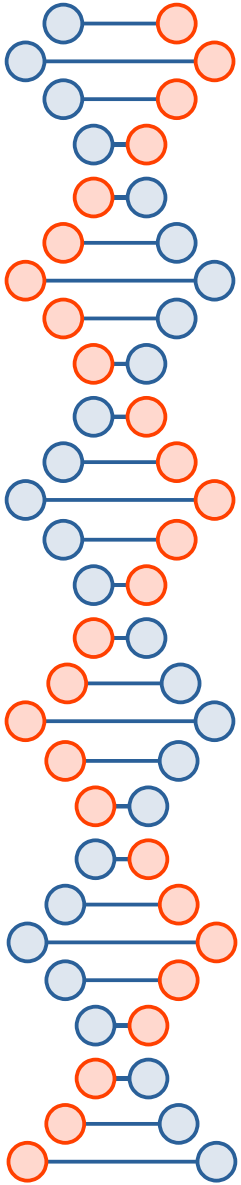
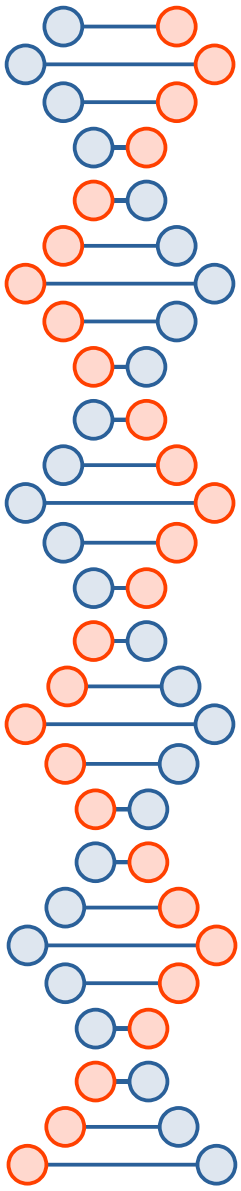


