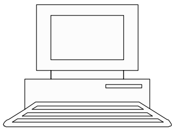
Shor's Algorithm

Peter Shor
AT&T Labs

Motivation

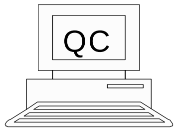
- It appears that the universe in which we live is governed by quantum mechanics
- Quantum information theory gives us a new avenue to study & test quantum mechanics
- Why do we want to build a quantum computer?

Why build a classical computer?

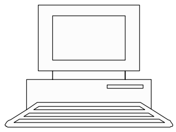


- They are able to perform calculations many orders of magnitude faster than can be done with pencil and paper.

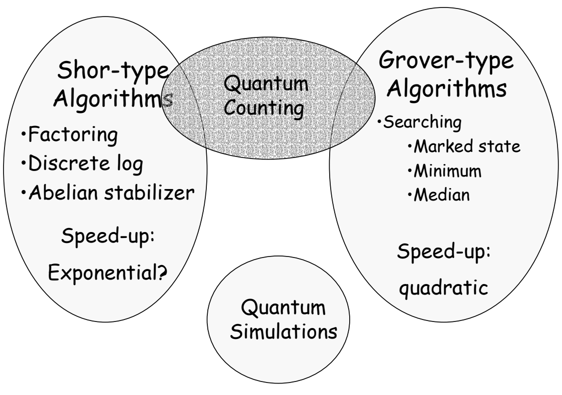
Why build a quantum computer?



- They should be able to perform calculations many orders of magnitude faster than can be done on a classical computer.



Quantum Algorithms



Overview

- Shor's factoring algorithm
  - Phase estimation algorithm
    - Quantum Fourier transform
    - Hadamard gate
    - Controlled-U gate
  - Equivalence of factoring and order finding
  - Solving order finding using PE
- Summary

Discrete Fourier Transform

- Given a sequence of N complex numbers,

$$x_0, x_1, \ldots x_{N-1}$$

- The DFT produces another sequence,

$$y_0, y_1, \ldots y_{N-1}$$

- where

$$y_k \equiv \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{2\pi ijk/N}$$

Discrete Fourier Transform

$$y_k \equiv \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \omega^{jk} \qquad \omega \equiv e^{2\pi i/N}$$

- It is not hard to show that the transform

$$x_j \equiv \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} y_k \omega^{-jk}$$

returns the original sequence.

Exercise: Verify the formula for x_j

Discrete Fourier Transform

- If we let x and y be N-by-1 vectors, then

$$y = Dx \qquad \text{and} \qquad x = D^{-1}y$$

- where

$$D = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & \\ 1 & \omega & \omega^2 & \omega^3 & \omega^4 & \\ 1 & \omega^2 & \omega^4 & \omega^6 & \omega^8 & \cdots \\ 1 & \omega^3 & \omega^6 & \omega^9 & \omega^{12} & \\ 1 & \omega^4 & \omega^8 & \omega^{12} & \omega^{16} & \\ & & \vdots & & & \ddots \end{bmatrix} \qquad D^{-1} = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & \\ 1 & \omega & \omega^{-2} & \omega^{-3} & \omega^{-4} & \\ 1 & \omega^{-2} & \omega^{-4} & \omega^{-6} & \omega^{-8} & \cdots \\ 1 & \omega^{-3} & \omega^{-6} & \omega^{-9} & \omega^{-12} & \\ 1 & \omega^{-4} & \omega^{-8} & \omega^{-12} & \omega^{-16} & \\ & & \vdots & & & \ddots \end{bmatrix}$$

- By inspection, $D^{-1} = D^{\dagger}$

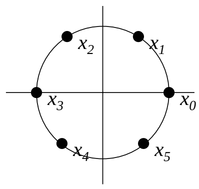
Discrete Fourier Transform

- Suppose

$$x_j = \frac{1}{\sqrt{N}} e^{\frac{2\pi\, ijk}{N}} \qquad k \in \{0, N-1\}$$

- Then

$$y_j = \partial_{j-k}$$



Exercise: Verify the formula for y_j

Quantum Fourier Transform

- The quantum Fourier transform is a DFT of the amplitudes of a quantum state.
- Suppose we have some state,

$$|\psi\rangle = x_0|0\rangle + x_1|1\rangle + \ldots + x_{N-1}|N-1\rangle$$

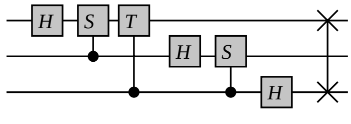
- The quantum Fourier transform produces the state

$$|\chi\rangle = y_0|0\rangle + y_1|1\rangle + \ldots + y_{N-1}|N-1\rangle$$

$$y = Dx$$

Quantum Fourier Transform

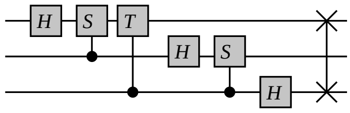
- The QFT
  - is unitary ✓
  - can be implemented very efficiently
- An example:



$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \qquad T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$$

Quantum Fourier Transform

$$|\psi\rangle = x_0|000\rangle + x_1|001\rangle + x_2|010\rangle + x_3|011\rangle$$
$$+ x_4|100\rangle + x_5|101\rangle + x_6|110\rangle + x_7|111\rangle$$

$|\psi\rangle \rightarrow$  $\rightarrow |\chi\rangle$

$$|\chi\rangle = y_0|000\rangle + y_1|001\rangle + y_2|010\rangle + y_3|011\rangle$$
$$+ y_4|100\rangle + y_5|101\rangle + y_6|110\rangle + y_7|111\rangle$$

$$y = Dx$$

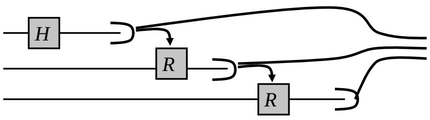
Quantum Fourier Transform

- In general,to perform the QFT on n qubits requires $O(n^2)$ one and two qubit gates
  - Reference: Cleve et al. (quant-ph/9708016)
- Transforming 2^n amplitudes with only n^2 operations
- The fastest we can do classically is $n2^n$
- However, QFT does not allow us to improve classical Fourier transforms
- There is no efficient way to extract the amplitudes of the state

$$|\chi\rangle = y_0|000\rangle + y_1|001\rangle + y_2|010\rangle + y_3|011\rangle$$
$$+ y_4|100\rangle + y_5|101\rangle + y_6|110\rangle + y_7|111\rangle$$

Quantum Fourier Transform

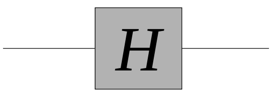
- Performing a QFT directly followed by a measurement is very easy
- In fact, if you wish to measure directly after applying the QFT, you only need n single qubit rotations!



Overview

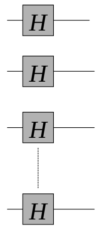
- Shor's factoring algorithm
  - Phase estimation algorithm
    - Quantum Fourier transform ✓
    - **Hadamard gate**
    - **Controlled-U gate**
  - Equivalence of factoring and order finding
  - Solving order finding using PE
- Summary

Hadamard gate

$|0\rangle$  $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Hadamard gate



$|0\rangle$ — $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$

$|0\rangle$ — $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$

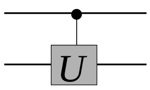
$|0\rangle$ — $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$

$|0\rangle$ — $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$

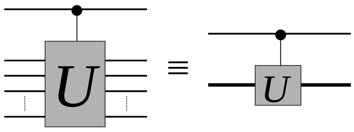
$$|0\rangle \longrightarrow \frac{1}{\sqrt{2^m}}\sum_{x=0}^{2^m-1}|x\rangle$$

Controlled-U gate

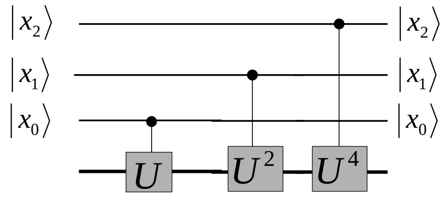
- Two-qubit controlled-U



- Multi-qubit controlled-U



Controlled-U gate



$$U^{x_2 x_1 x_0}$$

Phase estimation algorithm

- Given a unitary operator and an eigenstate of the operator
- The goal of the PE algorithm is to find the corresponding eigenvalue

