Finding factors in Berlekamp's Algebra

UQ Mathematics Student Society

Joel Richardson

The University of Queensland

April 2024

The Problem

Suppose p is prime. Let $\mathbb{Z}_p$ be the field of integers mod p.

Take a polynomial $f \in \mathbb{Z}_p[x]$.

Suppose, for simplicity, that f is squarefree[1].

How do we find the factors of f?

[1] has no repeated factors. The map $f \mapsto f/\gcd(f, f')$ deletes repeated factors.

The Problem

Suppose p is prime. Let $\mathbb{Z}_p$ be the field of integers mod p.

Take a polynomial $f \in \mathbb{Z}_p[x]$.

Suppose, for simplicity, that f is squarefree[1].

How do we find the factors of f?

[1] has no repeated factors. The map $f \mapsto f/\gcd(f, f')$ deletes repeated factors.

The Problem

Suppose p is prime. Let $\mathbb{Z}_p$ be the field of integers mod p.

Take a polynomial $f \in \mathbb{Z}_p[x]$.

Suppose, for simplicity, that f is squarefree[1].

How do we find the factors of f?

[1] has no repeated factors. The map $f \mapsto f/\gcd(f, f')$ deletes repeated factors.

The Problem

Suppose p is prime. Let $\mathbb{Z}_p$ be the field of integers mod p.

Take a polynomial $f \in \mathbb{Z}_p[x]$.

Suppose, for simplicity, that f is squarefree[1].

How do we find the factors of f?

[1] has no repeated factors. The map $f \mapsto f/\gcd(f, f')$ deletes repeated factors.

Problem Motivation

Ideas?

Let's get the obvious out of the way.

We can theoretically find the factors of f with a brute force search.

Ideas?

Let's get the obvious out of the way.

We can theoretically find the factors of f with a brute force search.

- this is boring

Ideas?

Let's get the obvious out of the way.

We can theoretically find the factors of f with a brute force search.

- this is boring (and slow)

Ideas?

Observation 1 We don't have to find every factor in one go.

Ideas?

Observation 1 We don't have to find every factor in one go.

If we can reliably produce even **just one** non-trivial divisor of f, then repeated application of our procedure will suffice to find every factor.

Non-Trivial Divisors

Suppose f has factors f_1, f_2, …, f_n

A non-trivial divisor g of f must contain at least one factor f_i of f

We might say that g splits f in two.

$$f = (f_1 f_2 f_3)(f_4 \dots f_n)$$

$$g = (f_1 f_2 f_3) \qquad f/g = (f_4 \dots f_n)$$

Non-Trivial Divisors

Suppose f has factors f_1, f_2, $\ldots$, f_n

A non-trivial divisor g of f must contain at least one factor f_i of f

We might say that g splits f in two.

$$f = (f_1 f_2 f_3)(f_4 \ldots f_n)$$

$$g = (f_1 f_2 f_3) \qquad f/g = (f_4 \ldots f_n)$$

Non-Trivial Divisors

Suppose f has factors f_1, f_2, $\ldots$, f_n

A non-trivial divisor g of f must contain at least one factor f_i of f

We might say that g splits f in two.

$$f = (f_1 f_2 f_3)(f_4 \ldots f_n)$$

$$g = (f_1 f_2 f_3) \qquad f/g = (f_4 \ldots f_n)$$

Non-Trivial Divisors

If we can come up with a `findNonTrivialDivisor` function, our factoring algorithm might look something like this:

```haskell
factor :: Polynomial -> Set Polynomial
factor f = case findNonTrivialDivisor f of
    Just g ->
        let h = f / g in
        Set.union ( factor g ) ( factor h )
    Nothing ->
        -- f is irreducible
        Set.singleton f
```

Non-Trivial Divisors

We want to find $g \in \mathbb{Z}_p[x]$ such that:

1. At least one factor of f divides g
2. At least one factor of f doesn't divide g
3. g divides f

Non-Trivial Divisors

We want to find $g \in \mathbb{Z}_p[x]$ such that:

1. At least one factor of f divides g
2. At least one factor of f doesn't divide g
3. g divides f

Non-Trivial Divisors

We want to find $g \in \mathbb{Z}_p[x]$ such that:

1. At least one factor of f divides g
2. At least one factor of f doesn't divide g
3. g divides f

Non-Trivial Divisors

We want to find $g \in \mathbb{Z}_p[x]$ such that:

1. At least one factor of f divides g
2. At least one factor of f doesn't divide g
3. g divides f

Non-Trivial Divisors

We want to find $g \in \mathbb{Z}_p[x]$ such that:

1. At least one factor of f divides g
2. At least one factor of f doesn't divide g
3. g divides f

Non-Trivial Divisors

We want to find $g \in \mathbb{Z}_p[x]$ such that:

1. At least one factor of f divides g
2. At least one factor of f doesn't divide g
3. All the factors of g divide f

Non-Trivial Divisors

We want to find $g \in \mathbb{Z}_p[x]$ such that:

1. At least one factor of f divides g
2. At least one factor of f doesn't divide g
3. All the factors of g divide f

the map $g \mapsto \gcd(f, g)$ deletes the factors of g that don't divide f!

Non-Trivial Divisors

We want to find $g \in \mathbb{Z}_p[x]$ such that:

1. At least one factor of f divides g
2. At least one factor of f doesn't divide g
3. ~~All the factors of g divide f~~ Apply $g \mapsto \gcd(f, g)$!

the map $g \mapsto \gcd(f, g)$ deletes the factors of g that don't divide f!

Non-Trivial Divisors

Where might we find such a polynomial g?

It's kinda stupid to look for g in all of $\mathbb{Z}_p[x]$

The majority of these polynomials have degree greater than that of f, and so do not divide it.

Instead, we should search only the set of polynomials whose degree is less than that of f

$$D_f \ = \ \{ \, h \in \mathbb{Z}_p[x] \ : \ \deg(h) < \deg(f) \, \}$$

Non-Trivial Divisors

Where might we find such a polynomial g?

It's kinda stupid to look for g in all of $\mathbb{Z}_p[x]$

The majority of these polynomials have degree greater than that of f, and so do not divide it.

Instead, we should search only the set of polynomials whose degree is less than that of f

$$D_f \;=\; \{\, h \in \mathbb{Z}_p[x] \;:\; \deg(h) < \deg(f) \,\}$$

Non-Trivial Divisors

Where might we find such a polynomial g?

It's kinda stupid to look for g in all of $\mathbb{Z}_p[x]$

The majority of these polynomials have degree greater than that of f, and so do not divide it.

Instead, we should search only the set of polynomials whose degree is less than that of f

$$D_f \ = \ \{ \, h \in \mathbb{Z}_p[x] \ : \ \deg(h) < \deg(f) \, \}$$

Non-Trivial Divisors

Where might we find such a polynomial g?

It's kinda stupid to look for g in all of $\mathbb{Z}_p[x]$

The majority of these polynomials have degree greater than that of f, and so do not divide it.

Instead, we should search only the set of polynomials whose degree is less than that of f

$$D_f = \{ h \in \mathbb{Z}_p[x] : \deg(h) < \deg(f) \}$$

Non-Trivial Divisors

Where might we find such a polynomial g?

It's kinda stupid to look for g in all of $\mathbb{Z}_p[x]$

The majority of these polynomials have degree greater than that of f, and so do not divide it.

Instead, we should search only the set of polynomials whose degree is less than that of f

$$D_f \;=\; \{\, h \in \mathbb{Z}_p[x] \;:\; \deg(h) < \deg(f) \,\}$$

Non-Trivial Divisors (Recap)

We've reduced our problem, finding all the factors of f, to the following problem:

Find any $g \in D_f$ such that:

1. At least one factor of f divides g
2. At least one factor of f doesn't divide g

We will call such polynomials **based**

Non-Trivial Divisors (Recap)

We've reduced our problem, finding all the factors of f, to the following problem:

Find any $g \in D_f$ such that:

1. At least one factor of f divides g
2. At least one factor of f doesn't divide g

We will call such polynomials **based**

- what's next?

What's next?

Observation 2 This feels like a quotient ring situation.

What's next?

Observation 2 This feels like a quotient ring situation.

