Phase Estimation and Factoring

Brief linear algebra review

Eigenvalues

If $M \in \mathbb{C}^{N \times N}$ is a matrix, $|\psi\rangle \in \mathbb{C}^N$ is a nonzero vector, and $\lambda \in \mathbb{C}$ satisfying

$$M|\psi\rangle = \lambda|\psi\rangle$$

then we say that $|\psi\rangle$ is an **eigenvector** of M with **eigenvalue** λ.

Eigenvalues of unitary matrices

Fact: The eigenvalues of a unitary matrix U are all of the form $e^{2\pi i\theta}$ for some $\theta \in [0, 1)$.

Eigenvalues of unitary matrices

Fact: The eigenvalues of a unitary matrix U are all of the form $e^{2\pi i\theta}$ for some $\theta \in [0, 1)$.

Proof: Suppose that $U|\psi\rangle = \lambda|\psi\rangle$ for some unit eigenvector $|\psi\rangle$ and some eigenvalue λ.

Eigenvalues of unitary matrices

Fact: The eigenvalues of a unitary matrix U are all of the form $e^{2\pi i\theta}$ for some $\theta \in [0, 1)$.

Proof: Suppose that $U|\psi\rangle = \lambda|\psi\rangle$ for some unit eigenvector $|\psi\rangle$ and some eigenvalue λ.

Taking inner products of $\lambda|\psi\rangle$ with itself, on one hand we get

$$(\lambda^*\langle\psi|)(\lambda|\psi\rangle) = |\lambda|^2\langle\psi|\psi\rangle = |\lambda|^2 .$$

Eigenvalues of unitary matrices

Fact: The eigenvalues of a unitary matrix U are all of the form $e^{2\pi i\theta}$ for some $\theta \in [0, 1)$.

Proof: Suppose that $U|\psi\rangle = \lambda|\psi\rangle$ for some unit eigenvector $|\psi\rangle$ and some eigenvalue λ.

Taking inner products of $\lambda|\psi\rangle$ with itself, on one hand we get

$$(\lambda^*\langle\psi|)(\lambda|\psi\rangle) = |\lambda|^2\langle\psi|\psi\rangle = |\lambda|^2 .$$

On the other hand,

$$(\lambda^*\langle\psi|)(\lambda|\psi\rangle) = (\langle\psi|U^\dagger)(U|\psi\rangle) = \langle\psi|U^\dagger U|\psi\rangle = \langle\psi|\psi\rangle = 1$$

because $U^\dagger U = I$ (one of definitions of being unitary).

Eigenvalues of unitary matrices

Fact: The eigenvalues of a unitary matrix U are all of the form $e^{2\pi i\theta}$ for some $0 \leq \theta < 1$.

Proof continued: Therefore

$$|\lambda|^2 = 1$$

and the only such λ's possible are of the form $e^{2\pi i\theta}$.

Some examples

Example: What are the eigenvalues and eigenvectors of

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Some examples

Example: What are the eigenvalues and eigenvectors of

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

We see that

$$Z\,|0\rangle = |0\rangle \qquad Z\,|1\rangle = -\,|1\rangle \ .$$

Therefore standard basis are the eigenvectors and ± 1 are corresponding eigenvalues.

Some examples

Example: What are the eigenvalues and eigenvectors of

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} .$$

Some examples

Example: What are the eigenvalues and eigenvectors of

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} .$$

We can compute this by hand, or we can also remember that

$$X\,|+\rangle = |+\rangle \qquad X\,|-\rangle = -\,|-\rangle$$

so the Hadamard basis are the eigenvectors and ± 1 are the corresponding eigenvalues.

Phase Estimation Algorithm

Phase Estimation

Phase Estimation Algorithm (PEA) is one of the most important subroutines in quantum computing.

Phase Estimation

Phase Estimation Algorithm (PEA) is one of the most important subroutines in quantum computing.

Goal of PEA:

- Ability to run controlled versions of U^k for $k = 1, 2, \ldots$.
- An **eigenstate** $|\psi\rangle$ where $U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$,

estimate θ.

Question: The eigenvalue $e^{2\pi i\theta}$ looks like a global phase... how can you possibly estimate it?

Question: The eigenvalue $e^{2\pi i\theta}$ looks like a global phase... how can you possibly estimate it?

