quantum bit = qubit  tensor value in \( \mathbb{C}^2 = \mathbb{C} \cdot V_s + \mathbb{C} \cdot V_i \)

\[ V_s = |0\rangle, \quad V_i = |1\rangle = \text{orthogonal basis vectors.} \]

quantum state space of a qubit = \( \mathbb{C}^2 \) = tensor product over \( n \) of state of \( n \) copies of one qubit.

Basis of \( \mathbb{C}^2 \): \( \{ v_{b_1, b_2, \ldots, b_n} = v_{b_1} \otimes v_{b_2} \otimes \ldots \otimes v_{b_n} \} \)

Each \( v_{b_i} \) = \( \text{basis vector for } i \text{th qubit} \)

In fact, classical computers = binning string \( S \) of length \( k \). (each term in string is a \( 0 \text{ or } 1 \).)

Make this into classical output of the state \( V_{s, 0_1, 0_2, \ldots} \).

The less mixed product in \( \mathbb{C}^n \) by \( \langle \alpha \mathbf{V}_0 + \beta \mathbf{V}_1, \gamma \mathbf{V}_0 + \delta \mathbf{V}_1 \rangle = \alpha \gamma + \beta \delta \).

Extend this to \( \mathbb{C}^2^n \) by \( \langle \mathbf{V}_{b_1, b_2, \ldots, b_n}, \mathbf{V'}_{b'_1, b'_2, \ldots, b'_n} \rangle = \sum_{b_1, b_2, \ldots, b_n} \langle v_{b_1}, v'_{b'_1} \rangle \langle v_{b_2}, v'_{b'_2} \rangle \langle v_{b_3}, v'_{b'_3} \rangle \ldots \langle v_{b_n}, v'_{b'_n} \rangle \).

\( \langle \alpha \mathbf{w}, \beta \mathbf{w} \rangle = \alpha \beta \langle \mathbf{w}, \mathbf{w} \rangle \).

\[ \| \mathbf{w} \| = \sqrt{\langle \mathbf{w}, \mathbf{w} \rangle} \]

Quantum states don't change under unitary, orthogonal \( U \in \mathbb{C}^2 \), or \( \mathbb{C}^n \).

can now solve from here to have \( \| \mathbf{w} \| = 1 \).

\( \mathbf{w} = \sum_{s \in \mathcal{S}_x} w_s v_s \) where \( v_s = |s\rangle \).

\( \sum_{s \in \mathcal{S}_x} |s\rangle = \mathbb{1} \).

\( \sum_{s \in \mathcal{S}_x} |s\rangle \langle s| = \mathbb{1} \).

\( w_s = |\text{probability amplitude of being in state } s \rangle \).

Measure what \( \mathcal{S}_x \) in \( \mathbf{w} \) produces state \( V_s \) with probability \( |w_s|^2 \).
Clasical gates:

\[ \text{AND} = \land \]

\[ \text{or} \]

\[ \text{or} \]

\[ \text{Not} = \neg \]

\[ \text{Possibilities} \]

\[ \text{Output} = 0 \]

\[ \text{Input} \]

Quantum wires correspond to one of the n qubits.

Each gate acts on one or 2 wires.

Transformation done by the gate = unitary transformation of corresponding state space of 1 or 2 qubits.

For one qubit gate, put 2x2 unitary matrix.

For 2 qubits, use tensor product of 2x2 matrices.

A gate gives a unitary transformation on a state space of n qubits by taking tensor product of given transform on 2 qubit space with identity on the other qubit spaces.
**CNOT quantum gate:** Negate second qubit iff first is 1.

\[ V_{z} \rightarrow V_{z'} \quad \text{when} \quad x, y, z \in \{0, 1\}, \quad z = x \land y \lor \neg z \]

On basic states:

- \[ V_{0} \otimes V_{0} \]
- \[ V_{0} \otimes V_{1} \]
- \[ V_{1} \otimes V_{0} \]
- \[ V_{1} \otimes V_{1} \]

**Matrix**

\[
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{pmatrix}
\]

Can make all quantum circuits from 1 bit gates + CNOT gates.

**Toffoli gate:** Negate 3rd bit iff first two are 1.

\[ V_{0} \otimes V_{0} \otimes V_{0} \]

**BG P =** Problems for an answer can be computed on a quantum computer in poly time, with correct answer \( \frac{2}{3} \) of time.

So: Need a Turing machine which on input \( n \) gives in poly time a description of the circuit for \( n \) inputs including the matrices of the quantum gates.
Can restrict to problems where output answer is \[\text{Yes or No}.\]

Quantum computer should find right state \[2/3\] of time for all inputs (of any length).

\[
\text{Physics: Spins = states of } n \text{ qubits}
\]

\[
\phi^* \text{ partial}
\]

Factoring

Factoring \(N\) by number field sieve takes \(O\left(\exp\left(\sqrt[3]{\log N}^{\frac{2}{3}}\right)\right)\) steps, if \(L = \#\text{ bits in } N \sim \log N\).

