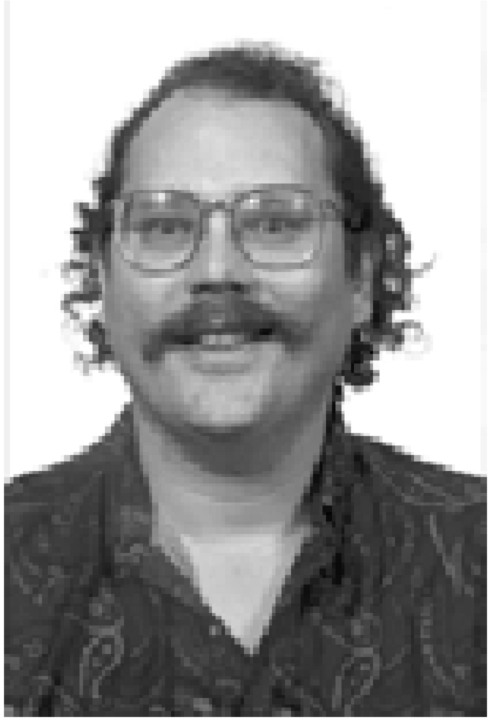
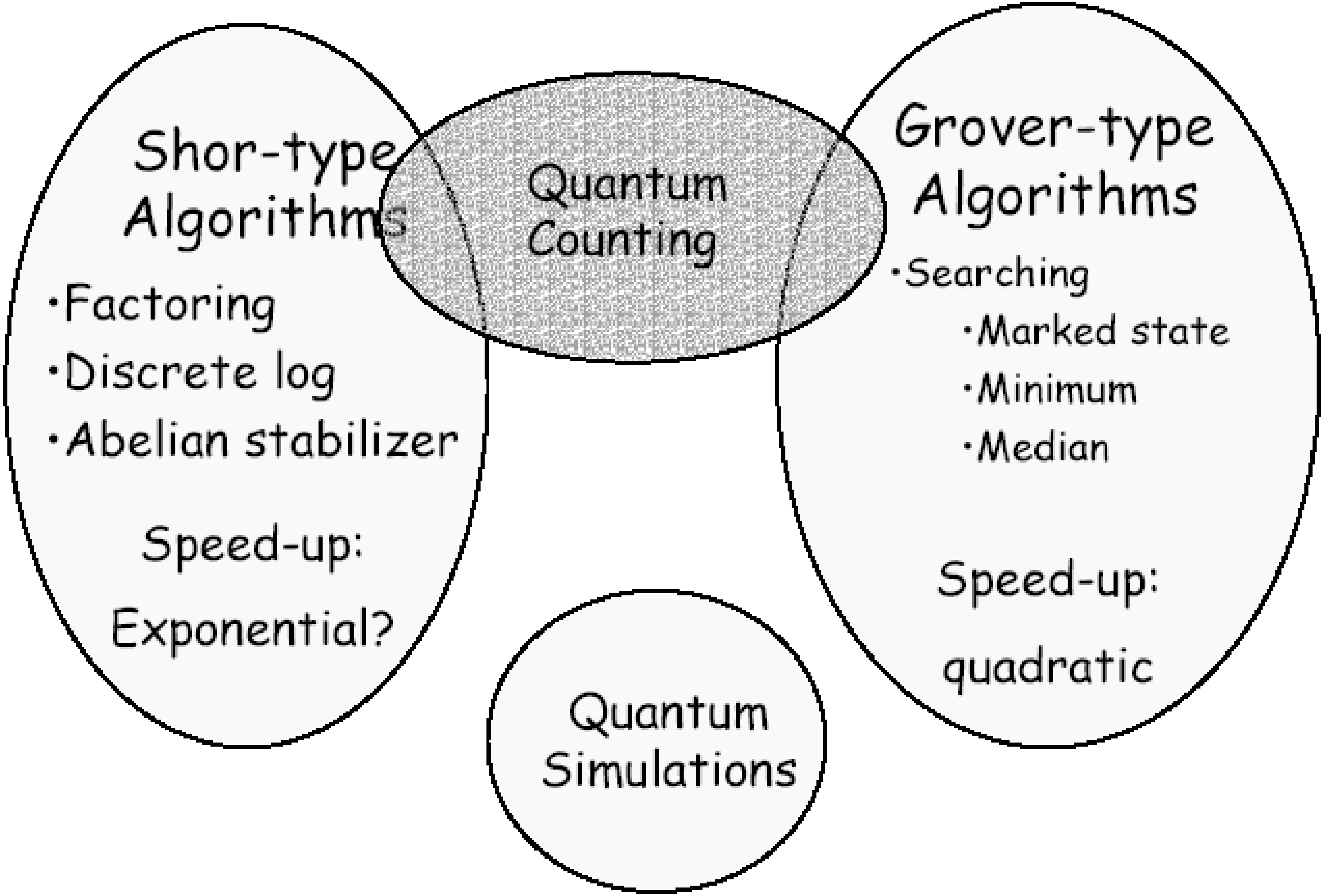
Shor's Algorithm

Anuj Dawar



Peter Shor
AT&T Labs

Quantum Algorithms



Shor-type Algorithms

- Factoring
- Discrete log
- Abelian stabilizer

Speed-up:

Exponential?

Quantum Counting

Grover-type Algorithms

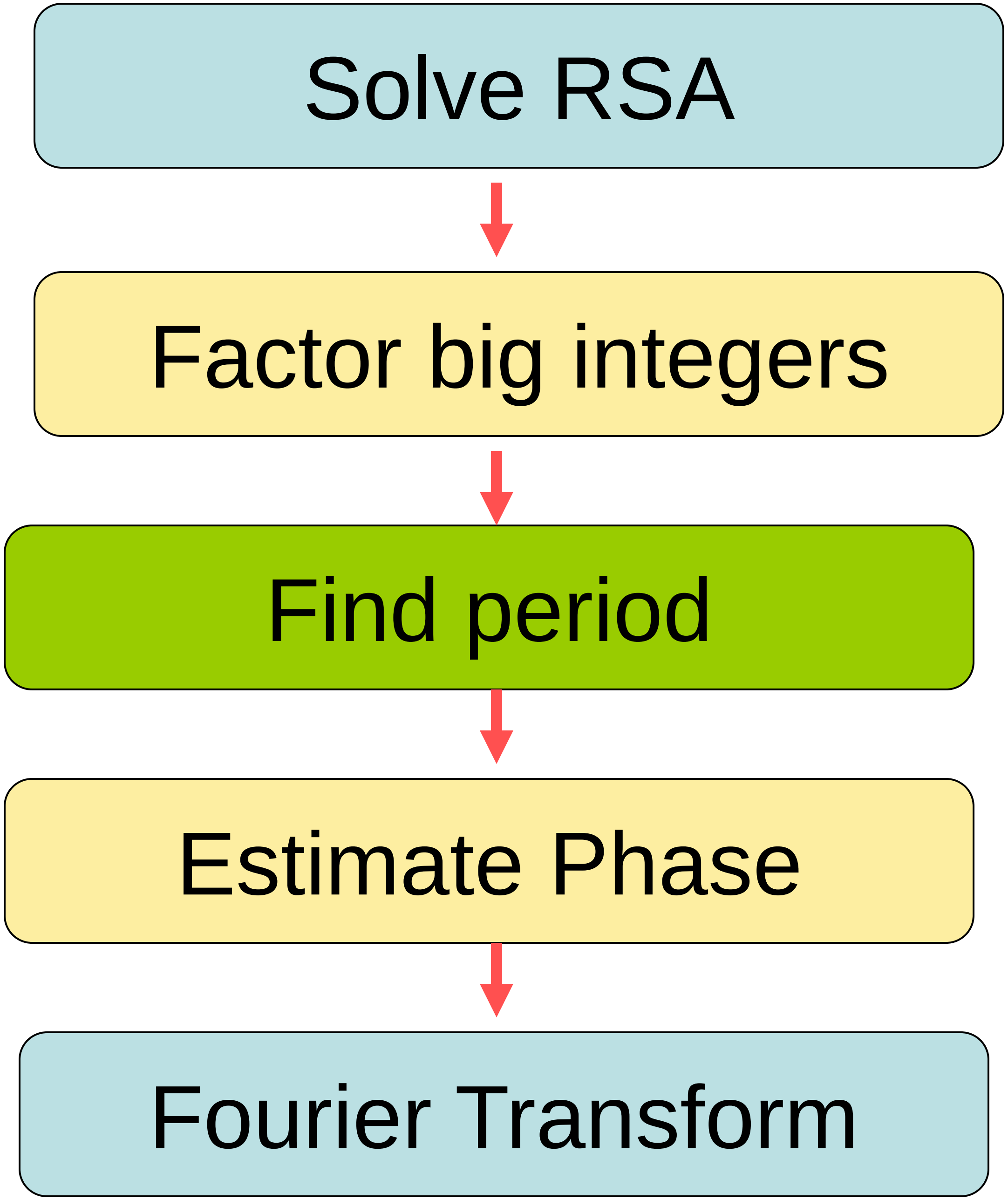
- Searching
  - Marked state
  - Minimum
  - Median

Speed-up:

quadratic

Quantum Simulations

Reductions



Solve RSA

Factor big integers

Find period

Estimate Phase

Fourier Transform

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

Discrete Fourier Transform

- Given a sequence of N complex numbers,

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

- The DFT produces another sequence,

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

- where

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

1. We transform a vector of complex numbers to another vector of complex numbers
2. This is a one-to-one mapping, so inverse transform exists
3. This is not the same condition as in standard Fourier Transform where we transform binary vectors to binary vectors

Discrete Fourier Transform

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

Rotations on a unit-circle

$$\omega \equiv e^{2\pi i / N}$$

- It is not hard to show that the transform

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

INVERSE FOURIER TRANSFORM

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,

$$D^{-1} = D^{\dagger}$$

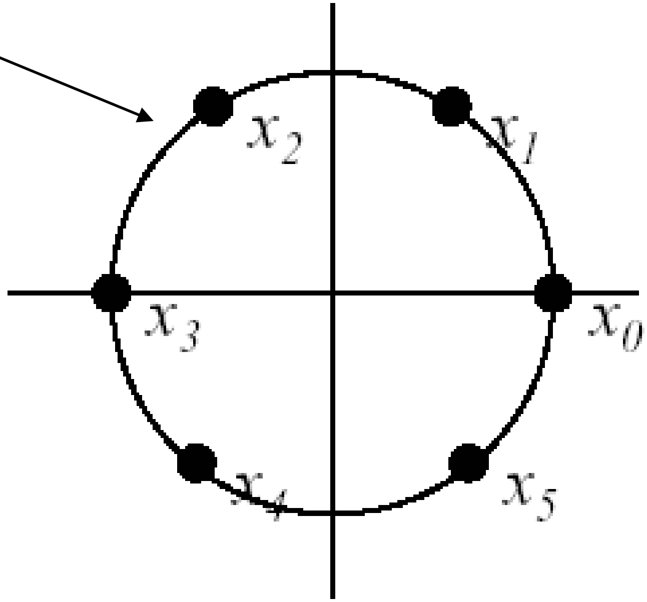
1. We can represent DFT and IDFT as matrix multiplication, but it would be wasteful.
2. We have butterflies in classical computing

ϖ From previous slide

Discrete Fourier Transform

- Suppose

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



- Then

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

Kronecker delta

Exercise: Verify the formula for y_j

Quantum Fourier Transform

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

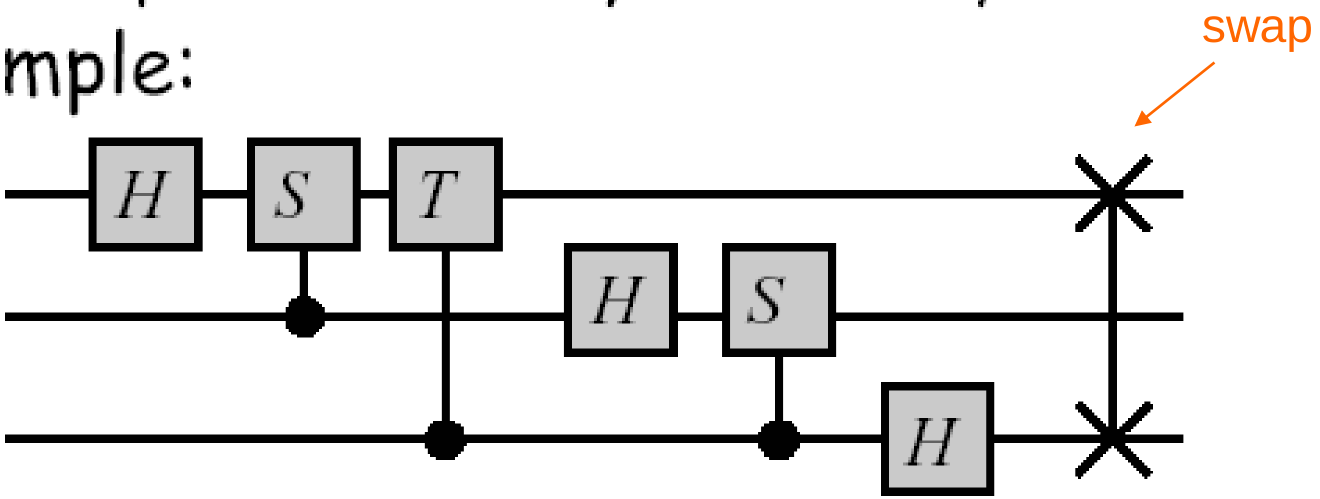
$$|\chi\rangle = y_0|0\rangle + y_1|1\rangle + \ldots + y_{N-1}|N-1\rangle$$

$$y = Dx$$

x_i, y_i, and D were derived earlier

Quantum Fourier Transform

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

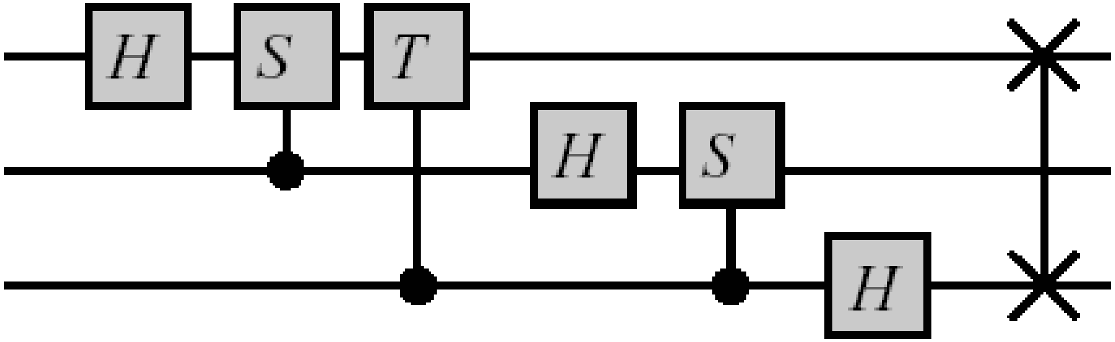


$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \qquad T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$$

Quantum Fourier Transform

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

$\left|\psi\right\rangle \rightarrow$



$\rightarrow \left|\chi\right\rangle$

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

$$y = Dx$$

x_i, y_i, and D were derived earlier

How people figure out this circuit?

1. You have enough knowledge how to analyze quantum circuits – even few methods.
2. When you know the gates and what they do, and you have understanding what is done by parallel composition and what is done by serial composition, you get skill to invent new circuits.
    1. Circuits give you ideas.
    2. Heisenberg notation helps you to verify numerically for small data.
    3. Dirac notation helps you to prove mathematically for arbitrary data.

1. The expansion shown at the bottom of last slide, that you already know from Deutsch, is a general form for all spectral transforms.

2. You can now invent new quantum transforms that correspond to well-known transforms from image processing and DSP

- So now we have a quantum Fourier Transform and a transform Inverse to it, but what can we do with them?

- We still have constraints typical to quantum computing.

1. Observe that we have an input as a quantum state, not as a binary of mv vector.

2. Also the output is a quantum state.

3. So we need special methods to use QFT, we cannot use it as it is, from vectors to vectors.

Quantum Fourier Transform

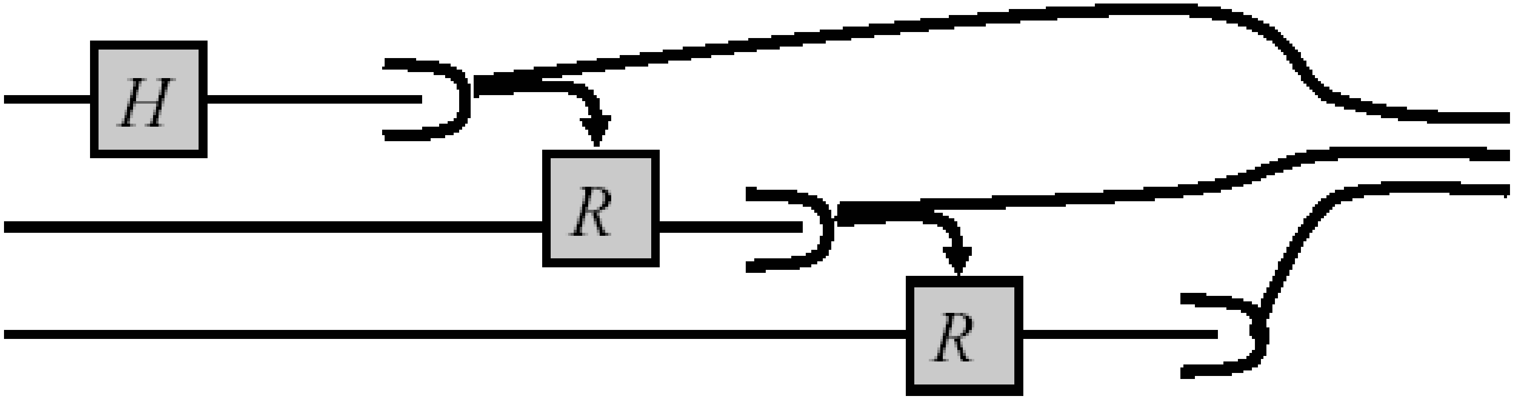
- In general, to perform the QFT on n qubits requires $O(n^2)$ one and two qubit gates
  - Reference: Cleve et al. (quant-ph/9708016)
- Transforming 2^n amplitudes with only n^2 operations
- The fastest we can do classically is $n2^n$
- However, QFT does not allow us to improve classical Fourier transforms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

$O(n^2)$

$$
\begin{aligned}
|\chi\rangle = {} & y_0|000\rangle + y_1|001\rangle + y_2|010\rangle + y_3|011\rangle \\
& + y_4|100\rangle + y_5|101\rangle + y_6|110\rangle + y_7|111\rangle
\end{aligned}
$$

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

Concluding on QFT

1. In quantum computing, the **quantum Fourier transform** is a linear transformation on quantum bits, and is the quantum analogue of the discrete Fourier transform.

2. The quantum Fourier transform is a part of many quantum algorithms, notably:
   1. Shor's algorithm for factoring
   2. computing the discrete logarithm,
   3. the quantum phase estimation algorithm for estimating the eigenvalues of a unitary operator,
   4. algorithms for the hidden subgroup problem.

Concluding on QFT

1. The quantum Fourier transform can be performed efficiently on a quantum computer, with a particular decomposition into a product of simpler unitary matrices.

2. Using a simple decomposition, the discrete Fourier transform can be implemented as a quantum circuit consisting of only $O(n^2)$ Hadamard gates and controlled phase shift gates, where n is the number of qubits.

3. This can be compared with the classical discrete Fourier transform,
   - which takes $O(n2^n)$ gates
   - (where n is the number of bits),
   - which is exponentially more than $O(n^2)$.

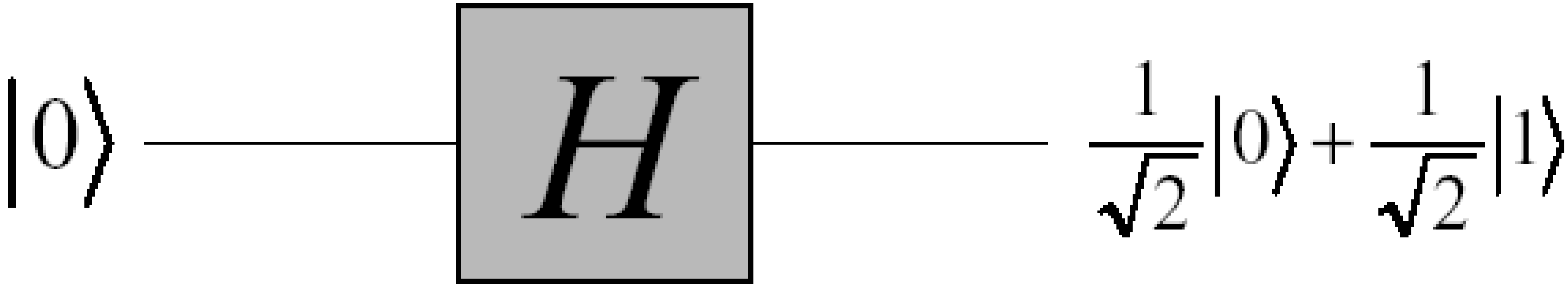
Concluding on QFT

1. However, the quantum Fourier transform acts on a quantum state,
   1. whereas the classical Fourier transform acts on a vector,
   2. so the quantum Fourier transform can not give a generic exponential speedup for any task which requires the classical Fourier transform.

2. The best quantum Fourier transform algorithms known today require only $O(n \log n)$ gates to achieve an efficient approximation.

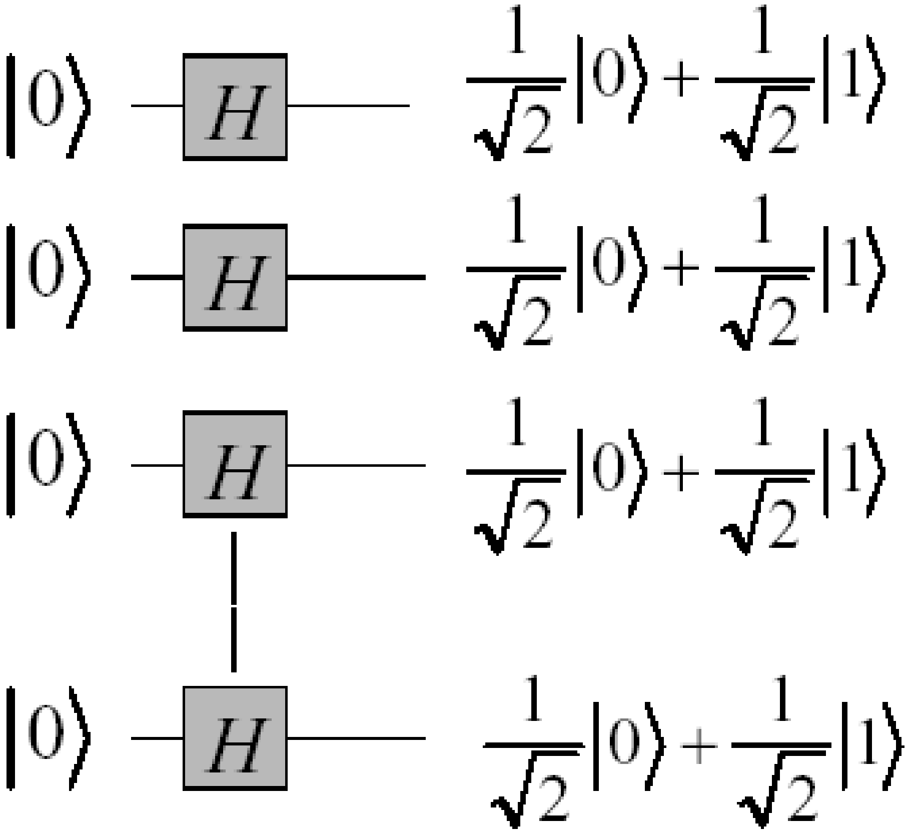
Hadamard Transform Review

Hadamard gate



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

Hadamard gate


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

QFT and vector of Hadamards are basic component blocks of Quantum Phase Estimation which we will discuss next

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