Shor's Algorithm: Factoring Numbers on Quantum Computers

Talha Ashraf

10th July 2025

What is Shor's Algorithm?

1. It's an algorithm that can find factors of composite numbers on a Quantum computer in polynomial time. This is exponentially faster than classical algorithms.
2. Shor's algorithm is based on the fact that factorization can be reduced to the problem of order finding (discussed soon).
3. Order finding in turn can be computed efficiently on a Quantum computer using an algorithm called Quantum Phase Estimation.

Order Finding

Definition

Consider a number N that we want to factorize and a second number $1 < x < N$ that is co-prime to x. The order r is the smallest integer such that $x^r \bmod N = 1$.

Theorem

Given $x^r \bmod N = 1$, *We can prove that* $\gcd(x^{\frac{r}{2}} - 1, N)$ *gives a non trivial factor of* N, *with the exception of two cases.*

Proof that Order Finding gives Factors

Proof.

$x^r \bmod N = 1$ is equivalent to the equation

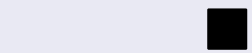
$$x^r = kN + 1 \implies x^r - 1 = kN \tag{1}$$

$$(x^{\frac{r}{2}} - 1)(x^{\frac{r}{2}} + 1) = kN \tag{2}$$

If $(x^{\frac{r}{2}} - 1)$ and $(x^{\frac{r}{2}} + 1)$ are integers, $(x^{\frac{r}{2}} - 1)$ is a factor of N, and $\gcd(x^{\frac{r}{2}} - 1, N)$ is a factor of N!

For them to be integers, we need 2 conditions to be satisfied:

1. $x^{\frac{r}{2}} - 1$ is an integer: r must be even.
2. We mustn't have $x^{\frac{r}{2}} = kN - 1$ or else the factor at the end is just N.



Order Finding Doesn't Fail Often

Theorem

Suppose a number $N = p_1^{s_1} p_2^{s^2} \dots p_n^{s_n}$, *and a randomly chosen number* x *which is coprime to it. Then their order* r *satisifes the conditions that* r *is even and* $x^{\frac{r}{2}} \neq 0 \mod N$ *with the probability*

$$P\left(r \text{ is even}, x^{\frac{r}{2}} \neq 0 \pmod{N}\right) \geq 1 - \frac{1}{2^n} \tag{3}$$

How do we find Orders?

1. In order to actually compute orders and find factors, we must develop an efficient Quantum algorithm for it.
2. Quantum algorithms rely heavily on a principle called Superposition, so let's discuss this next!

Superposition: The Heart of Quantum Technology

1. In a classical computer, we have bits: strings of 0s and 1s.
2. A Quantum computer has qubits. It also outputs 0s and 1s.
3. If a qubit in the state $|0\rangle$ is measured, we observe $|0\rangle$ and record it as a 0. If it's in the state $|1\rangle$ is measured, we observe $|1\rangle$ and record it as a 1.
4. The way qubits differ from normal bits is they can also be in a superosition of its states $|0\rangle$ and $|1\rangle$:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{4}$$

The rule is that when we measure this superposition, we will either see $|0\rangle$ with probability $|\alpha|^2$ or we'll see $|1\rangle$ with probability $|\beta|^2$. α and β are complex.

Transforming Qubits using Linear Algebra

1. We can use Quantum effects to manipulate qubit values to build algorithms.
2. Since a superposition can be fully specified by just α and β, we can write $|\psi\rangle$ as the vector

$$|\psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \alpha |0\rangle + \beta |1\rangle = \alpha \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

3. Quantum effects cause reversible linear linear transformations

$$U\Big(|x\rangle + |y\rangle \Big) = U|x\rangle + U|y\rangle$$

4. All linear transformations can be represented as a matrix. So Quantum gates are invertible matrices.
5. By the way, strings of multiple qubits are represented as follows:

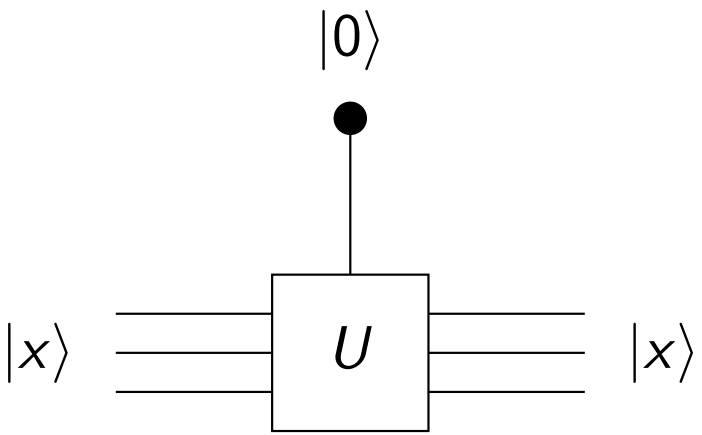
$$|\psi\rangle = |x_1\rangle \otimes |x_2\rangle \otimes |x_3\rangle \otimes \ldots |x_n\rangle$$

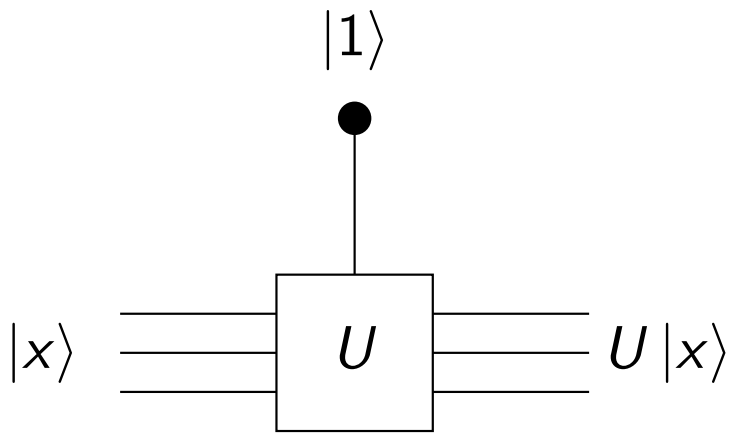
2 Useful Gates

1. The Hadamard gate is defined as

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad H \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \quad H \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

2. A controlled U gate applies the matrix U to the target bit if the control bit is $|1\rangle$ but does nothing if the control bit is $|0\rangle$.





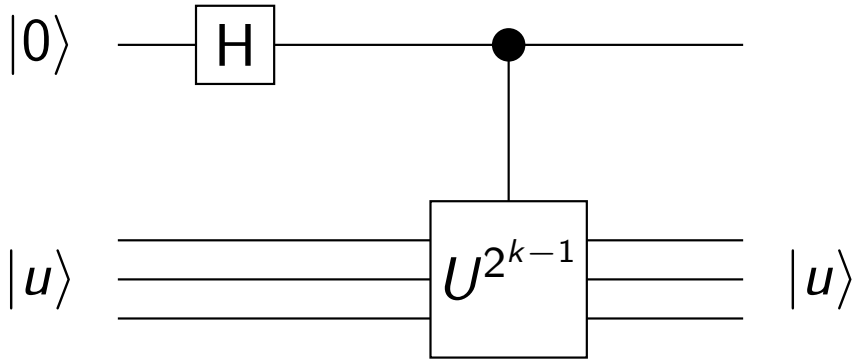
Quantum Phase Estimation

1. Quantum gates have eigenvalues of the form $e^{i2\pi\phi}$:

$$U|u\rangle = e^{i2\pi\phi}|u\rangle .$$

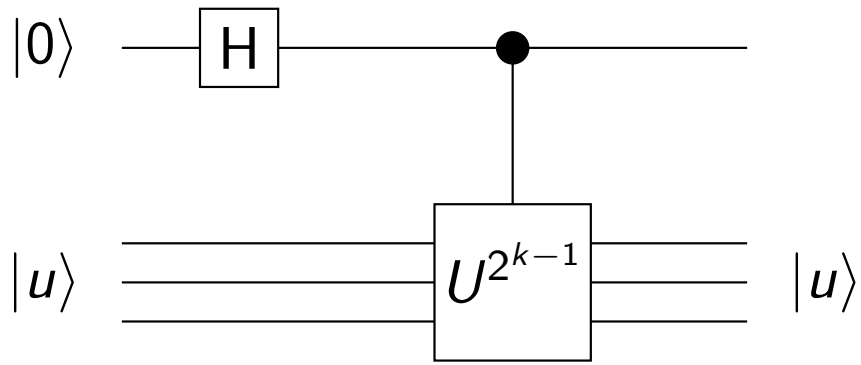
2. Quantum Phase Estimation is an algorithm that can estimate the phase ϕ for a given matrix of this form.
3. The QPE algorithm is essential for order finding. Let's go through its circuit together.

Quantum Phase Estimation 2



- Here the controlled Quantum gate $U^{2^{k-1}}$ applies the matrix U 2^{k-1} times to its eigenvector $|u\rangle$, but only if the control bit is $|0\rangle$ since it's a controlled gate

Quantum Phase Estimation 2



- This circuit is used repeatedly in QPE. It transforms the target qubit into

$$U^{2^{k-1}}\left[\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right]|u\rangle = \frac{1}{\sqrt{2}}\left(U^{2^{k-1}}|0\rangle\otimes|u\rangle + U^{2^{k-1}}|1\rangle\otimes|u\rangle\right)$$
$$= \frac{1}{\sqrt{2}}\left(|0\rangle\otimes|u\rangle + |1\rangle\otimes e^{i2\pi 2^{k-1}\phi}|u\rangle\right)$$
$$= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{i2\pi 2^{k-1}\phi}|1\rangle\right)\otimes|u\rangle. \quad (5)$$

Quantum Phase Estimation 3

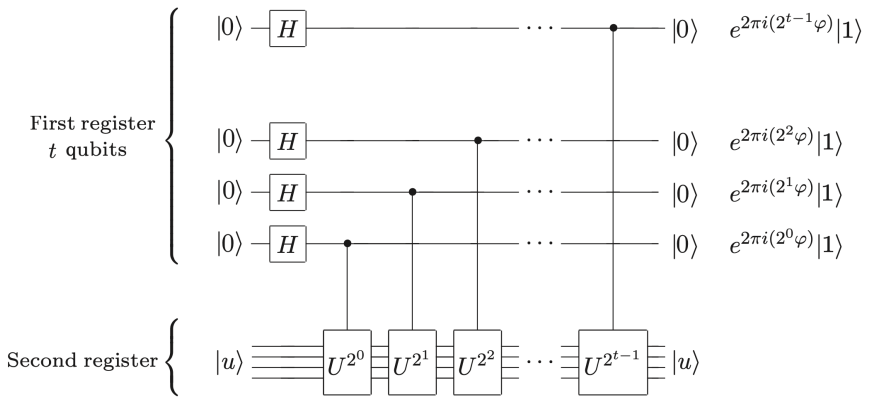


Figure: Credit to Nielson's Quantum Computing for the image.

1. The real QPE procedure is n of these circuits stacked together, for n qubits, as show in the figure.
2. Hence the final state is

$$\frac{1}{2^{\frac{n}{2}}}\left(|0\rangle + e^{i2\pi 2^{n-1}\phi}|1\rangle\right)\otimes\left(|0\rangle + e^{i2\pi 2^{n-2}\phi}|1\rangle\right)\otimes$$
$$\ldots\left(|0\rangle + e^{i2\pi 2^{1}\phi}|1\rangle\right)\otimes\left(|0\rangle + e^{i2\pi 2^{0}\phi}|1\rangle\right)\otimes|u\rangle\,.$$

Quantum Fourier Transform 1

1. Now we have n qubits in superposition, each containing the phase ϕ. But how do we actually recover the value of ϕ from the superpostion.
2. The last result

$$\frac{1}{2^{\frac{n}{2}}}\left(|0\rangle + e^{i2\pi 2^{n-1}\phi}|1\rangle\right) \otimes \left(|0\rangle + e^{i2\pi 2^{n-2}\phi}|1\rangle\right) \otimes \ldots \left(|0\rangle + e^{i2\pi 2^{0}\phi}|1\rangle\right)$$

   can be rewritten as

$$\frac{1}{2^{\frac{n}{3}}} \sum_{k=0}^{2^n-1} e^{i2\pi k\phi}|k\rangle .$$

3. But this looks so similar to the discrete Fourier Transform!

Quantum Fourier Transform 2

1. There is in fact another Quantum algorithm named the Quantum Fourier Transform that implements the exact expression we just saw:

$$U_{QFT} |x_1\rangle |x_2\rangle \dots |x_n\rangle = \frac{1}{2^{\frac{n}{3}}} \sum_{k=0}^{2^n-1} e^{i2\pi kx} |k\rangle .$$

2. Since all Quantum gates are reversible matrices, there is an inverse QFT U_{QFT}^{-1}.
3. If we apply the Quantum Phase Estimation procedure, then the inverse QFT will give us an estimate for ϕ with n qubits:

$$U_{QFT}^{-1} U_{QPE} |0\rangle |0\rangle \dots |0\rangle = U_{QFT}^{-1} \sum_{k=0}^{2^n-1} e^{i2\pi k\phi} |k\rangle = \boxed{\phi}.$$

Computing Orders

1. If we want to find the order of x and N, we can consider the Quantum gate
$$U|y\rangle = |xy \bmod N\rangle .$$
2. It can be proven this matrix has eigenvector $|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{i\frac{2\pi sk}{r}} |x^k \bmod N\rangle$ and eigenvalue $e^{i\frac{2\pi s}{r}}$.
3. Applying the phase estimation procedure to this gives us the value $\frac{s}{r}$ and hence tells us r, the order of x and N!
4. Finally $\gcd(x^{\frac{r}{2}} - 1, N)$ gives us a non trivial factor of N and hence we have succeeded in factoring N.

Shor's Algorithm

1. If N is even, return 2 as a factor.
2. Check if N is of the form $N = p^m$ a classical algorithm. If true, we return p and end our algorithm.
3. Choose a random integer in the range $1 < x < N$. If $\gcd(x, N) > 1$, then we have hit the jackpot and we can return $\gcd(x, N)$. If instead, $\gcd(x, N) = 1$, then x is coprime to N and we proceed to step 4.
4. We now find the order r of $x \mod N$ using the order finding procedure.
5. We check if r is even and if $x^{\frac{r}{2}} \neq -1 \mod N$. If these conditions hold, we can compute $\gcd(x^{\frac{r}{2}} - 1, N)$ and divide N by it to test if it is a non trivial factor of N. If it is, then we return this as a factor. If either condition does not hold, however, our algorithm has failed. We must try again with a new choice of x.