Error-Correcting Codes and The Power of Polynomials

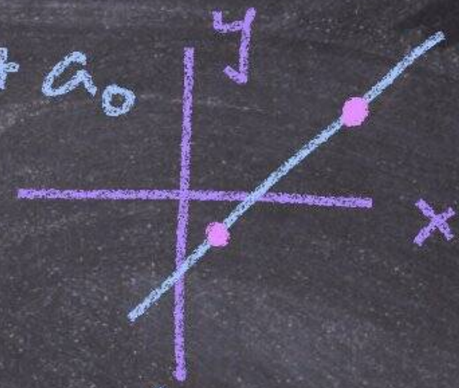
Manuel Sabin

Context (CS 70)

- Univariate polynomials e.g. $P(x) = 3x^2 + x + 5$
- Modular arithmetic e.g. $\mathbb{F}_p$
- Systems of linear equations e.g. $4a_2 + 2a_1 + a_0 = 5$
  $a_2 + a_1 + a_0 = 0$
  $a_0 = 3$

Warm Up

$P(x) = a_1 x + a_0$

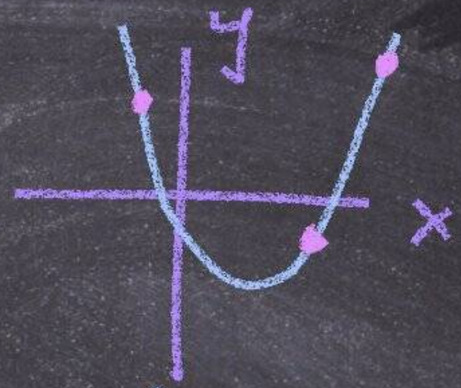


2 points uniquely define a line

3 points uniquely define a deg 2 polynomial

Warm Up

$P(x) = a_2x^2 + a_1x + a_0$



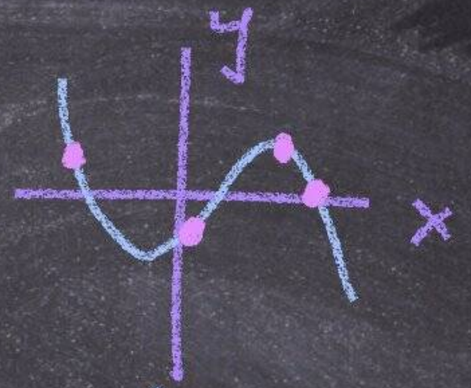
2 points uniquely define a line

3 points uniquely define a deg 2 polynomial

4 points uniquely define a deg 3 polynomial

Warm Up

$P(x) = a_3x^3 + a_2x^2 + a_1x + a_0$



2 points uniquely define a line

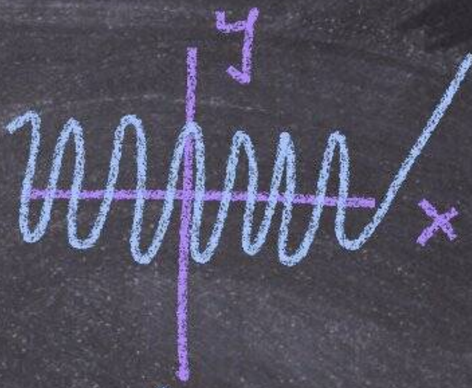
3 points uniquely define a deg 2 polynomial

4 points uniquely define a deg 3 polynomial

⋮

101 points uniquely define a deg 100 polynomial

Warm Up



2 points uniquely define a line

3 points uniquely define a deg 2 polynomial

4 points uniquely define a deg 3 polynomial

⋮

101 points uniquely define a deg 100 polynomial

⋮

d+1 points uniquely define a deg d polynomial

Degree d Polynomials

1) $d+1$ points uniquely define a deg d polynomial

cor If poly P, Q are deg d & $P(a) = Q(a)$ for $d+1$ inputs, a, then $P(x) = Q(x)$

2) Any non-zero deg d poly has $\leq d$ roots

- $x^2 - 11x + 30 = 0$
- $(x-5)(x-6) = 0 \leftsquigarrow a \cdot b = 0 \Rightarrow a = 0$ OR $b = 0$

Error-Correcting Codes and The Power of Polynomials

Manuel Sabin

Error-Correcting Codes

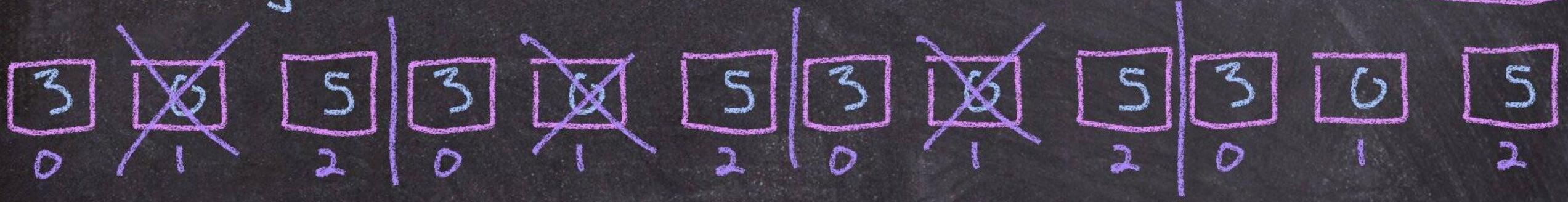
Message M

1 Erasure Error

Encoding C

Repitition Code

repeat k+1 times



k = 3 Erasure Errors

Repitition Code with k Erasure Errors

M is n symbols $\Rightarrow$ C is $n(k+1)$ symbols

What's wrong with this Erasure Code?