Quantum method involves finding \(\text{ord}_N(x)\) for \(x \in \mathbb{Z}/N\).

Suppose \(x^2 \equiv t^2 \mod N\), \(x \neq \pm t \mod N\). Then

\[
(s+t)(s-t) \\equiv 0 \mod N,
\]

\[
1 < \text{gcd}(s+t, N) = d = \text{proper divisor of } N
\]

\[
\text{Lemma: If } N \text{ not a prime } n. \text{ Then for } r = \frac{1}{2} \forall \text{ all } x \mod N \in \mathbb{Z}/N^*, \text{ if } r = \text{ord}_N(x) \text{ then } 2(r \mod x^{r/2} - t = 1 \mod N). \text{ So } s = x^{r/2}, t = 1 \text{ give}
\]

\[
1 < \text{gcd}(x^{r/2} + 1, N) = d | N \text{ and } d \neq N.
\]
So take $x$ at random mod $N$. Compute $q = d(x, N)$. If this is 1, then there is $\frac{1}{2}$ chance of getting a factor if you can find $x = d_0y(x)$. Do this for lots of $x$.

**Quantum Fourier Transform:**

$L$ qubits

[basis]

$V_a = \text{state } a = b_1 b_2 \ldots b_{L-1}$, thought of as integer $0 \leq a < 2^L$

(by binary expansion)

$$V_a \rightarrow \frac{1}{2^{L/2}} \sum_{b=0}^{2^{L/2} - 1} \exp\left(\frac{2\pi i a b}{2^L}\right) V_b \quad \text{for } 0 \leq a < 2^{L-1}$$

This is unitary:

$$\langle f(V_a), f(V_{a'}) \rangle = \frac{1}{2^L} \sum_{b=0}^{2^L - 1} e^{2\pi i a b/2^L} \hat{f}(V_b) \hat{f}(V_{b'}) = \frac{1}{2^L} \delta\left(\sum_{b=0}^{2^L - 1} e^{2\pi i a b/2^L} - 2\pi i a b\right) = \begin{cases} 1 & a = a' \\ 0 & a \neq a' \end{cases}$$

**Hard part:** Decompose $f$ into sequence of $2^{(L-1)/2}$ one + two bit quantum gates.
Factoring.

\[ N = \text{an } 2^L \text{-bit no.} \]

Make a quantum computer with 2L qubits in 2^{2L} registers. 2L qubits in \( 2^{2L} \) + some constant \( \# \) for work space.

Step 0: \( \text{Input } V_0 = \frac{1}{\sqrt{2}} \rho \). Put computer into state which is superposition of all possible values of 1st register and state \( V_0 = \frac{1}{\sqrt{2}} \rho \).

Step 1: Put computer into state which is superposition of all \( 2^{2L-1} \) qubits \( \frac{1}{2^{2L-1}} \sum_{a=0}^{2^{2L-1}} V_a \otimes V_0 \).

Do this by \( 2L \) gates, with each one taking \( 10 \) qubit in first register into state

\[ \frac{1}{\sqrt{2}} \left( |0\rangle + |1\rangle \right) \]

Step 2: Build a circuit which takes \( V_a \) in first register to \( x^a \mod N \) in second, sending computer to the state

\[ \frac{1}{2^L} \sum_{a=0}^{2^L-1} V_a \otimes V_{x^a \mod N} \]
Points: Each \( a = b_i \cdot b_k \) gives key bits.

Use binary exponentation to find \( x^a \mod N \)

\[
2_0 = x^2 \mod N
\]
\[
2_1 = x^{2^1} \mod N, \quad \text{successive products mod } N.
\]

There's a quantum algorithm for doing this.

**Partial products:**

\[
\begin{align*}
2^0 & = 1 \\
2^1 & = x \\
2^2 & = x^2 \\
& \vdots \\
2^k & \equiv x^a \mod N
\end{align*}
\]

Defined up a quantum gate which carries out this, classical comp
(gives correct value to only \( V_0 \))

(based on

**Step 3:** Do \( \text{if-} \) Fourier Transform on first register to get state

\[
\frac{1}{\sqrt{2^k}} \sum_{a=0}^{2-1} \sum_{h=0}^{2^k-1} \exp \left( 2\pi i ab / 2^k \right) V_b \otimes V_{a \mod N}.
\]

**Step 4:** Measure state. Output post \( V_b \otimes V_{x \mod N} \) is

\[
\frac{1}{2} \sum_{a=0}^{2^k-1} \exp \left( 2\pi i ab / 2^k \right)
\]

Note

Happened in class! 

