Shor's algorithm

Guojing Tian

Institute of Computing Technology, CAS

Outline

1. QFT
   - quantum Fourier transform
   - product representation
   - efficient circuit
   - complexity
2. phase estimation
   - phase estimation
   - three stages
   - intuition
   - performance and requirements
   - procedure
3. order finding and factoring
   - order finding
   - factoring
4. general applications

quantum Fourier transform

discrete Fourier transformation

input: a vector of complex numbers $x_0, \cdots, x_{N-1}$, for fixed N;
output: a vector of complex numbers $y_0, \cdots, y_{N-1}$, defined by

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

quantum Fourier transform

discrete Fourier transformation

input: a vector of complex numbers $x_0, \cdots, x_{N-1}$, for fixed N;
output: a vector of complex numbers $y_0, \cdots, y_{N-1}$, defined by

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

quantum Fourier transformation (QFT)

input: $|j\rangle$;
output:

$$\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi ijk/N} |k\rangle.$$

Equivalently, the action on an arbitrary state may be written

$$\sum_{j=0}^{N-1} x_j|j\rangle \rightarrow \sum_{k=0}^{N-1} y_k|k\rangle,$$

where the amplitudes y_k are the discrete Fourier transform of the amplitudes x_j.

Equivalently, the action on an arbitrary state may be written

$$\sum_{j=0}^{N-1} x_j|j\rangle \rightarrow \sum_{k=0}^{N-1} y_k|k\rangle,$$

where the amplitudes y_k are the discrete Fourier transform of the amplitudes x_j.

Q1: QFT is unitary?

product representation

For n-qubit quantum system, we have $N = 2^n$, and the basis $|0\rangle, |1\rangle, \cdots, |2^n - 1\rangle$ is the computational basis.

- $j = j_1 j_2 \cdots j_n$, i.e., $j = j_1 2^{n-1} + j_2 2^{n-2} + \cdots + j_n 2^0$.
- $0.j_l j_{l+1} \cdots j_m = j_l/2 + j_{l+1}/4 + \cdots + j_m/2^{m-l+1}$

product representation

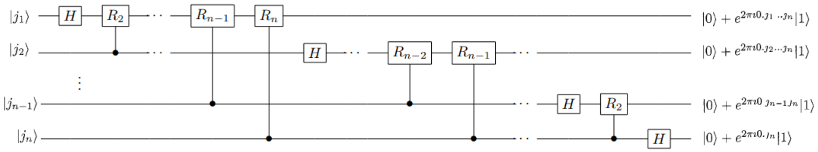
For n-qubit quantum system, we have $N = 2^n$, and the basis $|0\rangle, |1\rangle, \cdots, |2^n - 1\rangle$ is the computational basis.

- $j = j_1 j_2 \cdots j_n$, i.e., $j = j_1 2^{n-1} + j_2 2^{n-2} + \cdots + j_n 2^0$.
- $0.j_l j_{l+1} \cdots j_m = j_l/2 + j_{l+1}/4 + \cdots + j_m/2^{m-l+1}$

$$
\begin{aligned}
|j\rangle &\rightarrow \frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} e^{2\pi i jk/2^n} |k\rangle \\
&= \frac{1}{2^{n/2}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} e^{2\pi i j \left(\sum_{l=1}^{n} k_l 2^{-l}\right)} |k_1 \ldots k_n\rangle \\
&= \frac{1}{2^{n/2}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} \bigotimes_{l=1}^{n} e^{2\pi i j k_l 2^{-l}} |k_l\rangle \\
&= \frac{1}{2^{n/2}} \bigotimes_{l=1}^{n} \left[\sum_{k_l=0}^{1} e^{2\pi i j k_l 2^{-l}} |k_l\rangle \right] \\
&= \frac{1}{2^{n/2}} \bigotimes_{l=1}^{n} \left[|0\rangle + 2^{2\pi i j 2^{-l}} |1\rangle \right] \\
&= \frac{\left(|0\rangle + e^{2\pi i 0.j_n} |1\rangle\right) \left(|0\rangle + e^{2\pi i 0.j_{n-1} j_n} |1\rangle\right) \cdots \left(|0\rangle + e^{2\pi i 0.j_1 j_2 \cdots j_n} |1\rangle\right)}{2^{n/2}}
\end{aligned}
$$

efficient circuit

With $R_k \equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{bmatrix}$, we can derive the circuit for QFT just as follows.

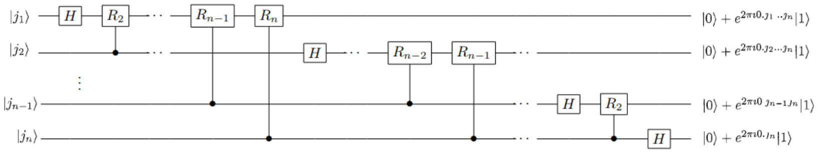


complexity

classical Fourier transformation: $O(N^2)$;
fast Fourier transform: $O(N \log N)$.

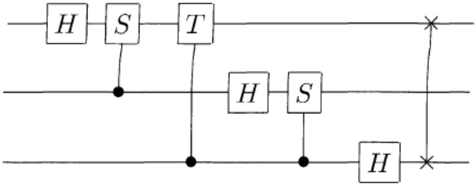
complexity

classical Fourier transformation: $O(N^2)$;
fast Fourier transform: $O(N \log N)$.

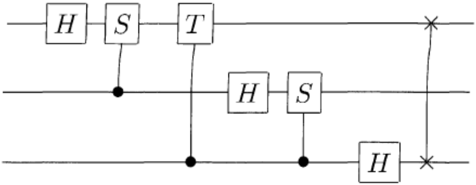


$O(n^2)$

Eg. Explicit circuit for 3-qubit QFT.



Eg. Explicit circuit for 3-qubit QFT.



$$\frac{1}{\sqrt{8}}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \omega^4 & \omega^5 & \omega^6 & \omega^7 \\ 1 & \omega^2 & \omega^4 & \omega^6 & 1 & \omega^2 & \omega^4 & \omega^6 \\ 1 & \omega^3 & \omega^6 & \omega^1 & \omega^4 & \omega^7 & \omega^2 & \omega^5 \\ 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 \\ 1 & \omega^5 & \omega^2 & \omega^7 & \omega^4 & \omega^1 & \omega^6 & \omega^3 \\ 1 & \omega^6 & \omega^4 & \omega^2 & 1 & \omega^6 & \omega^4 & \omega^2 \\ 1 & \omega^7 & \omega^6 & \omega^5 & \omega^4 & \omega^3 & \omega^2 & \omega^1 \end{bmatrix}$$

Q2: How to obtain this matrix?

phase estimation

Suppose a unitary operator U has an eigenvector $|u\rangle$ with eigenvalue $e^{2\pi i\varphi}$, where the value of φ is unknown.

Goal: to estimate φ
black boxes (oracles): preparing the state $|u\rangle$, performing the controlled-U^{2^j} operation, for non-negative integers j.

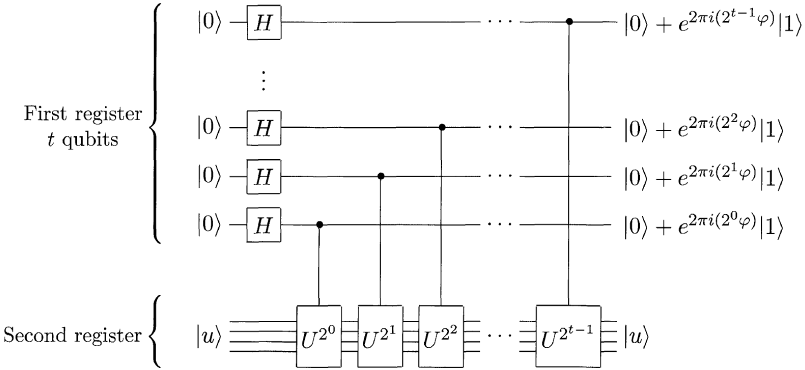
classical???

the first stage

- The first register contains t qubits initially in the state $|0\rangle$. [acurracy & probability]
- The second register begins in the state $|u\rangle$, and contains qubits which can store $|u\rangle$.

the first stage

- The first register contains t qubits initially in the state $|0\rangle$. [acurracy & probability]
- The second register begins in the state $|u\rangle$, and contains qubits which can store $|u\rangle$.



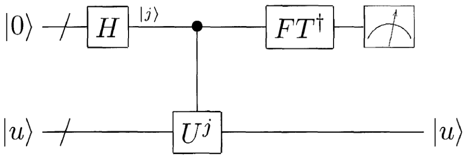
the second and third stage

- **The second stage**: inverse QFT
- **The third stage**: measure the first register in the computational basis

the second and third stage

- **The second stage**: inverse QFT
- **The third stage**: measure the first register in the computational basis

The schematic of the overall phase estimation is as follows.



intuition

Suppose φ may be expressed exactly in t bits, as $\varphi = 0.\varphi_1 \cdots \varphi_t$.

1. The state resulting from **the first stage** may by rewritten

$$\frac{1}{2^{t/2}}\left(|0\rangle + e^{2\pi i 0.\varphi_t}|1\rangle\right)\cdots\left(|0\rangle + e^{2\pi i 0.\varphi_1\varphi_2\cdots\varphi_t}|1\rangle\right).$$

2. The second stage is to apply the **inverse QFT** (heart), then the output state is the product state

$$\frac{1}{2^{t/2}}\sum_{j=0}^{2^t-1} e^{2\pi i\varphi j}|j\rangle|u\rangle \rightarrow |\varphi\rangle|u\rangle.$$

3. Thus a measurement in the computational basis gives us φ exactly.

performance and requirements

The above intuition based on the fact that φ can be written exactly in t bits.
What happens when this is not the case?

- Let $b \in [0, 2^t - 1]$, and $b/2^t = 0.b_1 \cdots b_t$ is the best t-bit approximation less than φ. (eg. the first t bits of φ)
- The difference $\delta \equiv \varphi - b/2^t$, and $0 \leq \delta \leq 2^{-t}$.
- Aim: produce a result which is close to b, thus to estimate φ accurately with high probability.

performance and requirements

The above intuition based on the fact that φ can be written exactly in t bits.
What happens when this is not the case?

- Let $b \in [0, 2^t - 1]$, and $b/2^t = 0.b_1 \cdots b_t$ is the best t-bit approximation less than φ. (eg. the first t bits of φ)
- The difference $\delta \equiv \varphi - b/2^t$, and $0 \leq \delta \leq 2^{-t}$.
- Aim: produce a result which is close to b, thus to estimate φ accurately with high probability.

t???

$$|\tilde{\varphi}\rangle = \frac{1}{2^t}\sum_{j,k=0}^{2^t-1} e^{\frac{-2\pi ijk}{2^t}} e^{2\pi ij\varphi}|k\rangle \quad = \frac{1}{2^t}\sum_{j,k=0}^{2^t-1} e^{\frac{2\pi ij(2^t\varphi-k)}{2^t}}|k\rangle$$

$$= \frac{1}{2^t}\sum_{k=0}^{2^t-1} \frac{1-e^{2\pi i\delta}}{1-e^{\frac{2\pi i\delta}{2^t}}}|k\rangle \quad (2^t\delta := 2^t\varphi - k)$$

- If the result is m, then $p(|m-b|>e) \leq \frac{1}{2(e-1)}$, where e is an integer satisfying

$$\frac{m}{2^t} - \frac{b}{2^t} < \frac{1}{2^n} \longrightarrow e = 2^{t-n} - 1.$$

$$|\tilde{\varphi}\rangle = \frac{1}{2^t}\sum_{j,k=0}^{2^t-1} e^{\frac{-2\pi ijk}{2^t}} e^{2\pi ij\varphi}|k\rangle \quad = \frac{1}{2^t}\sum_{j,k=0}^{2^t-1} e^{\frac{2\pi ij(2^t\varphi-k)}{2^t}}|k\rangle$$

$$= \frac{1}{2^t}\sum_{k=0}^{2^t-1} \frac{1-e^{2\pi i\delta}}{1-e^{\frac{2\pi i\delta}{2^t}}}|k\rangle \quad (2^t\delta := 2^t\varphi - k)$$

- If the result is m, then $p(|m-b|>e) \leq \frac{1}{2(e-1)}$, where e is an integer satisfying

$$\frac{m}{2^t} - \frac{b}{2^t} < \frac{1}{2^n} \longrightarrow e = 2^{t-n} - 1.$$

- Thus to obtain φ accurate to n-bits with succ. prob. at least $1-\epsilon$, we choose

$$t = n + \lceil \log(2 + \frac{1}{2\epsilon}) \rceil.$$

$$|\tilde{\varphi}\rangle = \frac{1}{2^t}\sum_{j,k=0}^{2^t-1} e^{\frac{-2\pi ijk}{2^t}} e^{2\pi ij\varphi}|k\rangle \quad = \frac{1}{2^t}\sum_{j,k=0}^{2^t-1} e^{\frac{2\pi ij(2^t\varphi-k)}{2^t}}|k\rangle$$

$$= \frac{1}{2^t}\sum_{k=0}^{2^t-1} \frac{1-e^{2\pi i\delta}}{1-e^{\frac{2\pi i\delta}{2^t}}}|k\rangle \quad (2^t\delta := 2^t\varphi - k)$$

- If the result is m, then $p(|m-b|>e) \le \frac{1}{2(e-1)}$, where e is an integer satisfying

$$\frac{m}{2^t} - \frac{b}{2^t} < \frac{1}{2^n} \longrightarrow e = 2^{t-n} - 1.$$

- Thus to obtain φ accurate to n-bits with succ. prob. at least $1-\epsilon$, we choose

$$t = n + \lceil \log(2 + \frac{1}{2\epsilon}) \rceil.$$

Q3: How to deal with the case of $|\psi\rangle = \sum_u c_u|u\rangle$? (**Ex. 5.8**)

procedure

Quantum phase estimation can be summarized below.

procedure

Quantum phase estimation can be summarized below.

Inputs: (1) A black box wich performs a controlled-U^j operation, for integer j, (2) an eigenstate $|u\rangle$ of U with eigenvalue $e^{2\pi i\varphi_u}$, and (3) $t = n + \lceil \log\left(2 + \frac{1}{2\epsilon}\right) \rceil$ qubits initialized to $|0\rangle$.

Outputs: An n-bit approximation $\widetilde{\varphi_u}$ to φ_u.

Runtime: $O(t^2)$ operations and one call to controlled-U^j black box. Succeeds with probability at least $1 - \epsilon$.

Procedure:

1.	$\|0\rangle\|u\rangle$	initial state
2.	$\rightarrow \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} \|j\rangle\|u\rangle$	create superposition
3.	$\rightarrow \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} \|j\rangle U^j \|u\rangle$	apply black box
	$= \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} e^{2\pi i j \varphi_u} \|j\rangle\|u\rangle$	result of black box
4.	$\rightarrow \|\widetilde{\varphi_u}\rangle\|u\rangle$	apply inverse Fourier transform
5.	$\rightarrow \widetilde{\varphi_u}$	measure first register

The fast quantum algorithms for these two problems are interesting for three reasons.

- providing evidence for the idea that "quantum computers may be inherently more powerful than classical ones"
- intrinsic worth to justify interest in any novel algorithm
- practical standpoint: to break the RSA public-key cryptosystem.

The fast quantum algorithms for these two problems are interesting for three reasons.

- providing evidence for the idea that "quantum computers may be inherently more powerful than classical ones"
- intrinsic worth to justify interest in any novel algorithm
- practical standpoint: to break the RSA public-key cryptosystem.

These two problems are in fact equivalent to one another.

1. explaining a quantum algorithm for solving the order-finding problem;
2. explaining how the order-finding problem implies the ability to factor.

order finding

- **Def:** For positive integers x and N, and $x < N$ with no common factors. **The order of x modulo N** is defined to be the least positive integer, r, such that $x^r = 1(\text{mod} N)$.

order finding

- **Def:** For positive integers x and N, and $x < N$ with no common factors. **The order of x modulo N** is defined to be the least positive integer, r, such that $x^r = 1(\mathrm{mod} N)$.
- **Goal:** to determine the order for some specified x and N.

order finding

- **Def:** For positive integers x and N, and $x < N$ with no common factors. **The order of x modulo N** is defined to be the least positive integer, r, such that $x^r = 1(\mathrm{mod} N)$.
- **Goal:** to determine the order for some specified x and N.
- **Hardness:** No classical algorithm is known to solve it using polynomial in the $O(L)$ bits, where $L \equiv \lceil log(N) \rceil$.

order finding

- **Def:** For positive integers x and N, and $x < N$ with no common factors. **The order of x modulo N** is defined to be the least positive integer, r, such that $x^r = 1(\text{mod} N)$.
- **Goal:** to determine the order for some specified x and N.
- **Hardness:** No classical algorithm is known to solve it using polynomial in the $O(L)$ bits, where $L \equiv \lceil log(N) \rceil$.
- The quantum algorithm for order-finding is just the phase estimation applied to the unitary operator

$$U|y\rangle \equiv |xy \, \text{mod} N\rangle,$$

with $y \in \{0,1\}^L$ and $0 \leq y \leq N-1$ for the action of mod N.

reduction to phase estimation

- Define states $|u_s\rangle, 0 \leq s \leq r-1$ as follows, i.e.,

$$|u_s\rangle \equiv \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} exp\left[\frac{-2\pi isk}{r}\right] |x^k \mathrm{mod} N\rangle.$$

- Then they are eigenstates of U, since

$$U|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} exp\left[\frac{-2\pi isk}{r}\right] |x^{k+1} \mathrm{mod} N\rangle = exp\left[\frac{-2\pi is}{r}\right] |u_s\rangle.$$

procedure

Inputs: (1) A black box $U_{x,N}$ which performs the transformation $|j\rangle|k\rangle \to |j\rangle|x^j k \bmod N\rangle$, for x co-prime to the L-bit number N, (2) $t = 2L + 1 + \lceil \log\left(2 + \frac{1}{2\epsilon}\right)\rceil$ qubits initialized to $|0\rangle$, and (3) L qubits initialized to the state $|1\rangle$.

Outputs: The least integer $r > 0$ such that $x^r = 1 \pmod N$.

Runtime: $O(L^3)$ operations. Succeeds with probability $O(1)$.

Procedure:

1.	$\|0\rangle\|1\rangle$	initial state
2.	$\to \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} \|j\rangle\|1\rangle$	create superposition
3.	$\to \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} \|j\rangle\|x^j \bmod N\rangle$	apply $U_{x,N}$
	$\approx \frac{1}{\sqrt{r2^t}} \sum_{s=0}^{r-1} \sum_{j=0}^{2^t-1} e^{2\pi i s j/r} \|j\rangle\|u_s\rangle$	
4.	$\to \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \|\widetilde{s/r}\rangle\|u_s\rangle$	apply inverse Fourier transform to first register
5.	$\to \widetilde{s/r}$	measure first register
6.	$\to r$	apply continued fractions algorithm

factoring

- Given a positive composite integer N, what prime numbers when multiplied together equal it?

factoring

- Given a positive composite integer N, what prime numbers when multiplied together equal it?
- reduction of factoring to order-finding

factoring

- Given a positive composite integer N, what prime numbers when multiplied together equal it?
- reduction of factoring to order-finding
- simple example of factoring 15

reduction

Inputs: A composite number N

Outputs: A non-trivial factor of N.

Runtime: $O((\log N)^3)$ operations. Succeeds with probability $O(1)$.

Procedure:

1. If N is even, return the factor 2.
2. Determine whether $N = a^b$ for integers $a \geq 1$ and $b \geq 2$, and if so return the factor a (uses the classical algorithm of Exercise 5.17).
3. Randomly choose x in the range 1 to $N-1$. If $\gcd(x, N) > 1$ then return the factor $\gcd(x, N)$.
4. Use the order-finding subroutine to find the order r of x modulo N.
5. If r is even and $x^{r/2} \neq -1 \pmod N$ then compute $\gcd(x^{r/2} - 1, N)$ and $\gcd(x^{r/2} + 1, N)$, and test to see if one of these is a non-trivial factor, returning that factor if so. Otherwise, the algorithm fails.

Eg. factoring 15

1. Choose a random number $x = 7$.

Eg. factoring 15

1. Choose a random number $x = 7$.
2. Compute the order r satisfying $x^r = 1 \mod N$

Eg. factoring 15

1. Choose a random number $x = 7$.
2. Compute the order r satisfying $x^r = 1 \mod N$
   2.1 begin with the state $|0_t\rangle|0_4\rangle$

Eg. factoring 15

1. Choose a random number $x = 7$.
2. Compute the order r satisfying $x^r = 1 \mod N$
   2.1 begin with the state $|0_t\rangle|0_4\rangle$
   2.2 apply H gates to the first register containing $t = 11$ qubits (ensuring $\epsilon \le 1/4$)

$$\frac{1}{\sqrt{2^t}}\sum_{k=0}^{2^t-1}|k\rangle|0_4\rangle = \frac{1}{\sqrt{2^t}}\left[|0\rangle + |1\rangle + |2\rangle + \cdots + |2^t-1\rangle\right]|0_4\rangle$$

Eg. factoring 15

1. Choose a random number $x = 7$.
2. Compute the order r satisfying $x^r = 1 \mod N$
   - 2.1 begin with the state $|0_t\rangle|0_4\rangle$
   - 2.2 apply H gates to the first register containing $t = 11$ qubits (ensuring $\epsilon \leq 1/4$)

$$\frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^t-1} |k\rangle|0_4\rangle = \frac{1}{\sqrt{2^t}} \left[|0\rangle + |1\rangle + |2\rangle + \cdots + |2^t - 1\rangle \right] |0_4\rangle$$

   - 2.3 compute $f(k) = x^k \bmod N$

$$\frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^t-1} |k\rangle|x^k \bmod N\rangle$$

$$= \frac{1}{\sqrt{2^t}} \left[|0\rangle|1\rangle + |1\rangle|7\rangle + |2\rangle|4\rangle + |3\rangle|13\rangle + |4\rangle|1\rangle + |5\rangle|7\rangle + |6\rangle|4\rangle + \cdots \right]$$

Eg. factoring 15

1. Choose a random number $x = 7$.
2. Compute the order r satisfying $x^r = 1 \mod N$
   - 2.1 begin with the state $|0_t\rangle|0_4\rangle$
   - 2.2 apply H gates to the first register containing $t = 11$ qubits (ensuring $\epsilon \leq 1/4$)

$$\frac{1}{\sqrt{2^t}}\sum_{k=0}^{2^t-1}|k\rangle|0_4\rangle = \frac{1}{\sqrt{2^t}}\left[|0\rangle + |1\rangle + |2\rangle + \cdots + |2^t - 1\rangle\right]|0_4\rangle$$

   - 2.3 compute $f(k) = x^k \bmod N$

$$\frac{1}{\sqrt{2^t}}\sum_{k=0}^{2^t-1}|k\rangle|x^k \bmod N\rangle$$

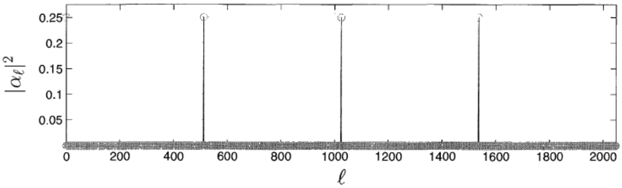
$$= \frac{1}{\sqrt{2^t}}\left[|0\rangle|1\rangle + |1\rangle|7\rangle + |2\rangle|4\rangle + |3\rangle|13\rangle + |4\rangle|1\rangle + |5\rangle|7\rangle + |6\rangle|4\rangle + \cdots\right]$$

   - 2.4 apply the inverse QFT to the first register and measure it
     measure the second register, obtaining a random result from 1, 7, 4 or 13.

2.5 suppose the result is 4 (r2), that means the state (r1) input to $FT^{\dagger}$ would have been $\sqrt{\frac{4}{2^t}}\left[|2\rangle + |6\rangle + |10\rangle + |14\rangle + \cdots\right]$

2.5 suppose the result is 4 (r2), that means the state (r1) input to $FT^{\dagger}$ would have been $\sqrt{\frac{4}{2^t}}\left[|2\rangle + |6\rangle + |10\rangle + |14\rangle + \cdots\right]$

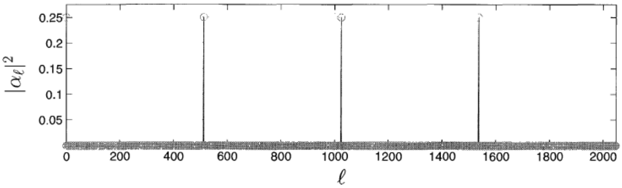
2.6 after applying $FT^{\dagger}$, we obtain some state $\sum_l \alpha_l |l\rangle$, with the probability distribution below



the final measurement gives either 0, 512, 1024, or 1536, and each with probability almost exactly 1/4.

2.5 suppose the result is 4 (r2), that means the state (r1) input to $FT^\dagger$ would have been $\sqrt{\frac{4}{2^t}}\left[|2\rangle + |6\rangle + |10\rangle + |14\rangle + \cdots\right]$

2.6 after applying $FT^\dagger$, we obtain some state $\sum_l \alpha_l |l\rangle$, with the probability distribution below

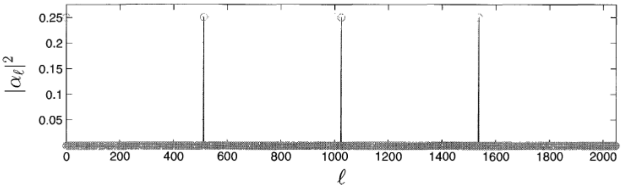


the final measurement gives either 0, 512, 1024, or 1536, and each with probability almost exactly 1/4.

3. suppose we obtain $l = 1536$, computing the continued fraction expansion gives $1536/2048 = 1/(1 + (1/3))$, so that $3/4$ occurs as a convergent in the expansion.

2.5 suppose the result is 4 (r2), that means the state (r1) input to $FT^\dagger$ would have been $\sqrt{\frac{4}{2^t}}\left[|2\rangle + |6\rangle + |10\rangle + |14\rangle + \cdots\right]$

2.6 after applying $FT^\dagger$, we obtain some state $\sum_l \alpha_l |l\rangle$, with the probability distribution below



the final measurement gives either 0, 512, 1024, or 1536, and each with probability almost exactly 1/4.

3. suppose we obtain $l = 1536$, computing the continued fraction expansion gives $1536/2048 = 1/(1 + (1/3))$, so that $3/4$ occurs as a convergent in the expansion.
4. r is even, and $x^{r/2} \mod N = 4 \neq -1 \mod 15$, so $\gcd(x^2 - 1, 15) = 3$ and $\gcd(x^2 + 1, 15) = 5$ are both non-trivial factors.

general applications of the QFT

- period finding
- discrete logarithms
- hidden subgroup problem

The readers interested in understanding all the details will have to work much harder, because the presentation in this section is rather more schematic and conceptual than earlier sections.

Summary

1. QFT
   - quantum Fourier transform
   - product representation
   - efficient circuit
   - complexity
2. phase estimation
   - phase estimation
   - three stages
   - intuition
   - performance and requirements
   - procedure
3. order finding and factoring
   - order finding
   - factoring
4. general applications