Error Correction

Satellite

GPS device
Error Correction

Satellite

3 packet message.

GPS device
Error Correction

Satellite

GPS device

3 packet message.

Corrupts 1 packets.
Error Correction

Satellite

Corrupts 1 packets.

3 packet message. Send 5.

GPS device
Error Correction

Satellite

3 packet message. Send 5.

Corrupts 1 packets.
Error Correction

Satellite

3 packet message. Send 5.

Corrupts 1 packets.
The Scheme.

**Problem:** Communicate $n$ packets $m_1, \ldots, m_n$ on noisy channel that corrupts $\leq k$ packets.
The Scheme.

**Problem:** Communicate \( n \) packets \( m_1, \ldots, m_n \) on noisy channel that corrupts \( \leq k \) packets.

**Reed-Solomon Code:**

1. Make a polynomial, \( P(x) \), of degree \( n - 1 \), that encodes message.
   - \( P(1) = m_1, \ldots, P(n) = m_n \).
   - Comment: could encode with packets as coefficients.
2. Send \( P(1), \ldots, P(n+2k) \).
   - After noisy channel: receive values \( R(1), \ldots, R(n+2k) \).

Properties:

1. \( P(i) = R(i) \) for at least \( n+k \) points \( i \),
2. \( P(x) \) is unique degree \( n - 1 \) polynomial that contains \( \geq n+k \) received points.
The Scheme.

**Problem:** Communicate \( n \) packets \( m_1, \ldots, m_n \) on noisy channel that corrupts \( \leq k \) packets.

**Reed-Solomon Code:**

1. Make a polynomial, \( P(x) \) of degree \( n - 1 \), that encodes message.
   
   ▶ \( P(1) = m_1, \ldots, P(n) = m_n \).
The Scheme.

**Problem:** Communicate $n$ packets $m_1, \ldots, m_n$ on noisy channel that corrupts $\leq k$ packets.

**Reed-Solomon Code:**

1. Make a polynomial, $P(x)$ of degree $n - 1$, that encodes message.
   - $P(1) = m_1, \ldots, P(n) = m_n$.
   - Comment: could encode with packets as coefficients.
The Scheme.

**Problem:** Communicate $n$ packets $m_1, \ldots, m_n$ on noisy channel that corrupts $\leq k$ packets.

**Reed-Solomon Code:**

1. Make a polynomial, $P(x)$ of degree $n - 1$, that encodes message.
   - $P(1) = m_1, \ldots, P(n) = m_n$.
   - Comment: could encode with packets as coefficients.

2. Send $P(1), \ldots, P(n + 2k)$.
The Scheme.

**Problem:** Communicate $n$ packets $m_1, \ldots, m_n$ on noisy channel that corrupts $\leq k$ packets.

**Reed-Solomon Code:**

1. Make a polynomial, $P(x)$ of degree $n - 1$, that encodes message.
   - $P(1) = m_1, \ldots, P(n) = m_n$.
   - Comment: could encode with packets as coefficients.

2. Send $P(1), \ldots, P(n + 2k)$.

**After noisy channel:** Recieve values $R(1), \ldots, R(n + 2k)$. 
The Scheme.

**Problem:** Communicate $n$ packets $m_1, \ldots, m_n$ on noisy channel that corrupts $\leq k$ packets.

**Reed-Solomon Code:**

1. Make a polynomial, $P(x)$ of degree $n - 1$, that encodes message.
   - $P(1) = m_1, \ldots, P(n) = m_n$.
   - Comment: could encode with packets as coefficients.

2. Send $P(1), \ldots, P(n + 2k)$.

**After noisy channel:** Receive values $R(1), \ldots, R(n + 2k)$.

**Properties:**

1. $P(i) = R(i)$ for at least $n + k$ points $i$, 
The Scheme.

**Problem:** Communicate $n$ packets $m_1, \ldots, m_n$ on noisy channel that corrupts $\leq k$ packets.

**Reed-Solomon Code:**

1. Make a polynomial, $P(x)$ of degree $n - 1$, that encodes message.
   - $P(1) = m_1, \ldots, P(n) = m_n$.
   - Comment: could encode with packets as coefficients.

2. Send $P(1), \ldots, P(n+2k)$.

**After noisy channel:** Recieve values $R(1), \ldots, R(n+2k)$.

**Properties:**

1. $P(i) = R(i)$ for at least $n + k$ points $i$,
2. $P(x)$ is unique degree $n - 1$ polynomial
The Scheme.

**Problem:** Communicate \( n \) packets \( m_1, \ldots, m_n \) on noisy channel that corrupts \( \leq k \) packets.

**Reed-Solomon Code:**

1. Make a polynomial, \( P(x) \) of degree \( n - 1 \), that encodes message.
   - \( P(1) = m_1, \ldots, P(n) = m_n \).
   - Comment: could encode with packets as coefficients.

2. Send \( P(1), \ldots, P(n + 2k) \).

**After noisy channel:** Recieve values \( R(1), \ldots, R(n + 2k) \).

**Properties:**

1. \( P(i) = R(i) \) for at least \( n + k \) points \( i \),
2. \( P(x) \) is unique degree \( n - 1 \) polynomial that contains \( \geq n + k \) received points.
Properties: proof.

\[ P(x): \text{degree } n - 1 \text{ polynomial.} \]
Properties: proof.

\[ P(x) \]: degree \( n - 1 \) polynomial.
Send \( P(1), \ldots, P(n + 2k) \)
Properties: proof.

\( P(x) \): degree \( n - 1 \) polynomial.
Send \( P(1), \ldots, P(n + 2k) \)
Receive \( R(1), \ldots, R(n + 2k) \)
Properties: proof.

\[ P(x) \]: degree \( n - 1 \) polynomial.
Send \( P(1), \ldots, P(n + 2k) \)
Receive \( R(1), \ldots, R(n + 2k) \)
At most \( k \) 'i's where \( P(i) \neq R(i) \).
Properties: proof.

\[ P(x) \]: degree \( n - 1 \) polynomial.
Send \( P(1), \ldots, P(n+2k) \)
Receive \( R(1), \ldots, R(n+2k) \)
At most \( k \) i’s where \( P(i) \neq R(i) \).

Properties:
(1) \( P(i) = R(i) \) for at least \( n + k \) points \( i \),
Properties: proof.

\( P(x) \): degree \( n - 1 \) polynomial.
Send \( P(1), \ldots, P(n + 2k) \)
Receive \( R(1), \ldots, R(n + 2k) \)
At most \( k \) i's where \( P(i) \neq R(i) \).

Properties:
(1) \( P(i) = R(i) \) for at least \( n + k \) points \( i \),
(2) \( P(x) \) is unique degree \( n - 1 \) polynomial
Properties: proof.

$P(x)$: degree $n - 1$ polynomial.
Send $P(1), \ldots, P(n+2k)$
Receive $R(1), \ldots, R(n+2k)$
At most $k$ i’s where $P(i) \neq R(i)$.

Properties:
(1) $P(i) = R(i)$ for at least $n+k$ points $i$,
(2) $P(x)$ is unique degree $n - 1$ polynomial
    that contains $\geq n+k$ received points.
Properties: proof.

\( P(x) \): degree \( n - 1 \) polynomial.
Send \( P(1), \ldots, P(n + 2k) \)
Receive \( R(1), \ldots, R(n + 2k) \)
At most \( k \) i’s where \( P(i) \neq R(i) \).

Properties:
(1) \( P(i) = R(i) \) for at least \( n + k \) points \( i \),
(2) \( P(x) \) is unique degree \( n - 1 \) polynomial
    that contains \( \geq n + k \) received points.

Proof:
Properties: proof.

\( P(x) \): degree \( n - 1 \) polynomial.
Send \( P(1), \ldots, P(n+2k) \)
Receive \( R(1), \ldots, R(n+2k) \)
At most \( k \) \( i \)'s where \( P(i) \neq R(i) \).

Properties:
(1) \( P(i) = R(i) \) for at least \( n+k \) points \( i \),
(2) \( P(x) \) is unique degree \( n - 1 \) polynomial
    that contains \( \geq n+k \) received points.

Proof:
(1) Sure.
Properties: proof.

\[ P(x) : \text{degree } n - 1 \text{ polynomial.} \]
Send \[ P(1), \ldots, P(n+2k) \]
Receive \[ R(1), \ldots, R(n+2k) \]
At most \( k \) \( i \)'s where \( P(i) \neq R(i) \).

Properties:
1. \( P(i) = R(i) \) for at least \( n + k \) points \( i \),
2. \( P(x) \) is unique degree \( n - 1 \) polynomial
   that contains \( \geq n + k \) received points.

Proof:
(1) Sure. Only \( k \) corruptions.
Properties: proof.

$P(x)$: degree $n-1$ polynomial.
Send $P(1), \ldots, P(n+2k)$
Receive $R(1), \ldots, R(n+2k)$
At most $k$ i’s where $P(i) \neq R(i)$.

Properties:
(1) $P(i) = R(i)$ for at least $n+k$ points $i$,
(2) $P(x)$ is unique degree $n-1$ polynomial
    that contains $\geq n+k$ received points.

Proof:
(1) Sure. Only $k$ corruptions.
(2) Degree $n-1$ polynomial $Q(x)$ consistent with $n+k$ points.
Properties: proof.

\( P(x) \): degree \( n - 1 \) polynomial.

Send \( P(1), \ldots, P(n+2k) \)

Receive \( R(1), \ldots, R(n+2k) \)

At most \( k \) i’s where \( P(i) \neq R(i) \).

Properties:

1. \( P(i) = R(i) \) for at least \( n + k \) points \( i \),
2. \( P(x) \) is unique degree \( n - 1 \) polynomial that contains \( \geq n + k \) received points.

Proof:

1. Sure. Only \( k \) corruptions.
2. Degree \( n - 1 \) polynomial \( Q(x) \) consistent with \( n + k \) points. 
   \( Q(x) \) agrees with \( R(i) \), \( n + k \) times.
Properties: proof.

\( P(x) \): degree \( n - 1 \) polynomial.
Send \( P(1), \ldots, P(n + 2k) \)
Receive \( R(1), \ldots, R(n + 2k) \)
At most \( k \) \( i \)'s where \( P(i) \neq R(i) \).

Properties:
(1) \( P(i) = R(i) \) for at least \( n + k \) points \( i \),
(2) \( P(x) \) is unique degree \( n - 1 \) polynomial
     that contains \( \geq n + k \) received points.

Proof:
(1) Sure. Only \( k \) corruptions.
(2) Degree \( n - 1 \) polynomial \( Q(x) \) consistent with \( n + k \) points.
    \( Q(x) \) agrees with \( R(i) \), \( n + k \) times.
    \( P(x) \) agrees with \( R(i) \), \( n + k \) times.
Properties: proof.

\( P(x) \): degree \( n - 1 \) polynomial.
Send \( P(1), \ldots, P(n + 2k) \)
Receive \( R(1), \ldots, R(n + 2k) \)
At most \( k \) i’s where \( P(i) \neq R(i) \).

Properties:

(1) \( P(i) = R(i) \) for at least \( n + k \) points \( i \),
(2) \( P(x) \) is unique degree \( n - 1 \) polynomial
    that contains \( \geq n + k \) received points.

Proof:

(1) Sure. Only \( k \) corruptions.
(2) Degree \( n - 1 \) polynomial \( Q(x) \) consistent with \( n + k \) points.
    \( Q(x) \) agrees with \( R(i), n + k \) times.
    \( P(x) \) agrees with \( R(i), n + k \) times.
    Total points contained by both: \( 2n + 2k \).
Properties: proof.

\( P(x) \): degree \( n - 1 \) polynomial.
Send \( P(1), \ldots, P(n + 2k) \)
Receive \( R(1), \ldots, R(n + 2k) \)
At most \( k \) i’s where \( P(i) \neq R(i) \).

Properties:
(1) \( P(i) = R(i) \) for at least \( n + k \) points \( i \),
(2) \( P(x) \) is unique degree \( n - 1 \) polynomial
that contains \( \geq n + k \) received points.

Proof:
(1) Sure. Only \( k \) corruptions.
(2) Degree \( n - 1 \) polynomial \( Q(x) \) consistent with \( n + k \) points.
\( Q(x) \) agrees with \( R(i), n + k \) times.
\( P(x) \) agrees with \( R(i), n + k \) times.
Total points contained by both: \( 2n + 2k \). \( P \) Pigeons.
Properties: proof.

\( P(x) \): degree \( n - 1 \) polynomial.
Send \( P(1), \ldots, P(n + 2k) \)
Receive \( R(1), \ldots, R(n + 2k) \)
At most \( k \) \( i \)’s where \( P(i) \neq R(i) \).

Properties:
(1) \( P(i) = R(i) \) for at least \( n + k \) points \( i \),
(2) \( P(x) \) is unique degree \( n - 1 \) polynomial
that contains \( \geq n + k \) received points.

Proof:
(1) Sure. Only \( k \) corruptions.
(2) Degree \( n - 1 \) polynomial \( Q(x) \) consistent with \( n + k \) points.
\( Q(x) \) agrees with \( R(i), n + k \) times.
\( P(x) \) agrees with \( R(i), n + k \) times.
Total points contained by both: \( 2n + 2k \). \( P \) Pigeons.
Total points to choose from : \( n + 2k \).
Properties: proof.

\[ P(x) \]: degree \( n - 1 \) polynomial.
Send \( P(1), \ldots, P(n+2k) \)
Receive \( R(1), \ldots, R(n+2k) \)
At most \( k \) i’s where \( P(i) \neq R(i) \).

Properties:
1. \( P(i) = R(i) \) for at least \( n+k \) points \( i \),
2. \( P(x) \) is unique degree \( n - 1 \) polynomial
   that contains \( \geq n + k \) received points.

Proof:
1. Sure. Only \( k \) corruptions.
2. Degree \( n - 1 \) polynomial \( Q(x) \) consistent with \( n + k \) points.
   \( Q(x) \) agrees with \( R(i), n+k \) times.
   \( P(x) \) agrees with \( R(i), n+k \) times.
   Total points contained by both: \( 2n+2k \).
   \( P \) Pigeons.
   Total points to choose from : \( n+2k \).
   \( H \) Holes.
Properties: proof.