Say 1 symbol is 1 byte, $k = \frac{1}{3}n$, and M is a small photo $n = 3\text{MB} = 3 \times 10^6$ bytes

$\Downarrow$

codeword C is $n(k+1) = 3\times10^6\left(\frac{1}{3}\cancel{3}\times10^6+1\right) \approx 3\times10^{12}$ bytes

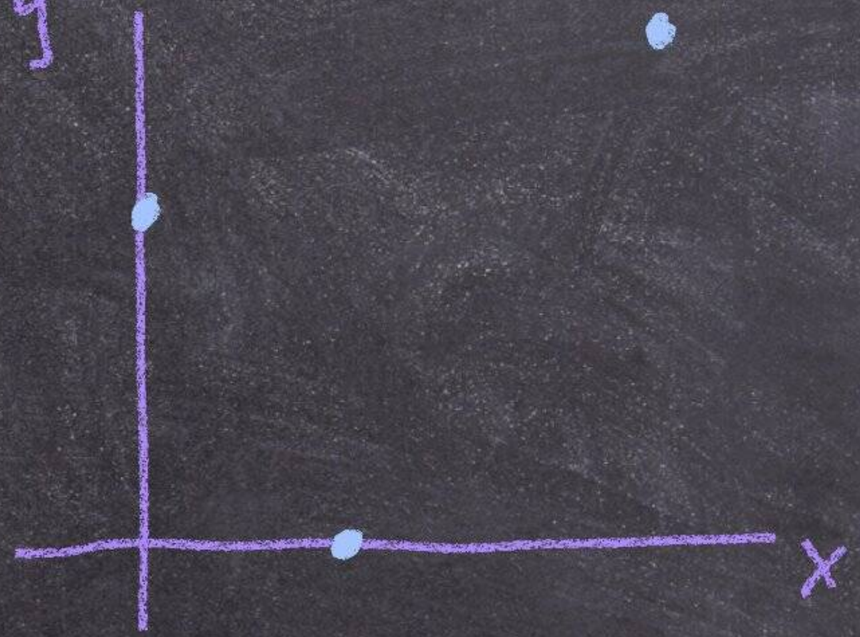
LARGE SSD 3 TB

Reed-Solomon Code for Erasure Errors

Message M

3	0	5
0	1	2

→ (0,3)
(1,0)
(2,5)



Degree d Polynomials

1) $d+1$ points uniquely define a deg d polynomial

cor If poly P, Q are deg d & $P(a) = Q(a)$ for $d+1$ inputs, a, then $P(x) = Q(x)$

2) Any non-zero deg d poly has $\leq d$ roots

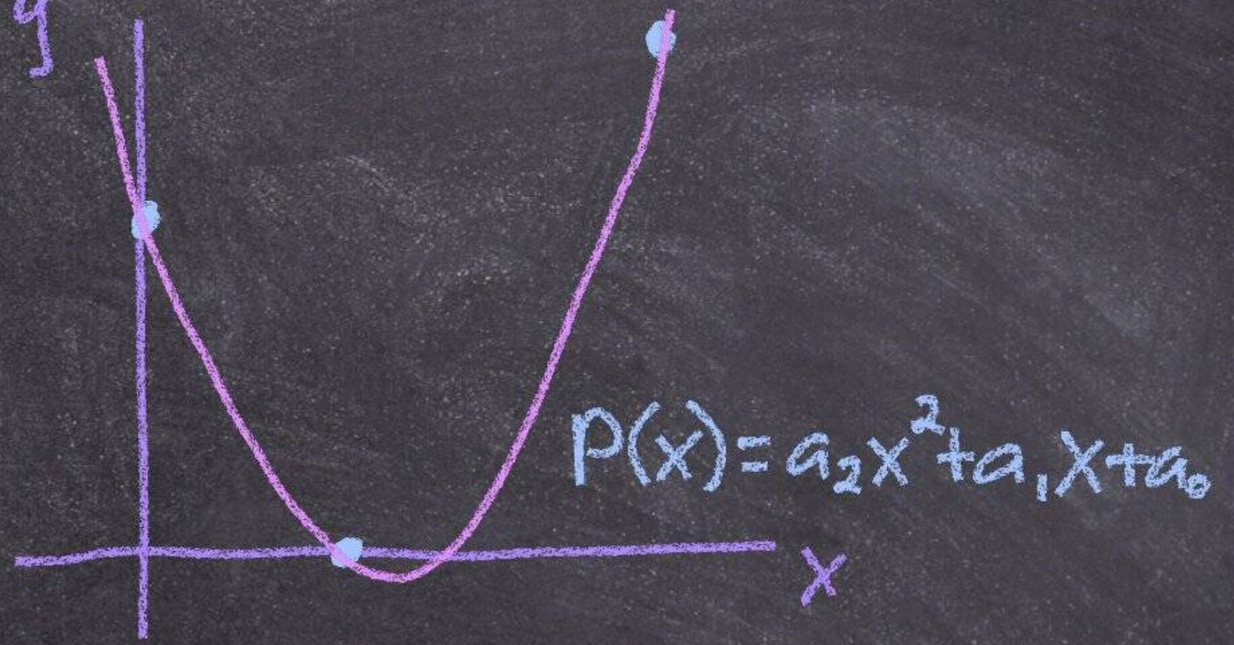
- $x^2 - 11x + 30 = 0$
- $(x-5)(x-6) = 0 \leftsquigarrow a \cdot b = 0 \Rightarrow a = 0$ OR $b = 0$

Reed-Solomon Code for Erasure Errors

Message M

3	0	5
0	1	2

→ (0,3) (1,0) (2,5)



↓

Encoding C

P(0)	P(1)	P(2)	P(3)
3	0	5	
0	1	2	3

Degree d Polynomials

1) $d+1$ points uniquely define a deg d polynomial

cor If poly P, Q are deg d & $P(a) = Q(a)$ for $d+1$ inputs, a, then $P(x) = Q(x)$