Overview of post-quantum cryptography
CSE 539

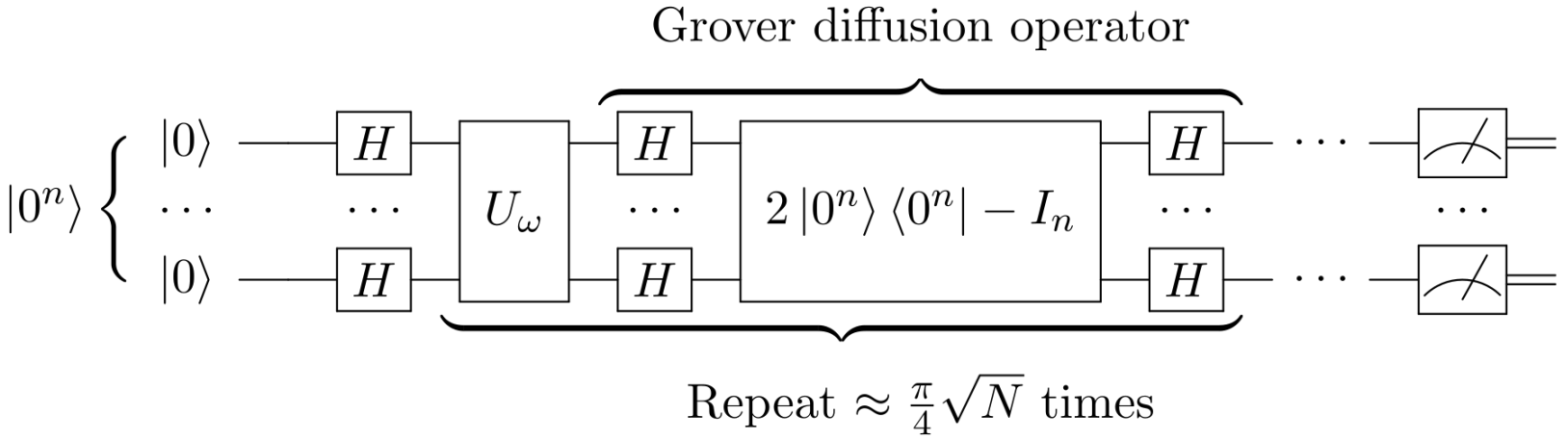
jedimaestro@asu.edu



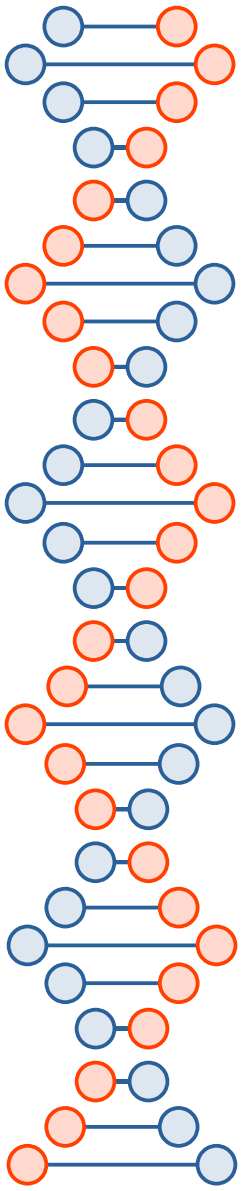
What we need

- Symmetric
  - Encryption
  - Authentication
  - Secure hashes
  - Others?
- Asymmetric
  - Encryption
  - Non-repudiability (signatures)
  - Key exchange
  - Others? (*e.g.*, homomorphic)

Grover’s algorithm

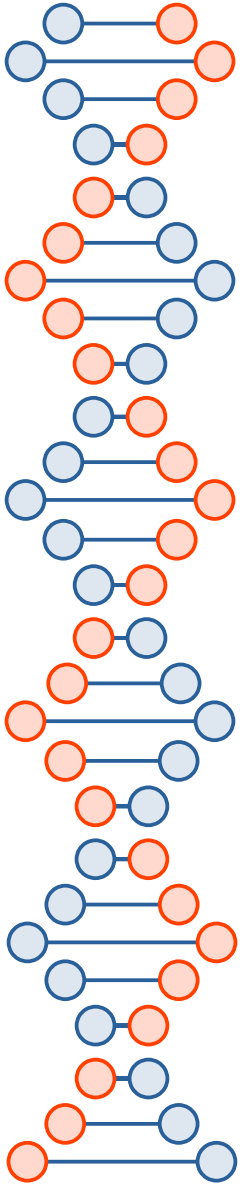


https://en.wikipedia.org/wiki/Grover%27s_algorithm#/media/File:Grover's_algorithm_circuit.svg



