

JPL



QUANTUM ALGORITHMS

Jonathan Dowling & Colin Williams

Quantum Information Project (QuIP)

Information & Computing Technologies Research, Section 365

NASA JET PROPULSION LABORATORY

California Institute of Technology

4800 Oak Grove Drive, Mail Stop 126-347, Pasadena, California 91109-8099

jonathan.p.dowling@jpl.nasa.gov

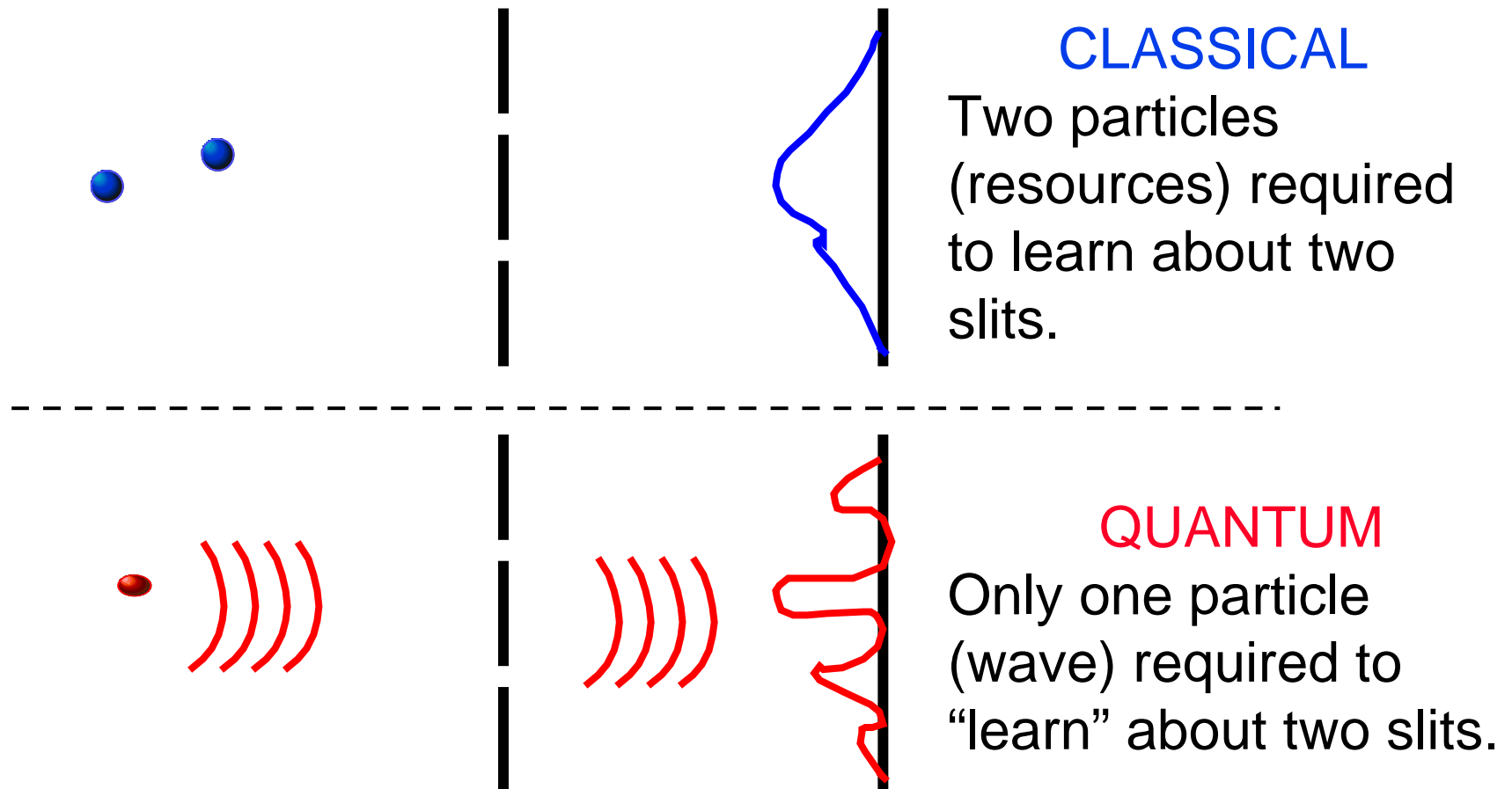
<http://home.earthlink.net/~jpdowling>



- Two-Slits & Quantum Nonlocality
- Traveling Salesmen & Quantum Parallelism
- The Role of Entanglement
- Quantum Factoring
- Bits vs. Qubits & Universality
- Shor's Algorithm For Exponentially Fast Factoring
- Grover's Algorithm for Quadratically Fast Searching