2) Any non-zero deg d poly has $\leq d$ roots

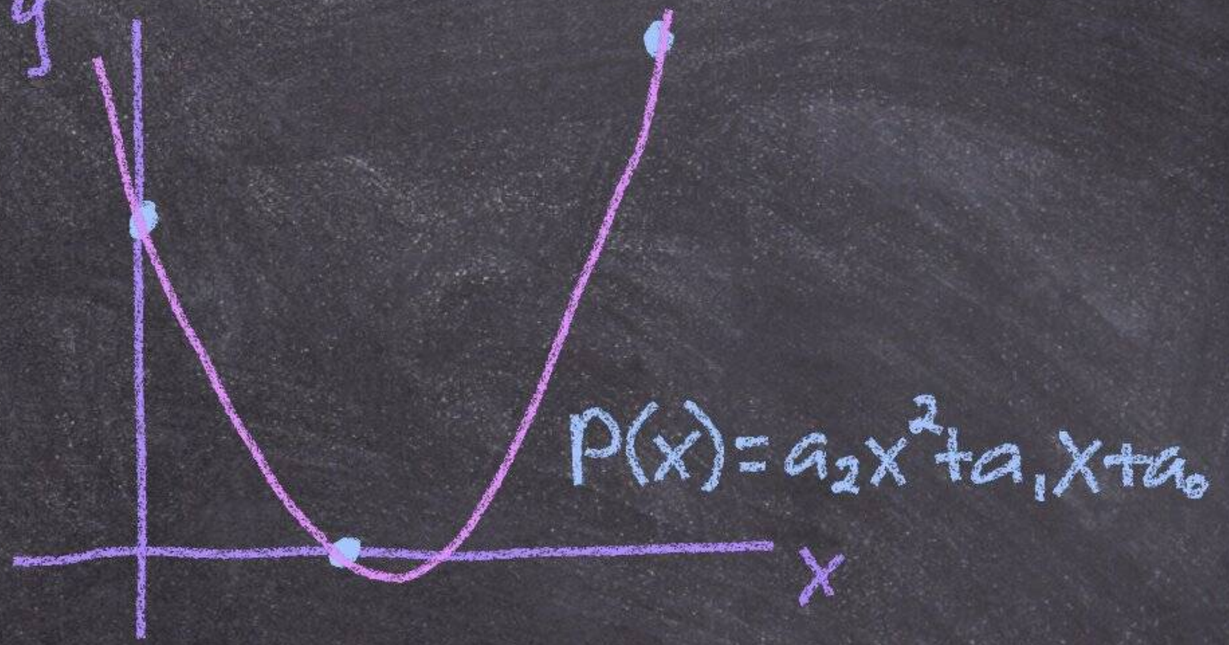
- $x^2 - 11x + 30 = 0$
- $(x-5)(x-6) = 0 \leftsquigarrow a \cdot b = 0 \Rightarrow a = 0$ OR $b = 0$

Reed-Solomon Code for Erasure Errors

Message M

3	0	5
0	1	2

→ (0,3) (1,0) (2,5)



↓

Encoding C

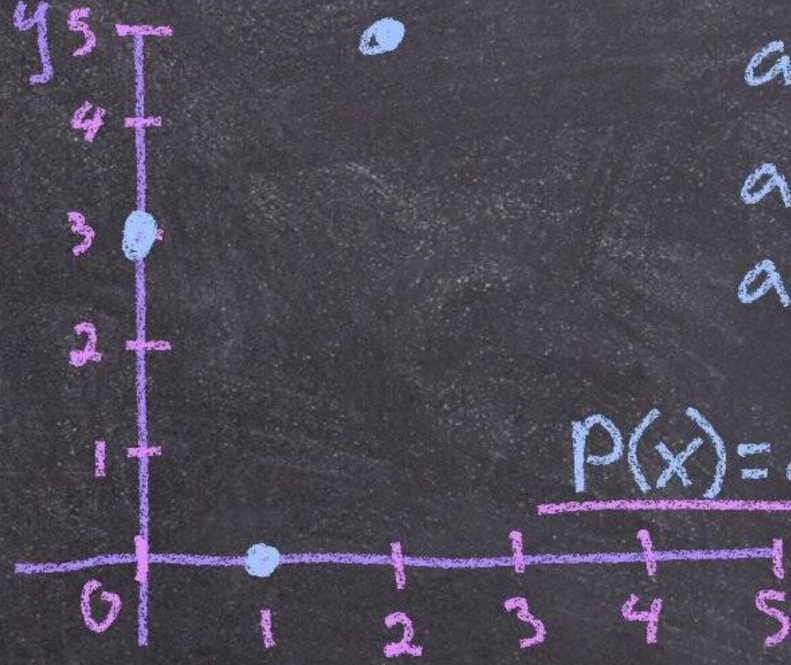
P(0)	~~P(1)~~	P(2)	P(3)
3	~~0~~	5	
0	1	2	3

Reed-Solomon Code for Erasure Errors

Message M

0	1	2
3	0	5

$\rightarrow$ (0,3) (1,0) (2,5)



