainder Theorem

Proof.

In fact, instead of going through the method twice, the same process does *both* these pairs of congruences: if we use Euclid's algorithm to give a solution to

$$rm_1 + sm_2 = 1,$$

in fact taking $z = rm_1$ and $y = sm_2$ gives us what we want:

- $sm_2 \equiv 1 \pmod{m_1}$ and $sm_2 \equiv 0 \pmod{m_2}$, while

Proving the Chinese Remainder Theorem

Proof.

In fact, instead of going through the method twice, the same process does *both* these pairs of congruences: if we use Euclid's algorithm to give a solution to

$$rm_1 + sm_2 = 1,$$

in fact taking $z = rm_1$ and $y = sm_2$ gives us what we want:

- $sm_2 \equiv 1 \pmod{m_1}$ and $sm_2 \equiv 0 \pmod{m_2}$, while
- $rm_1 \equiv 0 \pmod{m_1}$ and $rm_1 \equiv 1 \pmod{m_2}$.

Proving the Chinese Remainder Theorem

Proof.

Proving the Chinese Remainder Theorem

Proof.

What then of our original equations

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2}? \end{aligned}$$

Proving the Chinese Remainder Theorem

Proof.

What then of our original equations

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2}? \end{aligned}$$

I claim that if we take $x = a_1 y + a_2 z$, we have what we need.

Proving the Chinese Remainder Theorem

Proof.

What then of our original equations

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2}? \end{aligned}$$

I claim that if we take $x = a_1y + a_2z$, we have what we need. Indeed, since $y \equiv 1 \pmod{m_1}$ and $z \equiv 0 \pmod{m_1}$, we have

$$x = a_1y + a_2z \equiv a_1 \pmod{m_1},$$

Proving the Chinese Remainder Theorem

Proof.

What then of our original equations

$$x \equiv a_1 \pmod{m_1}$$
$$x \equiv a_2 \pmod{m_2}?$$

I claim that if we take $x = a_1 y + a_2 z$, we have what we need. Indeed, since $y \equiv 1 \pmod{m_1}$ and $z \equiv 0 \pmod{m_1}$, we have

$$x = a_1 y + a_2 z \equiv a_1 \pmod{m_1},$$

while, since $y \equiv 0 \pmod{m_2}$ and $z \equiv 1 \pmod{m_2}$, we have

$$x = a_1 y + a_2 z \equiv a_2 \pmod{m_1}.$$

Both of those are exactly what we needed. □

A worked example

A worked example

This gives us a new way of finding solutions, which I’ll show off:

A worked example

This gives us a new way of finding solutions, which I'll show off:
What are the solutions to:

$$\begin{aligned} x &\equiv 11 \pmod{14} \\ x &\equiv 10 \pmod{17}? \end{aligned}$$

A worked example

This gives us a new way of finding solutions, which I'll show off: What are the solutions to:

$$\begin{aligned} x &\equiv 11 \pmod{14} \\ x &\equiv 10 \pmod{17}? \end{aligned}$$

We'll use our "building blocks" from the proof of the Chinese Remainder Theorem. In order to find this, we discovered we needed to invert 17 mod 14:

A worked example

This gives us a new way of finding solutions, which I'll show off: What are the solutions to:

$$\begin{aligned} x &\equiv 11 \pmod{14} \\ x &\equiv 10 \pmod{17}? \end{aligned}$$

We'll use our "building blocks" from the proof of the Chinese Remainder Theorem. In order to find this, we discovered we needed to invert 17 mod 14: we need to solve

$$14r + 17s = 1.$$

A worked example

This gives us a new way of finding solutions, which I'll show off: What are the solutions to:

$$\begin{aligned} x &\equiv 11 \pmod{14} \\ x &\equiv 10 \pmod{17}? \end{aligned}$$

We'll use our "building blocks" from the proof of the Chinese Remainder Theorem. In order to find this, we discovered we needed to invert 17 mod 14: we need to solve

$$14r + 17s = 1.$$

This has a solution $5 \times 17 - 6 \times 14 = 1$.