Consider the ring of polynomials $\bmod f$

$$A_f \stackrel{\mathsf{def}}{=} \mathbb{Z}_p[x] \,/\, \langle\, f \,\rangle$$

What's next?

Observation 2 This feels like a quotient ring situation.

Consider the ring of polynomials $\bmod f$

$$A_f \stackrel{\text{def}}{=} \mathbb{Z}_p[x] \,/\, \langle\, f \,\rangle$$

Note. this is basically D_f but closed under multiplication.

The Quotient Ring

Does our new multiplication $(\cdot)$ on A_f preserve the properties we care about?

If g and h are based, is $g \cdot h$ based?

The Quotient Ring

Does our new multiplication $(\,\cdot\,)$ on A_f preserve the properties we care about?

If g and h are based, is $g \cdot h$ based?

The Quotient Ring

Suppose g is based

wlog write $\gcd(f, g) = f_1 \cdots f_k$

Now $h = f / \gcd(f, g) = f_{k+1} \cdots f_n$ is also based

Notice that every factor of f divides $g \cdot h$

So $g \cdot h$ isn't based

The Quotient Ring

Suppose g is based

wlog write $\gcd(f, g) = f_1 \cdots f_k$

Now $h = f / \gcd(f, g) = f_{k+1} \cdots f_n$ is also based

Notice that every factor of f divides $g \cdot h$

So $g \cdot h$ isn't based

The Quotient Ring

Suppose g is based

wlog write $\gcd(f, g) = f_1 \cdots f_k$

Now $h = f / \gcd(f, g) = f_{k+1} \cdots f_n$ is also based

Notice that every factor of f divides $g \cdot h$

So $g \cdot h$ isn't based

The Quotient Ring

Suppose g is based

wlog write $\gcd(f, g) = f_1 \cdots f_k$

Now $h = f / \gcd(f, g) = f_{k+1} \cdots f_n$ is also based

Notice that every factor of f divides $g \cdot h$

So $g \cdot h$ isn't based

The Quotient Ring

Suppose g is based

wlog write $\gcd(f, g) = f_1 \cdots f_k$

Now $h = f / \gcd(\, f, \, g\,) = f_{k+1} \cdots f_n$ is also based

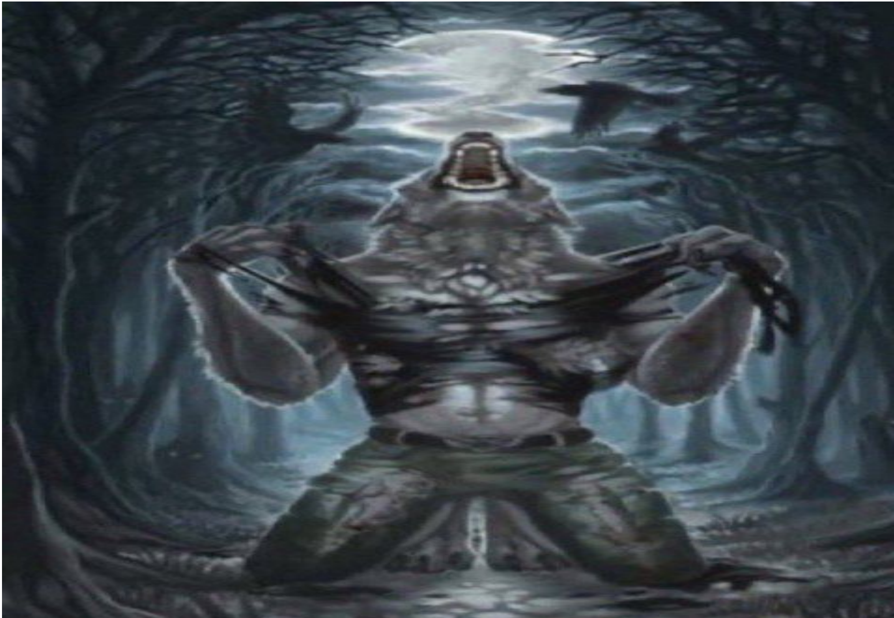
Notice that every factor of f divides $g \cdot h$

So $g \cdot h$ isn't based

The Quotient Ring

$g \cdot h$ isn't based.

