Shor's Algorithm

Fabio A. González
QCP 2020-2

Universidad Nacional de Colombia

2. Quantum Fourier Transform

Discrete Fourier Transform

$$DFT: \mathbb{C}^N \longrightarrow \mathbb{C}^N$$
$$(x_0, \dots x_{N-1}) \longmapsto (y_0, \dots y_{N-1})$$

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

Quantum Fourier Transform

$$QFT: \mathbb{C}^N \longrightarrow \mathbb{C}^N$$
$$\sum_{i=0}^{N-1} x_i |i\rangle \longmapsto \sum_{i=0}^{N-1} y_i |i\rangle$$

$$|x\rangle \longmapsto \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{2\pi i \frac{xy}{N}} |y\rangle$$

$$U_{QFT} = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} e^{2\pi i \frac{xy}{N}} |y\rangle\langle x|$$

$$\underbrace{|x\rangle}_{\text{Z basis}} \longmapsto \underbrace{|\tilde{x}\rangle}_{\text{Fourier Basis.}}$$

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \qquad N=2$$

$$|x\rangle \longmapsto \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{2\pi i \frac{xy}{N}} |y\rangle$$

$$U_{QFT}|\psi\rangle = \frac{\alpha}{\sqrt{2}}\left(\cancelto{1}{e^{2\pi i \frac{0\times 0}{2}}}|0\rangle + \cancelto{1}{e^{2\pi i \frac{0\times 1}{2}}}|1\rangle\right)$$

$$+ \frac{\beta}{\sqrt{2}}\left(\cancelto{1}{e^{2\pi i \frac{1\times 0}{2}}}|0\rangle + \cancelto{-1}{e^{2\pi i \frac{1\times 1}{2}}}|1\rangle\right)$$

$$e^{\pi i} = \cos\pi + i\sin\pi = -1$$

$$= \frac{1}{\sqrt{2}}\left((\alpha+\beta)|0\rangle + (\alpha-\beta)|1\rangle\right)$$

$$= H|\psi\rangle$$

QFT of n qubits $\quad N = 2^n$

$y = y_1 y_2 \dots y_n$

$y = y_1 2^{n-1} + y_2 2^{n-2} \dots y_1 2^0$
$= \sum_{k=1}^{n} y_k 2^{n-k}$

$|x\rangle = |x_1 \dots x_n\rangle$

most significant qubit

$y/2^n = \sum_{k=1}^{n} y_k 2^{\cancel{n}-k} / 2^{\cancel{n}}$

$$U_{QFT}|x\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{2\pi i x y/2^n} |y\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{2\pi i x \left(\sum_{k=1}^{n} y_k/2^k\right)} |y_1 \dots y_n\rangle$$

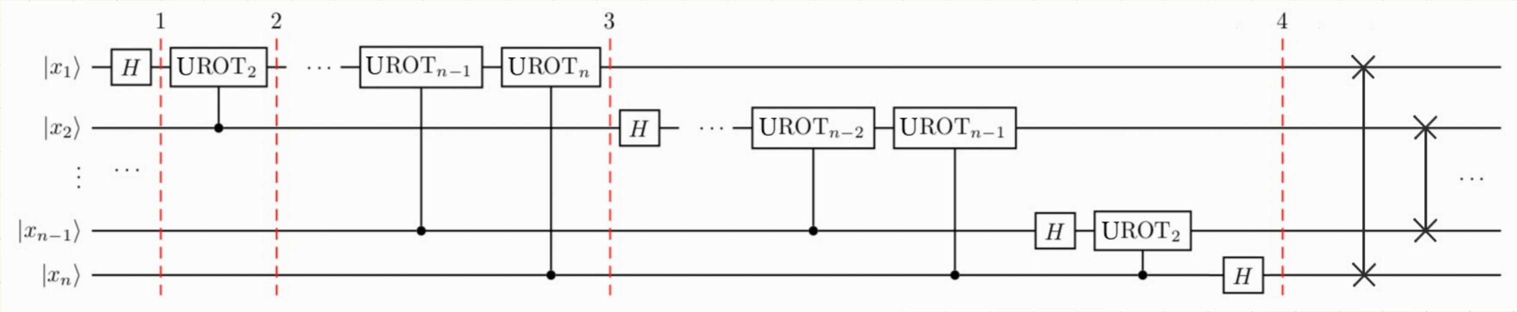
$$= \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} \prod_{k=1}^{n} e^{2\pi i x y_k/2^k} |y_1 \dots y_n\rangle$$

$$= \frac{1}{\sqrt{N}} \bigotimes_{k=1}^{n} \left[\sum_{y_k=0}^{1} e^{2\pi i x y_k/2^k} |y_k\rangle\right] = \frac{1}{\sqrt{N}} \bigotimes_{k=1}^{n} \left(|0\rangle + e^{2\pi i x/2^k} |1\rangle\right)$$

$$= \frac{1}{\sqrt{N}} \left(|0\rangle + e^{\frac{2\pi i}{2} x} |1\rangle\right) \otimes \left(|0\rangle + e^{\frac{2\pi i x}{2^2}} |1\rangle\right) \otimes \dots\dots$$

$$\otimes \left(|0\rangle + e^{\frac{2\pi i x}{2^{n-1}}} |1\rangle\right) \otimes \left(|0\rangle + e^{\frac{2\pi i}{2^n} x} |1\rangle\right)$$

2. Quantum Fourier Transform Circuit

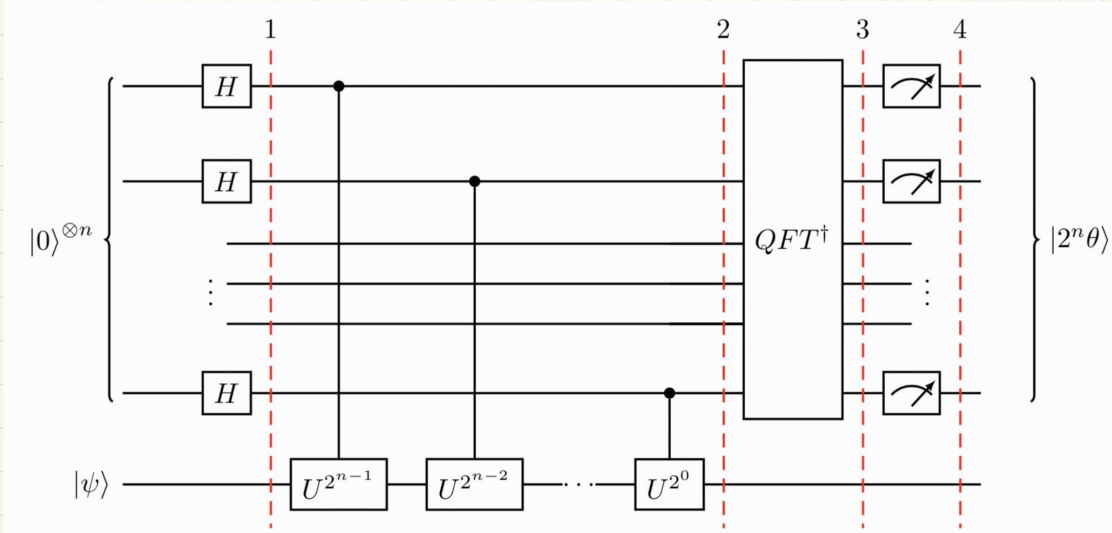


3. Quantum Phase Estimation

Given a unitary operator U with an eigenvector $|\psi\rangle$

$$U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$$

estimate θ.



1. Initialize

$$|\psi_0\rangle = |0\rangle^{\otimes n} \underbrace{|\psi\rangle}_{\text{Eigenvector}}$$

2. Apply Hadamard

$$|\psi_1\rangle = [H|0\rangle]^{\otimes n}|\psi\rangle$$

3. Apply several controlled U

Effect: Control qubit will turn (phase kickback) proportionally to the phase $e^{2i\pi\theta}$
The phase is encoded in the input qubits in the Fourier Basis

4. Apply inverse Quantum Fourier Transform $(QFT^\dagger)$ to convert to the z basis

5. Measure $\Longrightarrow |2^n\theta\rangle$

4. Shor's Algorithm

Problem: Given a and N, $a < N$, find the period of the function

$$f(x) = a^x \bmod N$$

Period: smallest r such that $a^r \bmod N = 1$

Example: $a = 7$ $N = 15$

$(a \times b) \bmod c = (a \bmod c) \times b \bmod c$

$f(1) = 7 \bmod 15 = 7$

$f(2) = 49 \bmod 15 = 4$

$f(3) = \underline{f(2)} \cdot 7 \bmod 15 = 13 = 7^3 \bmod 15$

$f(4) = f(3)\, 7 \bmod 15 = 91 \bmod 15 = 1$

$f(5) = 7$

$f(6) = 4$

$f(7) = 13$

The period is $r = 4$

Solution: Use QPE on

$$U|y\rangle = |ay \bmod N\rangle$$

$$U|\psi\rangle = e^{2\pi i \theta}|\psi\rangle \rightarrow \text{find } \theta$$

How is an Eigenstate of U?

$a = 7$ and $N = 15$ $\quad U|y\rangle = |7y \bmod 15\rangle$

$U|1\rangle = |7\rangle$ $\quad U|7\rangle = |4\rangle = U^2|1\rangle =$

$U^2|1\rangle = |4\rangle$

$U^3|1\rangle = |13\rangle$

$U^4|1\rangle = |1\rangle$

$U^5|1\rangle = |7\rangle$

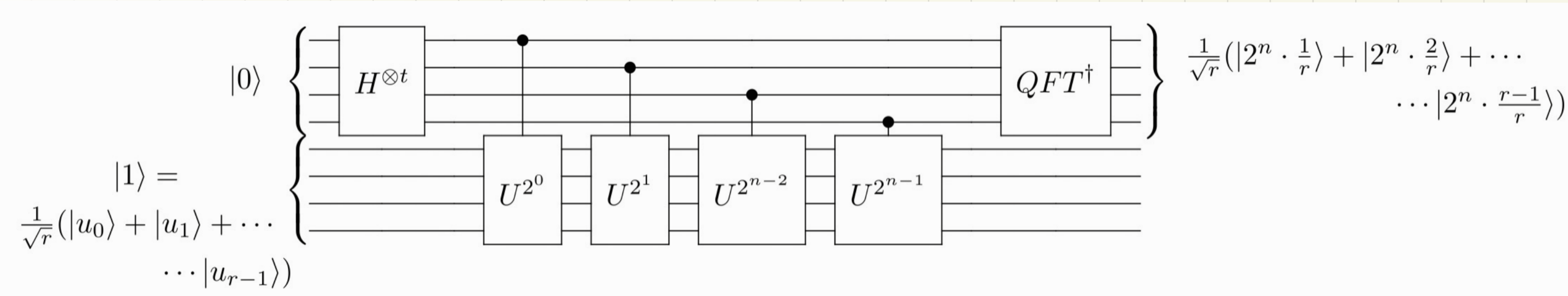
$$\rightarrow |u_0\rangle = \frac{1}{\sqrt{4}}\left(|7\rangle + |4\rangle + |13\rangle + |1\rangle\right)$$

$$U|u_0\rangle = \frac{1}{\sqrt{4}}\left(\overbrace{U|7\rangle}^{|4\rangle} + \overbrace{U|4\rangle}^{|13\rangle} + \overbrace{U|13\rangle}^{|1\rangle} + \overbrace{U|1\rangle}^{|7\rangle}\right)$$

$$= |u_0\rangle$$

Eigenvalue = 1

5. Shor's circuit



6. Factoring

Algorithm: Reduction of factoring to order-finding

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 $x^{r/2} \neq -1 (\mathrm{mod}\ N)$ 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.