Symmetric crypto

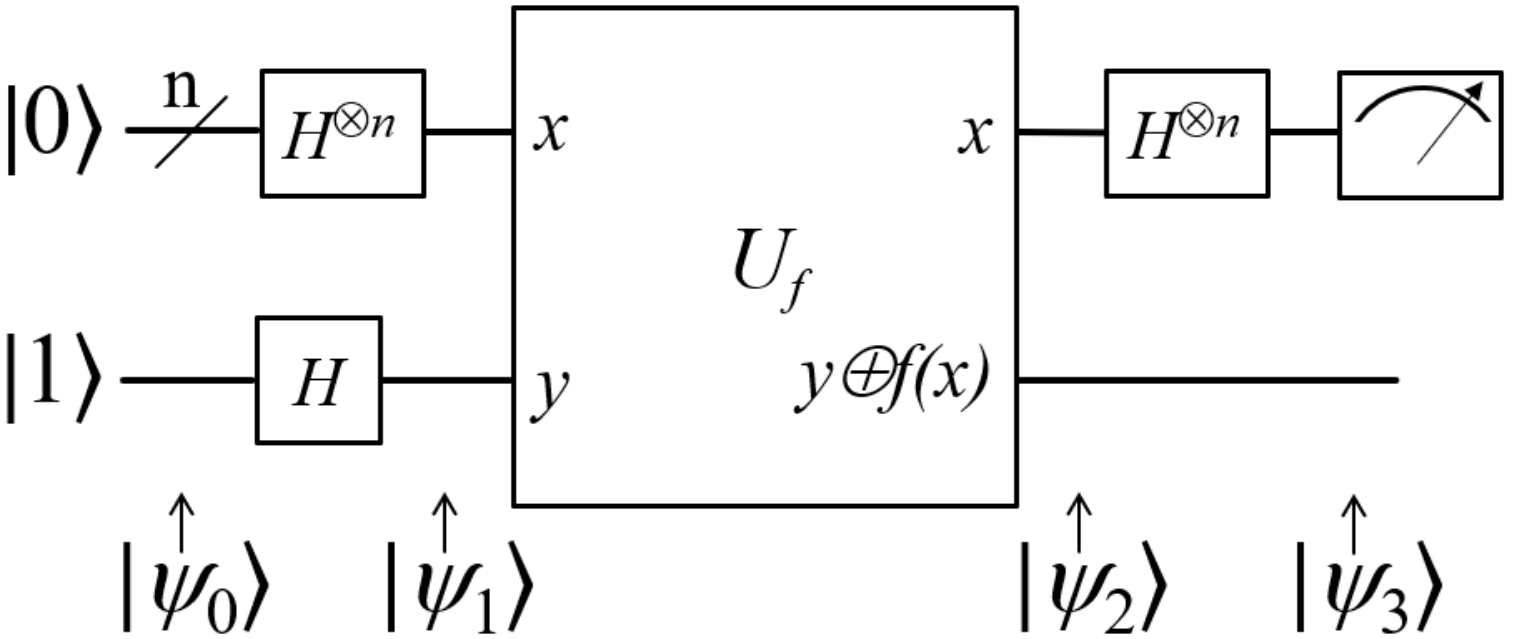
- Double the key size
  - sqrt(2^{2n}) = 2^{n}
  - sqrt(2^{256}) = 2^{128}



Asymmetric Crypto

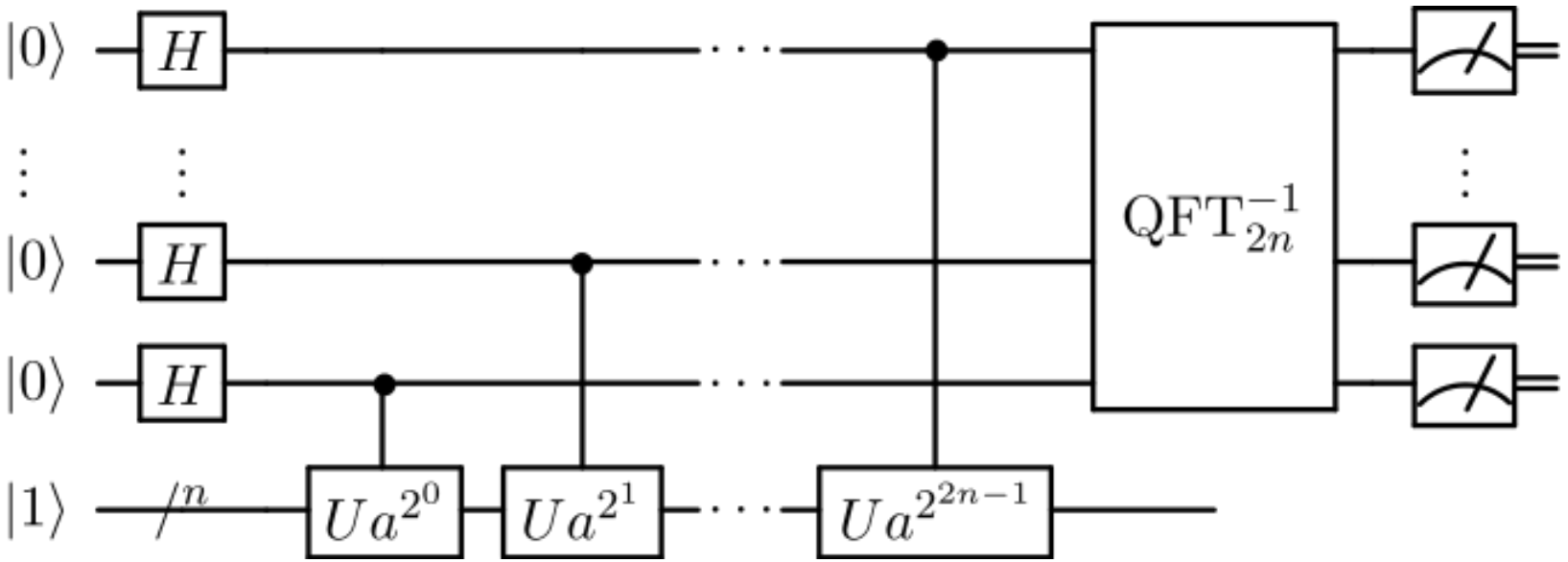
- Quantum computers seem to be good at the same kinds of things that make good trapdoor functions for asymmetric crypto (factorization, discrete log, *etc.*)
  - But not everything
    - Older schemes (*e.g.*, Merkle's signature scheme)
    - Newer schemes (*e.g.*, lattice-based)