\( P(x) \): degree \( n - 1 \) polynomial.
Send \( P(1), \ldots, P(n+2k) \)
Receive \( R(1), \ldots, R(n+2k) \)
At most \( k \) 's where \( P(i) \neq R(i) \).

Properties:
1. \( P(i) = R(i) \) for at least \( n+k \) points \( i \),
2. \( P(x) \) is unique degree \( n - 1 \) polynomial that contains \( \geq n+k \) received points.

Proof:
1. Sure. Only \( k \) corruptions.
2. Degree \( n - 1 \) polynomial \( Q(x) \) consistent with \( n+k \) points.
   \( Q(x) \) agrees with \( R(i), n+k \) times.
   \( P(x) \) agrees with \( R(i), n+k \) times.

Total points contained by both: \( 2n+2k \). \( P \) Pigeons.
Total points to choose from : \( n+2k \). \( H \) Holes.
Points contained by both : \( \geq n \).
Properties: proof.

\( P(x) \): degree \( n - 1 \) polynomial.
Send \( P(1), \ldots, P(n+2k) \)
Receive \( R(1), \ldots, R(n+2k) \)
At most \( k \) i's where \( P(i) \neq R(i) \).

Properties:
(1) \( P(i) = R(i) \) for at least \( n+k \) points \( i \),
(2) \( P(x) \) is unique degree \( n - 1 \) polynomial
    that contains \( \geq n+k \) received points.

Proof:
(1) Sure. Only \( k \) corruptions.
(2) Degree \( n - 1 \) polynomial \( Q(x) \) consistent with \( n+k \) points.
    \( Q(x) \) agrees with \( R(i), n+k \) times.
    \( P(x) \) agrees with \( R(i), n+k \) times.
    Total points contained by both: \( 2n+2k \). \( P \) Pigeons.
    Total points to choose from : \( n+2k \). \( H \) Holes.
    Points contained by both : \( \geq n \). \( \geq P - H \) Collisions.
    \( \implies Q(i) = P(i) \) at \( n \) points.
Properties: proof.

\( P(x) \): degree \( n - 1 \) polynomial.
Send  \( P(1), \ldots, P(n+2k) \)
Receive \( R(1), \ldots, R(n+2k) \)
At most \( k \) 's where \( P(i) \neq R(i) \).

**Properties:**
(1) \( P(i) = R(i) \) for at least \( n+k \) points \( i \),
(2) \( P(x) \) is unique degree \( n - 1 \) polynomial
    that contains \( \geq n+k \) received points.

**Proof:**
(1) Sure. Only \( k \) corruptions.
(2) Degree \( n - 1 \) polynomial \( Q(x) \) consistent with \( n+k \) points.

\( Q(x) \) agrees with \( R(i), n+k \) times.
\( P(x) \) agrees with \( R(i), n+k \) times.
Total points contained by both: \( 2n+2k \).  \( P \) Pigeons.
Total points to choose from : \( n+2k \).  \( H \) Holes.
Points contained by both : \( \geq n \). \( \geq P - H \) Collisions.

\[ Q(i) = P(i) \text{ at } n \text{ points.} \]
\[ Q(x) = P(x). \]
Properties: proof.

\[ P(x) : \text{degree } n - 1 \text{ polynomial.} \]
Send \( P(1), \ldots, P(n + 2k) \)
Receive \( R(1), \ldots, R(n + 2k) \)
At most \( k \) i’s where \( P(i) \neq R(i) \).

Properties:
(1) \( P(i) = R(i) \) for at least \( n + k \) points \( i \),
(2) \( P(x) \) is unique degree \( n - 1 \) polynomial
    that contains \( \geq n + k \) received points.

Proof:
(1) Sure. Only \( k \) corruptions.
(2) Degree \( n - 1 \) polynomial \( Q(x) \) consistent with \( n + k \) points.
    \( Q(x) \) agrees with \( R(i) \), \( n + k \) times.
    \( P(x) \) agrees with \( R(i) \), \( n + k \) times.
    Total points contained by both: \( 2n + 2k \).
    Total points to choose from : \( n + 2k \).
    Points contained by both : \( \geq n \).
    \( \geq P - H \) Collisions.
    \( \implies Q(i) = P(i) \) at \( n \) points.
    \( \implies Q(x) = P(x). \)
Example.

Message: 3, 0, 6.
Example.

Message: 3, 0, 6.

Reed Solomon Code: \( P(x) = x^2 + x + 1 \) (mod 7) has \( P(1) = 3, P(2) = 0, P(3) = 6 \) modulo 7.
Example.

Message: 3, 0, 6.
Reed Solomon Code: $P(x) = x^2 + x + 1 \pmod{7}$ has $P(1) = 3, P(2) = 0, P(3) = 6 \pmod{7}$.
Send: $P(1) = 3, P(2) = 0, P(3) = 6$, 

Example.

Message: 3, 0, 6.

Reed Solomon Code: $P(x) = x^2 + x + 1 \pmod{7}$ has $P(1) = 3, P(2) = 0, P(3) = 6$ modulo 7.

Send: $P(1) = 3, P(2) = 0, P(3) = 6, P(4) = 0, P(5) = 3$. 
Example.

Message: 3, 0, 6.

Reed Solomon Code: \( P(x) = x^2 + x + 1 \) (mod 7) has \( P(1) = 3, P(2) = 0, P(3) = 6 \) modulo 7.

Send: \( P(1) = 3, P(2) = 0, P(3) = 6, P(4) = 0, P(5) = 3 \).

(Aside: Message in plain text!)
Example.

Message: 3, 0, 6.

Reed Solomon Code: \( P(x) = x^2 + x + 1 \) (mod 7) has 
\( P(1) = 3, P(2) = 0, P(3) = 6 \) modulo 7.

Send: \( P(1) = 3, P(2) = 0, P(3) = 6, P(4) = 0, P(5) = 3. \)

(Aside: Message in plain text!)

Receive \( R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3. \)
Example.

Message: 3, 0, 6.

Reed Solomon Code: $P(x) = x^2 + x + 1 \pmod{7}$ has $P(1) = 3, P(2) = 0, P(3) = 6$ modulo 7.

Send: $P(1) = 3, P(2) = 0, P(3) = 6, P(4) = 0, P(5) = 3$.

(Aside: Message in plain text!)

Receive $R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3$.

$P(i) = R(i)$ for $n + k = 3 + 1 = 4$ points.
Slow solution.

**Brute Force:**
For each subset of $n + k$ points
Brute Force:
For each subset of \( n + k \) points
   Fit degree \( n - 1 \) polynomial, \( Q(x) \), to \( n \) of them.
Slow solution.

**Brute Force:**
For each subset of $n+k$ points
- Fit degree $n-1$ polynomial, $Q(x)$, to $n$ of them.
- Check if consistent with $n+k$ of the total points.
**Brute Force:**
For each subset of $n + k$ points
   - Fit degree $n - 1$ polynomial, $Q(x)$, to $n$ of them.
   - Check if consistent with $n + k$ of the total points.
   - If yes, output $Q(x)$. 
Slow solution.

**Brute Force:**
For each subset of $n+k$ points
- Fit degree $n-1$ polynomial, $Q(x)$, to $n$ of them.
- Check if consistent with $n+k$ of the total points.
  - If yes, output $Q(x)$.

- For subset of $n+k$ pts where $R(i) = P(i)$, method will reconstruct $P(x)$!
Slow solution.

**Brute Force:**
For each subset of $n+k$ points
- Fit degree $n-1$ polynomial, $Q(x)$, to $n$ of them.
- Check if consistent with $n+k$ of the total points.
  If yes, output $Q(x)$.

- For subset of $n+k$ pts where $R(i) = P(i)$, method will reconstruct $P(x)$!
- For any subset of $n+k$ pts,
Slow solution.

**Brute Force:**
For each subset of \( n + k \) points
Fit degree \( n - 1 \) polynomial, \( Q(x) \), to \( n \) of them.
Check if consistent with \( n + k \) of the total points.
If yes, output \( Q(x) \).

- For subset of \( n + k \) pts where \( R(i) = P(i) \),
  method will reconstruct \( P(x) \)!

- For any subset of \( n + k \) pts,
  1. there is unique degree \( n - 1 \) polynomial \( Q(x) \) that fits \( n \) of them
**Slow solution.**

**Brute Force:**
For each subset of $n+k$ points
  Fit degree $n-1$ polynomial, $Q(x)$, to $n$ of them.
  Check if consistent with $n+k$ of the total points.
  If yes, output $Q(x)$.

- For subset of $n+k$ pts where $R(i) = P(i)$, method will reconstruct $P(x)$!
- For any subset of $n+k$ pts,
  1. there is unique degree $n-1$ polynomial $Q(x)$ that fits $n$ of them
  2. and where $Q(x)$ is consistent with $n+k$ points
Slow solution.

**Brute Force:**
For each subset of \( n + k \) points
   Fit degree \( n - 1 \) polynomial, \( Q(x) \), to \( n \) of them.
   Check if consistent with \( n + k \) of the total points.
   If yes, output \( Q(x) \).

- For subset of \( n + k \) pts where \( R(i) = P(i) \),
  method will reconstruct \( P(x) \)!

- For any subset of \( n + k \) pts,
  1. there is unique degree \( n - 1 \) polynomial \( Q(x) \) that fits \( n \) of them
  2. and where \( Q(x) \) is consistent with \( n + k \) points
      \( \implies P(x) = Q(x) \).
Slow solution.

**Brute Force:**
For each subset of $n+k$ points
- Fit degree $n-1$ polynomial, $Q(x)$, to $n$ of them.
- Check if consistent with $n+k$ of the total points.
  - If yes, output $Q(x)$.

- For subset of $n+k$ pts where $R(i) = P(i)$, method will reconstruct $P(x)$!

- For any subset of $n+k$ pts,
  1. there is unique degree $n-1$ polynomial $Q(x)$ that fits $n$ of them
  2. and where $Q(x)$ is consistent with $n+k$ points
     \[ \implies P(x) = Q(x). \]

Reconstructs $P(x)$ and only $P(x)$!!
Example.

Received $R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3$
Example.

Received \( R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3 \)
Find \( P(x) = p_2 x^2 + p_1 x + p_0 \) that contains \( n + k = 3 + 1 \) points.
Example.

Received $R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3$
Find $P(x) = p_2x^2 + p_1x + p_0$ that contains $n + k = 3 + 1$ points.

All equations..

$$p_2 + p_1 + p_0 \equiv 3 \pmod{7}$$
$$4p_2 + 2p_1 + p_0 \equiv 1 \pmod{7}$$
$$2p_2 + 3p_1 + p_0 \equiv 6 \pmod{7}$$
$$2p_2 + 4p_1 + p_0 \equiv 0 \pmod{7}$$
$$1p_2 + 5p_1 + p_0 \equiv 3 \pmod{7}$$
Example.

Received $R(1) = 3, \ R(2) = 1, \ R(3) = 6, \ R(4) = 0, \ R(5) = 3$

Find $P(x) = p_2x^2 + p_1x + p_0$ that contains $n + k = 3 + 1$ points.

All equations..

\[
\begin{align*}
  p_2 + p_1 + p_0 & \equiv 3 \pmod{7} \\
  4p_2 + 2p_1 + p_0 & \equiv 1 \pmod{7} \\
  2p_2 + 3p_1 + p_0 & \equiv 6 \pmod{7} \\
  2p_2 + 4p_1 + p_0 & \equiv 0 \pmod{7} \\
  p_2 + 5p_1 + p_0 & \equiv 3 \pmod{7}
\end{align*}
\]

Assume point 1 is wrong

No consistent solution!

Assume point 2 is wrong

Consistent solution!
Example.

Received $R(1) = 3$, $R(2) = 1$, $R(3) = 6$, $R(4) = 0$, $R(5) = 3$

Find $P(x) = p_2 x^2 + p_1 x + p_0$ that contains $n + k = 3 + 1$ points.

All equations.

\[
\begin{align*}
p_2 + p_1 + p_0 & \equiv 3 \pmod{7} \\
4p_2 + 2p_1 + p_0 & \equiv 1 \pmod{7} \\
2p_2 + 3p_1 + p_0 & \equiv 6 \pmod{7} \\
2p_2 + 4p_1 + p_0 & \equiv 0 \pmod{7} \\
1p_2 + 5p_1 + p_0 & \equiv 3 \pmod{7}
\end{align*}
\]

Assume point 1 is wrong and solve..
Example.

Received $R(1) = 3$, $R(2) = 1$, $R(3) = 6$, $R(4) = 0$, $R(5) = 3$

Find $P(x) = p_2x^2 + p_1x + p_0$ that contains $n+k = 3 + 1$ points.

All equations..

\[
\begin{align*}
p_2 + p_1 + p_0 & \equiv 3 \pmod{7} \\
4p_2 + 2p_1 + p_0 & \equiv 1 \pmod{7} \\
2p_2 + 3p_1 + p_0 & \equiv 6 \pmod{7} \\
2p_2 + 4p_1 + p_0 & \equiv 0 \pmod{7} \\
1p_2 + 5p_1 + p_0 & \equiv 3 \pmod{7}
\end{align*}
\]

Assume point 1 is wrong and solve...no consistent solution!
Example.

Received $R(1) = 3$, $R(2) = 1$, $R(3) = 6$, $R(4) = 0$, $R(5) = 3$
Find $P(x) = p_2x^2 + p_1x + p_0$ that contains $n + k = 3 + 1$ points.
All equations..

\[
\begin{align*}
p_2 + p_1 + p_0 & \equiv 3 \pmod{7} \\
4p_2 + 2p_1 + p_0 & \equiv 1 \pmod{7} \\
2p_2 + 3p_1 + p_0 & \equiv 6 \pmod{7} \\
2p_2 + 4p_1 + p_0 & \equiv 0 \pmod{7} \\
p_2 + 5p_1 + p_0 & \equiv 3 \pmod{7}
\end{align*}
\]

Assume point 1 is wrong and solve..no consistent solution!
Assume point 2 is wrong
Example.

Received $R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3$

Find $P(x) = p_2 x^2 + p_1 x + p_0$ that contains $n + k = 3 + 1$ points.

All equations..

\[
\begin{align*}
p_2 + p_1 + p_0 & \equiv 3 \pmod{7} \\
4p_2 + 2p_1 + p_0 & \equiv 1 \pmod{7} \\
2p_2 + 3p_1 + p_0 & \equiv 6 \pmod{7} \\
2p_2 + 4p_1 + p_0 & \equiv 0 \pmod{7} \\
1p_2 + 5p_1 + p_0 & \equiv 3 \pmod{7}
\end{align*}
\]

Assume point 1 is wrong and solve.. no consistent solution!
Assume point 2 is wrong and solve...
Example.

Received \( R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3 \)

Find \( P(x) = p_2 x^2 + p_1 x + p_0 \) that contains \( n + k = 3 + 1 \) points.

All equations..

\[
\begin{align*}
p_2 + p_1 + p_0 & \equiv 3 \pmod{7} \\
4p_2 + 2p_1 + p_0 & \equiv 1 \pmod{7} \\
2p_2 + 3p_1 + p_0 & \equiv 6 \pmod{7} \\
2p_2 + 4p_1 + p_0 & \equiv 0 \pmod{7} \\
p_2 + 5p_1 + p_0 & \equiv 3 \pmod{7} \\
\end{align*}
\]

Assume point 1 is wrong and solve..no consistent solution!
Assume point 2 is wrong and solve...consistent solution!
In general..

\[ P(x) = p_{n-1}x^{n-1} + \cdots p_0 \] and receive \( R(1), \ldots R(m = n + 2k) \).
In general,

\[ P(x) = p_{n-1}x^{n-1} + \cdots + p_0 \] and receive \( R(1), \ldots, R(m = n+2k) \).

\[ p_{n-1} + \cdots + p_0 \equiv R(1) \pmod{p} \]
In general,

\[ P(x) = p_{n-1}x^{n-1} + \cdots + p_0 \] and receive \( R(1), \ldots, R(m = n+2k) \).

\[ p_{n-1} + \cdots + p_0 \equiv R(1) \pmod{p} \]

\[ p_{n-1}2^{n-1} + \cdots + p_0 \equiv R(2) \pmod{p} \]
In general,

\[ P(x) = p_{n-1}x^{n-1} + \cdots + p_0 \text{ and receive } R(1), \ldots R(m = n + 2k). \]

\[
\begin{align*}
p_{n-1} + \cdots + p_0 & \equiv R(1) \pmod{p} \\
p_{n-1}2^{n-1} + \cdots + p_0 & \equiv R(2) \pmod{p} \\
& \quad \vdots \\
p_{n-1}i^{n-1} + \cdots + p_0 & \equiv R(i) \pmod{p} \\
& \quad \vdots \\
p_{n-1}(m)^{n-1} + \cdots + p_0 & \equiv R(m) \pmod{p}
\end{align*}
\]
In general,

\[ P(x) = p_{n-1}x^{n-1} + \cdots p_0 \] and receive \( R(1), \ldots R(m = n + 2k) \).

\[
\begin{align*}
p_{n-1} + \cdots p_0 & \equiv R(1) \pmod{p} \\
p_{n-1}2^{n-1} + \cdots p_0 & \equiv R(2) \pmod{p} \\
\vdots & \\
p_{n-1}i^{n-1} + \cdots p_0 & \equiv R(i) \pmod{p} \\
\vdots & \\
p_{n-1}(m)^{n-1} + \cdots p_0 & \equiv R(m) \pmod{p}
\end{align*}
\]

Error!!
In general,

\[ P(x) = p_{n-1}x^{n-1} + \cdots p_0 \] and receive \( R(1), \ldots, R(m = n + 2k) \).

\[
\begin{align*}
    p_{n-1} + \cdots p_0 & \equiv R(1) \pmod{p} \\
p_{n-1}2^{n-1} + \cdots p_0 & \equiv R(2) \pmod{p} \\
\vdots & \\
p_{n-1}i^{n-1} + \cdots p_0 & \equiv R(i) \pmod{p} \\
\vdots & \\
p_{n-1}(m)^{n-1} + \cdots p_0 & \equiv R(m) \pmod{p}
\end{align*}
\]

Error!! .... Where???
In general..

\[ P(x) = p_{n-1}x^{n-1} + \cdots + p_0 \] and receive \( R(1), \ldots, R(m = n + 2k) \).

\[
\begin{align*}
p_{n-1} + \cdots + p_0 & \equiv R(1) \pmod{p} \\
p_{n-1}2^{n-1} + \cdots + p_0 & \equiv R(2) \pmod{p} \\
\vdots & \\
p_{n-1}i^{n-1} + \cdots + p_0 & \equiv R(i) \pmod{p} \\
\vdots & \\
p_{n-1}(m)^{n-1} + \cdots + p_0 & \equiv R(m) \pmod{p}
\end{align*}
\]

Error!! .... Where???
Could be anywhere!!!
In general..

\[ P(x) = p_{n-1}x^{n-1} + \cdots p_0 \] and receive \( R(1), \ldots R(m = n + 2k) \).

\[
\begin{align*}
  p_{n-1} + \cdots p_0 & \equiv R(1) \pmod{p} \\
  p_{n-1}2^{n-1} + \cdots p_0 & \equiv R(2) \pmod{p} \\
  \vdots \\
  p_{n-1}i^{n-1} + \cdots p_0 & \equiv R(i) \pmod{p} \\
  \vdots \\
  p_{n-1}(m)^{n-1} + \cdots p_0 & \equiv R(m) \pmod{p}
\end{align*}
\]

Error!! .... Where???
Could be anywhere!!! ...so try everywhere.
In general,

\[ P(x) = p_{n-1}x^{n-1} + \cdots + p_0 \] and receive \( R(1), \ldots, R(m = n + 2k) \).

\[
\begin{align*}
p_{n-1} + \cdots + p_0 & \equiv R(1) \pmod{p} \\
p_{n-1}2^{n-1} + \cdots + p_0 & \equiv R(2) \pmod{p} \\
\vdots & \\
p_{n-1}i^{n-1} + \cdots + p_0 & \equiv R(i) \pmod{p} \\
\vdots & \\
p_{n-1}(m)^{n-1} + \cdots + p_0 & \equiv R(m) \pmod{p}
\end{align*}
\]

Error!! .... Where???
Could be anywhere!!! ...so try everywhere.

**Runtime:** \( \binom{n+2k}{k} \) possibilities.
In general,

\[ P(x) = p_{n-1}x^{n-1} + \cdots p_0 \] and receive \( R(1), \ldots R(m = n+2k) \).

\[
\begin{align*}
p_{n-1} + \cdots p_0 &\equiv R(1) \pmod{p} \\
p_{n-1}2^{n-1} + \cdots p_0 &\equiv R(2) \pmod{p} \\
&\vdots \\
p_{n-1}i^{n-1} + \cdots p_0 &\equiv R(i) \pmod{p} \\
&\vdots \\
p_{n-1}(m)^{n-1} + \cdots p_0 &\equiv R(m) \pmod{p}
\end{align*}
\]

Error!! .... Where???
Could be anywhere!!! ...so try everywhere.

**Runtime:** \( \binom{n+2k}{k} \) possibilities.

Something like \( (n/k)^k \) ...Exponential in \( k \).
In general,

\[ P(x) = p_{n-1}x^{n-1} + \cdots p_0 \] and receive \( R(1), \ldots, R(m = n + 2k) \).

\[
\begin{align*}
p_{n-1} + \cdots + p_0 & \equiv R(1) \pmod{p} \\
p_{n-1}2^{n-1} + \cdots + p_0 & \equiv R(2) \pmod{p} \\
& \quad \vdots \\
p_{n-1}i^{n-1} + \cdots + p_0 & \equiv R(i) \pmod{p} \\
& \quad \vdots \\
p_{n-1}(m)^{n-1} + \cdots + p_0 & \equiv R(m) \pmod{p}
\end{align*}
\]

Error!! .... Where???
Could be anywhere!!! ...so try everywhere.

**Runtime:** \( \binom{n+2k}{k} \) possibilities.

Something like \((n/k)^k\) ...Exponential in \(k\)!

How do we find where the bad packets are efficiently?!?!?!
Ditty...
Ditty...

Where oh where
Ditty...

Where oh where can my bad packets be ...
Ditty...

Where oh where can my bad packets be ...
Ditty...

Where oh where can my bad packets be ... 
Today.
Where oh where can my bad packets be?

\[(p_{n-1} + \cdots + p_0) \equiv R(1) \pmod{p}\]
Where oh where can my bad packets be?

\[(p_{n-1} + \cdots p_0) \equiv R(1) \pmod{p}\]
\[(p_{n-1}2^{n-1} + \cdots p_0) \equiv R(2) \pmod{p}\]
\[\vdots\]
\[(p_{n-1}(m)^{n-1} + \cdots p_0) \equiv R(n+2k) \pmod{p}\]
Where oh where can my bad packets be?

\[(p_{n-1} + \cdots + p_0) \equiv R(1) \pmod{p}\]

\[(p_{n-1}2^{n-1} + \cdots + p_0) \equiv R(2) \pmod{p}\]

\[\vdots\]

\[(p_{n-1}(m)^{n-1} + \cdots + p_0) \equiv R(n + 2k) \pmod{p}\]

Idea: Multiply equation \(i\) by 0 if and only if \(P(i) \neq R(i)\).
Where oh where can my bad packets be?

\[(p_{n-1} + \cdots + p_0) \equiv R(1) \quad (\text{mod } p)\]

\[0 \times (p_{n-1}2^{n-1} + \cdots + p_0) \equiv R(2) \quad (\text{mod } p)\]

\[\vdots\]

\[(p_{n-1}(m)^{n-1} + \cdots + p_0) \equiv R(n + 2k) \quad (\text{mod } p)\]

**Idea:** Multiply equation \(i\) by 0 if and only if \(P(i) \neq R(i)\).
All equations satisfied!!!!!
Where oh where can my bad packets be?

\[
(p_{n-1} + \cdots p_0) \equiv R(1) \pmod{p} \\
(p_{n-1} 2^{n-1} + \cdots p_0) \equiv R(2) \pmod{p} \\
\vdots \\
(p_{n-1} (m)^{n-1} + \cdots p_0) \equiv R(n+2k) \pmod{p}
\]

**Idea:** Multiply equation \(i\) by 0 if and only if \(P(i) \neq R(i)\). All equations satisfied!!!!!

But which equations should we multiply by 0?
Where oh where can my bad packets be?

\[(\rho_{n-1} + \cdots \rho_0) \equiv R(1) \pmod{p}\]
\[(\rho_{n-1}2^{n-1} + \cdots \rho_0) \equiv R(2) \pmod{p}\]
\[
\vdots
\]
\[(\rho_{n-1}(m)^{n-1} + \cdots \rho_0) \equiv R(n+2k) \pmod{p}\]

**Idea:** Multiply equation \(i\) by 0 if and only if \(P(i) \neq R(i)\).
All equations satisfied!!!!!!

But which equations should we multiply by 0? Where oh where...
Where oh where can my **bad packets** be?

\[
(p_{n-1} + \cdots p_0) \equiv R(1) \pmod{p}
\]
\[
(p_{n-1} 2^{n-1} + \cdots p_0) \equiv R(2) \pmod{p}
\]
\[\vdots\]
\[
(p_{n-1} (m)^{n-1} + \cdots p_0) \equiv R(n+2k) \pmod{p}
\]

*Idea:* Multiply equation \(i\) by 0 if and only if \(P(i) \neq R(i)\). All equations satisfied!!!!!

But which equations should we multiply by 0? Where oh where...???
Where oh where can my bad packets be?

\[(p_{n-1} + \cdots + p_0) \equiv R(1) \pmod{p} \]
\[(p_{n-1}2^{n-1} + \cdots + p_0) \equiv R(2) \pmod{p} \]
\[\vdots\]
\[(p_{n-1}(m)^{n-1} + \cdots + p_0) \equiv R(n+2k) \pmod{p} \]

**Idea:** Multiply equation $i$ by 0 if and only if $P(i) \neq R(i)$. All equations satisfied!!!!!

But which equations should we multiply by 0? Where oh where...??

We will use a polynomial!!!
Where oh where can my **bad packets** be?

\[
(p_{n-1} + \cdots p_0) \equiv R(1) \pmod{p} \\
(p_{n-1}2^{n-1} + \cdots p_0) \equiv R(2) \pmod{p} \\
\vdots \\
(p_{n-1}(m)^{n-1} + \cdots p_0) \equiv R(n + 2k) \pmod{p}
\]

**Idea:** Multiply equation \( i \) by 0 if and only if \( P(i) \neq R(i) \).

All equations satisfied!!!!!

But which equations should we multiply by 0? Where oh where...??

We will use a polynomial!!! That we don’t know.
Where oh where can my **bad packets** be?

\[
\begin{align*}
(p_{n-1} + \cdots + p_0) & \equiv R(1) \pmod{p} \\
(p_{n-1}2^{n-1} + \cdots + p_0) & \equiv R(2) \pmod{p} \\
& \vdots \\
(p_{n-1}(m)^{n-1} + \cdots + p_0) & \equiv R(n+2k) \pmod{p}
\end{align*}
\]

**Idea:** Multiply equation \( i \) by 0 if and only if \( P(i) \neq R(i) \).
All equations satisfied!!!!!

But which equations should we multiply by 0? Where oh where...??

We will use a polynomial!!! That we don’t know. But can find!
Where oh where can my bad packets be?

\[(p_{n-1} + \cdots p_0) \equiv R(1) \pmod{p}\]
\[(p_{n-1}2^{n-1} + \cdots p_0) \equiv R(2) \pmod{p}\]
\[\vdots\]
\[(p_{n-1}(m)^{n-1} + \cdots p_0) \equiv R(n + 2k) \pmod{p}\]

Idea: Multiply equation \(i\) by 0 if and only if \(P(i) \neq R(i)\).
All equations satisfied!!!!!

But which equations should we multiply by 0? Where oh where...??

We will use a polynomial!!! That we don’t know. But can find!

Errors at points \(e_1, \ldots, e_k\). (In diagram above, \(e_1 = 2\).)
Where oh where can my bad packets be?

\[(p_{n-1} + \cdots p_0) \equiv R(1) \pmod{p}\]
\[(p_{n-1}2^{n-1} + \cdots p_0) \equiv R(2) \pmod{p}\]
\[\vdots\]
\[(p_{n-1}(m)^{n-1} + \cdots p_0) \equiv R(n+2k) \pmod{p}\]

**Idea:** Multiply equation \(i\) by 0 if and only if \(P(i) \neq R(i)\). All equations satisfied!!!!!

But which equations should we multiply by 0? Where oh where...??

We will use a polynomial!!! That we don’t know. But can find!

Errors at points \(e_1, \ldots, e_k\). (In diagram above, \(e_1 = 2\).)

**Error locator polynomial:** \(E(x) = (x - e_1)\)
Where oh where can my bad packets be?

\[
(p_{n-1} + \cdots p_0) \equiv R(1) \pmod{p} \\
(p_{n-1}2^{n-1} + \cdots p_0) \equiv R(2) \pmod{p} \\
\vdots \\
(p_{n-1}(m)^{n-1} + \cdots p_0) \equiv R(n+2k) \pmod{p}
\]

**Idea:** Multiply equation \(i\) by 0 if and only if \(P(i) \neq R(i)\). All equations satisfied!!!!!

But which equations should we multiply by 0? Where oh where...??

We will use a polynomial!!! That we don’t know. But can find!

Errors at points \(e_1, \ldots, e_k\). (In diagram above, \(e_1 = 2\).)

**Error locator polynomial:** \(E(x) = (x - e_1)(x - e_2)\)
Where oh where can my bad packets be?

\[
(p_{n-1} + \cdots + p_0) \equiv R(1) \pmod{p} \\
(p_{n-1}2^{n-1} + \cdots + p_0) \equiv R(2) \pmod{p} \\
\vdots \\
(p_{n-1}(m)^{n-1} + \cdots + p_0) \equiv R(n+2k) \pmod{p}
\]

**Idea:** Multiply equation \(i\) by 0 if and only if \(P(i) \neq R(i)\).

All equations satisfied!!!!!!

But which equations should we multiply by 0? Where oh where...??

We will use a polynomial!!! That we don’t know. But can find!

Errors at points \(e_1, \ldots, e_k\). (In diagram above, \(e_1 = 2\).

**Error locator polynomial:** \(E(x) = (x - e_1)(x - e_2)\ldots\)
Where oh where can my **bad packets** be?

\[
(p_{n-1} + \cdots + p_0) \equiv R(1) \pmod{p} \\
(p_{n-1} 2^{n-1} + \cdots + p_0) \equiv R(2) \pmod{p} \\
\vdots \\
(p_{n-1} (m)^{n-1} + \cdots + p_0) \equiv R(n + 2k) \pmod{p}
\]

**Idea:** Multiply equation \(i\) by 0 if and only if \(P(i) \neq R(i)\).

All equations satisfied!!!!!

But which equations should we multiply by 0? **Where oh where...??**

**We will use a polynomial!!!** That we don’t know. But can find!

Errors at points \(e_1, \ldots, e_k\). (In diagram above, \(e_1 = 2\).)

**Error locator polynomial:** \(E(x) = (x - e_1)(x - e_2) \cdots (x - e_k)\).
Where oh where can my bad packets be?

\[(p_{n-1} + \cdots + p_0) \equiv R(1) \pmod{p}\]
\[(p_{n-1} 2^{n-1} + \cdots + p_0) \equiv R(2) \pmod{p}\]
\[\vdots\]
\[(p_{n-1} (m)^{n-1} + \cdots + p_0) \equiv R(n+2k) \pmod{p}\]

**Idea:** Multiply equation \(i\) by 0 if and only if \(P(i) \neq R(i)\). All equations satisfied!!!!

But which equations should we multiply by 0? Where oh where...??

**We will use a polynomial!!!** That we don’t know. But can find!

Errors at points \(e_1, \ldots, e_k\). (In diagram above, \(e_1 = 2\).)

**Error locator polynomial:** \(E(x) = (x - e_1)(x - e_2)\ldots(x - e_k)\).

\(E(i) = 0\) if and only if \(e_j = i\) for some \(j\)
Where oh where can my bad packets be?

\[
E(1)(\rho_{n-1} + \cdots \rho_0) \equiv R(1)E(1) \pmod{p} \\
E(2)(\rho_{n-1}2^{n-1} + \cdots \rho_0) \equiv R(2)E(2) \pmod{p} \\
\vdots \\
E(m)(\rho_{n-1}(m)^{n-1} + \cdots \rho_0) \equiv R(n+2k)E(m) \pmod{p}
\]

**Idea:** Multiply equation \(i\) by 0 if and only if \(P(i) \neq R(i)\). All equations satisfied!!!!!!

But which equations should we multiply by 0? Where oh where...??

We will use a polynomial!!! That we don’t know. But can find!

Errors at points \(e_1, \ldots, e_k\). (In diagram above, \(e_1 = 2\).)

**Error locator polynomial:** \(E(x) = (x - e_1)(x - e_2)\cdots(x - e_k)\).

\(E(i) = 0\) if and only if \(e_j = i\) for some \(j\)

Multiply equations by \(E(\cdot)\).
Where oh where can my bad packets be?

\[
E(1)(\rho_{n-1} + \cdots + p_0) \equiv R(1)E(1) \pmod p
\]
\[
E(2)(\rho_{n-1}2^{n-1} + \cdots + p_0) \equiv R(2)E(2) \pmod p
\]
\[
\vdots
\]
\[
E(m)(\rho_{n-1}(m)^{n-1} + \cdots + p_0) \equiv R(n+2k)E(m) \pmod p
\]

**Idea:** Multiply equation \( i \) by 0 if and only if \( P(i) \neq R(i) \).

All equations satisfied!!!!!

But which equations should we multiply by 0? Where oh where...??

We will use a polynomial!!! That we don’t know. But can find!

Errors at points \( e_1, \ldots, e_k \). (In diagram above, \( e_1 = 2 \).)

**Error locator polynomial:** \( E(x) = (x - e_1)(x - e_2) \cdots (x - e_k) \).

\( E(i) = 0 \) if and only if \( e_j = i \) for some \( j \)

Multiply equations by \( E(\cdot) \). (Above \( E(x) = (x-2) \).)
Where oh where can my **bad packets** be?

\[
E(1)(p_{n-1} + \cdots p_0) \equiv R(1)E(1) \pmod{p} \\
E(2)(p_{n-1}2^{n-1} + \cdots p_0) \equiv R(2)E(2) \pmod{p} \\
\vdots \\
E(m)(p_{n-1}(m)^{n-1} + \cdots p_0) \equiv R(n+2k)E(m) \pmod{p}
\]

**Idea:** Multiply equation \(i\) by 0 if and only if \(P(i) \neq R(i)\).

All equations satisfied!!!!!

But which equations should we multiply by 0? Where oh where...??

**We will use a polynomial!!!** That we don’t know. But can find!

Errors at points \(e_1, \ldots, e_k\). (In diagram above, \(e_1 = 2\).)

**Error locator polynomial:** 
\[
E(x) = (x - e_1)(x - e_2) \cdots (x - e_k).
\]

\(E(i) = 0\) if and only if \(e_j = i\) for some \(j\)

Multiply equations by \(E(\cdot)\). (Above \(E(x) = (x-2)\).)

All equations satisfied!!
Example.

Received $R(1) = 3$, $R(2) = 1$, $R(3) = 6$, $R(4) = 0$, $R(5) = 3$
Example.

Received $R(1) = 3$, $R(2) = 1$, $R(3) = 6$, $R(4) = 0$, $R(5) = 3$

Find $P(x) = p_2 x^2 + p_1 x + p_0$ that contains $n + k = 3 + 1$ points.
Example.

Received $R(1) = 3$, $R(2) = 1$, $R(3) = 6$, $R(4) = 0$, $R(5) = 3$

Find $P(x) = p_2 x^2 + p_1 x + p_0$ that contains $n + k = 3 + 1$ points.

Plugin points...

\[
\begin{align*}
(p_2 + p_1 + p_0) & \equiv (3) \pmod{7} \\
(4p_2 + 2p_1 + p_0) & \equiv (1) \pmod{7} \\
(2p_2 + 3p_1 + p_0) & \equiv (6) \pmod{7} \\
(2p_2 + 4p_1 + p_0) & \equiv (0) \pmod{7} \\
(4p_2 + 5p_1 + p_0) & \equiv (3) \pmod{7}
\end{align*}
\]
Example.

Received $R(1) = 3$, $R(2) = 1$, $R(3) = 6$, $R(4) = 0$, $R(5) = 3$

Find $P(x) = p_2 x^2 + p_1 x + p_0$ that contains $n + k = 3 + 1$ points.

Plugin points...

\[
(p_2 + p_1 + p_0) \equiv (3) \pmod{7}
\]
\[
(4p_2 + 2p_1 + p_0) \equiv (1) \pmod{7}
\]
\[
(2p_2 + 3p_1 + p_0) \equiv (6) \pmod{7}
\]
\[
(2p_2 + 4p_1 + p_0) \equiv (0) \pmod{7}
\]
\[
(4p_2 + 5p_1 + p_0) \equiv (3) \pmod{7}
\]

Error locator polynomial: $(x - 2)$. 
Example.

Received $R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3$

Find $P(x) = p_2 x^2 + p_1 x + p_0$ that contains $n + k = 3 + 1$ points.

Plugin points...

\[
(1 - 2)(p_2 + p_1 + p_0) \equiv (3)(1 - 2) \pmod{7}
\]
\[
(2 - 2)(4p_2 + 2p_1 + p_0) \equiv (1)(2 - 2) \pmod{7}
\]
\[
(3 - 2)(2p_2 + 3p_1 + p_0) \equiv (6)(3 - 2) \pmod{7}
\]
\[
(4 - 2)(2p_2 + 4p_1 + p_0) \equiv (0)(4 - 2) \pmod{7}
\]
\[
(5 - 2)(4p_2 + 5p_1 + p_0) \equiv (3)(5 - 2) \pmod{7}
\]

Error locator polynomial: $(x - 2)$.

Multiply equation $i$ by $(i - 2)$.
Example.

Received $R(1) = 3$, $R(2) = 1$, $R(3) = 6$, $R(4) = 0$, $R(5) = 3$

Find $P(x) = p_2 x^2 + p_1 x + p_0$ that contains $n + k = 3 + 1$ points.

Plugin points...

\[
\begin{align*}
(1 - 2)(p_2 + p_1 + p_0) & \equiv (3)(1 - 2) \pmod{7} \\
(2 - 2)(4p_2 + 2p_1 + p_0) & \equiv (1)(2 - 2) \pmod{7} \\
(3 - 2)(2p_2 + 3p_1 + p_0) & \equiv (6)(3 - 2) \pmod{7} \\
(4 - 2)(2p_2 + 4p_1 + p_0) & \equiv (0)(4 - 2) \pmod{7} \\
(5 - 2)(4p_2 + 5p_1 + p_0) & \equiv (3)(5 - 2) \pmod{7}
\end{align*}
\]

Error locator polynomial: $(x - 2)$.

Multiply equation $i$ by $(i - 2)$. All equations satisfied!
Example.

Received $R(1) = 3$, $R(2) = 1$, $R(3) = 6$, $R(4) = 0$, $R(5) = 3$

Find $P(x) = p_2 x^2 + p_1 x + p_0$ that contains $n + k = 3 + 1$ points.

Plugin points...

\[
(1 - 2)(p_2 + p_1 + p_0) \equiv (3)(1 - 2) \pmod{7}
\]
\[
(2 - 2)(4p_2 + 2p_1 + p_0) \equiv (1)(2 - 2) \pmod{7}
\]
\[
(3 - 2)(2p_2 + 3p_1 + p_0) \equiv (6)(3 - 2) \pmod{7}
\]
\[
(4 - 2)(2p_2 + 4p_1 + p_0) \equiv (0)(4 - 2) \pmod{7}
\]
\[
(5 - 2)(4p_2 + 5p_1 + p_0) \equiv (3)(5 - 2) \pmod{7}
\]

Error locator polynomial: $(x - 2)$.

Multiply equation $i$ by $(i - 2)$. All equations satisfied!

But don’t know error locator polynomial!
Example.

Received $R(1) = 3$, $R(2) = 1$, $R(3) = 6$, $R(4) = 0$, $R(5) = 3$

Find $P(x) = p_2x^2 + p_1x + p_0$ that contains $n + k = 3 + 1$ points.

Plugin points...

\[
\begin{align*}
(1 - 2)(p_2 + p_1 + p_0) & \equiv (3)(1 - 2) \pmod{7} \\
(2 - 2)(4p_2 + 2p_1 + p_0) & \equiv (1)(2 - 2) \pmod{7} \\
(3 - 2)(2p_2 + 3p_1 + p_0) & \equiv (6)(3 - 2) \pmod{7} \\
(4 - 2)(2p_2 + 4p_1 + p_0) & \equiv (0)(4 - 2) \pmod{7} \\
(5 - 2)(4p_2 + 5p_1 + p_0) & \equiv (3)(5 - 2) \pmod{7}
\end{align*}
\]

Error locator polynomial: $(x - 2)$.

Multiply equation $i$ by $(i - 2)$. All equations satisfied!

But don’t know error locator polynomial! Do know form:
Example.

Received $R(1) = 3$, $R(2) = 1$, $R(3) = 6$, $R(4) = 0$, $R(5) = 3$

Find $P(x) = p_2x^2 + p_1x + p_0$ that contains $n + k = 3 + 1$ points.

Plugin points...

$$(1 - 2)(p_2 + p_1 + p_0) \equiv (3)(1 - 2) \pmod{7}$$

$$(2 - 2)(4p_2 + 2p_1 + p_0) \equiv (1)(2 - 2) \pmod{7}$$

$$(3 - 2)(2p_2 + 3p_1 + p_0) \equiv (6)(3 - 2) \pmod{7}$$

$$(4 - 2)(2p_2 + 4p_1 + p_0) \equiv (0)(4 - 2) \pmod{7}$$

$$(5 - 2)(4p_2 + 5p_1 + p_0) \equiv (3)(5 - 2) \pmod{7}$$

Error locator polynomial: $(x - 2)$.

Multiply equation $i$ by $(i - 2)$. All equations satisfied!

But don’t know error locator polynomial! Do know form: $(x - e)$. 
Example.

Received $R(1) = 3$, $R(2) = 1$, $R(3) = 6$, $R(4) = 0$, $R(5) = 3$

Find $P(x) = p_2 x^2 + p_1 x + p_0$ that contains $n + k = 3 + 1$ points.

Plugin points...

\[
\begin{align*}
(1 - e)(p_2 + p_1 + p_0) & \equiv (3)(1 - e) \pmod{7} \\
(2 - e)(4p_2 + 2p_1 + p_0) & \equiv (1)(2 - e) \pmod{7} \\
(3 - e)(2p_2 + 3p_1 + p_0) & \equiv (3)(3 - e) \pmod{7} \\
(4 - e)(2p_2 + 4p_1 + p_0) & \equiv (0)(4 - e) \pmod{7} \\
(5 - e)(4p_2 + 5p_1 + p_0) & \equiv (3)(5 - e) \pmod{7}
\end{align*}
\]

Error locator polynomial: $(x - 2)$.

Multiply equation $i$ by $(i - 2)$. All equations satisfied!

But don’t know error locator polynomial! Do know form: $(x - e)$. 
Example.

Received $R(1) = 3$, $R(2) = 1$, $R(3) = 6$, $R(4) = 0$, $R(5) = 3$

Find $P(x) = p_2 x^2 + p_1 x + p_0$ that contains $n + k = 3 + 1$ points.

Plugin points...

\[
\begin{align*}
(1 - e)(p_2 + p_1 + p_0) & \equiv (3)(1 - e) \pmod{7} \\
(2 - e)(4p_2 + 2p_1 + p_0) & \equiv (1)(2 - e) \pmod{7} \\
(3 - e)(2p_2 + 3p_1 + p_0) & \equiv (3)(3 - e) \pmod{7} \\
(4 - e)(2p_2 + 4p_1 + p_0) & \equiv (0)(4 - e) \pmod{7} \\
(5 - e)(4p_2 + 5p_1 + p_0) & \equiv (3)(5 - e) \pmod{7}
\end{align*}
\]

Error locator polynomial: $(x - 2)$.

Multiply equation $i$ by $(i - 2)$. All equations satisfied!

But don’t know error locator polynomial! Do know form: $(x - e)$.

4 unknowns ($p_0, p_1, p_2$ and $e$),
Example.

Received $R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3$

Find $P(x) = p_2 x^2 + p_1 x + p_0$ that contains $n + k = 3 + 1$ points.

Plugin points...

\[ (1 - e)(p_2 + p_1 + p_0) \equiv (3)(1 - e) \pmod{7} \]
\[ (2 - e)(4p_2 + 2p_1 + p_0) \equiv (1)(2 - e) \pmod{7} \]
\[ (3 - e)(2p_2 + 3p_1 + p_0) \equiv (3)(3 - e) \pmod{7} \]
\[ (4 - e)(2p_2 + 4p_1 + p_0) \equiv (0)(4 - e) \pmod{7} \]
\[ (5 - e)(4p_2 + 5p_1 + p_0) \equiv (3)(5 - e) \pmod{7} \]

Error locator polynomial: $(x - 2)$.

Multiply equation $i$ by $(i - 2)$. All equations satisfied!

But don’t know error locator polynomial! Do know form: $(x - e)$.

4 unknowns $(p_0, p_1, p_2$ and $e)$, 5 nonlinear equations.
..turn their heads each day,

\[
(p_{n-1} + \cdots p_0) \equiv R(1) \pmod{p}
\]
\[
(p_{n-1}i^{n-1} + \cdots p_0) \equiv R(i) \pmod{p}
\]
\[
(p_{n-1}(n+2k)^{n-1} + \cdots p_0) \equiv R(m) \pmod{p}
\]
..turn their heads each day,

\[ E(1)(p_{n-1} + \cdots p_0) \equiv R(1)E(1) \pmod{p} \]

\[ \vdots \]

\[ E(i)(p_{n-1}i^{n-1} + \cdots p_0) \equiv R(i)E(i) \pmod{p} \]

\[ \vdots \]

\[ E(m)(p_{n-1}(n+2k)^{n-1} + \cdots p_0) \equiv R(m)E(m) \pmod{p} \]

...so satisfied, I’m on my way.
..turn their heads each day,

\[
E(1)(p_{n-1} + \cdots p_0) \equiv R(1)E(1) \pmod{p} \\
\vdots \\
E(i)(p_{n-1}i^{n-1} + \cdots p_0) \equiv R(i)E(i) \pmod{p} \\
\vdots \\
E(m)(p_{n-1}(n+2k)^{n-1} + \cdots p_0) \equiv R(m)E(m) \pmod{p}
\]

...so satisfied, I’m on my way.

\[m = n + 2k \] satisfied equations,
..turn their heads each day,

\[
E(1)(\rho_{n-1} + \cdots + \rho_0) \equiv R(1)E(1) \pmod{p} \\
\vdots \\
E(i)(\rho_{n-1}i^{n-1} + \cdots + \rho_0) \equiv R(i)E(i) \pmod{p} \\
\vdots \\
E(m)(\rho_{n-1}(n+2k)^{n-1} + \cdots + \rho_0) \equiv R(m)E(m) \pmod{p}
\]

...so satisfied, I’m on my way.

\[m = n + 2k\] satisfied equations, \(n + k\) unknowns.
..turn their heads each day,

\[
E(1)(\rho_{n-1} + \cdots + \rho_0) \equiv R(1)E(1) \pmod{p}
\]

\[
\vdots
\]

\[
E(i)(\rho_{n-1}i^{n-1} + \cdots + \rho_0) \equiv R(i)E(i) \pmod{p}
\]

\[
\vdots
\]

\[
E(m)(\rho_{n-1}(n+2k)^{n-1} + \cdots + \rho_0) \equiv R(m)E(m) \pmod{p}
\]

...so satisfied, I'm on my way.

\[m = n + 2k\] satisfied equations, \(n + k\) unknowns. But nonlinear!
..turn their heads each day,

\[
E(1)(p_{n-1} + \cdots p_0) \equiv R(1)E(1) \pmod{p} \\
\vdots \\
E(i)(p_{n-1} i^{n-1} + \cdots p_0) \equiv R(i)E(i) \pmod{p} \\
\vdots \\
E(m)(p_{n-1}(n+2k)^{n-1} + \cdots p_0) \equiv R(m)E(m) \pmod{p}
\]

...so satisfied, I’m on my way.

\( m = n + 2k \) satisfied equations, \( n + k \) unknowns. But nonlinear!

Let \( Q(x) = E(x)P(x) = a_{n+k-1}x^{n+k-1} + \cdots a_0 \).
..turn their heads each day,

\[ E(1)(p_{n-1} + \cdots p_0) \equiv R(1)E(1) \pmod{p} \]
\[ \vdots \]
\[ E(i)(p_{n-1}i^{n-1} + \cdots p_0) \equiv R(i)E(i) \pmod{p} \]
\[ \vdots \]
\[ E(m)(p_{n-1}(n+2k)^{n-1} + \cdots p_0) \equiv R(m)E(m) \pmod{p} \]

...so satisfied, I’m on my way.

\[ m = n + 2k \text{ satisfied equations, } n + k \text{ unknowns. But nonlinear!} \]

Let \( Q(x) = E(x)P(x) = a_{n+k-1}x^{n+k-1} + \cdots a_0. \)

Equations:

\[ Q(i) = R(i)E(i). \]
..turn their heads each day,

\[
E(1)(p_{n-1} + \cdots p_0) \equiv R(1)E(1) \pmod{p} \\
\vdots \\
E(i)(p_{n-1}i^{n-1} + \cdots p_0) \equiv R(i)E(i) \pmod{p} \\
\vdots \\
E(m)(p_{n-1}(n+2k)^{n-1} + \cdots p_0) \equiv R(m)E(m) \pmod{p}
\]

...so satisfied, I’m on my way.

\[m = n + 2k\] satisfied equations, \(n + k\) unknowns. But nonlinear!

Let \(Q(x) = E(x)P(x) = a_{n+k-1}x^{n+k-1} + \cdots a_0\).

Equations:

\[Q(i) = R(i)E(i).\]
..turn their heads each day,

\[ E(1)(p_{n-1} + \cdots + p_0) \equiv R(1)E(1) \pmod{p} \]
\[ \vdots \]
\[ E(i)(p_{n-1}i^{n-1} + \cdots + p_0) \equiv R(i)E(i) \pmod{p} \]
\[ \vdots \]
\[ E(m)(p_{n-1}(n+2k)^{n-1} + \cdots + p_0) \equiv R(m)E(m) \pmod{p} \]

...so satisfied, I’m on my way.

\[ m = n + 2k \text{ satisfied equations, } n + k \text{ unknowns. But nonlinear!} \]

Let \( Q(x) = E(x)P(x) = a_{n+k-1}x^{n+k-1} + \cdots a_0. \)

Equations:

\[ Q(i) = R(i)E(i). \]
..turn their heads each day,

\[
E(1)(p_{n-1} + \cdots + p_0) \equiv R(1)E(1) \pmod{p}
\]

\[
\vdots
\]

\[
E(i)(p_{n-1}i^{n-1} + \cdots + p_0) \equiv R(i)E(i) \pmod{p}
\]

\[
\vdots
\]

\[
E(m)(p_{n-1}(n+2k)^{n-1} + \cdots + p_0) \equiv R(m)E(m) \pmod{p}
\]

...so satisfied, I’m on my way.

\(m = n + 2k\) satisfied equations, \(n + k\) unknowns. But nonlinear!

Let \(Q(x) = E(x)P(x) = a_{n+k-1}x^{n+k-1} + \cdots + a_0.\)

Equations:

\[
Q(i) = R(i)E(i).
\]

and linear in \(a_i\) and coefficients of \(E(x)\)!
Finding $Q(x)$ and $E(x)$?

$E(x)$ has degree $k$...

$E(x) = x^k + b_{k-1}x^{k-1} + \cdots + b_0$.

$Q(x) = P(x)E(x)$ has degree $n + k - 1$ ...

$Q(x) = a_{n+k-1}x^{n+k-1} + a_{n+k-2}x^{n+k-2} + \cdots + a_0$. 
Finding $Q(x)$ and $E(x)$?

- $E(x)$ has degree $k$
Finding $Q(x)$ and $E(x)$?

- $E(x)$ has degree $k$ ...

\[ E(x) = x^k + b_{k-1}x^{k-1} \cdots b_0. \]
Finding $Q(x)$ and $E(x)$?

- $E(x)$ has degree $k$ ...

$$E(x) = x^k + b_{k-1}x^{k-1} \cdots b_0.$$  

- $Q(x) = P(x)E(x)$ has degree $n + k - 1$
Finding $Q(x)$ and $E(x)$?

- $E(x)$ has degree $k$ ...

$$E(x) = x^k + b_{k-1}x^{k-1} \cdots b_0.$$ 

- $Q(x) = P(x)E(x)$ has degree $n + k - 1$ ...

$$Q(x) = a_{n+k-1}x^{n+k-1} + a_{n+k-2}x^{n+k-2} + \cdots a_0$$
Solving for $Q(x)$ and $E(x)$...

For all points $1, \ldots, i, n+2k$,

$$Q(i) = R(i)E(i) \pmod{p}$$
Solving for $Q(x)$ and $E(x)$...

For all points $1, \ldots, i, n + 2k$,

$$Q(i) = R(i)E(i) \pmod{p}$$

Gives $n + 2k$ linear equations.
Solving for $Q(x)$ and $E(x)$...

For all points $1, \ldots, i, n+2k$,

$$Q(i) = R(i)E(i) \pmod{p}$$

Gives $n+2k$ linear equations.

$$a_{n+k-1} + \cdots + a_0 \equiv R(1)(1 + b_{k-1} \cdots b_0) \pmod{p}$$
Solving for $Q(x)$ and $E(x)$...

For all points $1, \ldots, i, n+2k$,

$$Q(i) = R(i)E(i) \pmod{p}$$

Gives $n+2k$ linear equations.

$$a_{n+k-1} + \ldots a_0 \equiv R(1)(1 + b_{k-1} \cdots b_0) \pmod{p}$$

$$a_{n+k-1}(2)^{n+k-1} + \ldots a_0 \equiv R(2)((2)^k + b_{k-1}(2)^{k-1} \cdots b_0) \pmod{p}$$

$$\vdots$$
Solving for $Q(x)$ and $E(x)$...

For all points $1, \ldots, i, n+2k$,

$$Q(i) = R(i)E(i) \pmod{p}$$

Gives $n+2k$ linear equations.

$$a_{n+k-1} + \ldots a_0 \equiv R(1)(1 + b_{k-1} \ldots b_0) \pmod{p}$$

$$a_{n+k-1}(2)^{n+k-1} + \ldots a_0 \equiv R(2)((2)^k + b_{k-1}(2)^{k-1} \ldots b_0) \pmod{p}$$

$$\vdots$$

$$a_{n+k-1}(m)^{n+k-1} + \ldots a_0 \equiv R(m)((m)^k + b_{k-1}(m)^{k-1} \ldots b_0) \pmod{p}$$
Solving for $Q(x)$ and $E(x)$...

For all points $1, \ldots, i, n+2k$,

$$Q(i) = R(i)E(i) \pmod{p}$$

Gives $n+2k$ linear equations.

$$a_{n+k-1} + \ldots + a_0 \equiv R(1)(1 + b_{k-1} \ldots b_0) \pmod{p}$$

$$a_{n+k-1}(2)^{n+k-1} + \ldots + a_0 \equiv R(2)((2)^k + b_{k-1}(2)^{k-1} \ldots b_0) \pmod{p}$$

$$\vdots$$

$$a_{n+k-1}(m)^{n+k-1} + \ldots + a_0 \equiv R(m)((m)^k + b_{k-1}(m)^{k-1} \ldots b_0) \pmod{p}$$

..and $n+2k$ unknown coefficients of $Q(x)$ and $E(x)$!
Solving for $Q(x)$ and $E(x)$...

For all points $1, \ldots, i, n+2k$,

$$Q(i) = R(i)E(i) \pmod{p}$$

Gives $n+2k$ linear equations.

$$a_{n+k-1} + \ldots a_0 \equiv R(1)(1 + b_{k-1} \ldots b_0) \pmod{p}$$

$$a_{n+k-1}(2)^{n+k-1} + \ldots a_0 \equiv R(2)((2)^k + b_{k-1}(2)^{k-1} \ldots b_0) \pmod{p}$$

$$\vdots$$

$$a_{n+k-1}(m)^{n+k-1} + \ldots a_0 \equiv R(m)((m)^k + b_{k-1}(m)^{k-1} \ldots b_0) \pmod{p}$$

..and $n+2k$ unknown coefficients of $Q(x)$ and $E(x)$!

Solve for coefficients of $Q(x)$ and $E(x)$. 
Solving for $Q(x)$ and $E(x)$...and $P(x)$

For all points $1, \ldots, i, n+2k$,

$$Q(i) = R(i)E(i) \pmod{p}$$

Gives $n+2k$ linear equations.

$$a_{n+k-1} + \ldots a_0 \equiv R(1)(1 + b_{k-1} \ldots b_0) \pmod{p}$$

$$a_{n+k-1}(2)^{n+k-1} + \ldots a_0 \equiv R(2)((2)^k + b_{k-1}(2)^{k-1} \ldots b_0) \pmod{p}$$

$$\vdots$$

$$a_{n+k-1}(m)^{n+k-1} + \ldots a_0 \equiv R(m)((m)^k + b_{k-1}(m)^{k-1} \ldots b_0) \pmod{p}$$

..and $n+2k$ unknown coefficients of $Q(x)$ and $E(x)$!

Solve for coefficients of $Q(x)$ and $E(x)$.

Find $P(x) = Q(x)/E(x)$. 
Solving for $Q(x)$ and $E(x)$...and $P(x)$

For all points $1, \ldots, i, n+2k$,

$$Q(i) = R(i)E(i) \pmod{p}$$

Gives $n+2k$ linear equations.

$$a_{n+k-1} + \cdots + a_0 \equiv R(1)(1 + b_{k-1} \cdots b_0) \pmod{p}$$

$$a_{n+k-1}(2)^{n+k-1} + \cdots + a_0 \equiv R(2)((2)^k + b_{k-1}(2)^{k-1} \cdots b_0) \pmod{p}$$

$$a_{n+k-1}(m)^{n+k-1} + \cdots + a_0 \equiv R(m)((m)^k + b_{k-1}(m)^{k-1} \cdots b_0) \pmod{p}$$

..and $n+2k$ unknown coefficients of $Q(x)$ and $E(x)$!

Solve for coefficients of $Q(x)$ and $E(x)$.

Find $P(x) = Q(x)/E(x)$. 

Solving for $Q(x)$ and $E(x)$...and $P(x)$

For all points $1, \ldots, i, n+2k$,

$$Q(i) = R(i)E(i) \pmod{p}$$

Gives $n+2k$ linear equations.

$$a_{n+k-1} + \ldots a_0 \equiv R(1)(1 + b_{k-1} \cdots b_0) \pmod{p}$$

$$a_{n+k-1}(2)^{n+k-1} + \ldots a_0 \equiv R(2)((2)^k + b_{k-1}(2)^{k-1} \cdots b_0) \pmod{p}$$

$$\vdots$$

$$a_{n+k-1}(m)^{n+k-1} + \ldots a_0 \equiv R(m)((m)^k + b_{k-1}(m)^{k-1} \cdots b_0) \pmod{p}$$

..and $n+2k$ unknown coefficients of $Q(x)$ and $E(x)$!

Solve for coefficients of $Q(x)$ and $E(x)$.

Find $P(x) = Q(x)/E(x)$. 
Solving for $Q(x)$ and $E(x)$...and $P(x)$

For all points $1, \ldots, i, n+2k$,

$$Q(i) = R(i)E(i) \pmod{p}$$

Gives $n+2k$ linear equations.

$$a_{n+k-1} + \ldots a_0 \equiv R(1)(1 + b_{k-1} \ldots b_0) \pmod{p}$$

$$a_{n+k-1}(2)^{n+k-1} + \ldots a_0 \equiv R(2)((2)^k + b_{k-1}(2)^{k-1} \ldots b_0) \pmod{p}$$

$$\vdots$$

$$a_{n+k-1}(m)^{n+k-1} + \ldots a_0 \equiv R(m)((m)^k + b_{k-1}(m)^{k-1} \ldots b_0) \pmod{p}$$

..and $n+2k$ unknown coefficients of $Q(x)$ and $E(x)$!

Solve for coefficients of $Q(x)$ and $E(x)$.

Find $P(x) = Q(x)/E(x)$. 
Example.

Received $R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3$
Example.

Received \( R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3 \)

\[
Q(x) = E(x) P(x) = a_3 x^3 + a_2 x^2 + a_1 x + a_0
\]
Example.

Received $R(1) = 3$, $R(2) = 1$, $R(3) = 6$, $R(4) = 0$, $R(5) = 3$

$Q(x) = E(x)P(x) = a_3 x^3 + a_2 x^2 + a_1 x + a_0$

$E(x) = x - b_0$
Example.

Received $R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3$

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

$E(x) = x - b_0$

$Q(i) = R(i)E(i)$. 
Example.

Received $R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3$

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

$E(x) = x - b_0$

$Q(i) = R(i)E(i)$.

\[ a_3 + a_2 + a_1 + a_0 \equiv 3(1 - b_0) \pmod{7} \]
Example.

Received $R(1) = 3$, $R(2) = 1$, $R(3) = 6$, $R(4) = 0$, $R(5) = 3$

$Q(x) = E(x)P(x) = a_3 x^3 + a_2 x^2 + a_1 x + a_0$

$E(x) = x - b_0$

$Q(i) = R(i)E(i)$.

\[
\begin{align*}
a_3 + a_2 + a_1 + a_0 & \equiv 3(1 - b_0) \pmod{7} \\
a_3 + 4a_2 + 2a_1 + a_0 & \equiv 1(2 - b_0) \pmod{7}
\end{align*}
\]
Example.

Received $R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3$

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

$E(x) = x - b_0$

$Q(i) = R(i)E(i)$.

\[
\begin{align*}
    a_3 + a_2 + a_1 + a_0 & \equiv 3(1 - b_0) \pmod{7} \\
    a_3 + 4a_2 + 2a_1 + a_0 & \equiv 1(2 - b_0) \pmod{7} \\
    6a_3 + 2a_2 + 3a_1 + a_0 & \equiv 6(3 - b_0) \pmod{7} \\
    a_3 + 2a_2 + 4a_1 + a_0 & \equiv 0(4 - b_0) \pmod{7} \\
    6a_3 + 4a_2 + 5a_1 + a_0 & \equiv 3(5 - b_0) \pmod{7}
\end{align*}
\]
Example.

Received $R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3$

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

$E(x) = x - b_0$

$Q(i) = R(i)E(i)$.

\[
\begin{align*}
a_3 + a_2 + a_1 + a_0 & \equiv 3(1 - b_0) \pmod{7} \\
a_3 + 4a_2 + 2a_1 + a_0 & \equiv 1(2 - b_0) \pmod{7} \\
6a_3 + 2a_2 + 3a_1 + a_0 & \equiv 6(3 - b_0) \pmod{7} \\
a_3 + 2a_2 + 4a_1 + a_0 & \equiv 0(4 - b_0) \pmod{7} \\
6a_3 + 4a_2 + 5a_1 + a_0 & \equiv 3(5 - b_0) \pmod{7}
\end{align*}
\]

$a_3 = 1, a_2 = 6, a_1 = 6, a_0 = 5$ and $b_0 = 2$. 
Example.

Received $R(1) = 3, R(2) = 1, R(3) = 6, R(4) = 0, R(5) = 3$

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

$E(x) = x - b_0$

$Q(i) = R(i)E(i)$.

\[
\begin{align*}
  a_3 + a_2 + a_1 + a_0 & \equiv 3(1 - b_0) \pmod{7} \\
  a_3 + 4a_2 + 2a_1 + a_0 & \equiv 1(2 - b_0) \pmod{7} \\
  6a_3 + 2a_2 + 3a_1 + a_0 & \equiv 6(3 - b_0) \pmod{7} \\
  a_3 + 2a_2 + 4a_1 + a_0 & \equiv 0(4 - b_0) \pmod{7} \\
  6a_3 + 4a_2 + 5a_1 + a_0 & \equiv 3(5 - b_0) \pmod{7}
\end{align*}
\]

$a_3 = 1, a_2 = 6, a_1 = 6, a_0 = 5$ and $b_0 = 2$.

$Q(x) = x^3 + 6x^2 + 6x + 5$. 
Example.

Received $R(1) = 3$, $R(2) = 1$, $R(3) = 6$, $R(4) = 0$, $R(5) = 3$

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

$E(x) = x - b_0$

$Q(i) = R(i)E(i)$.

\[
\begin{align*}
  a_3 + a_2 + a_1 + a_0 & \equiv 3(1 - b_0) \pmod{7} \\
  a_3 + 4a_2 + 2a_1 + a_0 & \equiv 1(2 - b_0) \pmod{7} \\
  6a_3 + 2a_2 + 3a_1 + a_0 & \equiv 6(3 - b_0) \pmod{7} \\
  a_3 + 2a_2 + 4a_1 + a_0 & \equiv 0(4 - b_0) \pmod{7} \\
  6a_3 + 4a_2 + 5a_1 + a_0 & \equiv 3(5 - b_0) \pmod{7}
\end{align*}
\]

$a_3 = 1$, $a_2 = 6$, $a_1 = 6$, $a_0 = 5$ and $b_0 = 2$.

$Q(x) = x^3 + 6x^2 + 6x + 5$.

$E(x) = x - 2$. 
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{c}
\frac{x - 2}{x^3 + 6x^2 + 6x + 5} \\
\end{array}
\]

Message is

\[ P(x) = x^2 + x + 1 \]

\[ P(1) = 3, \quad P(2) = 0, \quad P(3) = 6. \]

What is \( x - 2 \)?

Except at \( x = 2 \)?

Hole there?
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{c}
1 \ x^2 \\
\hline
x - 2 ) x^3 + 6x^2 + 6x + 5 \\
x^3 - 2x^2 \\
1x^2 + 6x + 5 \\
x^2 - 2x^2
\end{array}
\]

Message is

\[ P(x) = x^2 + x + 1 \]

\[ P(1) = 3, \ P(2) = 0, \ P(3) = 6. \]

What is \( x - 2 \)?

Except at \( x = 2 \)?

Hole there?
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{c}
1 \times^2 \\
\hline
x - 2 ) x^3 + 6x^2 + 6x + 5 \\
x^3 - 2 \times^2 \\
\hline
1 \times^2 + 6x + 5
\end{array}
\]

\[ P(x) = x^2 + x + 1 \]
\[ P(1) = 3, \quad P(2) = 0, \quad P(3) = 6. \]

What is \( x - 2 \)? Except at \( x = 2 \)?
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{c}
1 \\
\hline
x - 2 \quad ) \quad x^3 + 6x^2 + 6x + 5 \\
\hline
x^3 - 2x^2 \\
\hline
1x^2 + 6x + 5 \\
1x^2 - 2x \\
\hline
0
\end{array}
\]

Message is

\[ P(x) = x^2 + x + 1 \]

\[ P(1) = 3, \quad P(2) = 0, \quad P(3) = 6. \]

What is \[ x - 2 \quad x - 2 \quad x - 2 \] except at \[ x = 2? \] Hole there?
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{c}
1 \ x^2 + 1 \ x \\
\hline
x - 2 \ ) \ x^3 + 6 \ x^2 + 6 \ x + 5 \\
x^3 - 2 \ x^2 \\
\hline
1 \ x^2 + 6 \ x + 5 \\
1 \ x^2 - 2 \ x \\
\hline
x + 5
\end{array}
\]

\[ P(x) = x^2 + x + 1. \]
Message is \[ P(1) = 3, \ P(2) = 0, \ P(3) = 6. \]
What is \[ x - 2 \ x - 2 \] except at \[ x = 2? \]
Hole there?
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[ \begin{array}{c}
1 \ x^2 + 1 \ x + 1 \\
\hline
x - 2 \\ \\
\hline
\end{array} \]

\[ \begin{array}{c}
x^3 + 6x^2 + 6x + 5 \\
x^3 - 2x^2 \\
\hline
1 \ x^2 + 6 \ x + 5 \\
1 \ x^2 - 2 \ x \\
\hline
\end{array} \]

\[ x + 5 \]
\[ x - 2 \]
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{c}
1 \ x^2 + 1 \ x + 1 \\
-------------------
\end{array}
\]

\[
\begin{array}{c}
x - 2 ) x^3 + 6x^2 + 6x + 5 \\
x^3 - 2x^2 \\
----------
1 \ x^2 + 6 \ x + 5 \\
1 \ x^2 - 2 \ x \\
----------
\end{array}
\]

\[
\begin{array}{c}
x + 5 \\
x - 2 \\
-----
0
\end{array}
\]

Message is \[ P(x) = x^2 + x + 1 \]

P(1) = 3,
P(2) = 0,
P(3) = 6.

What is \( x - 2 \)?

Except at \( x = 2 \)?

Hole there?
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{r}
1 \ x^2 + 1 \ x + 1 \\
\hline
x - 2) \ x^3 + 6 \ x^2 + 6 \ x + 5 \\
\hline
x^3 - 2 \ x^2 \\
\hline
1 \ x^2 + 6 \ x + 5 \\
1 \ x^2 - 2 \ x \\
\hline
x + 5 \\
\hline
x - 2 \\
\hline
0
\end{array}
\]

\[ P(x) = x^2 + x + 1 \]
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{r}
1 \ x^2 + 1 \ x + 1 \\
\hline
x - 2 \ \) \ x^3 + 6 \ x^2 + 6 \ x + 5 \\
x^3 - 2 \ x^2 \\
\hline
1 \ x^2 + 6 \ x + 5 \\
1 \ x^2 - 2 \ x \\
\hline
x + 5 \\
x - 2 \\
\hline
0
\end{array}
\]

\[ P(x) = x^2 + x + 1 \]
Message is \( P(1) = 3, P(2) = 0, P(3) = 6. \)
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{c}
1 \ x^2 + 1 \ x + 1 \\
\hline
\end{array}
\]

\[
\begin{array}{c}
x - 2 \\
\hline
\end{array}
\]

\[
\begin{array}{c}
1 \ x^3 + 6 \ x^2 + 6 \ x + 5 \\
1 \ x^3 - 2 \ x^2 \\
\hline
1 \ x^2 + 6 \ x + 5 \\
1 \ x^2 - 2 \ x \\
\hline
x + 5 \\
x - 2 \\
\hline
0 \\
\end{array}
\]

\[ P(x) = x^2 + x + 1 \]

Message is \( P(1) = 3, P(2) = 0, P(3) = 6. \)

What is \( \frac{x - 2}{x - 2} \)?
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{cccc}
1 & x^2 & + & 1 & x & + & 1 \\
\hline 
x - 2 & | & x^3 & + & 6x^2 & + & 6x & + & 5 \\
   & | & x^3 & - & 2x^2 & & & & \\
   & |-------------------- & \\
   & | 1x^2 & + & 6x & + & 5 \\
   & | 1x^2 & - & 2x & & & & \\
   & |---------------------- & \\
   & | x & + & 5 \\
   & | x & - & 2 & & & & \\
   & |---------- & \\
   & | 0 & & & & & & \\
\end{array}
\]

\[ P(x) = x^2 + x + 1 \]
Message is \( P(1) = 3, P(2) = 0, P(3) = 6. \)

What is \( \frac{x-2}{x-2} \)? 1  Except at \( x = 2 \)?
Example: finishing up.

\[ Q(x) = x^3 + 6x^2 + 6x + 5. \]
\[ E(x) = x - 2. \]

\[
\begin{array}{c}
1 \ x^2 + 1 \ x + 1 \\
\hline
x - 2 ) x^3 + 6x^2 + 6x + 5 \\
\hline
x^3 - 2x^2 \\
\hline
x^2 + 6x + 5 \\
\hline
1 \ x^2 - 2x \\
\hline
x + 5 \\
\hline
\end{array}
\]

\[ P(x) = x^2 + x + 1 \]
Message is \( P(1) = 3, P(2) = 0, P(3) = 6. \)
What is \( \frac{x-2}{x-2} \)? 1  Except at \( x = 2 \)? Hole there?
Error Correction: Berlekamp-Welsh

Message: $m_1, \ldots, m_n$.

Sender:
1. Form degree $n-1$ polynomial $P(x)$ where $P(i) = m_i$.
2. Send $P(1), \ldots, P(n+2k)$.

Receiver:
1. Receive $R(1), \ldots, R(n+2k)$.
2. Solve $n+2k$ equations, $Q(i) = E(i)R(i)$ to find $Q(x) = E(x)P(x)$ and $E(x)$.
3. Compute $P(x) = Q(x)/E(x)$.
4. Compute $P(1), \ldots, P(n)$. 
Check your understanding.

You have error locator polynomial!
Check your undersanding.

You have error locator polynomial!
Where oh where can my bad packets be?...
Check your understanding.

You have error locator polynomial!
Where oh where can my bad packets be?...
Factor?
You have error locator polynomial!
Where oh where can my bad packets be?...
Factor? Sure.
Check your understanding.

You have error locator polynomial!
Where oh where can my bad packets be?...
Factor? Sure.
Check all values?
Check your understanding.

You have error locator polynomial!
Where oh where can my bad packets be?...
Factor? Sure.
Check all values? Sure.
Check your understanding.

You have error locator polynomial!

Where oh where can my bad packets be?...

Factor? Sure.

Check all values? Sure.
Check your understanding.

You have error locator polynomial!
Where oh where can my bad packets be?...
Factor? Sure.
Check all values? Sure.
Efficiency?
Check your understanding.

You have error locator polynomial!
Where oh where can my bad packets be?...
Factor? Sure.
Check all values? Sure.
Efficiency? Sure.
Check your understanding.

You have error locator polynomial!
Where oh where can my bad packets be?...
Factor? Sure.
Check all values? Sure.
Efficiency? Sure. Only $n + k$ values.
Check your understanding.

You have error locator polynomial!
Where oh where can my bad packets be?...
Factor? Sure.
Check all values? Sure.
Efficiency? Sure. Only $n + k$ values.
   See where it is 0.
Hmmm...

Is there one and only one $P(x)$ from Berlekamp-Welsh procedure?
Is there one and only one $P(x)$ from Berlekamp-Welsh procedure?

**Existence:** there is a $P(x)$ and $E(x)$ that satisfy equations.
Unique solution for $P(x)$

**Uniqueness:** any solution $Q'(x)$ and $E'(x)$ have

$$\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x).$$  \hspace{1cm} (1)
Unique solution for $P(x)$

**Uniqueness:** any solution $Q'(x)$ and $E'(x)$ have

$$\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x).$$

(1)

**Proof:**

Cross divide.
Unique solution for $P(x)$

**Uniqueness:** any solution $Q'(x)$ and $E'(x)$ have

$$\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x).$$ (1)

**Proof:**
We claim
Unique solution for $P(x)$

**Uniqueness:** any solution $Q'(x)$ and $E'(x)$ have

\[
\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x).
\]  

(1)

**Proof:**

We claim

\[
Q'(x)E(x) = Q(x)E'(x) \text{ on } n + 2k \text{ values of } x.
\]

(2)
Unique solution for $P(x)$

**Uniqueness:** any solution $Q'(x)$ and $E'(x)$ have

$$
\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x).
$$

(1)

**Proof:**
We claim

$$
Q'(x)E(x) = Q(x)E'(x) \text{ on } n + 2k \text{ values of } x.
$$

(2)

Equation 2 implies 1:
Unique solution for $P(x)$

**Uniqueness:** any solution $Q'(x)$ and $E'(x)$ have

$$
\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x).
$$

(1)

**Proof:**
We claim

$$
Q'(x)E(x) = Q(x)E'(x) \text{ on } n + 2k \text{ values of } x.
$$

(2)

Equation 2 implies 1:

$Q'(x)E(x) \text{ and } Q(x)E'(x)$ are degree $n + 2k - 1$
Unique solution for $P(x)$

**Uniqueness:** any solution $Q'(x)$ and $E'(x)$ have

$$
\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x).
$$

(1)

**Proof:**
We claim

$$Q'(x)E(x) = Q(x)E'(x) \text{ on } n+2k \text{ values of } x.
$$

(2)

Equation 2 implies 1:

$Q'(x)E(x)$ and $Q(x)E'(x)$ are degree $n+2k-1$
and agree on $n+2k$ points
Unique solution for $P(x)$

**Uniqueness:** any solution $Q'(x)$ and $E'(x)$ have

$$\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x). \quad (1)$$

**Proof:**

We claim

$$Q'(x)E(x) = Q(x)E'(x) \text{ on } n+2k \text{ values of } x. \quad (2)$$

Equation 2 implies 1:

$Q'(x)E(x)$ and $Q(x)E'(x)$ are degree $n+2k-1$ and agree on $n+2k$ points

$\implies Q'(x)E(x) = Q(x)E'(x)$. 
Unique solution for $P(x)$

**Uniqueness:** any solution $Q'(x)$ and $E'(x)$ have

$$\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x).$$ \hspace{1cm} (1)

**Proof:**
We claim

$$Q'(x)E(x) = Q(x)E'(x) \text{ on } n + 2k \text{ values of } x.$$ \hspace{1cm} (2)

Equation 2 implies 1:

$Q'(x)E(x)$ and $Q(x)E'(x)$ are degree $n + 2k - 1$

and agree on $n + 2k$ points

$$\implies Q'(x)E(x) = Q(x)E'(x).$$

Cross divide.
**Unique solution for** \( P(x) \)

**Uniqueness:** any solution \( Q'(x) \) and \( E'(x) \) have

\[
\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x). 
\]

(1)

**Proof:**

We claim

\[
Q'(x)E(x) = Q(x)E'(x) \text{ on } n+2k \text{ values of } x. 
\]

(2)

Equation 2 implies 1:

\( Q'(x)E(x) \) and \( Q(x)E'(x) \) are degree \( n+2k-1 \)

and agree on \( n+2k \) points

\[
\implies Q'(x)E(x) = Q(x)E'(x). 
\]

Cross divide.

\( \Box \)
Fact: $Q'(x)E(x) = Q(x)E'(x)$ on $n + 2k$ values of $x$. 
Fact: $Q'(x)E(x) = Q(x)E'(x)$ on $n + 2k$ values of $x$.

Proof:
Fact: $Q'(x)E(x) = Q(x)E'(x)$ on $n + 2k$ values of $x$.

Proof: Construction implies that
Fact: \( Q'(x)E(x) = Q(x)E'(x) \) on \( n + 2k \) values of \( x \).

Proof: Construction implies that

\[
\begin{align*}
Q(i) &= R(i)E(i) \\
Q'(i) &= R(i)E'(i)
\end{align*}
\]
Fact: \( Q'(x)E(x) = Q(x)E'(x) \) on \( n + 2k \) values of \( x \).

Proof: Construction implies that

\[
Q(i) = R(i)E(i) \\
Q'(i) = R(i)E'(i)
\]

for \( i \in \{1, \ldots, n + 2k\} \).
Fact: \( Q'(x)E(x) = Q(x)E'(x) \) on \( n+2k \) values of \( x \).

Proof: Construction implies that

\[
Q(i) = R(i)E(i) \\
Q'(i) = R(i)E'(i)
\]

for \( i \in \{1, \ldots, n+2k\} \).

If \( E(i) = 0 \), then \( Q(i) = 0 \).
**Fact:** \( Q'(x)E(x) = Q(x)E'(x) \) on \( n + 2k \) values of \( x \).

**Proof:** Construction implies that

\[
Q(i) = R(i)E(i) \\
Q'(i) = R(i)E'(i)
\]

for \( i \in \{1, \ldots n+2k\} \).

If \( E(i) = 0 \), then \( Q(i) = 0 \). If \( E'(i) = 0 \), then \( Q'(i) = 0 \).
Fact: \( Q'(x)E(x) = Q(x)E'(x) \) on \( n + 2k \) values of \( x \).

Proof: Construction implies that

\[
Q(i) = R(i)E(i) \\
Q'(i) = R(i)E'(i)
\]

for \( i \in \{1, \ldots n + 2k\} \).

If \( E(i) = 0 \), then \( Q(i) = 0 \). If \( E'(i) = 0 \), then \( Q'(i) = 0 \).

\[
\implies Q(i)E'(i) = Q'(i)E(i) \text{ holds when } E(i) \text{ or } E'(i) \text{ are zero.}
\]
Fact: \( Q'(x)E(x) = Q(x)E'(x) \) on \( n + 2k \) values of \( x \).

Proof: Construction implies that

\[
Q(i) = R(i)E(i) \\
Q'(i) = R(i)E'(i)
\]

for \( i \in \{1, \ldots n + 2k\} \).

If \( E(i) = 0 \), then \( Q(i) = 0 \). If \( E'(i) = 0 \), then \( Q'(i) = 0 \).

\[\Rightarrow Q(i)E'(i) = Q'(i)E(i) \]
holds when \( E(i) \) or \( E'(i) \) are zero.

When \( E'(i) \) and \( E(i) \) are not zero
Fact: $Q'(x)E(x) = Q(x)E'(x)$ on $n + 2k$ values of $x$.

Proof: Construction implies that

$$Q(i) = R(i)E(i)$$
$$Q'(i) = R(i)E'(i)$$

for $i \in \{1, \ldots n + 2k\}$.

If $E(i) = 0$, then $Q(i) = 0$. If $E'(i) = 0$, then $Q'(i) = 0$.

$$\Rightarrow Q(i)E'(i) = Q'(i)E(i)$$ holds when $E(i)$ or $E'(i)$ are zero.

When $E'(i)$ and $E(i)$ are not zero

$$\frac{Q'(i)}{E'(i)} = \frac{Q(i)}{E(i)} = R(i).$$
Fact: \( Q'(x)E(x) = Q(x)E'(x) \) on \( n + 2k \) values of \( x \).

Proof: Construction implies that

\[
Q(i) = R(i)E(i) \\
Q'(i) = R(i)E'(i)
\]

for \( i \in \{1, \ldots n + 2k \} \).

If \( E(i) = 0 \), then \( Q(i) = 0 \). If \( E'(i) = 0 \), then \( Q'(i) = 0 \). \( \implies \) \( Q(i)E'(i) = Q'(i)E(i) \) holds when \( E(i) \) or \( E'(i) \) are zero.

When \( E'(i) \) and \( E(i) \) are not zero

\[
\frac{Q'(i)}{E'(i)} = \frac{Q(i)}{E(i)} = R(i).
\]

Cross multiplying gives equality in fact for these points.
Fact: \( Q'(x)E(x) = Q(x)E'(x) \) on \( n + 2k \) values of \( x \).

Proof: Construction implies that

\[
Q(i) = R(i)E(i) \\
Q'(i) = R(i)E'(i)
\]

for \( i \in \{1, \ldots n + 2k\} \).

If \( E(i) = 0 \), then \( Q(i) = 0 \). If \( E'(i) = 0 \), then \( Q'(i) = 0 \).

\[ \implies Q(i)E'(i) = Q'(i)E(i) \]
holds when \( E(i) \) or \( E'(i) \) are zero.

When \( E'(i) \) and \( E(i) \) are not zero

\[
\frac{Q'(i)}{E'(i)} = \frac{Q(i)}{E(i)} = R(i).
\]

Cross multiplying gives equality in fact for these points. \( \square \)
Fact: $Q'(x)E(x) = Q(x)E'(x)$ on $n + 2k$ values of $x$.

Proof: Construction implies that

$$Q(i) = R(i)E(i)$$
$$Q'(i) = R(i)E'(i)$$

for $i \in \{1, \ldots n + 2k\}$.

If $E(i) = 0$, then $Q(i) = 0$. If $E'(i) = 0$, then $Q'(i) = 0$.

$$\implies Q(i)E'(i) = Q'(i)E(i)$$

holds when $E(i)$ or $E'(i)$ are zero.

When $E'(i)$ and $E(i)$ are not zero

$$\frac{Q'(i)}{E'(i)} = \frac{Q(i)}{E(i)} = R(i).$$

Cross multiplying gives equality in fact for these points.

Points to polynomials, have to deal with zeros!
Fact: $Q'(x)E(x) = Q(x)E'(x)$ on $n + 2k$ values of $x$.

Proof: Construction implies that

$$Q(i) = R(i)E(i)$$
$$Q'(i) = R(i)E'(i)$$

for $i \in \{1, \ldots, n + 2k\}$.

If $E(i) = 0$, then $Q(i) = 0$. If $E'(i) = 0$, then $Q'(i) = 0$.

$\implies Q(i)E'(i) = Q'(i)E(i)$ holds when $E(i)$ or $E'(i)$ are zero.

When $E'(i)$ and $E(i)$ are not zero

$$\frac{Q'(i)}{E'(i)} = \frac{Q(i)}{E(i)} = R(i).$$

Cross multiplying gives equality in fact for these points.

Points to polynomials, have to deal with zeros!

Example: dealing with $\frac{x-2}{x-2}$ at $x = 2$. 
Berlekamp-Welsh algorithm decodes correctly when $k$ errors!
Communicate $n$ packets, with $k$ erasures.

Reconstruct $P(x)$ with any $n$ points.

Why?

$k$ changes to make different messages overlap.

How to encode?

With polynomial, $P(x)$.

Of degree?

$n - 1$.

Recover?

Reconstruct error polynomial, $E(x)$, and $P(x)$!

Nonlinear equations.

Reconstruct $E(x)$ and $Q(x) = E(x)P(x)$.

Linear Equations.

Polynomial division! $P(x) = Q(x)/E(x)$!
Communicate $n$ packets, with $k$ erasures.

How many packets?
Summary. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets? $n + k$
Summary. Error Correction.

Communicate $n$ packets, with $k$ erasures.

- How many packets? $n + k$
- How to encode?

Reed-Solomon codes. Welsh-Berlekamp Decoding. Perfection!
Summary. Error Correction.

Communicate $n$ packets, with $k$ erasures.

- How many packets? $n + k$
- How to encode? With polynomial, $P(x)$.
Summary. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets? $n + k$
How to encode? With polynomial, $P(x)$.
Of degree?

Reed-Solomon codes.
Welsh-Berlekamp Decoding.
Perfection!
Summary. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets? $n + k$
How to encode? With polynomial, $P(x)$.
Of degree? $n - 1$
Communicate $n$ packets, with $k$ erasures.

How many packets? $n + k$
How to encode? With polynomial, $P(x)$.
Of degree? $n - 1$
Recover?
Summary. Error Correction.

Communicate $n$ packets, with $k$ erasures.

- How many packets? $n + k$
- How to encode? With polynomial, $P(x)$.
- Of degree? $n - 1$
- Recover? Reconstruct $P(x)$ with any $n$ points!
Summary. Error Correction.

Communicate \( n \) packets, with \( k \) erasures.

- How many packets? \( n + k \)
- How to encode? With polynomial, \( P(x) \).
- Of degree? \( n - 1 \)
- Recover? Reconstruct \( P(x) \) with any \( n \) points!

Communicate \( n \) packets, with \( k \) errors.

Why? \( k \) changes to make different messages overlap.

How to encode? With polynomial, \( P(x) \).

Of degree? \( n - 1 \).

Recover? Reconstruct error polynomial, \( E(x) \), and \( P(x) \)!

Reconstruct \( E(x) \) and \( Q(x) = E(x)P(x) \).

Why? \( P(x) \) is divisible by \( E(x) \).

Reed-Solomon codes.

Welsh-Berlekamp Decoding.

Perfection!
Communicate $n$ packets, with $k$ erasures.

- How many packets? $n + k$
- How to encode? With polynomial, $P(x)$.
- Of degree? $n - 1$
- Recover? Reconstruct $P(x)$ with any $n$ points!

Communicate $n$ packets, with $k$ errors.

- How many packets?
Summary. Error Correction.

Communicate $n$ packets, with $k$ erasures.

- How many packets? $n + k$
- How to encode? With polynomial, $P(x)$.
- Of degree? $n - 1$
- Recover? Reconstruct $P(x)$ with any $n$ points!

Communicate $n$ packets, with $k$ errors.

- How many packets? $n + 2k$
Summary. Error Correction.

Communicate $n$ packets, with $k$ erasures.

- How many packets? $n + k$
- How to encode? With polynomial, $P(x)$.
- Of degree? $n - 1$
- Recover? Reconstruct $P(x)$ with any $n$ points!

Communicate $n$ packets, with $k$ errors.

- How many packets? $n + 2k$
- Why?
Summary. Error Correction.

Communicate \( n \) packets, with \( k \) erasures.

- How many packets? \( n + k \)
- How to encode? With polynomial, \( P(x) \).
- Of degree? \( n - 1 \)
- Recover? Reconstruct \( P(x) \) with any \( n \) points!

Communicate \( n \) packets, with \( k \) errors.

- How many packets? \( n + 2k \)
- Why?
  - \( k \) changes to make diff. messages overlap
Summary. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets? $n + k$
How to encode? With polynomial, $P(x)$.
Of degree? $n - 1$
Recover? Reconstruct $P(x)$ with any $n$ points!

Communicate $n$ packets, with $k$ errors.

How many packets? $n + 2k$
Why?
  $k$ changes to make diff. messages overlap
How to encode?
Summary. Error Correction.

Communicate $n$ packets, with $k$ erasures.
How many packets? $n + k$
How to encode? With polynomial, $P(x)$.
Of degree? $n - 1$
Recover? Reconstruct $P(x)$ with any $n$ points!

Communicate $n$ packets, with $k$ errors.
How many packets? $n + 2k$
Why?
$\quad k$ changes to make diff. messages overlap
How to encode? With polynomial, $P(x)$. 
Summary. Error Correction.

Communicate $n$ packets, with $k$ erasures.

- How many packets? $n + k$
- How to encode? With polynomial, $P(x)$.
- Of degree? $n - 1$
- Recover? Reconstruct $P(x)$ with any $n$ points!

Communicate $n$ packets, with $k$ errors.

- How many packets? $n + 2k$
- Why?
  - $k$ changes to make diff. messages overlap
- How to encode? With polynomial, $P(x)$. Of degree?
Communicate $n$ packets, with $k$ erasures.

- How many packets? $n + k$
- How to encode? With polynomial, $P(x)$.
- Of degree? $n - 1$
- Recover? Reconstruct $P(x)$ with any $n$ points!

Communicate $n$ packets, with $k$ errors.

- How many packets? $n + 2k$
- Why?
  - $k$ changes to make diff. messages overlap

Reed-Solomon codes.

Welsh-Berlekamp Decoding.

Perfection!
Summary. Error Correction.

Communicate \( n \) packets, with \( k \) erasures.

- How many packets? \( n + k \)
- How to encode? With polynomial, \( P(x) \).
- Of degree? \( n - 1 \)
- Recover? Reconstruct \( P(x) \) with any \( n \) points!

Communicate \( n \) packets, with \( k \) errors.

- How many packets? \( n + 2k \)
- Why?
  - \( k \) changes to make diff. messages overlap
- How to encode? With polynomial, \( P(x) \).
- Of degree? \( n - 1 \)
- Recover?
Summary. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets? $n + k$
How to encode? With polynomial, $P(x)$.
Of degree? $n - 1$
Recover? Reconstruct $P(x)$ with any $n$ points!

Communicate $n$ packets, with $k$ errors.

How many packets? $n + 2k$
Why?
\[ k \text{ changes to make diff. messages overlap} \]
How to encode? With polynomial, $P(x)$. Of degree? $n - 1$.
Recover?
Summary. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets? $n + k$
How to encode? With polynomial, $P(x)$.
Of degree? $n - 1$
Recover? Reconstruct $P(x)$ with any $n$ points!

Communicate $n$ packets, with $k$ errors.

How many packets? $n + 2k$
Why?
$k$ changes to make diff. messages overlap
How to encode? With polynomial, $P(x)$. Of degree? $n - 1$.
Recover?
Reconstruct error polynomial, $E(X)$, and $P(x)$!
Summary. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets? $n + k$
How to encode? With polynomial, $P(x)$.
Of degree? $n - 1$
Recover? Reconstruct $P(x)$ with any $n$ points!

Communicate $n$ packets, with $k$ errors.

How many packets? $n + 2k$
Why?
  $k$ changes to make diff. messages overlap
How to encode? With polynomial, $P(x)$. Of degree? $n - 1$.
Recover?
  Reconstruct error polynomial, $E(X)$, and $P(x)$!
  Nonlinear equations.
Summary. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets? $n + k$
How to encode? With polynomial, $P(x)$.
Of degree? $n - 1$
Recover? Reconstruct $P(x)$ with any $n$ points!

Communicate $n$ packets, with $k$ errors.

How many packets? $n + 2k$
Why?
   $k$ changes to make diff. messages overlap
How to encode? With polynomial, $P(x)$. Of degree? $n - 1$.
Recover?
   Reconstruct error polynomial, $E(X)$, and $P(x)$!
      Nonlinear equations.
Reconstruct $E(x)$ and $Q(x) = E(x)P(x)$. 
Summary. Error Correction.

Communicate $n$ packets, with $k$ erasures.

How many packets? $n + k$
How to encode? With polynomial, $P(x)$.
Of degree? $n - 1$
Recover? Reconstruct $P(x)$ with any $n$ points!

Communicate $n$ packets, with $k$ errors.

How many packets? $n + 2k$
Why?
  $k$ changes to make diff. messages overlap
How to encode? With polynomial, $P(x)$. Of degree? $n - 1$.
Recover?
  Reconstruct error polynomial, $E(X)$, and $P(x)$!
    Nonlinear equations.
Reconstruct $E(x)$ and $Q(x) = E(x)P(x)$. Linear Equations.
Summary. Error Correction.

Communicate $n$ packets, with $k$ erasures.

- How many packets? $n + k$
- How to encode? With polynomial, $P(x)$.
- Of degree? $n - 1$
- Recover? Reconstruct $P(x)$ with any $n$ points!

Communicate $n$ packets, with $k$ errors.

- How many packets? $n + 2k$
- Why?
  - $k$ changes to make different messages overlap
- Recover?
  - Reconstruct error polynomial, $E(X)$, and $P(x)$!
    - Nonlinear equations.
  - Reconstruct $E(x)$ and $Q(x) = E(x)P(x)$. Linear Equations.
  - Polynomial division!
Summary. Error Correction.

Communicate \( n \) packets, with \( k \) erasures.

- How many packets? \( n + k \)
- How to encode? With polynomial, \( P(x) \).
- Of degree? \( n - 1 \)
- Recover? Reconstruct \( P(x) \) with any \( n \) points!

Communicate \( n \) packets, with \( k \) errors.

- How many packets? \( n + 2k \)
- Why?
  - \( k \) changes to make diff. messages overlap
- How to encode? With polynomial, \( P(x) \). Of degree? \( n - 1 \).
- Recover?
  - Reconstruct error polynomial, \( E(X) \), and \( P(x) \)!
    - Nonlinear equations.
  - Reconstruct \( E(x) \) and \( Q(x) = E(x)P(x) \). Linear Equations.
  - Polynomial division! \( P(x) = Q(x)/E(x) \)!
Summary. Error Correction.

Communicate $n$ packets, with $k$ erasures.

- How many packets? $n + k$
- How to encode? With polynomial, $P(x)$.
- Of degree? $n - 1$
- Recover? Reconstruct $P(x)$ with any $n$ points!

Communicate $n$ packets, with $k$ errors.

- How many packets? $n + 2k$
- Why?
  - $k$ changes to make diff. messages overlap
- Recover?
  - Reconstruct error polynomial, $E(X)$, and $P(x)$!
    - Nonlinear equations.
  - Reconstruct $E(x)$ and $Q(x) = E(x)P(x)$. Linear Equations.
  - Polynomial division! $P(x) = Q(x)/E(x)$!

Reed-Solomon codes.
Summary. Error Correction.

Communicate \( n \) packets, with \( k \) erasures.

How many packets? \( n + k \)
How to encode? With polynomial, \( P(x) \).
Of degree? \( n – 1 \)
Recover? Reconstruct \( P(x) \) with any \( n \) points!

Communicate \( n \) packets, with \( k \) errors.

How many packets? \( n + 2k \)
Why?
\( k \) changes to make diff. messages overlap
How to encode? With polynomial, \( P(x) \). Of degree? \( n – 1 \).
Recover?
Reconstruct error polynomial, \( E(X) \), and \( P(x) \)!
Nonlinear equations.
Reconstruct \( E(x) \) and \( Q(x) = E(x)P(x) \). Linear Equations.
Polynomial division! \( P(x) = Q(x)/E(x) \)!

Reed-Solomon codes. Welsh-Berlekamp Decoding.
Summary. Error Correction.

Communicate $n$ packets, with $k$ erasures.

- How many packets? $n + k$
- How to encode? With polynomial, $P(x)$.
- Of degree? $n - 1$
- Recover? Reconstruct $P(x)$ with any $n$ points!

Communicate $n$ packets, with $k$ errors.

- How many packets? $n + 2k$
- Why?
  - $k$ changes to make diff. messages overlap
- Recover?
  - Reconstruct error polynomial, $E(X)$, and $P(x)$!
    - Nonlinear equations.
  - Reconstruct $E(x)$ and $Q(x) = E(x)P(x)$. Linear Equations.
    - Polynomial division! $P(x) = Q(x)/E(x)$!

Reed-Solomon codes. Welsh-Berlekamp Decoding. Perfection!