A worked example

This gives us a new way of finding solutions, which I'll show off: What are the solutions to:

$$\begin{aligned} x &\equiv 11 \pmod{14} \\ x &\equiv 10 \pmod{17}? \end{aligned}$$

We'll use our "building blocks" from the proof of the Chinese Remainder Theorem. In order to find this, we discovered we needed to invert 17 mod 14: we need to solve

$$14r + 17s = 1.$$

This has a solution $5 \times 17 - 6 \times 14 = 1$.
As a result $5 \times 17 = 85$ is congruent to 1 mod 14 and 0 modulo 17,

A worked example

This gives us a new way of finding solutions, which I'll show off: What are the solutions to:

$$\begin{aligned} x &\equiv 11 \pmod{14} \\ x &\equiv 10 \pmod{17}? \end{aligned}$$

We'll use our "building blocks" from the proof of the Chinese Remainder Theorem. In order to find this, we discovered we needed to invert 17 mod 14: we need to solve

$$14r + 17s = 1.$$

This has a solution $5 \times 17 - 6 \times 14 = 1$.
As a result $5 \times 17 = 85$ is congruent to 1 mod 14 and 0 modulo 17, and $-6 \times 14 = -84$ is congruent to 0 mod 14 and 1 modulo 17.

A worked example

This gives us a new way of finding solutions, which I'll show off: What are the solutions to:

$$\begin{aligned} x &\equiv 11 \pmod{14} \\ x &\equiv 10 \pmod{17}? \end{aligned}$$

We'll use our "building blocks" from the proof of the Chinese Remainder Theorem. In order to find this, we discovered we needed to invert 17 mod 14: we need to solve

$$14r + 17s = 1.$$

This has a solution $5 \times 17 - 6 \times 14 = 1$.
As a result $5 \times 17 = 85$ is congruent to 1 mod 14 and 0 modulo 17, and $-6 \times 14 = -84$ is congruent to 0 mod 14 and 1 modulo 17.
Hence our solution is

$$11 \times 85 + 10 \times (-84) \equiv 95 \pmod{238}.$$

Comments

Comments

The bit in the statement which says that the moduli have to be coprime is definitely important!

Comments

The bit in the statement which says that the moduli have to be coprime is definitely important!
Consider the following:

$$\begin{aligned} x &\equiv 2 \pmod{5} \\ x &\equiv 2 \pmod{5} \end{aligned}$$

These have a solution, but it's $x \equiv 2 \pmod{5}$, and not modulo 25.

Comments

The bit in the statement which says that the moduli have to be coprime is definitely important!
Consider the following:

$$x \equiv 2 \pmod{5}$$
$$x \equiv 2 \pmod{5}$$

These have a solution, but it's $x \equiv 2 \pmod{5}$, and not modulo 25.
On the other hand, these

$$x \equiv 2 \pmod{5}$$
$$x \equiv 3 \pmod{5}$$

don't have any solution, modulo anything.

More comments

More comments

Similarly, you can check that

$$\begin{aligned} x &\equiv 17 \pmod{30} \\ x &\equiv 7 \pmod{20} \end{aligned}$$

have a solution, which is $x \equiv 47 \pmod{60}$.
We won't prove it, but the rules are this:

More comments

Similarly, you can check that

$$\begin{aligned} x &\equiv 17 \pmod{30} \\ x &\equiv 7 \pmod{20} \end{aligned}$$

have a solution, which is $x \equiv 47 \pmod{60}$.
We won't prove it, but the rules are this:

- You can check whether two congruences with moduli m_1 and m_2 agree by looking what they say modulo $\gcd(m_1, m_2)$

More comments

Similarly, you can check that

$$
\begin{aligned}
x &\equiv 17 \pmod{30} \\
x &\equiv 7 \pmod{20}
\end{aligned}
$$

have a solution, which is $x \equiv 47 \pmod{60}$.
We won't prove it, but the rules are this:

- You can check whether two congruences with moduli m_1 and m_2 agree by looking what they say modulo $\gcd(m_1, m_2)$ (for example, the two above agree, because they both say $x \equiv 7 \pmod{10}$);

More comments

Similarly, you can check that

$$\begin{aligned} x &\equiv 17 \pmod{30} \\ x &\equiv 7 \pmod{20} \end{aligned}$$

have a solution, which is $x \equiv 47 \pmod{60}$.
We won't prove it, but the rules are this:

- You can check whether two congruences with moduli m_1 and m_2 agree by looking what they say modulo $\gcd(m_1, m_2)$ (for example, the two above agree, because they both say $x \equiv 7 \pmod{10}$);
- If two congruences agree, they have a common solution modulo $\mathrm{lcm}(m_1, m_2)$.

More arithmetic mod p

More arithmetic mod p

Earlier, we pointed out that modular arithmetic modulo primes is very well-behaved: every nonzero residue is invertible.

More arithmetic mod p

Earlier, we pointed out that modular arithmetic modulo primes is very well-behaved: every nonzero residue is invertible.
We'll going to go on and use that.

More arithmetic mod p

Earlier, we pointed out that modular arithmetic modulo primes is very well-behaved: every nonzero residue is invertible.

We'll going to go on and use that.

The first thing we'll talk about is *exponentiation* in modular arithmetic.

Exponentiation mod p

Exponentiation mod p

In integer arithmetic, it's usually stupid to try to calculate very large powers: for example, 3^{1234} has a huge number of digits (589 of them, to be precise).

Exponentiation mod p

In integer arithmetic, it's usually stupid to try to calculate very large powers: for example, 3^{1234} has a huge number of digits (589 of them, to be precise).

But, in modular arithmetic there are no large numbers. For example 3^{1234} will be congruent to something between 0 and 9 modulo 10, and it's reasonable to ask what.

Exponentiation mod p

In integer arithmetic, it's usually stupid to try to calculate very large powers: for example, 3^{1234} has a huge number of digits (589 of them, to be precise).

But, in modular arithmetic there are no large numbers. For example 3^{1234} will be congruent to something between 0 and 9 modulo 10, and it's reasonable to ask what.

One very stupid way of working it out would be to do the multiplication in the integers, then divide by 10 and find the remainder.

Exponentiation mod p

In integer arithmetic, it's usually stupid to try to calculate very large powers: for example, 3^{1234} has a huge number of digits (589 of them, to be precise).

But, in modular arithmetic there are no large numbers. For example 3^{1234} will be congruent to something between 0 and 9 modulo 10, and it's reasonable to ask what.

One very stupid way of working it out would be to do the multiplication in the integers, then divide by 10 and find the remainder.

We can do better, by doing our arithmetic all modulo 10 in the first place. So:

Exponentiation mod p

In integer arithmetic, it's usually stupid to try to calculate very large powers: for example, 3^{1234} has a huge number of digits (589 of them, to be precise).

But, in modular arithmetic there are no large numbers. For example 3^{1234} will be congruent to something between 0 and 9 modulo 10, and it's reasonable to ask what.

One very stupid way of working it out would be to do the multiplication in the integers, then divide by 10 and find the remainder.

We can do better, by doing our arithmetic all modulo 10 in the first place. So:

$$3^2 = 3 \times 3 \equiv 9 \pmod{10},$$

Exponentiation mod p

In integer arithmetic, it's usually stupid to try to calculate very large powers: for example, 3^{1234} has a huge number of digits (589 of them, to be precise).

But, in modular arithmetic there are no large numbers. For example 3^{1234} will be congruent to something between 0 and 9 modulo 10, and it's reasonable to ask what.

One very stupid way of working it out would be to do the multiplication in the integers, then divide by 10 and find the remainder.

