Problem: Want to send a message with $n$ packets.
Channel: Lossy channel: loses $k$ packets.
Question: Can you send $n + k$ packets and recover message?
A degree $n - 1$ polynomial determined by any $n$ points!

Erasure Coding Scheme: message $= m_0, m_2, \ldots, m_{n-1}$.
1. Choose prime $p \approx 2^b$ for packet size $b$.
2. $P(x) = m_{n-1}x^{n-1} + \cdots + m_0 \pmod{p}$.
3. Send $P(1), \ldots, P(n + k)$.
Any $n$ of the $n + k$ packets gives polynomial ... and message!

Complexity Issues.
Size: Can choose a prime between $2^{b-1}$ and $2^b$.
(Lose at most 1 bit per packet.)
In practice, $O(n)$ operations with almost the same redundancy.

Satellite
GPS device
$n$ packet message. Send $n + k$!
Lose $k$ packets.
Any $n$ packets is enough!
$n$ packet message.
Optimal.

Polynomials.
▶ ..give Secret Sharing.
▶ ..give Erasure Codes.

Error Correction:
Noisy Channel: corrupts $k$ packets. (rather than loses.)
Additional Challenge: Finding which packets are corrupt.

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 $R(1), \ldots, R(n + 2k)$.
Property: $P(i) = R(i)$ for at least $n + k$ points $i$. 

The Scheme.
Brute Force: For each subset of $n + k$ points fit degree $n - 1$ polynomial to them.

- For subset of $n + k$ pts where $R(i) = P(i)$, method will reconstruct $P(x)$!
- For any subset of $n + k$ pts,
  1. at least $n$ pts are correct since only $k$ errors,
  2. $P(x)$ is only degree $n - 1$ polynomial that contains the $n$ correct points.

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

Details..

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

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

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

$$\vdots$$

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

Error!! ... Where??

Could be anywhere!!! ...so try everything.

Runtime: \((n+2k)^k\) possibilities.

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

Reed Solomon codes of size $n + 2k$ can tolerate $k$ errors!
The scheme is information optimal:
Must send $n + 2k$ packets to recover from $k$ errors.

- Any two codewords must be different by $2k$.
- Total information must equal original $n$ packets, plus which $k$ packets are corrupted.
  $n + 2k$ packets, $k$ garbage packets: $n + k$ packets of information.

Runtime is bad: exponential in $k$.

Find the errors faster?
...on Wednesday.