Deutsh-Jozsa algorithm

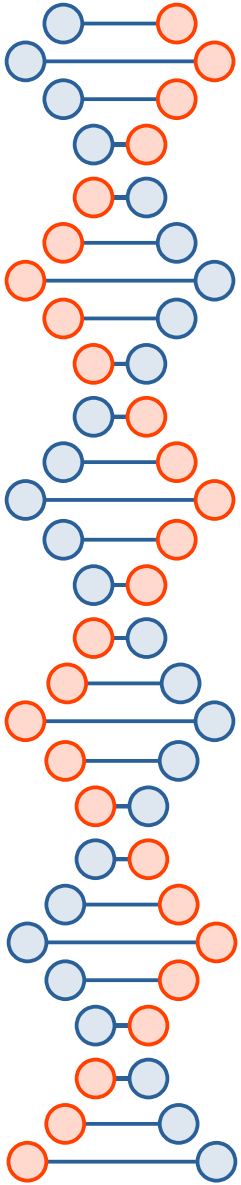


https://en.wikipedia.org/wiki/Deutsch%E2%80%93Jozsa_algorithm#/media/File:Deutsch-Jozsa-algorithm-quantum-circuit.png

Shor’s algorithm

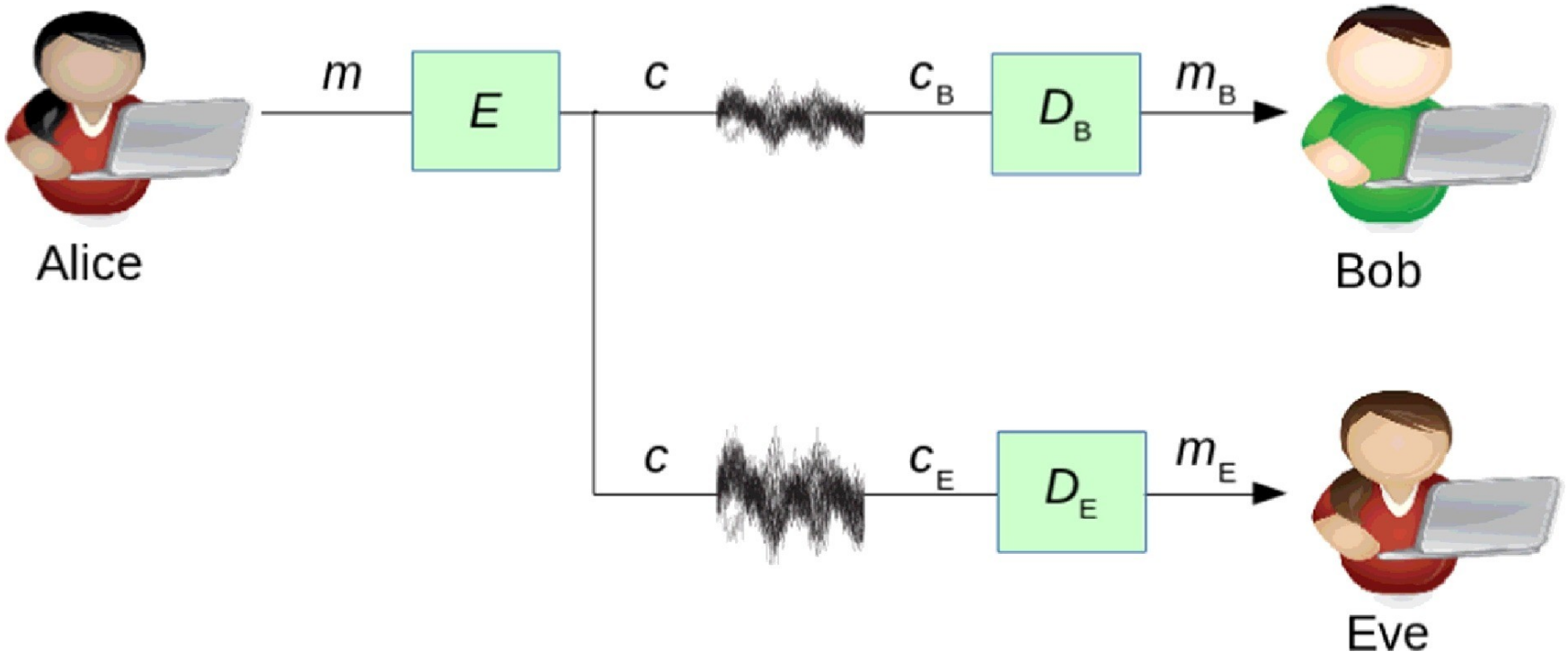


https://en.wikipedia.org/wiki/Shor%27s_algorithm#/media/File:Shor's_algorithm.svg



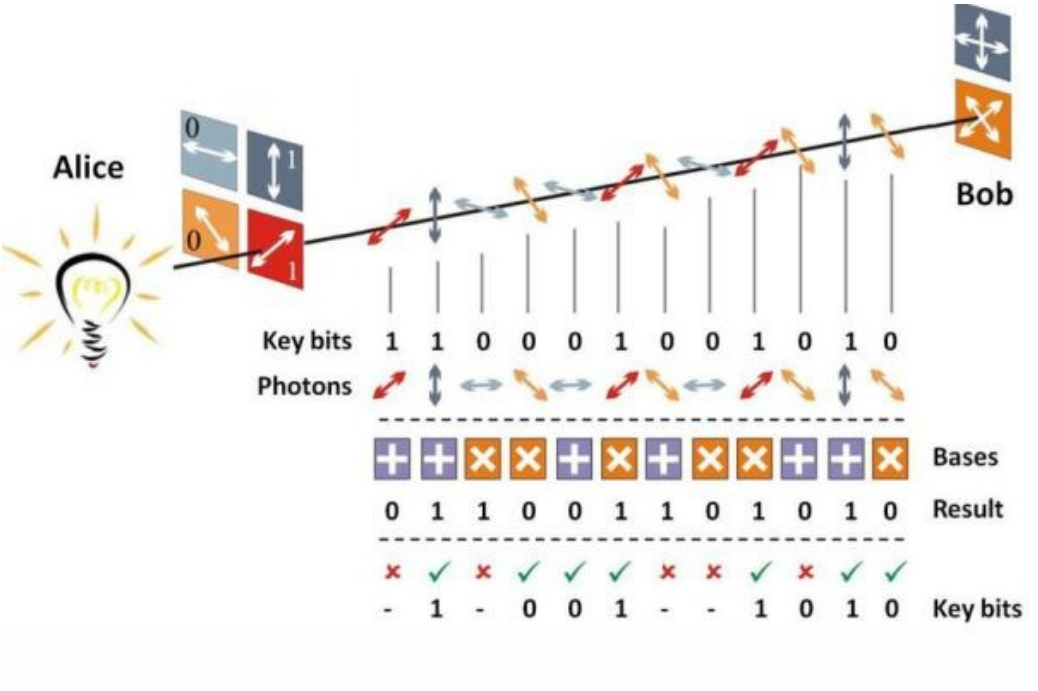
RSA, DH, ECDH, DSA, *etc.* all broken. Need something else instead...