We can do better, by doing our arithmetic all modulo 10 in the first place. So:

$$\begin{aligned} 3^2 &= 3 \times 3 \equiv 9 \pmod{10}, \\ 3^3 &= 3 \times 3^2 \equiv 3 \times 9 \equiv 7 \pmod{10}, \end{aligned}$$

Exponentiation mod p

In integer arithmetic, it's usually stupid to try to calculate very large powers: for example, 3^{1234} has a huge number of digits (589 of them, to be precise).

But, in modular arithmetic there are no large numbers. For example 3^{1234} will be congruent to something between 0 and 9 modulo 10, and it's reasonable to ask what.

One very stupid way of working it out would be to do the multiplication in the integers, then divide by 10 and find the remainder.

We can do better, by doing our arithmetic all modulo 10 in the first place. So:

$$\begin{aligned} 3^2 &= 3 \times 3 \equiv 9 \pmod{10}, \\ 3^3 &= 3 \times 3^2 \equiv 3 \times 9 \equiv 7 \pmod{10}, \\ 3^4 &= 3 \times 3^3 \equiv 3 \times 7 \equiv 1 \pmod{10} \ldots \end{aligned}$$

Exponentiation mod p

In integer arithmetic, it's usually stupid to try to calculate very large powers: for example, 3^{1234} has a huge number of digits (589 of them, to be precise).

But, in modular arithmetic there are no large numbers. For example 3^{1234} will be congruent to something between 0 and 9 modulo 10, and it's reasonable to ask what.

One very stupid way of working it out would be to do the multiplication in the integers, then divide by 10 and find the remainder.

We can do better, by doing our arithmetic all modulo 10 in the first place. So:

$$\begin{aligned} 3^2 &= 3 \times 3 \equiv 9 \pmod{10}, \\ 3^3 &= 3 \times 3^2 \equiv 3 \times 9 \equiv 7 \pmod{10}, \\ 3^4 &= 3 \times 3^3 \equiv 3 \times 7 \equiv 1 \pmod{10} \ldots \end{aligned}$$

That's still going to be a lot of multiplication, if we keep multiplying by 3 (modulo 10) more than a thousand times!

Better yet?

Better yet?

There are considerably more intelligent ways.

Better yet?

There are considerably more intelligent ways. For example, we can square modulo 10 quite quickly.

Better yet?

There are considerably more intelligent ways. For example, we can square modulo 10 quite quickly.

That lets us do some exponents by repeated squaring. For example,

$$3^8 = 3^{2\times 4} = (3^2)^4 = (3^2)^{2\times 2} = \left((3^2)^2\right)^2.$$

Better yet?

There are considerably more intelligent ways. For example, we can square modulo 10 quite quickly.
That lets us do some exponents by repeated squaring. For example,

$$3^8 = 3^{2\times4} = (3^2)^4 = (3^2)^{2\times2} = \left((3^2)^2\right)^2.$$

1234 isn't quite as nice, but we can do

3^{1234}

Better yet?

There are considerably more intelligent ways. For example, we can square modulo 10 quite quickly.
That lets us do some exponents by repeated squaring. For example,

$$3^8 = 3^{2\times 4} = (3^2)^4 = (3^2)^{2\times 2} = \left((3^2)^2\right)^2.$$

1234 isn't quite as nice, but we can do

$$3^{1234} \equiv 3^{2\times 617}$$

Better yet?

There are considerably more intelligent ways. For example, we can square modulo 10 quite quickly.

That lets us do some exponents by repeated squaring. For example,

$$3^8 = 3^{2\times 4} = (3^2)^4 = (3^2)^{2\times 2} = \left((3^2)^2\right)^2.$$

1234 isn't quite as nice, but we can do

$$3^{1234} \equiv 3^{2\times 617} \equiv (3^2)^{617}$$

Better yet?

There are considerably more intelligent ways. For example, we can square modulo 10 quite quickly.