- New Results
- Summary and Confusions



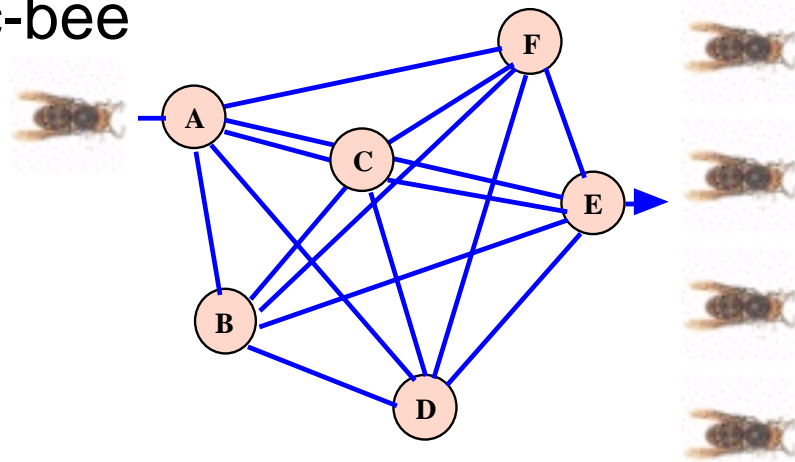
“... a phenomenon which is impossible, absolutely impossible, to explain in any classical way, and which has in it the heart of quantum mechanics. In reality, it contains the only mystery. We cannot make the mystery go away by ‘explaining’ how it works. We will just tell you how it works. In telling you how it works we will have told you about the basic peculiarities of all quantum mechanics.” -Feynman



Traveling SALESMEN & QUANTUM Parallelism



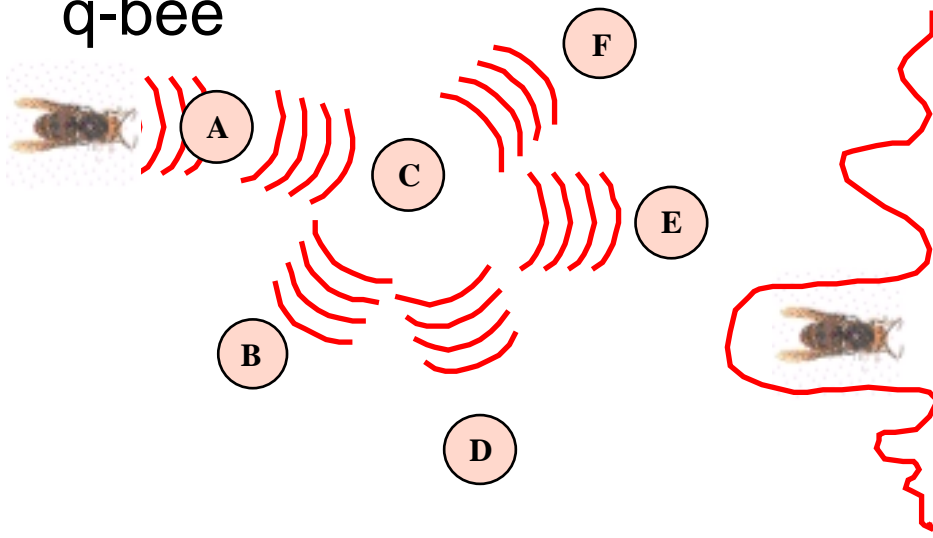
c-bee



CLASSICAL

Exponential number of classical “bees” (particles/resources) needed to examine $N!$ paths.