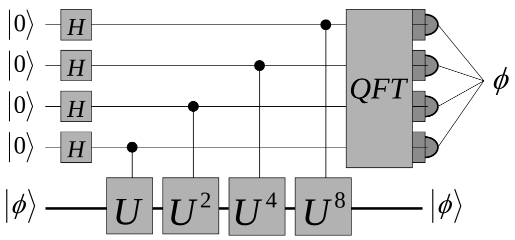
Phase

$$\hat{U}\,|\phi\rangle = e^{i\phi}|\phi\rangle$$

Phase estimation algorithm

- The PE algorithm uses two registers of qubits
  - The target register, to which U can be applied
  - The index register, which will be used to store the eigenvalue of U

Phase estimation algorithm



Quantum circuit diagram

Phase estimation algorithm

- We initially start with the system in the state

$$|0\rangle|\phi\rangle$$

- Performing the Hadamard gates on the index register creates the state

$$\frac{1}{\sqrt{2^m}}\sum_{x=0}^{2^m-1}|x\rangle|\phi\rangle$$

- Performing the series of controlled-U gates gives

$$\hat{U}^{\hat{x}}\frac{1}{\sqrt{2^m}}\sum_{x=0}^{2^m-1}|x\rangle|\phi\rangle$$

Phase estimation algorithm

- We can move the U inside the summation

$$\frac{1}{\sqrt{2^m}}\sum_{x=0}^{2^m-1}|x\rangle\hat{U}^x|\phi\rangle$$

- And replace U with $e^{i\phi}$

$$\frac{1}{\sqrt{2^m}}\sum_{x=0}^{2^m-1}|x\rangle e^{ix\varphi}|\phi\rangle$$

Phase estimation algorithm

- Rearranging,

$$|\phi\rangle\frac{1}{\sqrt{2^m}}\sum_{x=0}^{2^m-1}e^{ix\varphi}|x\rangle \quad \text{if} \quad \phi=\frac{2\pi k}{2^m}$$

then $$|\phi\rangle\frac{1}{\sqrt{2^m}}\sum_{x=0}^{2^m-1}e^{\frac{2\pi ixk}{2^m}}|x\rangle$$

Applying the quantum Fourier transform gives

$$|\phi\rangle|k\rangle$$

Phase estimation algorithm

- Generally, k will not be an integer
- With high probability we will obtain the nearest integer to k
- Thus, we have an m-bit approximation to ϕ.

RSA encryption

- Named after Rivest, Shamir and Adleman, who came up with the scheme

$$m_1 \times m_2 = N$$

Primes

- Based on the ease with which N can be calculated from m_1 and m_2
- And the difficulty of calculating m_1 and m_2 from N

RSA encryption

- N is made publicly available, and is used to encrypt data
- m_1 and m_2 are the secret keys which enable you to decrypt the data
- To crack the code, a code-breaker needs to factor N
- Best current cracking method on a classical computer
  - Number field sieve
  - Requires $\exp(O(n^{1/3}\log^{2/3} n))$
  - n is the length of N

A little number theory

$$m_1 \times m_2 = N$$

Modular Arithmetic

$$a \equiv b \bmod N$$

Simply means

$$a = b + kN$$

k is any integer

and $b < N$

Smallest

$$a^r \equiv 1 \bmod N$$

Co-prime

$$\gcd(a, N) = 1$$

Greatest Common Divisor

No factors in common!

A little number theory

$$m_1 \times m_2 = N \quad \Leftrightarrow \quad a^r \equiv 1 \bmod N$$

Consider the equation

$$y^2 \equiv 1 \bmod N$$
$$y^2 - 1 \equiv 0 \bmod N$$
$$(y+1)(y-1) \equiv 0 \bmod N$$
$$(y+1)(y-1) = kN$$

A little number theory

$$m_1 \times m_2 = N \quad \Leftrightarrow \quad a^r \equiv 1 \bmod N$$

$$(y+1)(y-1) = km_1m_2$$

$$\gcd(y+1, N) = N$$
$$\gcd(y-1, N) = 1$$

Trivial solutions

$$\gcd(y+1, N) = m_1$$
$$\gcd(y-1, N) = m_2$$

- gcd can be calculated very efficiently
- Euclid's algorithm
- 300 BC

A little number theory

$$m_1 \times m_2 = N \quad \Leftrightarrow \quad a^r \equiv 1 \bmod N$$

$$y^2 \equiv 1 \bmod N$$

- If we can find r
- → And the r is even
- Then

$$m_1 = \gcd(a^{r/2} + 1, N)$$
$$m_2 = \gcd(a^{r/2} - 1, N)$$

- → Provided we don't get trivial solutions

A little number theory

$$m_1 \times m_2 = N \quad \Leftrightarrow \quad a^r \equiv 1 \bmod N$$

- What about the ifs and buts ?!?