That lets us do some exponents by repeated squaring. For example,

$$3^8 = 3^{2\times 4} = (3^2)^4 = (3^2)^{2\times 2} = \left((3^2)^2\right)^2.$$

1234 isn't quite as nice, but we can do

$$3^{1234} \equiv 3^{2\times 617} \equiv (3^2)^{617} \equiv 9^{617}$$

Better yet?

There are considerably more intelligent ways. For example, we can square modulo 10 quite quickly.

That lets us do some exponents by repeated squaring. For example,

$$3^8 = 3^{2\times 4} = \left(3^2\right)^4 = \left(3^2\right)^{2\times 2} = \left(\left(3^2\right)^2\right)^2.$$

1234 isn't quite as nice, but we can do

$$3^{1234} \equiv 3^{2\times 617} \equiv (3^2)^{617} \equiv 9^{617} \equiv 9^{2\times 308+1}$$

Better yet?

There are considerably more intelligent ways. For example, we can square modulo 10 quite quickly.
That lets us do some exponents by repeated squaring. For example,

$$3^8 = 3^{2\times 4} = (3^2)^4 = (3^2)^{2\times 2} = \left((3^2)^2\right)^2.$$

1234 isn't quite as nice, but we can do

$$3^{1234} \equiv 3^{2\times 617} \equiv (3^2)^{617} \equiv 9^{617} \equiv 9^{2\times 308+1} \equiv (9^2)^{308} 9$$

Better yet?

There are considerably more intelligent ways. For example, we can square modulo 10 quite quickly.
That lets us do some exponents by repeated squaring. For example,

$$3^8 = 3^{2\times 4} = (3^2)^4 = (3^2)^{2\times 2} = \left((3^2)^2\right)^2.$$

1234 isn't quite as nice, but we can do

$$3^{1234} \equiv 3^{2\times 617} \equiv (3^2)^{617} \equiv 9^{617} \equiv 9^{2\times 308+1} \equiv (9^2)^{308} 9$$

and end up getting the answer.

Better yet?

There are considerably more intelligent ways. For example, we can square modulo 10 quite quickly.

That lets us do some exponents by repeated squaring. For example,

$$3^8 = 3^{2\times 4} = (3^2)^4 = (3^2)^{2\times 2} = \left((3^2)^2\right)^2.$$

1234 isn't quite as nice, but we can do

$$3^{1234} \equiv 3^{2\times 617} \equiv (3^2)^{617} \equiv 9^{617} \equiv 9^{2\times 308+1} \equiv (9^2)^{308} 9$$

and end up getting the answer.

We'll end up only multiplying about twenty times if we do it this way: that's much less!

Even better yet

Even better yet

But, in fact, there's a method that's even faster still for this situation.

Even better yet

But, in fact, there's a method that's even faster still for this situation. We've just computed that

$$3^4 \equiv 1 \pmod{10}.$$

Even better yet

But, in fact, there's a method that's even faster still for this situation. We've just computed that

$$3^4 \equiv 1 \pmod{10}.$$

That does almost all the work for us, as

$$3^{4k}$$

Even better yet

But, in fact, there's a method that's even faster still for this situation. We've just computed that

$$3^4 \equiv 1 \pmod{10}.$$

That does almost all the work for us, as

$$3^{4k} = (3^4)^k$$

Even better yet

But, in fact, there's a method that's even faster still for this situation. We've just computed that

$$3^4 \equiv 1 \pmod{10}.$$

That does almost all the work for us, as

$$3^{4k} = (3^4)^k \equiv 1^k$$

Even better yet

But, in fact, there's a method that's even faster still for this situation. We've just computed that

$$3^4 \equiv 1 \pmod{10}.$$

That does almost all the work for us, as

$$3^{4k} = (3^4)^k \equiv 1^k = 1 \pmod{10}.$$

Even better yet

But, in fact, there's a method that's even faster still for this situation. We've just computed that

$$3^4 \equiv 1 \pmod{10}.$$

That does almost all the work for us, as

$$3^{4k} = (3^4)^k \equiv 1^k = 1 \pmod{10}.$$

Hence

3^{1234}

Even better yet

But, in fact, there's a method that's even faster still for this situation. We've just computed that