frick



The Quotient Ring

Suppose g is based

wlog write $\gcd(f, g) = f_1 \cdots f_k$

Now $h = f / \gcd(f, g) = f_{k+1} \cdots f_n$ is also based

Notice that every factor of f divides $g \cdot h$

So $g \cdot h$ isn't based

The Quotient Ring

Theorem (Chinese remainder)

Suppose R is a commutative ring, and $I_1, I_2, \ldots, I_n$ are ideals of R such that for every pair i, j of non-equal indices

$$R = \{ ax + by : a \in I_i, b \in I_j, x, y \in R \}$$

Then, the function

$$\sigma : R/\bigcap_{i=1}^{n} I_i \to R/I_1 \times R/I_2 \times \cdots \times R/I_n$$
$$[r] \mapsto ([r], [r], \ldots, [r])$$

is a ring isomorphism.

The Quotient Ring

Theorem (Chinese remainder)

Suppose R is a commutative ring, and $I_1, I_2, \ldots, I_n$ are ideals of R such that for every pair i, j of non-equal indices

$$R = \{ ax + by : a \in I_i, b \in I_j, x, y \in R \}$$

Then, the function

$$: R / \bigcap_{i=1}^{n} I_i \to R/I_1 \times R/I_2 \times \cdots \times R/I_n$$

$$[r] \mapsto ([r], [r], \ldots, [r])$$

is a ring isomorphism.

The Quotient Ring

Notice that

$$\langle f \rangle = \langle f_1 \rangle \cap \langle f_2 \rangle \cap \cdots \cap \langle f_n \rangle$$

Moreover, it follows from the Chinese remainder theorem that there is an isomorphism

$$\sigma : A_f \to \mathbb{Z}_p[x]/\langle f_1 \rangle \times \mathbb{Z}_p[x]/\langle f_2 \rangle \times \cdots \times \mathbb{Z}_p[x]/\langle f_n \rangle$$

The Quotient Ring

The map

$$\sigma \ : \ g \ \mapsto \ (\ g \bmod f_1, \ g \bmod f_2, \ \ldots, \ g \bmod f_n \ )$$

is an isomorphism

The Quotient Ring

Now it's obvious that if g and h are based, then $g \cdot h$ is either based or zero.

$$\begin{aligned} \sigma(g \cdot h) &= \sigma(g) \cdot \sigma(h) \\ &= (\, 0, *, \ldots, * \,) \cdot (\, *, 0, \ldots, 0 \,) \\ &= (\, 0 \cdot *, * \cdot 0, \ldots, * \cdot 0 \,) \\ &= (\, 0, 0, \ldots, 0 \,) \end{aligned}$$

Moreover, if $g \cdot h$ is based or zero then at least one of g or h is based or zero.

- if $\sigma(g)$ is zero in the i^{th} component, then so is $\sigma(g \cdot h)$

The Quotient Ring

Notice that if g is based, then g^k is based for every positive k.

$$\sigma(g^k) = \sigma(g)^k = (\,*, \cdots, 0, \cdots, *\,)^k = (\,*, \cdots, 0, \cdots, *\,)$$

The set of based polynomials is closed under exponentiation.

The Quotient Ring

Give a name to the exponentiation map:

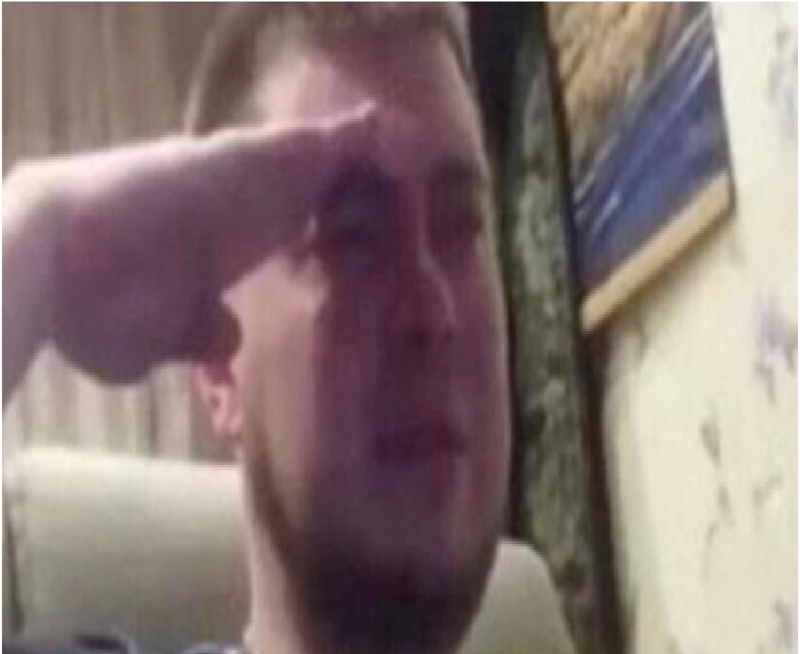
$$\begin{aligned} Q_k : A_f &\to A_f \\ g &\mapsto g^k \end{aligned}$$

The Quotient Ring

At this point we must make a sacrifice.

We must abandon some based polynomials.

frick



The Quotient Ring

We're going to pick some k and look at the subset in A_f that is fixed by Q_k

The Quotient Ring

We're going to pick some k and look at the subset in A_f that is fixed by Q_k

- which k should we pick?

The Quotient Ring

We're going to pick some k and look at the subset in A_f that is fixed by Q_k

- which k should we pick?
- what properties do we want?

The Quotient Ring

Suppose g and h are in fix(Q_k)

$$Q_k(g \cdot h) = (g \cdot h)^k = g^k \cdot h^k = g \cdot h$$

The Quotient Ring

Suppose g and h are in fix(Q_k)

$$Q_k(g \cdot h) = (g \cdot h)^k = g^k \cdot h^k = g \cdot h$$

- fix(Q_k) is closed under multiplication

The Quotient Ring

Suppose g and h are in fix(Q_k)

$$Q_k(g \cdot h) = (g \cdot h)^k = g^k \cdot h^k = g \cdot h$$

- fix(Q_k) is closed under multiplication
- is it closed under addition?

The Quotient Ring

Suppose g and h are in fix(Q_k)

$$Q_k(g \cdot h) = (g \cdot h)^k = g^k \cdot h^k = g \cdot h$$

- fix(Q_k) is closed under multiplication
- is it closed under addition?

$$Q_k(g + h) = (g + h)^k = g^k + h^k = g + h$$

The Quotient Ring

Suppose g and h are in fix(Q_k)

$$Q_k(g \cdot h) = (g \cdot h)^k = g^k \cdot h^k = g \cdot h$$

- fix(Q_k) is closed under multiplication
- is it closed under addition?

$$Q_k(g + h) = (g + h)^k = g^k + h^k = g + h$$

The Quotient Ring

Suppose g and h are in fix(Q_k)

$$Q_k(g \cdot h) = (g \cdot h)^k = g^k \cdot h^k = g \cdot h$$

- fix(Q_k) is closed under multiplication
- is it closed under addition?

$$Q_k(g + h) = (g + h)^k = g^k + h^k = g + h$$

- set $k = p$ (the same p as in $\mathbb{Z}_p[x]$)

The Quotient Ring

$$B_f \stackrel{\mathsf{def}}{=} \mathsf{fix}(Q_p)$$

The Quotient Ring

$$B_f \stackrel{\mathsf{def}}{=} \mathsf{fix}(Q_p)$$

Turns out B_f is kinda epic.

The Quotient Ring

$$B_f \overset{\text{def}}{=} \mathsf{fix}(Q_p)$$

Turns out B_f is kinda epic.

So epic, infact, that it has a name.

The Quotient Ring

$$B_f \stackrel{\text{def}}{=} \text{fix}(Q_p)$$

Turns out B_f is kinda epic.

So epic, infact, that it has a name.

B_f is called the Berlekamp subalgebra.