$a_2 = 4$
$a_1 = 0$
$a_0 = 3$

$P(x) = a_2x^2 + a_1x + a_0 \pmod 7$

$\downarrow$

Encoding C

P(0)	P(1)	P(2)	P(3)
3	0	5	
0	1	2	3

$P(0) = a_2 0^2 + a_1 0 + a_0 = 3$

$P(1) = a_2 1^2 + a_1 1 + a_0 = 0$

$P(2) = a_2 2^2 + a_1 2 + a_0 = 5$

$a_0 = 3$

$\Rightarrow a_2 + a_1 + a_0 = 0$

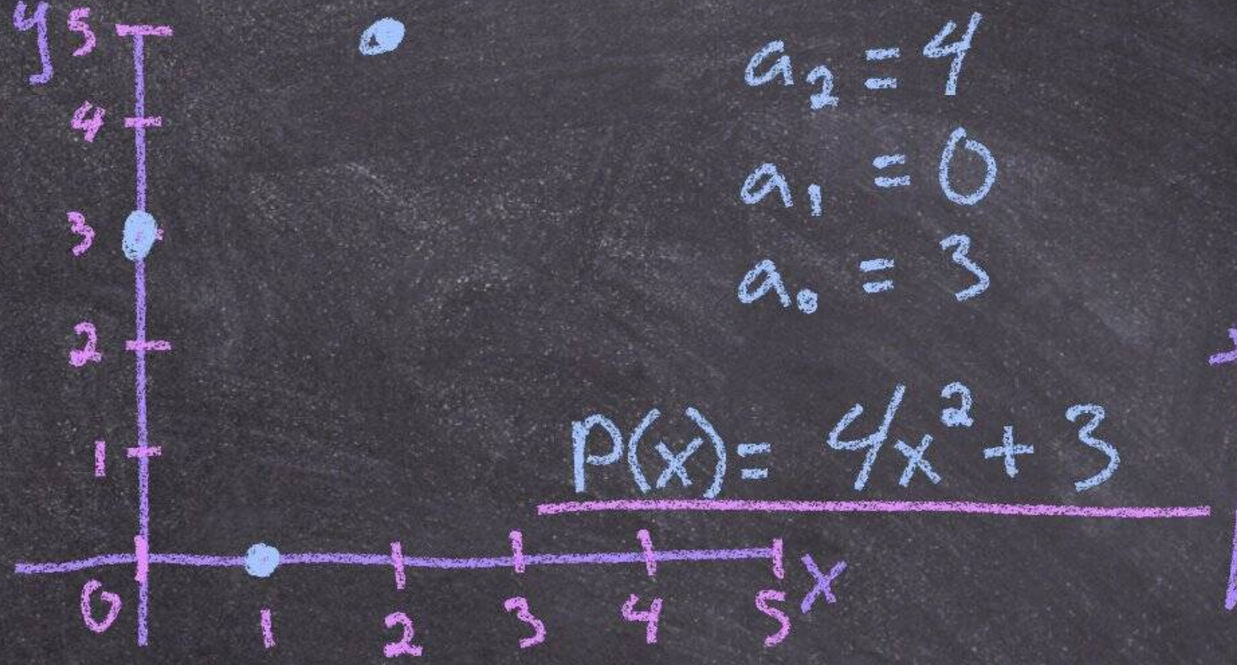
$4a_2 + 2a_1 + a_0 = 5$

Reed-Solomon Code for Erasure Errors

Message M

3	0	5
0	1	2

→ (0,3) (1,0) (2,5)



$a_2 = 4$
$a_1 = 0$
$a_0 = 3$

$P(x) = 4x^2 + 3 \pmod 7$

↓

Encoding C

P(0)	P(1)	P(2)	
3	0	5	P(3)
0	1	2	3

$P(0) = a_2 0^2 + a_1 0 + a_0 = 3$

$P(1) = a_2 1^2 + a_1 1 + a_0 = 0$

$P(2) = a_2 2^2 + a_1 2 + a_0 = 5$

$a_0 = 3$

$\Rightarrow a_2 + a_1 + a_0 = 0$

$4a_2 + 2a_1 + a_0 = 5$

Reed-Solomon Code for Erasure Errors

Message M

3	0	5
0	1	2

→ (0,3)
(1,0)
(2,5)

$P(x) = 4x^2 + 3 \pmod 7$

↓

Encoding C

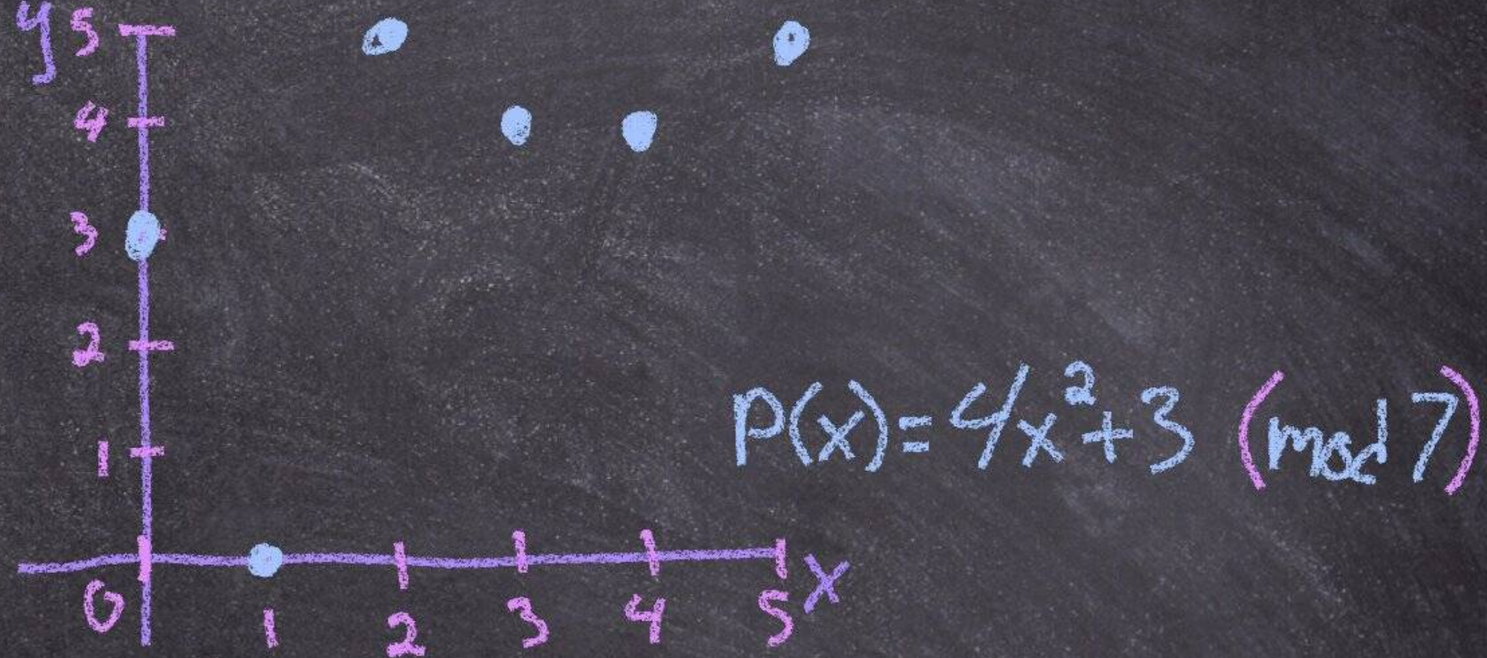
P(0)	P(1)	P(2)	
3	0	5	P(3)
0	1	2	3

Reed-Solomon Code for Erasure Errors

Message M

3	0	5
0	1	2

→ (0,3) (1,0) (2,5)



$P(x) = 4x^2 + 3 \pmod 7$

↓

Encoding C

P(0)	P(1)	P(2)	P(3)	P(4)	P(5)
3	0	5	4	4	5
0	1	2	3	4	5

k errors

~~(n)k~~ points of degree n-1 polynomial

Polys are data structures w/ structure

Reed-Solomon Code for Erasure Errors

M is n symbols $\Rightarrow$ C is n+k symbols

Why is this better than Repitition Codes?

Say 1 symbol is 1 byte, $k = \frac{1}{3}n$, and M is a small photo $n = 3MB = 3 \times 10^6$ bytes

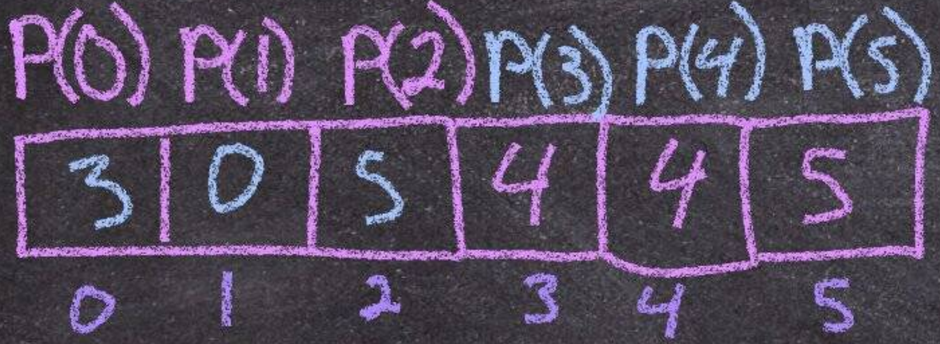
$\Downarrow$

codeword C is $n+k = 3 \times 10^6 + \frac{1}{3}3 \times 10^6 = 4 \times 10^6$ bytes

small photo 4MB

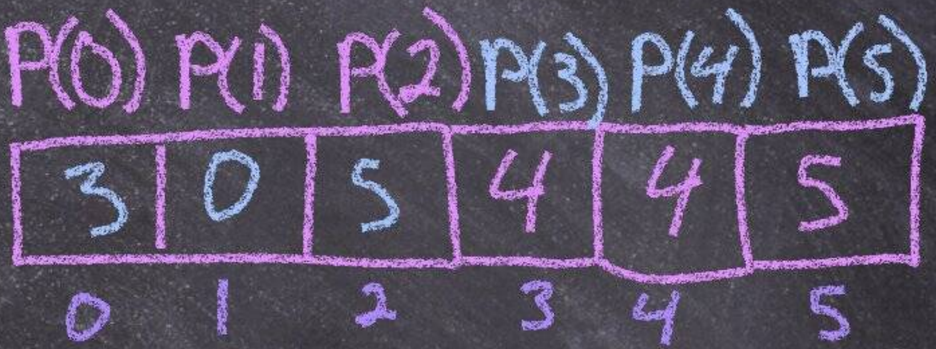
Reed-Solomon Code for General Errors

C



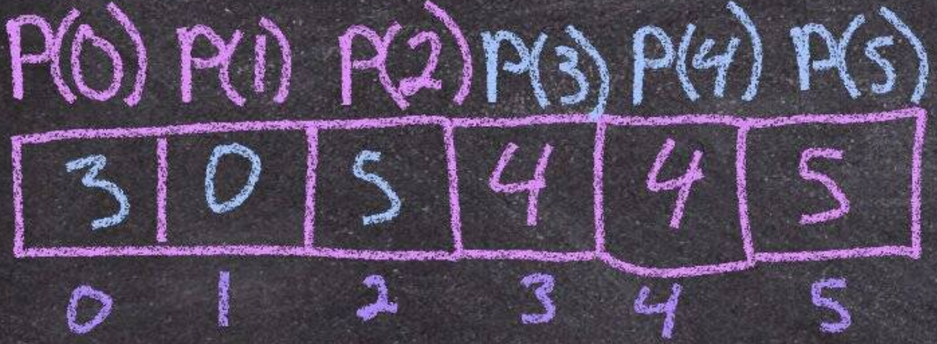
3 General Errors

R



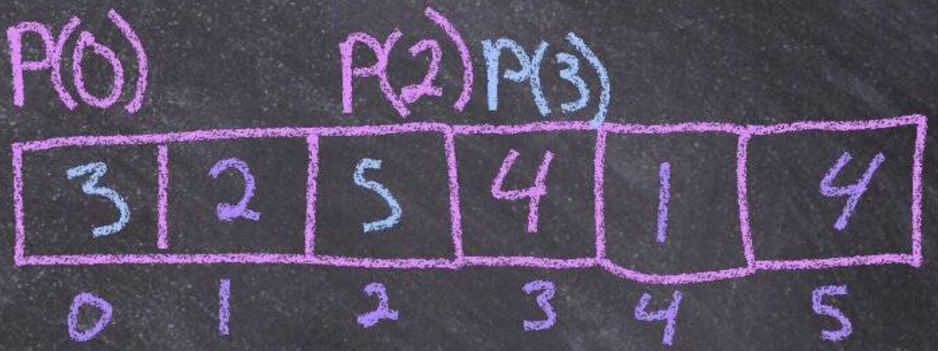
Reed-Solomon Code for General Errors

C



3 General Errors

R

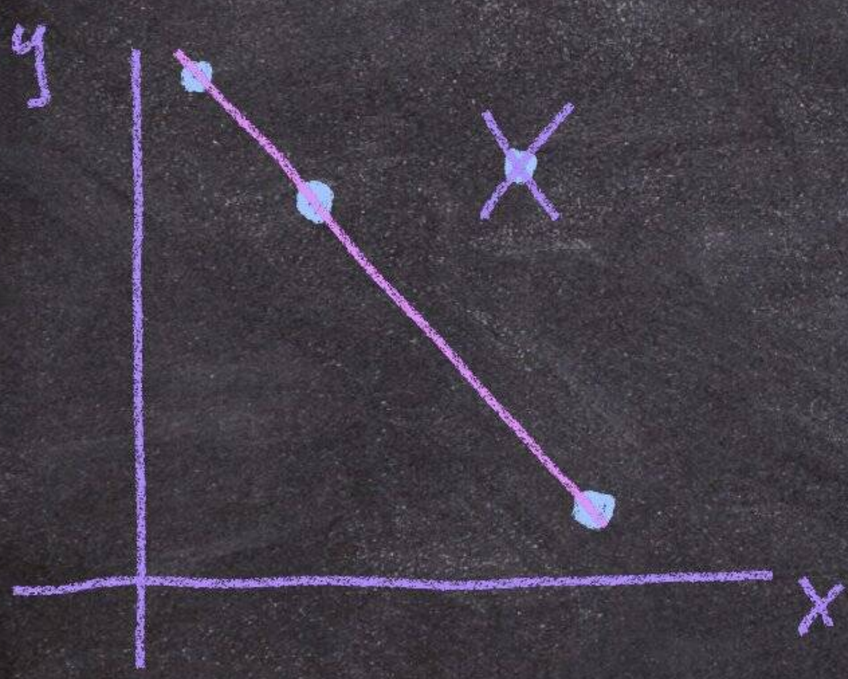


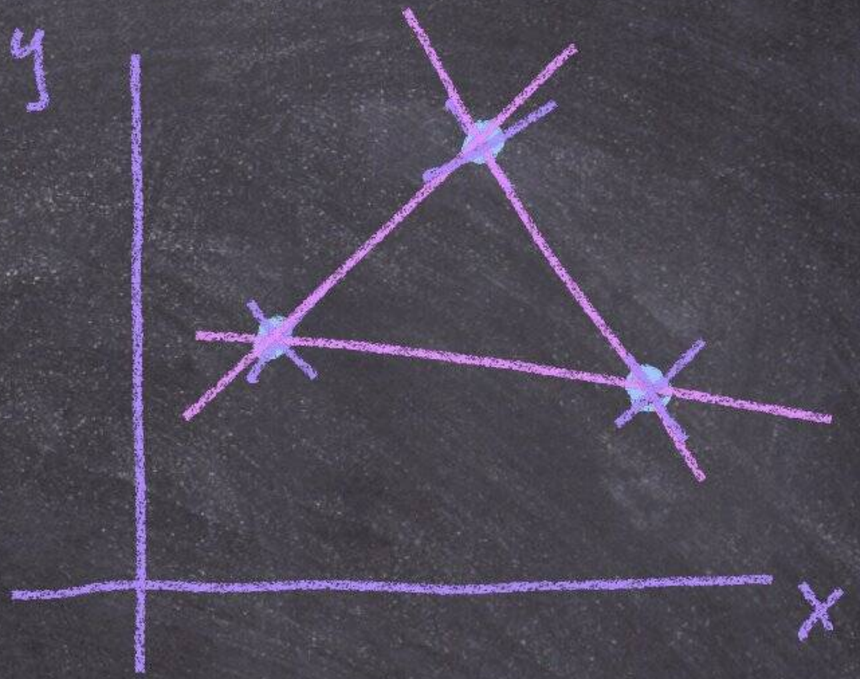
Context: Streams of Data

- Deep space communication
- CDs / DVDs
- QR Codes

Reed-Solomon Code for General Errors

n=2 i.e. line
k=1 error





What's special about passing through 3 points?

$3 = 2 + 1$ $k = 1$

Used 4 points

Reed-Solomon Code for General Errors

$n = 3$ i.e. parabola
$k = 2$ errors

What's special about passing through 5 points?

$5 = 3 + \cancel{2}$ $k = 2$

Reed-Solomon Code for General Errors

$n=2$ $k=1$ $\Rightarrow$ 4 symbol code word

$n=3$ $k=2$ $\Rightarrow$ 7 symbol code word

$n+2k$ symbol codeword for k errors $\Rightarrow$

1)

Reed-Solomon Code for General Errors

$n=2$ $k=1$ $\Rightarrow$ 4 symbol code word

$n=3$ $k=2$ $\Rightarrow$ 7 symbol code word

$n+k+k$ symbol codeword for k errors $\Rightarrow$

1) $n+k$ of the received points are uncorrupted
   i.e. There is a deg $\leq n-1$ poly passing through $n+k$ points
2) If a deg $\leq n-1$ poly passes through $n+k$ points, then it must be the codeword poly $P(x)$

Reed-Solomon Code for General Errors

$n+k$ symbol codeword for k errors $\Rightarrow$

1) $n+k$ of the received points are uncorrupted

i.e. There is a deg $\leq n-1$ poly passing through $n+k$ points

2) If a deg $\leq n-1$ poly passes through $n+k$ points, then it must be the codeword poly $P(x)$

pf

Say $Q(x)$ a deg $\leq n-1$ poly s.t. Q passes through $n+k$ points

Only k errors $\Rightarrow \geq n$ points are uncorrupted

$\Rightarrow Q$ passes through n points of $P \Rightarrow Q(x) = P(x)$

Degree d Polynomials

1) $d+1$ points uniquely define a deg d polynomial

cor If poly P, Q are deg d & $P(a) = Q(a)$ for $d+1$ inputs, a, then $P(x) = Q(x)$

2) Any non-zero deg d poly has $\leq d$ roots

- $x^2 - 11x + 30 = 0$
- $(x-5)(x-6) = 0 \leftsquigarrow a \cdot b = 0 \Rightarrow a = 0$ OR $b = 0$

Reed-Solomon Code for General Errors

$n+k$ symbol codeword for k errors $\Rightarrow$

1) $n+k$ of the received points are uncorrupted

i.e. There is a deg $\leq n-1$ poly passing through $n+k$ points

2) If a deg $\leq n-1$ poly passes through $n+k$ points, then it must be the codeword poly $P(x)$

pf

Say $Q(x)$ a deg $\leq n-1$ poly s.t. Q passes through $n+k$ points

Only k errors $\Rightarrow \geq n$ points are uncorrupted

$\Rightarrow Q$ passes through n points of $P \Rightarrow Q(x) = P(x)$ ✗

Reed-Solomon Code for General Errors

$n+2k$ symbol codeword for k errors $\Rightarrow$

Decode: Find deg $\leq n-1$ poly passing through $n+k$ points

Brute Force:

1) Guess k errors $\quad {}_{n+2k}C_k \approx (n+2k)^k \Rightarrow O(n^n)$

$k = \frac{1}{3}n$

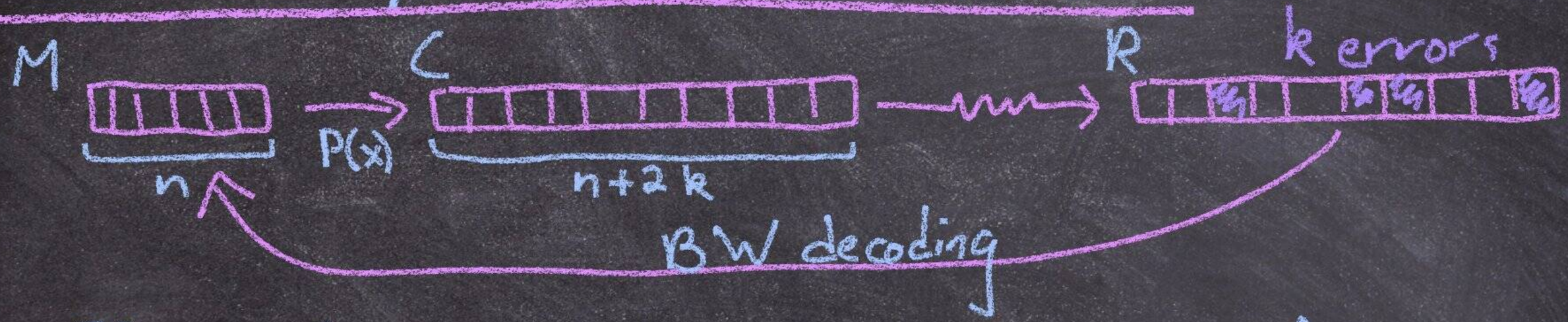
2) Interpolate remaining $n+k$ points for poly $Q(x)$

3) - If $\deg(Q) \leq n-1$, then $Q(x) = P(x)$ Done!!

- Else, go to Step 1

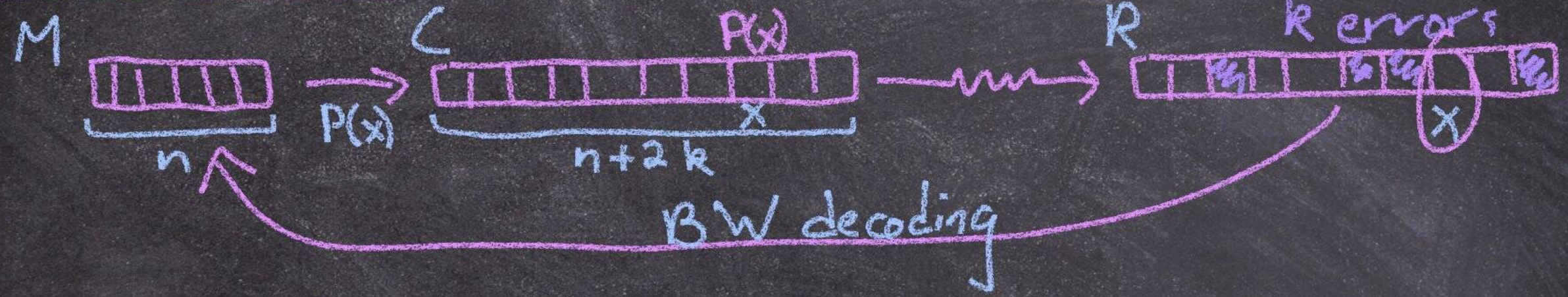
Find errors faster?

Berlekamp-Welch Decoder



- Efficiently corrects up to k general errors for n+2k RS
- Identify where the errors occurred
- Tell you when R is too corrupted to recover M

Berlekamp-Welch Decoder



Natural Language

R_x was uncorrupted $\longrightarrow$

Polynomial Land

$P(x) = R_x \iff P(x) - R_x = 0$

Berlekamp-Welch Decoder

Natural Language

R_x was uncorrupted $\longrightarrow$

Polynomial Land

$P(x) = R_x \iff P(x) - R_x = 0$

R_x was uncorrupted

(OR) $\longrightarrow$

There was an error at index x

Degree d Polynomials

1) $d+1$ points uniquely define a deg d polynomial

cor If poly P, Q are deg d & $P(a) = Q(a)$ for $d+1$ inputs, a, then $P(x) = Q(x)$

2) Any non-zero deg d poly has $\leq d$ roots

- $x^2 - 11x + 30 = 0$
- $(x-5)(x-6) = 0 \leftsquigarrow a \cdot b = 0 \Rightarrow a = 0$ OR $b = 0$

Berlekamp-Welch Decoder

Natural Language

R_x was uncorrupted $\longrightarrow$

Polynomial Land

$P(x) = R_x \iff P(x) - R_x = 0$

R_x was uncorrupted

(OR)

There was an error at index x

$\longrightarrow$

$E(x)(P(x) - R_x) = 0$

$\iff$

$P(x) - R_x = 0$

OR

$E(x) = 0$

Berlekamp-Welch Decoder

Natural Language

R_x was uncorrupted

OR

There was an error at index x

deg k polynomial

Polynomial Land

$$E(x)(P(x) - R_x) = 0$$

$$\Leftrightarrow$$

$$P(x) - R_x = 0$$

OR

$$E(x) = 0$$

Berlekamp-Welch Decoder

Natural Language

R_x was uncorrupted

OR

There was an error at index x

deg k polynomial

Polynomial Land

$$E(x)(P(x) - R_x) = 0$$

$\Leftrightarrow$

$P(x) - R_x = 0$

OR

$E(x) = 0$

If $P(x) \neq R_x$, then $E(x) = 0$

$\Leftrightarrow$ Roots of E are errors

Degree d Polynomials

1) $d+1$ points uniquely define a deg d polynomial

cor If poly P, Q are deg d & $P(a) = Q(a)$ for $d+1$ inputs, a, then $P(x) = Q(x)$