Wiretap channel

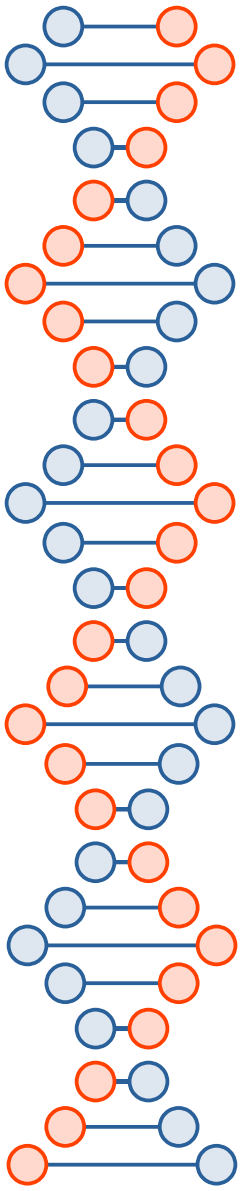


https://www.sciencedirect.com/science/article/pii/S1389128616302146

Quantum Key Distribution

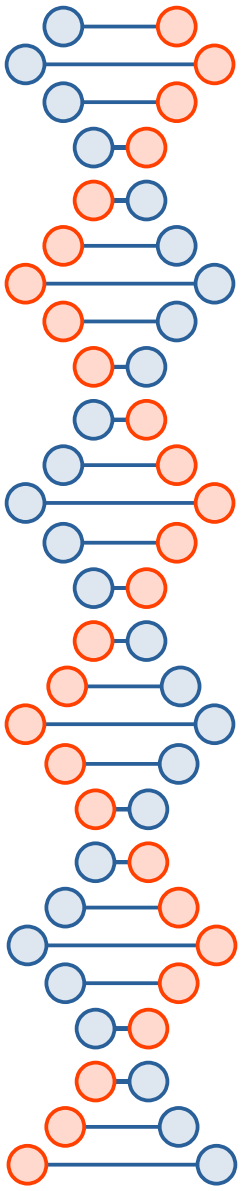


https://imrmedia.in/quantum-key-distribution-test-successfully-demonstrated/



Themes

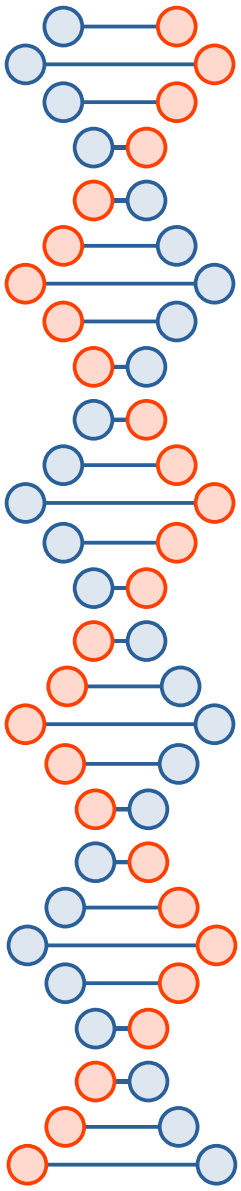
- In schemes based on information theory or physics the eavesdropper has some noise or uncertainty the receiver doesn't have
  - We'll see this in post-quantum crypto (*e.g.*, learning with errors)
- Quantum computers aren't necessarily faster at everything
  - There's usually a "trick at the end" where all the quantum information gets destroyed but the classical information measured still means something



Lamport signature (1979)

- How to sign a 256-bit message digest...
  - Generate 512 random 256-bit integers (256 pairs of them)
    - Private key
  - For all 512 generate corresponding hash
    - Public key (single use)
  - When you want to sign something, reveal one unhashed private version per pair for corresponding to the bit being 0 or 1 (*i.e.*, the first of the pair for 0, the other for 1)
    - 64 Kbits

https://en.wikipedia.org/wiki/Lamport_signature



Watch these three videos...

- https://www.youtube.com/watch?v=_C5dkUiiQnw
- https://www.youtube.com/watch?v=QDdOoYdb748
- https://www.youtube.com/watch?v=K026C5YaB3A