Answer: It becomes a **relative** phase once you run the controlled-U gate in superposition:

$$\begin{aligned} cU\left|+\right\rangle\left|\psi\right\rangle &= \frac{1}{\sqrt{2}}(\left|0\right\rangle\left|\psi\right\rangle + \left|1\right\rangle U\left|\psi\right\rangle) \\ &= \frac{1}{\sqrt{2}}(\left|0\right\rangle\left|\psi\right\rangle + e^{2\pi i\theta}\left|1\right\rangle\left|\psi\right\rangle) \\ &= \frac{1}{\sqrt{2}}(\left|0\right\rangle + e^{2\pi i\theta}\left|1\right\rangle)\left|\psi\right\rangle \end{aligned}$$

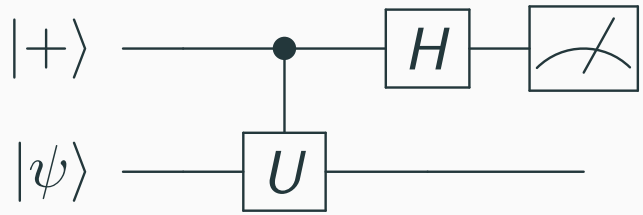
Warmup towards Phase Estimation

Let U be a unitary with an eigenvector $|\psi\rangle$ whose corresponding eigenvalue is either $+1$ or -1. How to tell which is the case, given one copy of $|\psi\rangle$ and the ability to apply controlled versions of U?

Warmup towards Phase Estimation

Let U be a unitary with an eigenvector $|\psi\rangle$ whose corresponding eigenvalue is either $+1$ or -1. How to tell which is the case, given one copy of $|\psi\rangle$ and the ability to apply controlled versions of U?

A "baby" form of phase estimation:



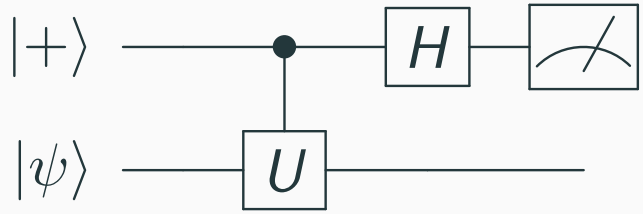
When $|\psi\rangle$ is a $+1$-eigenvector of U, the output is $|0\rangle$. When it is a -1-eigenvector, the output is $|1\rangle$.

Warmup to Phase Estimation

What if the phase were $\exp(2\pi i\theta)$ for some $0 \leq \theta < 1$?

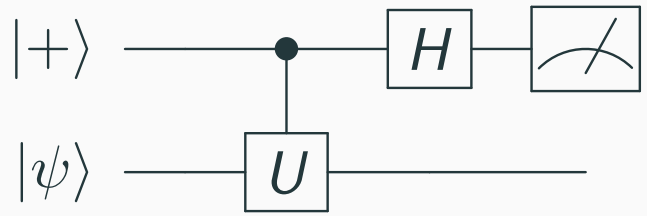
Warmup to Phase Estimation

What if the phase were $\exp(2\pi i\theta)$ for some $0 \leq \theta < 1$? We can analyze the same circuit:



Warmup to Phase Estimation

What if the phase were $\exp(2\pi i\theta)$ for some $0 \leq \theta < 1$? We can analyze the same circuit:

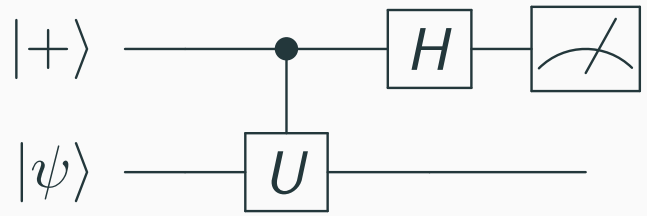


The state of top qubit before measurement is:

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

Warmup to Phase Estimation

What if the phase were $\exp(2\pi i\theta)$ for some $0 \leq \theta < 1$? We can analyze the same circuit:



The state of top qubit before measurement is:

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

Measuring this qubit yields

$$\Pr[|0\rangle] = \left|\frac{1+e^{2\pi i\theta}}{2}\right|^2 = \cdots \text{ high school trig } \cdots = \cos^2(\pi\theta) \ .$$

The state $|\psi\rangle$ is undisturbed after running the circuit. So we can repeat it multiple times with fresh ancilla qubits to get an estimate of θ.

By repeating the phase estimation circuit $O(1/\epsilon^2)$ times, can obtain an estimate of $\cos^2(\pi\theta) \pm \epsilon$ with constant confidence. Does this uniquely identify θ?

The state $|\psi\rangle$ is undisturbed after running the circuit. So we can repeat it multiple times with fresh ancilla qubits to get an estimate of θ.

By repeating the phase estimation circuit $O(1/\epsilon^2)$ times, can obtain an estimate of $\cos^2(\pi\theta) \pm \epsilon$ with constant confidence. Does this uniquely identify θ?

No: There is ambiguity between θ and $1 - \theta$:

$$\cos^2(\pi\theta) = \cos^2(\pi(1-\theta)) \ .$$

In other words, this estimation procedure cannot distinguish between whether θ is smaller or bigger than $\frac{1}{2}$.

How to uniquely identify θ?