q-bee



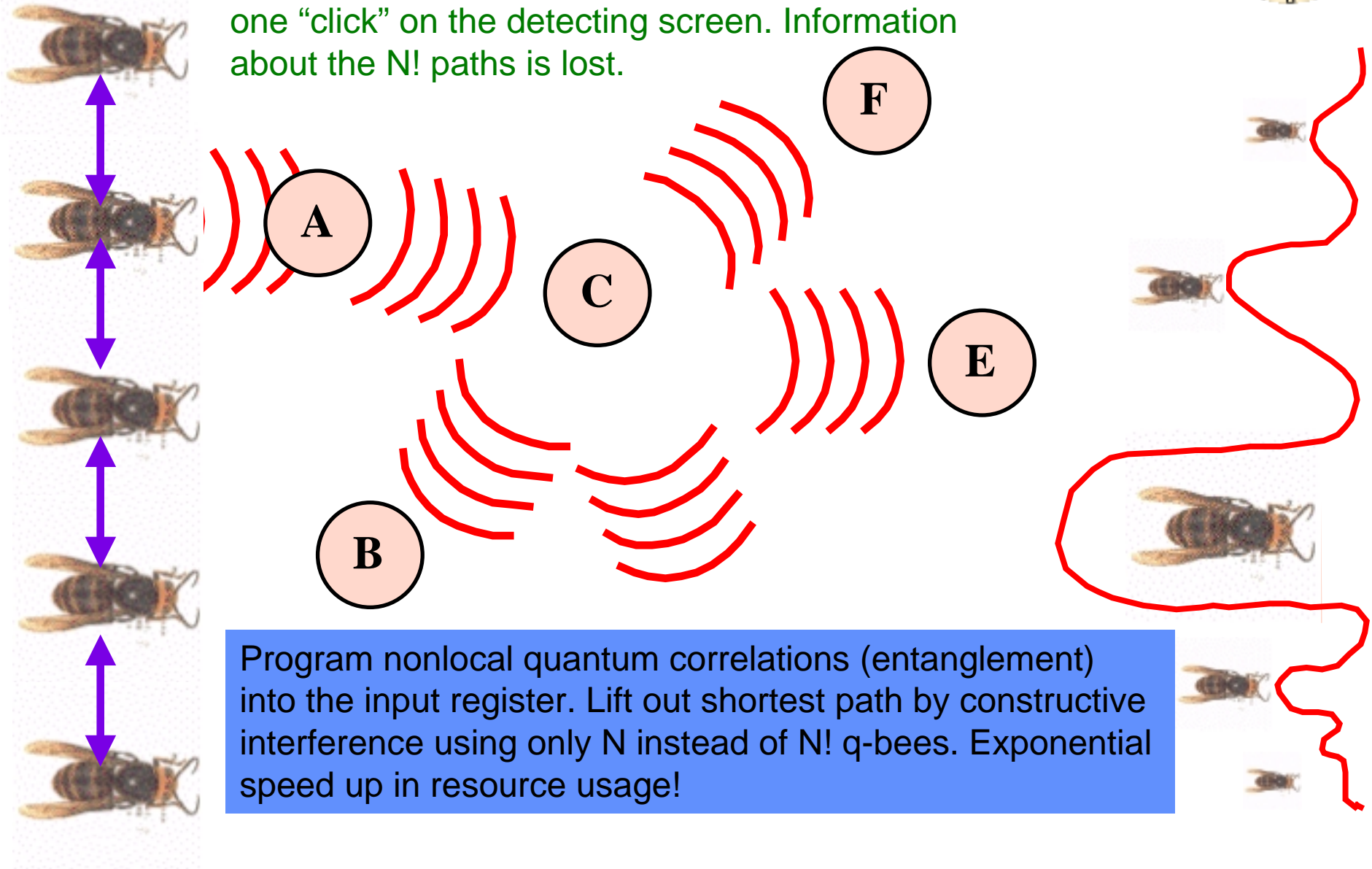
QUANTUM

Only one quantum bee (particle/wave) required to “learn” about $N!$ paths.