Berlekamp's Algebra

Some facts:

Berlekamp's Algebra

Some facts:

1. $\mathbb{Z}_p \subset B_f \subset A_f \subset \mathbb{Z}_p[x]$

Berlekamp's Algebra

Some facts:

1. $\mathbb{Z}_p \subset B_f \subset A_f \subset \mathbb{Z}_p[x]$
2. Q_p is a linear map on A_f

Berlekamp's Algebra

Some facts:

1. $\mathbb{Z}_p \subset B_f \subset A_f \subset \mathbb{Z}_p[x]$
2. Q_p is a linear map on A_f
3. $B_f \ = \ \mathsf{fix}(Q_p)$

Berlekamp's Algebra

Some facts:

1. $\mathbb{Z}_p \subset B_f \subset A_f \subset \mathbb{Z}_p[x]$
2. Q_p is a linear map on A_f
3. $B_f \;=\; \mathsf{fix}(Q_p) = \{\, x \in A_f \;:\; x^p = x \,\}$

Berlekamp's Algebra

Some facts:

1. $\mathbb{Z}_p \subset B_f \subset A_f \subset \mathbb{Z}_p[x]$
2. Q_p is a linear map on A_f
3. $B_f \ = \ \mathsf{fix}(Q_p) \ = \{\, x \in A_f \ : \ x^p - x = 0 \,\}$

Berlekamp's Algebra

Some facts:

1. $\mathbb{Z}_p \subset B_f \subset A_f \subset \mathbb{Z}_p[x]$
2. Q_p is a linear map on A_f
3. $B_f \; = \; \mathsf{fix}(Q_p) \; = \{ \, x \in A_f \; : \; x^p - x = 0 \, \} = \ker(Q_p - \mathsf{id})$

Berlekamp's Algebra

We can encode $Q_f - \mathrm{id}$ as an A_f valued matrix.

[2] other methods are available

Berlekamp's Algebra

We can encode $Q_f - \mathrm{id}$ as an A_f valued matrix.

Then we can use Gaussian elimination[2] to produce a basis for its nullspace.

[2] other methods are available

Berlekamp's Algebra

We can encode $Q_f - \text{id}$ as an A_f valued matrix.

Then we can use Gaussian elimination[2] to produce a basis for its nullspace.

The members of B_f are precisely the linear combinations of the elements of this basis!

[2] other methods are available

Berlekamp's Algebra

Lock in guys.

You ain't ready for this.

Berlekamp's Algebra

Have you seen this equality? For every $h \in \mathbb{Z}_p[x]$

$$\prod_{c \in \mathbb{Z}_p} (h + c) = h^p - h$$

proof on joelrichardson.au if u want

Berlekamp's Algebra

For every $h \in B_f$

$$\prod_{c \in \mathbb{Z}_p} (h + c) = 0 \mod f$$

Berlekamp's Algebra

For every $h \in B_f$

$$\prod_{c \in \mathbb{Z}_p} (h + c) = 0 \mod f$$

- the product of the $(h + c)$s is based or zero

Berlekamp's Algebra

For every $h \in B_f$

$$\prod_{c \in \mathbb{Z}_p} (h + c) = 0 \mod f$$

- the product of the $(h + c)$s is based or zero
- at least one $h + c$ is based or zero!

Berlekamp's Algebra

For every $h \in B_f$

$$\prod_{c \in \mathbb{Z}_p} (h + c) = 0 \mod f$$

- the product of the $(h + c)$s is based or zero
- at least one $h + c$ is based ~~or zero~~!

Berlekamp's Algebra

For every $h \in B_f$

$$\prod_{c \in \mathbb{Z}_p} (h + c) = 0 \mod f$$

- the product of the $(h + c)$s is based or zero
- at least one $h + c$ is based!

Berlekamp's Algebra

For every $h \in B_f$

$$\prod_{c \in \mathbb{Z}_p} (h + c) = 0 \mod f$$

- the product of the $(h + c)$s is based or zero
- at least one $h + c$ is based!
- at least one $\gcd(f, h + c)$ is a non-trivial divisor of f!