2) Any non-zero deg d poly has $\leq d$ roots

- $x^2 - 11x + 30 = 0$
- $(x-5)(x-6) = 0 \leftsquigarrow a \cdot b = 0 \Rightarrow a = 0$ OR $b = 0$

Berlekamp-Welch Decoder

Natural Language

R_x was uncorrupted

OR

There was an error at index x

deg k polynomial

Polynomial Land

$$E(x)(P(x) - R_x) = 0$$

$$\Leftrightarrow \quad P(x) - R_x = 0 \quad \text{OR} \quad E(x) = 0$$

If $P(x) \neq R_x$, then $E(x) = 0$

$\Leftrightarrow$ Roots of E are errors $e_1, \ldots, e_k$

$\Leftrightarrow E(x) = (x - e_1)(x - e_2) \cdots (x - e_k)$

Berlekamp-Welch Decoder

Natural Language

R_x was uncorrupted

OR

There was an error at index e_1 OR e_2 OR ⋮ OR e_k

Polynomial Land

deg k polynomial

$$E(x)(P(x)-R_x)=0$$

$$P(x)-R_x=0 \quad \text{OR} \quad E(x)=0$$

If $P(x) \neq R_x$, then $E(x)=0$

$\Leftrightarrow$ Roots of E are errors $e_1, \ldots, e_k$

$\Leftrightarrow E(x)=(x-e_1)(x-e_2)\cdots(x-e_k)$

Berlekamp-Welch Decoder

deg k polynomial

$$E(x)(P(x)-R_x) = 0 \iff \underbrace{E(x)P(x)}_{Q(x)} = R_x E(x)$$

$Q(x)$: deg $n+k-1$; $E(x)$: deg k; $P(x)$: deg $n-1$

$$\Rightarrow E(x) = (x-e_1)(x-e_2)\cdots(x-e_k) \text{ for the errors } e_1, \ldots, e_k$$

Error Locator Polynomial

$$Q(x) = a_{n+k-1}x^{n+k-1} + \cdots + a_2x^2 + a_1x + a_0$$

$$R_x E(x) = R_x(b_k x^k + \cdots b_2x^2 + b_1x + b_0)$$

Berlekamp-Welch Decoder

deg k polynomial

$$E(x)(P(x)-R_x) = 0 \iff \underbrace{E(x)P(x)}_{Q(x),\ \deg n+k-1} = R_x E(x)$$

($E(x)$: deg k; $P(x)$: deg $n-1$)

$$\Rightarrow E(x) = (x-e_1)(x-e_2)\cdots(x-e_k) \text{ for the errors } e_1, \ldots, e_k$$

Error Locator Polynomial

$$\underbrace{a_{n+k-1}x^{n+k-1} + \cdots + a_2x^2 + a_1x + a_0}_{n+k \text{ unknowns}} = R_x\underbrace{(b_kx^k + \cdots b_2x^2 + b_1x + b_0)}_{k+1 \text{ unknows}}$$

$n+2k$ unknowns

Berlekamp-Welch Decoder

$$Q(x) = E(x)P(x) = R_x E(x)$$

$$a_{n+k-1}x^{n+k-1} + \cdots + a_2x^2 + a_1x + a_0 = R_x(x^k + \cdots b_2x^2 + b_1x + b_0)$$

n+2k unknowns

Berlekamp-Welch Decoder

$$Q(x) = E(x)P(x) = R_x E(x)$$

$$a_{n+k-1}0^{n+k-1} + \cdots + a_2 0^2 + a_1 + a_0 = R_0(0^k + \cdots b_2 0^2 + b_1 0 + b_0)$$

$$a_{n+k-1}1^{n+k-1} + \cdots + a_2 1^2 + a_1 1 + a_0 = R_1(1^k + \cdots b_2 1^2 + b_1 1 + b_0)$$

$$\vdots$$

$$a_{n+k-1}(n+2k)^{n+k-1} + \cdots + a_2(n+2k)^2 + a_1(n+2k) + a_0 = R_{n+2k}((n+2k)^k + \cdots b_2(n+2k)^2 + b_1(n+2k) + b_0)$$

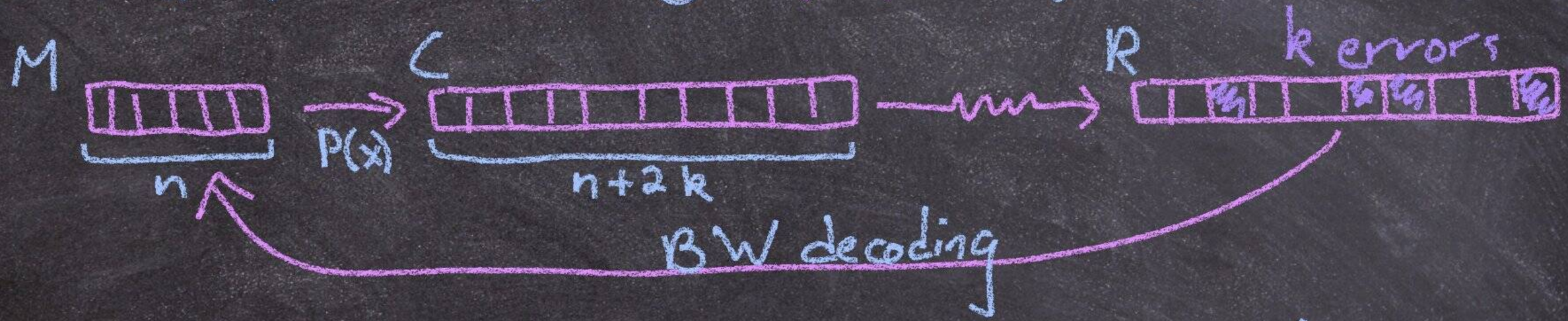
$n+2k$ unknowns

$n+2k$ equations

$\Rightarrow$ Solve for Q & E $\Rightarrow$ $P(x) = \frac{Q(x)}{E(x)}$

Berlekamp-Welch Decoder

$$E(x)(P(x)-R_x) = 0 \iff E(x)P(x) = R_x E(x)$$



- Efficiently corrects up to k general errors for n+2k RS
- Identify where the errors occurred
- Tell you when R is too corrupted to recover M