THE ROLE OF ENTANGLEMENT



The readout problem: one q-bee makes one "click" on the detecting screen. Information about the $N!$ paths is lost.



Program nonlocal quantum correlations (entanglement) into the input register. Lift out shortest path by constructive interference using only N instead of $N!$ q-bees. Exponential speed up in resource usage!

JPL

QUANTUM FACTORING



$$C = P \cdot Q$$

encrypt decrypt

Security of the public key cryptography depends on the assumed inability of a classical computer to factor C (public key) into P and Q (private key) rapidly. Classical sieving is exponentially slow.

$$C=10^N \rightarrow 10^N \text{ steps}$$

$$O(\exp[(64/9)^{1/3}(\ln N)^{1/3}(\ln \ln N)^{2/3}])$$

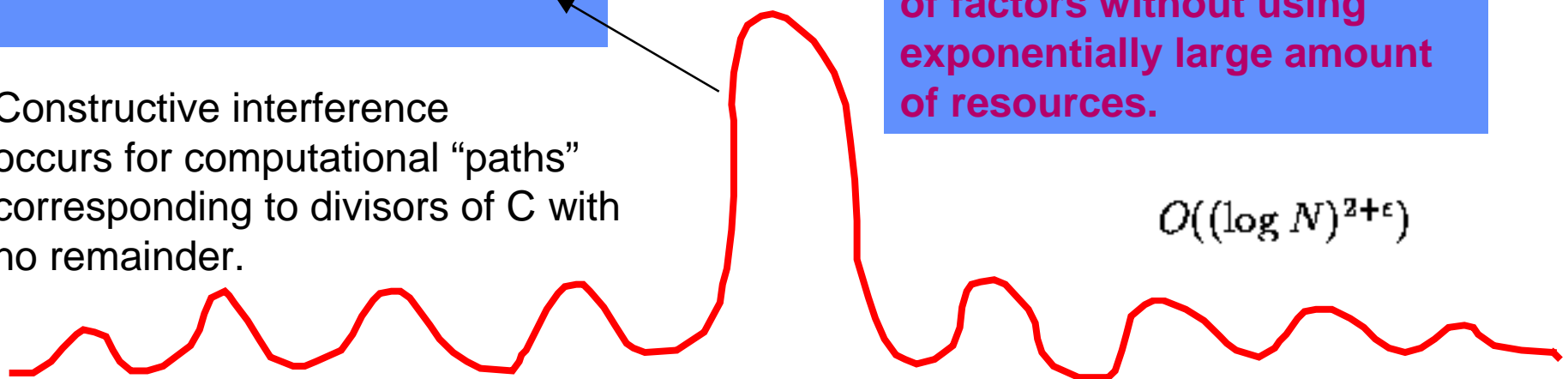
$C/2$ $C/3$ $C/5$ $C/7$ $C/13$ $C/17$...

$$\Psi = \sum a_n | \exp(2 \pi i C/n)$$

A quantum state can test an exponentially large number of factors without using exponentially large amount of resources.

Constructive interference occurs for computational "paths" corresponding to divisors of C with no remainder.

$$O((\log N)^{2+\epsilon})$$



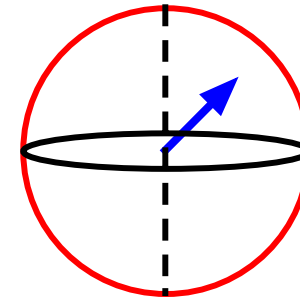


- Qubits can be in a quantum superposition of $\{0\}$ and $\{1\}$.
- The general state can be represented by a vector on the unit sphere.

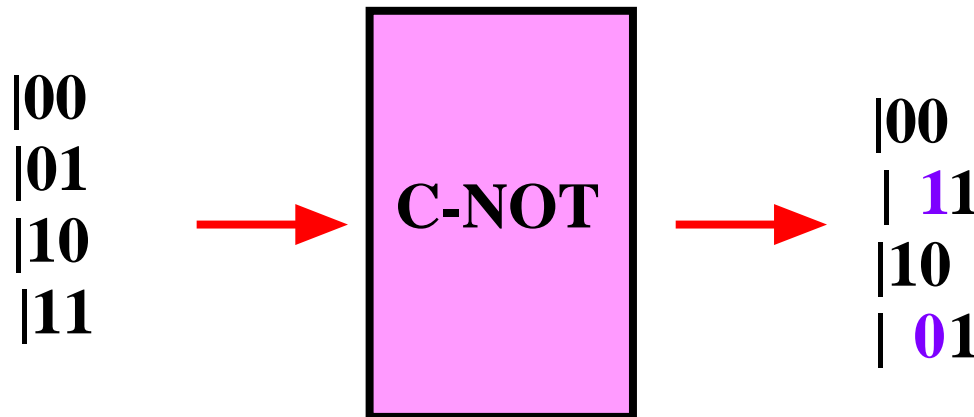
$$U_{\pi}|0\rangle = -|1\rangle, \text{ and } U_{\pi}|1\rangle = |0\rangle$$

$$U_{-\pi/2}|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$ 0\rangle = \downarrow\rangle$
$ 1\rangle = \uparrow\rangle$



A universal quantum computer can be constructed using rotations on Hilbert space (sphere) and a Controlled-NOT gate (XOR).



SHOR'S ALGORITHM: PERIOD OF A SEQUENCE



Sequence:

$$f(0), f(1), \dots, f(q-1)$$

$q \equiv 2^k$

Initial Register:

$$|0; 0\rangle = |\downarrow, \downarrow, \dots; \downarrow, \downarrow, \dots\rangle$$

Initialize Superposition:

$$U_{-\pi/2} \longrightarrow \frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a; 0\rangle$$

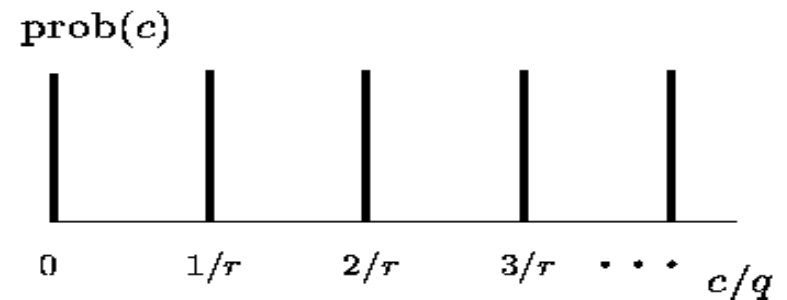
Define Discrete
Fourier transform :

$$|a\rangle \longrightarrow \frac{1}{\sqrt{q}} \sum_{c=0}^{q-1} e^{2\pi i ac/q} |c\rangle$$

Apply Transform:

$$\longrightarrow \frac{1}{q} \sum_{a=0}^{q-1} \sum_{c=0}^{q-1} e^{2\pi i ac/q} |c; f(a)\rangle$$

Constructive interference peaks
at multiples of the inverse
period of the sequence $1/r$.



SHOR'S ALGORITHM: FACTORING NUMBERS



We wish to factor the number N . First, select a number x , and consider the sequence.

$$f(a) = x^a \pmod{N}$$

$$1, x, \dots, x^{r-1}, x^r, x^{r+1}, \dots$$

$$\underbrace{1, x, \dots}_{r\text{-terms}} \quad \underbrace{1, x, \dots}_{r\text{-terms}} \quad \underbrace{1, x, \dots}_{r\text{-terms}}$$

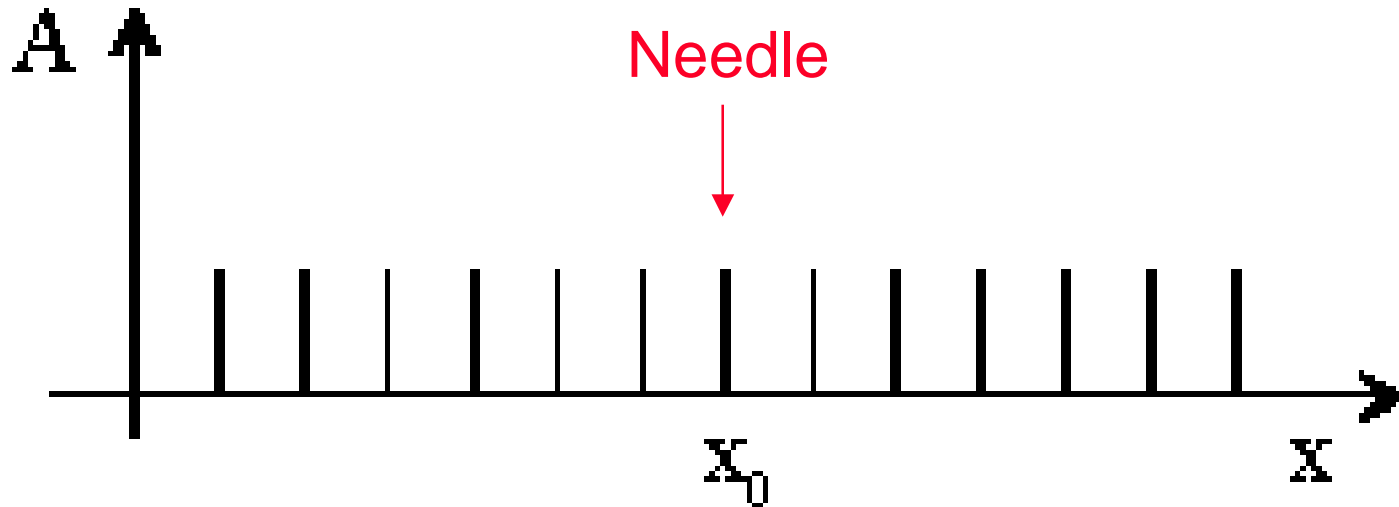
Choose x such that r is even.

$$x^r \equiv 1 \pmod{N}$$

Factor, mod N .

$$(x^{r/2} + 1)(x^{r/2} - 1) \equiv 0 \pmod{N}$$

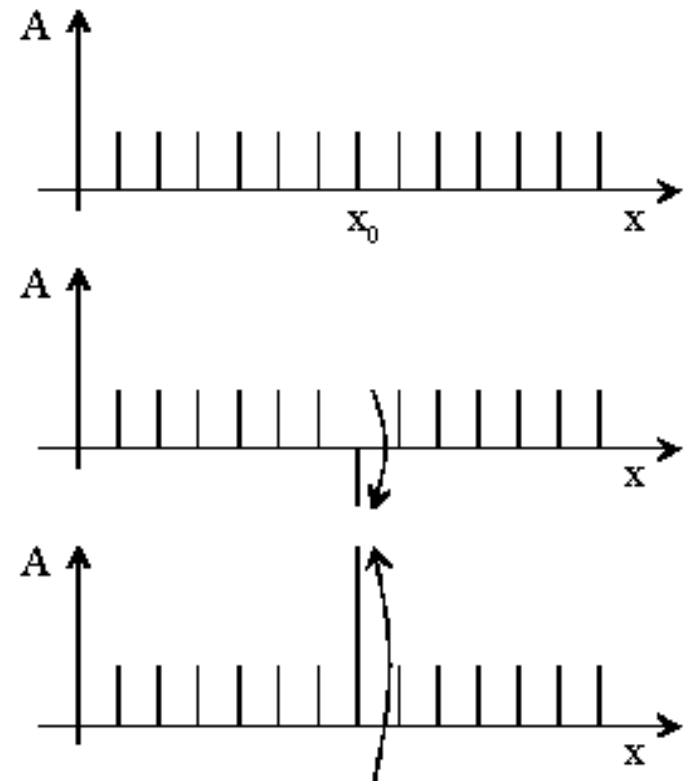
One or the other of these terms must have a factor in common with N . The final step in the algorithm then is to calculate the greatest common divisor of these terms individually with N ; any non-trivial common divisor will be a factor we have sought, thus completing our search. Number of steps scales (quantum probabilistic) polynomial in N .



Best classical algorithm: Start at one end and prick your finger with one straw at a time until you yell "OUCH!" at the needle. Sometimes you'll be lucky and the needle will be near the beginning, sometimes not and it will be near the end. On average, $N/2$ tries will be needed to find it. Best you can do.



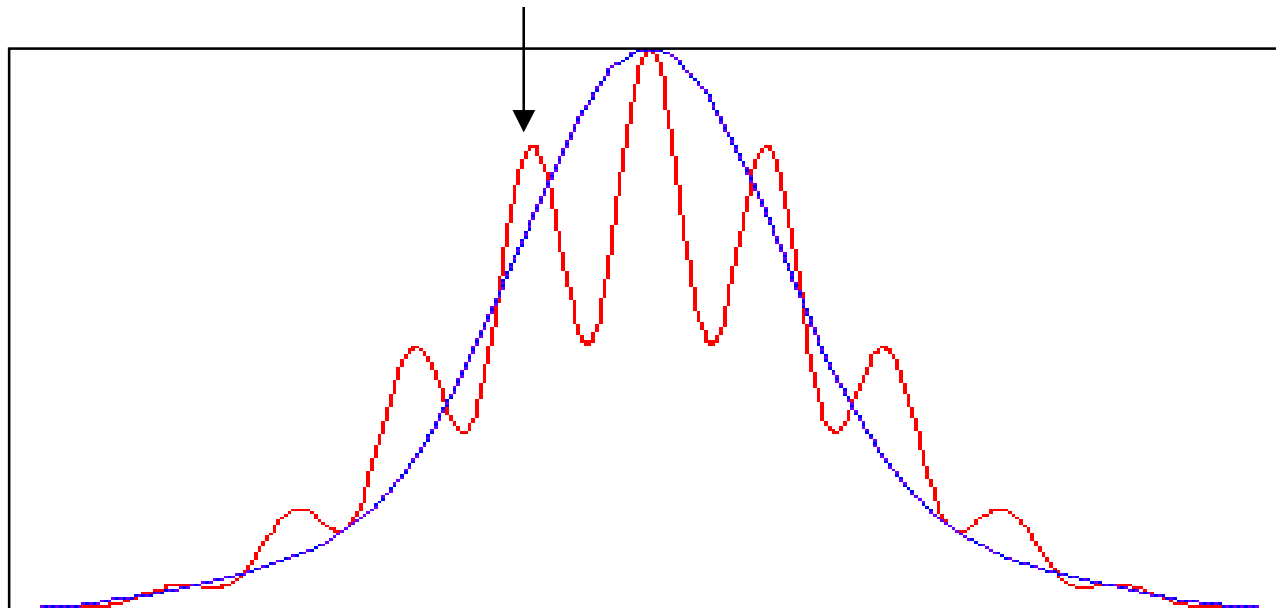
- **Goal: find $x_n \mid f(x_n) = 1, f(x_m) = 0$**
- Initialize L bit registers
- Prepare superposition of states
- Apply operator that rotates phase by π if $f(x) = 1$
- Invert about average
- Repeat $O(N^{1/2})$ times
- **Measure state**



Note from classical antenna theory: Power from an unphased array of N antennas scales like N , but when phased scales like N^2 .



Prob



Number of Iterations

The probability amplification unitary operator has an exact form in terms of Chebyshev Polynomials, allowing you to stop at fewer than $\text{Sqrt}[N]$ iterations, with high probability of success. On average, can do better than Grover's result.



- Traveling Salesmen Still Unsolved.
- Entanglement in Mixed States Not Understood.
- Need New (Useful) Algorithms
- Need Hardware (Kimble's Talk)
- What Are the Limitations?
- What Can Be Done on "Analog" Quantum Computers
- What Can be Done without Universality
- Tailor Algorithm Development to Hardware