[Diagram of quantum circuit]
\begin{align*}
\sum_{a \equiv j \mod d} \exp \left( \frac{2\pi i a b}{2^L} \right) \\
&= \exp \left( \frac{2\pi i j b}{2^L} \right) \sum_{\omega = 0}^{2^L - 1} \exp \left( \frac{2\pi i \omega b}{2^L} \right) \\
\text{In the value this is} \\
\left| \sum_{\omega = 0}^{2^L - 1} \omega^j \right| &= \frac{\left[ \frac{2^L - 1}{2^L} \right] + 1}{2^L - 1} \\
\text{where} \quad j &= \frac{2\pi i b}{2^L} \\
\text{Note} \quad j &= \frac{2^L - 1}{2^L} \quad \frac{2^L}{2^L - 1} \\
\text{The probability of seeing the state} \quad \nu_x \otimes \nu_{x^d \mod N} \quad \text{is} \\
\left| \sum_{\omega = 0}^{2^L - 1} \omega^j \right|^2 \\
\left| \sum_{\omega = 0}^{2^L - 1} \omega^j \right|^2 &= \frac{\left[ \frac{2^L - 1}{2^L} \right] + 1}{2^L - 1} \\
\text{Show prove that if this is large.} \quad \frac{1}{2^L - 1} \quad \text{for some integer} \quad d. \\
\text{Show proved elsewhere that if} \quad \left| \frac{\omega}{2^L} - \frac{d}{r} \right| \quad \leq \frac{1}{2^L} \\
\text{Then the probability is at least} \quad \frac{2^L - 1}{\pi^2} > \frac{1}{3^4}.
\end{align*}
(You'd expect something like this, since \( S = 1 \),
\[
\frac{1}{\frac{1}{2 \sqrt{c}}} \sim \frac{1}{\tau^2}.
\]

Now suppose we observe this large probability \( \tau \), so we know \( \tau \leq 2 \leq 2^L \), and we want to find \( \nu \), where
\[
\gamma \leq 2 \nu \sim \nu.
\]

Since
\[
\left| \frac{b}{2^{2L}} - \frac{d}{\nu} \right| \leq \frac{1}{2 \cdot 2^{2L}} \leq \frac{1}{2\tau^2},
\]

we know \( \frac{d}{\nu} \) is a convergent \( \tau \) and we can find this quickly from \( b \) and \( 2^{2L} \).

Then we get \( \nu \).

**Theorem:** If \( \left| \frac{b}{2^{2L}} - \frac{d}{\nu} \right| < \frac{1}{2 \tau^2 \cdot 2^{2L}} \)

Then with probability that the state \( V \otimes V \) is observed is \( \frac{1}{2 \tau^2} \) and we can use this observation as above to find \( \tau \).
Next we can use our observed states of this kind will let us find \( r \). For a prime to \( r \), there is a unique \( \frac{b}{2^k} \) with \( 0 \leq b < 2^k \) and

\[
\left| \frac{b}{2^k} - \frac{d}{r} \right| < \frac{1}{2 \cdot 2^k} \quad \text{Gaps in between}
\]

\[
\text{ults. of } \frac{1}{2^k} \text{ are}
\]

\[
\text{of size } \frac{1}{2^k}
\]

So we get \( \Psi(r) \) possible \( b^k \)'s, and we can let \( j \) vary from 0 to \( j-1 \). So there are

\[ \Psi(r) \cdot r \text{ states } \forall b \in \mathbb{Z}/N \text{ which will lead to } \]

\[ \left| \frac{b}{2^k} - \frac{d}{r} \right| < \frac{1}{2 \cdot 2^k} \quad \text{and the determination of } r = \Theta(N) \]

Each one has probability of being observed of \( \geq \frac{1}{3 r^2} \), so the probability one will occur is \( \Psi(r) \cdot r > \frac{5}{3 r^2} = \frac{5}{3 \log \log r} \).

Where \( \delta > 0 \) is absolute. Now in

\[ O \left( \log \log r \right) < O \left( \log \log N \right) \] steps we should have a \( \Theta \) possible constant and finding \( r \).
\[ a \to x^a \mod N. \]

Periodic function of \( a \) of period \( r = \text{ord}_N(x) \),

\[ \log_2 L = \text{the binary digits in } N. \]

\[ N < 2^L \]

\[ a = 0, 1, \ldots, 2^L - 1 \]

\[ N^2 < 2^{L-1} \]

\[ f \in \text{Map}(\mathbb{Z}/2^L, C) \]

\[ e_j : \quad a \to e^{2\pi i a / 2^L} \]

\[ f = \sum_{j=0}^{2^L-1} f(j) e_j \]

If \( f \) has period \( r \), the quotients \( \hat{f}(a) \) are assigned

to \( e_j \) of period nearly \( r \).

\[ \frac{2^L}{r} \sim d \text{ integers} \]

\[ \left( \frac{j}{2^L} \right) \sim \frac{d}{r} \]

If we know \( \frac{j}{2^L} \)

Then \( \frac{d}{r} \) should be a

convergent to continued form of \( \frac{j}{2^L} \)