Berlekamp's Algebra

Recall our earlier code block:

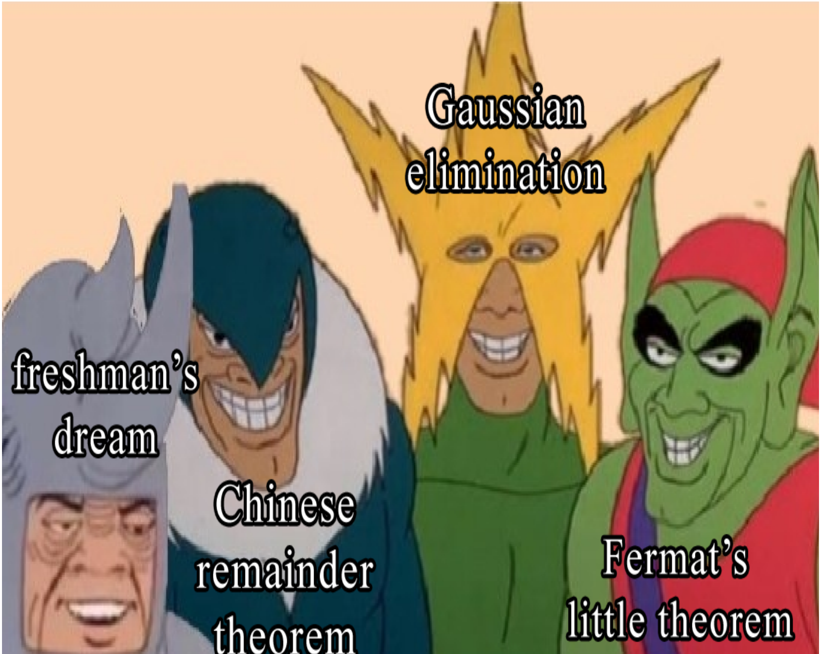
```
factor :: Polynomial -> Set Polynomial
factor f = case findNonTrivialDivisor f of
    Just g ->
        let h = f / g in
        Set.union ( factor g ) ( factor h )
    Nothing ->
        -- f is irreducible
        Set.singleton f
```

Berlekamp's Algebra

We can now implement `findNonTrivialDivisor`

```
findNonTrivialDivisor :: Polynomial -> Maybe Polynomial
findNonTrivialDivisor f = case nullspaceBasis (berlekampMatrix f) of
    basis | length basis < 2 ->
        Nothing
    basis ->
        let h = head basis in
        find ( isNonZeroNonUnit ) [ gcd f (h + c) | c <- field ]
        -- dont forget to apply the ^^^ gcd we talked about earlier!
```

Any Questions???



more.

WE CAN DO BETTER

WE CAN DO BETTER

We know that the following is a multiple of f

$$\prod_{c \in \mathbb{Z}_p} h + c$$

WE CAN DO BETTER

We know that the following is a multiple of f

$$\prod_{c \in \mathbb{Z}_p} h + c$$

So,

$$f \;=\; \gcd\left(f, \prod_{c \in \mathbb{Z}_p} h + c\right)$$

WE CAN DO BETTER

We know that the following is a multiple of f

$$\prod_{c \in \mathbb{Z}_p} h + c$$

So,

$$f \;=\; \gcd\left(f,\, \prod_{c \in \mathbb{Z}_p} h + c\right)$$

Pretty easy to show that

$$f \;=\; \prod_{c \in \mathbb{Z}_p} \gcd\left(f,\, h + c\right)$$

Berlekamp's Algorithm

```
factorBerlekamp :: Polynomial -> Set Polynomial
factorBerlekamp f = case nullspaceBasis (berlekampMatrix f) of
    basis | length basis > 1 ->
        let h = head basis in -- element of B_f
        let terms = filter
                ( isNonZeroNonUnit )
                [ gcd f (h + c) | c <- field ]
        in Set.unionMap factorBerlekamp terms
    basis ->
        basis -- f is irreducible
```

Berlekamp's Algorithm

```haskell
factorBerlekamp :: Polynomial -> Set Polynomial
factorBerlekamp f = case nullspaceBasis (berlekampMatrix f) of
    basis | length basis > 1 ->
        let h = head basis in -- element of B_f
        let terms = filter
                ( isNonZeroNonUnit )
                [ gcd f (h + c) | c <- field ]
        in Set.unionMap factorBerlekamp terms
    basis ->
        basis -- f is irreducible
```

funny: the dimension of B_f is the number of factors of f.