Theorem:

Let $N = m_1 m_2$, where m_1 and m_2 are prime numbers not equal to 2. Suppose a is chosen at random from the set $\{a : 1 < a < N, \gcd(a,N) = 1\}$. Let r be the order of y mod N. Then the probability

$$\text{Prob}(r \text{ is even and non-trivial}) \geq \frac{1}{2}$$

Proof: long, boring and complicated

A little number theory

$$m_1 \times m_2 = N \quad \Leftrightarrow \quad a^r \equiv 1 \bmod N$$

- Finding r is equivalent to factoring N
- Why can't we use a classical computer to find r?
  - It takes $O(2^n)$ operations

Exercise: Using the reduction of factoring to order-finding, and the fact that 10 is co-prime to 21, factor 21

Choosing a U

- Consider the operator,

$$\boxed{a^r \equiv 1 \bmod N}$$

$$U|x\rangle \rightarrow |ax \bmod N\rangle$$

- As a and N are co-prime, this operator is unitary
- Can be efficiently implemented on a quantum computer
- What about U^2, U^4, U^8, …, U^{2^j}

$$U^2|x\rangle \rightarrow |a^2 x \bmod N\rangle$$

Choosing an initial state

$$a^r \equiv 1 \bmod N$$

- Consider the state,

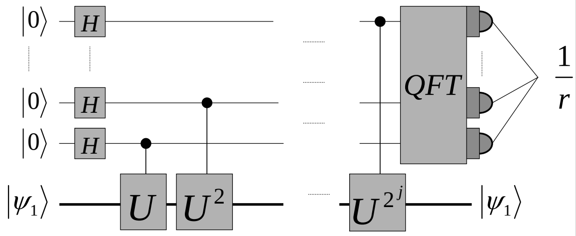
$$|\psi_1\rangle = \sum_{j=0}^{r-1} e^{\frac{-2\pi ij}{r}} |a^j \bmod N\rangle$$

- $|\psi_1\rangle$ is an eigenstate of U, with eigenvalue

$$e^{2\pi i(\frac{1}{r})}$$

- Therefore, if we could prepare $|\psi_1\rangle$, we can use the PE algorithm to efficiently find r, and hence factor N.

Choosing an initial state



- Therefore, if we could prepare $|\psi_1\rangle$, we can use the PE algorithm to efficiently find r, and hence factor N.

Choosing an initial state

$$a^r \equiv 1 \bmod N$$

- Consider the states,

$$|\psi_1\rangle = \sum_{j=0}^{r-1} e^{\frac{-2\pi ij}{r}} |a^j \bmod N\rangle$$

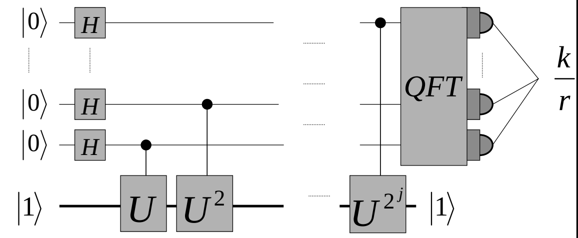
$$k \in \{1, \ldots r\}$$

- $|\psi_k\rangle$ is an eigenstate of U, with eigenvalue

$$e^{2\pi i(\frac{k}{r})}$$

Exercise: Show $|1\rangle = \sum_{k=1}^{r} |\psi_k\rangle$

Choosing an initial state



$$|1\rangle = \sum_{k=1}^{r} |\psi_k\rangle$$

Choosing an initial state

$$a^r \equiv 1 \bmod N$$

- Therefore, using the PE algorithm, we can efficiently calculate

$$\frac{k}{r}$$

- Where k and r are unknown
- If k and r are co-prime, then canceling to an irreducible fraction will yield r.
- If k and r are not co-prime, we try again.

Summary

- We want to find $m_1 \times m_2 = N$
- Equivalent to solving $a^r \equiv 1 \bmod N$
- Use two qubit registers, initially in the state $$|0\rangle|1\rangle$$
- Calculate circuits for U, U^2, .. $U^{2^{\wedge}2n}$
- Apply the phase estimation algorithm
- Repeat if required