$$3^4 \equiv 1 \pmod{10}.$$

That does almost all the work for us, as

$$3^{4k} = (3^4)^k \equiv 1^k = 1 \pmod{10}.$$

Hence

$3^{1234} = 3^{4\times 308+2}$

Even better yet

But, in fact, there's a method that's even faster still for this situation. We've just computed that

$$3^4 \equiv 1 \pmod{10}.$$

That does almost all the work for us, as

$$3^{4k} = (3^4)^k \equiv 1^k = 1 \pmod{10}.$$

Hence

$3^{1234} = 3^{4\times 308+2} = 3^{4\times 308}3^2$

Even better yet

But, in fact, there's a method that's even faster still for this situation. We've just computed that

$$3^4 \equiv 1 \pmod{10}.$$

That does almost all the work for us, as

$$3^{4k} = (3^4)^k \equiv 1^k = 1 \pmod{10}.$$

Hence

$3^{1234} = 3^{4 \times 308 + 2} = 3^{4 \times 308} 3^2 = (3^4)^{308} 3^2$

Even better yet

But, in fact, there's a method that's even faster still for this situation. We've just computed that

$$3^4 \equiv 1 \pmod{10}.$$

That does almost all the work for us, as

$$3^{4k} = (3^4)^k \equiv 1^k = 1 \pmod{10}.$$

Hence

$$3^{1234} = 3^{4\times 308+2} = 3^{4\times 308}3^2 = (3^4)^{308}3^2 \equiv 1^{308}3^2$$

Even better yet

But, in fact, there's a method that's even faster still for this situation. We've just computed that

$$3^4 \equiv 1 \pmod{10}.$$

That does almost all the work for us, as

$$3^{4k} = (3^4)^k \equiv 1^k = 1 \pmod{10}.$$

Hence

$$3^{1234} = 3^{4\times308+2} = 3^{4\times308}3^2 = (3^4)^{308}3^2 \equiv 1^{308}3^2 \equiv 3^2$$

Even better yet

But, in fact, there's a method that's even faster still for this situation. We've just computed that

$$3^4 \equiv 1 \pmod{10}.$$

That does almost all the work for us, as

$$3^{4k} = (3^4)^k \equiv 1^k = 1 \pmod{10}.$$

Hence

$$3^{1234} = 3^{4\times 308+2} = 3^{4\times 308}3^2 = (3^4)^{308}3^2 \equiv 1^{308}3^2 \equiv 3^2 \equiv 9 \pmod{10}$$

Even better yet

But, in fact, there's a method that's even faster still for this situation. We've just computed that

$$3^4 \equiv 1 \pmod{10}.$$

That does almost all the work for us, as

$$3^{4k} = (3^4)^k \equiv 1^k = 1 \pmod{10}.$$

Hence

$$3^{1234} = 3^{4\times 308+2} = 3^{4\times 308}3^2 = (3^4)^{308}3^2 \equiv 1^{308}3^2 \equiv 3^2 \equiv 9 \pmod{10}$$

That makes the whole thing easy.

Making 1 as a power

Making 1 as a power

The relevant observation here was really that there was some integer n such that $3^n \equiv 1 \pmod{10}$.

Making 1 as a power

The relevant observation here was really that there was some integer n such that $3^n \equiv 1 \pmod{10}$. So two obvious questions are:

1. When does there exist such an n?

Making 1 as a power

The relevant observation here was really that there was some integer n such that $3^n \equiv 1 \pmod{10}$. So two obvious questions are:

1. When does there exist such an n?
2. When it does exist, can we compute it?

Making 1 as a power

The relevant observation here was really that there was some integer n such that $3^n \equiv 1 \pmod{10}$. So two obvious questions are:

1. When does there exist such an n?
2. When it does exist, can we compute it?

Our answer to the first is not too difficult: