Shor's Algorithm

Tiffany Ding

April 23, 2025

1 Quantum Primer

Qubits and quantum states. We use $|v\rangle$ to refer to a quantum state called "v".

1. *Qubit.* There are two special qubits:

$$|0\rangle := \begin{bmatrix}1\\0\end{bmatrix} \qquad |1\rangle := \begin{bmatrix}0\\1\end{bmatrix}$$

   It is worth emphasizing that $|0\rangle$ and $|1\rangle$ are two-dimensional vectors, and $|0\rangle$ does *not* refer to the all 0s vector! A general qubit $|\psi\rangle$ is a *superposition* of the two special states:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \quad \text{where } \alpha, \beta \in \mathbb{C},\ |\alpha|^2 + |\beta|^2 = 1$$

2. n-qubit quantum states. The joint state of qubit $|u\rangle$ and qubit $|v\rangle$ is represented as $|u\rangle \otimes |v\rangle$, where $\otimes$ denotes the *tensor product*[1].The state of n qubits lives in a 2^n-dimensional vector space.

   - *A note on notation*: The following are equivalent

$$|0\rangle \otimes |0\rangle\,,\ \ |0\rangle \otimes |1\rangle\,,\ \ |1\rangle \otimes |0\rangle\,,\ \ |1\rangle \otimes |1\rangle$$

$$|0\rangle\,|0\rangle\,,\ \ |0\rangle\,|1\rangle\,,\ \ |1\rangle\,|0\rangle\,,\ \ |1\rangle\,|1\rangle$$

$$|0\rangle\,,\ \ |1\rangle\,,\ \ |2\rangle\,,\ \ |3\rangle$$

$$|0\rangle_2\,,\ \ |1\rangle_2\,,\ \ |2\rangle_2\,,\ \ |3\rangle_2$$

     Similar equivalences can be extracted to n-qubit quantum state for $n > 2$.

Operations on qubits.

- $\langle v|$ is the *conjugate transpose*[2] of $|v\rangle$
- $\langle u|\,|v\rangle$ is the *inner product* of $|u\rangle$ and $|v\rangle$
- $|u\rangle\,\langle v|$ is the *outer product* of $|u\rangle$ and $|v\rangle$

[1] Let $|a\rangle \in \mathbb{C}^m$ and $|b\rangle \in \mathbb{C}^n$. Their tensor product $|a\rangle \otimes |b\rangle \in \mathbb{C}^{mn}$ is defined as:

$$|a\rangle \otimes |b\rangle = \begin{bmatrix} a_1b_1 \\ a_1b_2 \\ \vdots \\ a_2b_1 \\ \vdots \\ a_mb_n \end{bmatrix}$$

[2] Recall that the complex conjugate of $(u + iv) \in \mathbb{C}$ is $u - iv$ (where $u, v \in R$).

Measurement. Measuring a state $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ yields

$$\begin{cases} |0\rangle & \text{with probability } |\alpha|^2 \\ |1\rangle & \text{with probability } |\beta|^2 \end{cases}$$

Gates. Gates map quantum states to quantum states. There are a few important one-qubit gates. One of them is the Hadamard gate

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

When applied to $|0\rangle$ and $|1\rangle$, you get $H\,|0\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and $H\,|1\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}}$. When applied to an arbitrary state $|x\rangle$, you get $H^{\otimes n}\,|x\rangle = \frac{1}{\sqrt{2^n}}\sum_{z\in\{0,1\}^n}(-1)^{x\cdot z}\,|z\rangle$.

An important property of H is that applying it to each qubit in $|0\rangle^{\otimes n}$ (n sequential $|0\rangle$'s) yields the uniform superposition over all possible binary strings of length h:

$$H{\otimes}n\,|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle$$

An important two-qubit cate is the CNOT gate (Controlled-NOT):

$$\text{CNOT}\,|c\rangle\,|t\rangle = |c\rangle\,|t\oplus c\rangle$$

where $\oplus$ denotes *bit-wise exclusive OR (XOR)*, i.e., the i-th entry of $x\oplus y$ is 1 iff exactly one of x_i and y_i is 1.

Registers and unitary encodings of functions. In many quantum algorithms, we are given access to a classical function $f : \{0,1\}^n \to \{0,1\}^m$. Quantum transformations have to be reversible, so we can construct an *unitary encoding of* f, denoted U_f, which is a unitary operator that acts on two quantum registers: an input register and an output register. It is defined as:

$$U_f : |x\rangle\,|0\rangle \mapsto |x\rangle\,|f(x)\rangle$$

where $x \in \{0,1\}^n$ and $y \in \{0,1\}^m$. See p. 36-37 of Mermin (2007).

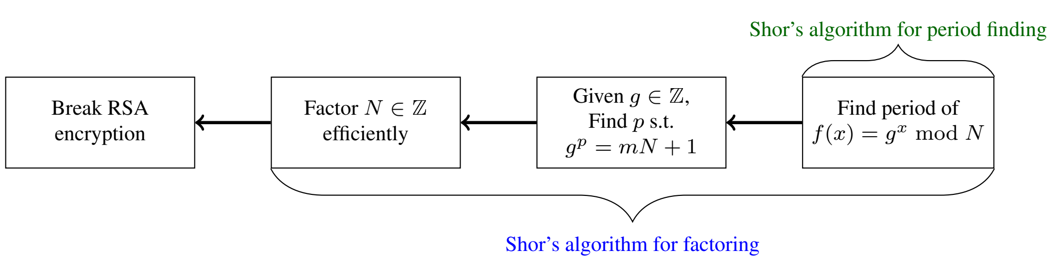
2 Shor's Algorithm

Shor's algorithm provides a way to efficiently factor integers using a quantum computer. This can be used to break RSA encryption.

Remark. "Shor's algorithm" could refer to two closely related algorithms:

1. Shor's algorithm for period-finding (all of the quantum magic happens here!)
2. Shor's algorithm for factoring = Shor's algorithm for period-finding + some (classical) operations based on number theory.

Generally, people mean the latter when they say "Shor's algorithm."



Preliminaries.

- For integers m and n, let $\gcd(m, n)$ denote the largest integer that divides both of them ("greatest common divisor"). This can be computed efficiently using Euclid's algorithm.[3]
- *Modular arithmetic.*
$$a \equiv b \mod N \quad \text{means} \quad a \mod N = b \mod N$$

Shor's algorithm for factoring.

- *Goal:* Find factors of an integer N (excluding 1 and N)[4].
- *Algorithm*
  1. Uniformly at random pick an integer $a \in \{2, 3, \ldots, N-1\}$. Compute $\gcd(a, N)$. If $\gcd(a, N) > 1$, we are done: we have found a factor!
  2. Apply Shor's algorithm for period-finding to the function $f(x) = a^x \mod N$ to find its period r.
  3. Check if r is odd. If so, return to Step 1. (Otherwise, Step 5 would yield non-integer factors)
  4. Check if $a^{r/2} \equiv -1 \mod N$ (equivalently, $a^{r/2} \mod N = N - 1$). If so, return to Step 1. (Otherwise, Step 5 would yield the trivial factor N)
  5. Given that r is even and $a^{r/2} \not\equiv -1 \mod N$,
  $$\gcd(a^{r/2} - 1, N) \quad \text{and} \quad \gcd(a^{r/2} + 1, N)$$
  yield non-trivial integer factors of N.
- *Why does this work?* When a and N share no factors, it is guaranteed that if you multiply a by itself enough times, you will eventually get a number that is a multiple of N + 1. In other words, there exists $p, m \in \mathbb{Z}$ such that
$$a^p = mN + 1.$$
Rearranging, we get $(a^p - 1) = mN$, so
$$(a^{p/2} - 1)(a^{p/2} + 1) = mN.$$
In other words, $a^{p/2} - 1$ and $a^{p/2} + 1$ are factors of mN. Taking the gcd of these with N yields fators of N.

Shor's algorithm for period-finding. We are given a function $f : \mathbb{Z}_N \to S$ such that[5]
$$f(x) = f(y) \iff x \equiv y \mod r.$$
Our goal is to find the period r efficiently.

1. *Determine the necessary register sizes.* The 2nd register should be large enough to store any integer in $\{0, 1, \ldots, N-1\}$ = $\log_2 N := n$ bits. The 1st register should be large enough so that we have sufficient precision to recover r using measurements of the 1st register in the later steps of the algorithm. For now let's denote the number of 1st register bits by m, which can represent integers up to $M - 1$ where $M := 2^m$. We will talk about how m is determined later.
2. *Initialize both registers to 0.*
$$|\psi_0\rangle = |0\rangle_m |0\rangle_n$$
3. *Apply Hadamards to first register to create a uniform superposition*[6].
$$|\psi_1\rangle = \frac{1}{\sqrt{M}} \sum_{x=0}^{M-1} |x\rangle |0\rangle_n$$

[3] The key idea of Euclid's algorithm is the greatest common divisor of two numbers does not change if the larger number is replaced by its difference with the smaller number.

[4] Classically, factoring is believed to require super-polynomial time (no known classical polynomial-time algorithm exists), and it forms the basis of RSA cryptography. Shor's algorithm solves the factoring problem in *polynomial time* on a quantum computer.

[5] Roughly, $\mathbb{Z}_N = \{0, 1, \ldots, N-1\}$ and S is any finite set.

[6] This can also be accomplished by taking the QFT of the first register. We will define QFT shortly!

4. *Apply the unitary encoding of f, that is, $U_f : |x\rangle |0\rangle \mapsto |x\rangle |f(x)\rangle$*

$$|\psi_2\rangle = \frac{1}{\sqrt{M}} \sum_{x=0}^{M-1} |x\rangle |f(x)\rangle$$

5. *Measure the second register.* This collapses the second register to some value $|f(x_0)\rangle$ and also reveals information about $|x\rangle$ because now $|x\rangle$ is constrained to the pre-image of $|f(x_0)\rangle$. Because f is periodic, the pre-image of $f(x_0)$ is $\{x_0, x_0 + r, x_0 + 2r, \ldots, x_0 + (q-1)r\}$ where $q :=$ largest integer such that $x_0 + (q-1)r < N$. Thus, the first register is now in uniform superposition over the pre-image of $|f(x_0)\rangle$, so we have

$$|\psi_3\rangle = \frac{1}{\sqrt{q}} \sum_{j=0}^{q-1} |x_0 + jr\rangle |f(x_0)\rangle$$

6. *Apply the quantum Fourier transform (QFT) to the first register.*

   - **Background on quantum Fourier transforms.**

     – *Definition.* The *quantum Fourier transform (QFT)* is a unitary transformation U_{FT}, where its action on a basis state $|j\rangle_m \in \{|0\rangle, |1\rangle, |2\rangle, \ldots, |M-1\rangle\}$, where $M := 2^m$, is defined as

     $$U_{\text{FT}} |j\rangle_m = \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} e^{2\pi ijk/M} |k\rangle_m$$

     Since $|j\rangle$ is a basis state, the jk in the superscript is simply normal multiplication. Since QFT is a *linear* transformation, the QFT of a generic quantum state $|x\rangle = \sum_{j=0}^{M-1} \alpha_j |j\rangle$ is

     $$\begin{aligned} U_{\text{FT}} |x\rangle &= \sum_{j=0}^{M-1} \alpha_j U_{\text{FT}} |j\rangle \\ &= \sum_{j=0}^{M-1} \alpha_j \left(\frac{1}{\sqrt{N}} \sum_{k=0}^{M-1} e^{2\pi ijk/M} |k\rangle \right) \end{aligned}$$

     Swapping the order of summation,

     $$= \sum_{k=0}^{M-1} \left(\frac{1}{\sqrt{M}} \sum_{j=0}^{M-1} \alpha_j e^{2\pi ijk/M} \right) |k\rangle$$

     – *Time complexity.* Importantly, computing the QFT of an n-bit qubit only takes $O(n^2)$ time.

   - **Applying the QFT to the first register**

$$\begin{aligned} U_{\text{FT}} \frac{1}{\sqrt{q}} \sum_{j=0}^{q-1} |x_0 + jr\rangle &= \frac{1}{\sqrt{q}} \sum_{j=0}^{q-1} U_{\text{FT}} |x_0 + jr\rangle && (1) \\ &= \frac{1}{\sqrt{q}} \sum_{j=0}^{q-1} \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} e^{2\pi i(x_0+jr)k/M} |k\rangle_m && (2) \\ &= \frac{1}{\sqrt{Mq}} \sum_{k=0}^{M-1} \sum_{j=0}^{q-1} e^{2\pi i(x_0+jr)k/M} |k\rangle_m \qquad \text{swapping order of summations} && (3) \\ &= \frac{1}{\sqrt{Mq}} \sum_{k=0}^{M-1} A(k) |k\rangle_m \qquad \text{where } A(k) := \sum_{j=0}^{q-1} e^{2\pi i(x_0+jr)k/M} && (4) \end{aligned}$$

We will show that the probability amplitudes $A(k)$ are large only for $|k\rangle$ satisfying $\frac{rK}{M} \in \mathbb{Z}$. We can write

$$A(k) = e^{2\pi i x_0 k/M} \underbrace{\sum_{j=0}^{q-1} e^{2\pi i j r k/M}}_{\substack{\text{behavior of } A(k) \text{ is} \\ \text{determined by this sum}}}$$

We will simplify this sum by noting the following two facts:

$$\text{(Fact 1)} \qquad 1 + a + a^2 + \cdots + a^q = \begin{cases} \frac{1-a^q}{1-a} & \text{for } |a| < 1 \\ q & \text{for } a = 1 \end{cases}$$

$$\text{(Fact 2)} \qquad e^{\frac{2\pi i j r k}{M}} = 1 \text{ for all } j = 0, 1, \ldots, q \qquad \iff \qquad \frac{rk}{M} \in \mathbb{Z}$$

Together, these two facts tell us that

$$\sum_{j=0}^{q-1} e^{2\pi i j r k/M} = \begin{cases} q & \text{if } \frac{rk}{M} \in \mathbb{Z} \\ \frac{1-e^{\frac{2\pi i r k}{M} q}}{1-e^{\frac{2\pi i r k}{M}}} & \text{otherwise} \end{cases}$$

The $\frac{rk}{M} \in \mathbb{Z}$ case corresponds to *constructive interference*. The $\frac{rk}{M} \notin \mathbb{Z}$ corresponds to *destructive interference*. Using Euler's formula + some algebra, we can show that $\frac{1-e^{\frac{2\pi i r k}{M} q}}{1-e^{\frac{2\pi i r k}{M}}}$ is small compared to q.

Main takeaway: After applying the QFT to the first register, we get a superposition where the probability amplitudes of $|k\rangle$ for $k \approx$ multiple of $\frac{M}{r}$ (i.e., $k \approx \frac{\ell M}{r}$ for $\ell \in \mathbb{Z}$) are large and all other amplitudes are close to 0. As a result, measuring the first register is highly likely to yield a value that is (close to) a multiple of $\frac{M}{r}$.

7. *Measure the first register.* Call the obtained measurement y.
8. *Repeat Steps 2-7 several times to obtain samples* $y_1, y_2, y_3, \ldots$. We know that w.h.p. all of these y_i's satisfy

$$y_i \approx \frac{\ell_i}{M} r \quad \text{for some } \ell_i \in \mathbb{Z}$$
$$\iff \quad \frac{y_i}{M} \approx \frac{\ell_i}{r}$$

To attempt to recover $\frac{l_i}{r}$ from $\frac{y_i}{M}$, we apply the *continued fraction algorithm*.

$$\frac{y_i}{M} \xrightarrow{\text{continued frac.}} \frac{b_i}{c_i} \qquad \text{for some } b_i, c_i \in \mathbb{Z}$$

We will state the following fact without proof:

- **Fact:** If $\left|\frac{y_i}{M} - (\text{multiple of } \frac{1}{r})\right| \leq \frac{1}{2r^2}$, the continued fraction algorithm recovers $c_i = r$ (up to lost factors).[7]

From each y_i, we obtain a c_i. W.h.p., the least common multiple of $c_1, c_2, \ldots$ will be equal to r.

9. Repeat Steps 2-8 to make sure we have indeed recovered r.

References. This minutephysics video and online textbook give a good high-level overview. The presentation in these CMU lecture notes is also nice. For an in-depth description, see Chapter 3 of Mermin (2007),

References

N David Mermin. *Quantum computer science: an introduction*. Cambridge University Press, 2007.

[7]This is where the size of the first register is important. To ensure that this inequality is satisfied, it can be shown that $m = 2n$ bits is sufficient.