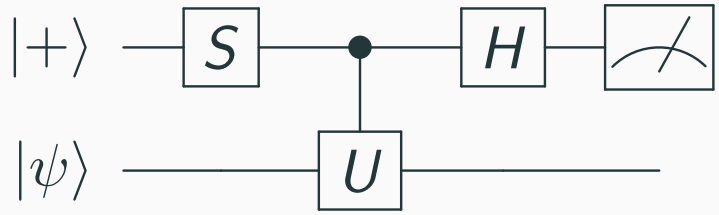
How to uniquely identify θ?

Suppose, in addition to having a good estimate of $\cos^2(\pi\theta)$, we also knew (a good estimate of)

$$\cos^2(\pi\theta + \frac{\pi}{4}) \ .$$

This is enough to recover θ! (Proof by picture on board).

Thus, after estimating $\cos^2(\pi\theta)$ using the first circuit, we can run a different circuit to get an estimate of $\cos^2(\pi\theta + \frac{\pi}{4})$:



where $S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$. You can show that the probability of getting $|0\rangle$ after measurement is

$$\Pr[|0\rangle] = \cos^2(\pi\theta + \frac{\pi}{4})$$

as desired.

Using the two circuits above, if we get charged \$1 each time we query controlled U, we can obtain $\pm\epsilon$ approximations of θ by spending $O(1/\epsilon^2)$ dollars (with constant confidence). This is fine for many applications, but for Shor's factoring algorithm, we need something much, much cheaper.

Using the two circuits above, if we get charged \$1 each time we query controlled U, we can obtain $\pm\epsilon$ approximations of θ by spending $O(1/\epsilon^2)$ dollars (with constant confidence). This is fine for many applications, but for Shor's factoring algorithm, we need something much, much cheaper.

If instead we have the ability to query controlled U^k for arbitrarily large k for \$1, then we can get $\pm\epsilon$ approximations of θ using $O(\log 1/\epsilon)$ dollars. Exponentially cheaper!

Using the two circuits above, if we get charged \$1 each time we query controlled U, we can obtain $\pm\epsilon$ approximations of θ by spending $O(1/\epsilon^2)$ dollars (with constant confidence). This is fine for many applications, but for Shor's factoring algorithm, we need something much, much cheaper.

If instead we have the ability to query controlled U^k for arbitrarily large k for \$1, then we can get $\pm\epsilon$ approximations of θ using $O(\log 1/\epsilon)$ dollars. Exponentially cheaper!

Main idea: estimate θ bit-by-bit.

Phase Estimation Algorithm

Assume for simplicity that θ can be represented using exactly t bits. In other words the binary representation of θ looks like

$$\theta = 0.\theta_1\theta_2\cdots\theta_t$$

where $\theta_1, \theta_2, \ldots \in \{0, 1\}$. This is equivalent to

$$\theta = \frac{\theta_1}{2} + \frac{\theta_2}{2^2} + \cdots + \frac{\theta_t}{2^t}.$$

Phase Estimation Algorithm

First we will estimate $\theta_t \in \{0, 1\}$. Let $k = 2^{t-1}$. Since $U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$, we have

$$U^k|\psi\rangle = e^{2\pi ik\theta}|\psi\rangle \ .$$

Phase Estimation Algorithm

First we will estimate $\theta_t \in \{0, 1\}$. Let $k = 2^{t-1}$. Since $U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$, we have

$$U^k|\psi\rangle = e^{2\pi ik\theta}|\psi\rangle \ .$$

But notice that

$$k\theta = \frac{k\theta_1}{2} + \frac{k\theta_2}{2^2} + \cdots + \frac{k\theta_t}{2^t} = \underbrace{2^{t-2}\theta_1 + \cdots + \theta_{t-1}}_{\text{integer}} + \frac{\theta_t}{2} \ .$$

Phase Estimation Algorithm

First we will estimate $\theta_t \in \{0, 1\}$. Let $k = 2^{t-1}$. Since $U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$, we have

$$U^k|\psi\rangle = e^{2\pi ik\theta}|\psi\rangle \ .$$

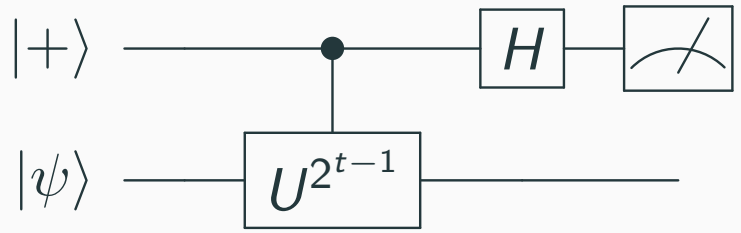
But notice that

$$k\theta = \frac{k\theta_1}{2} + \frac{k\theta_2}{2^2} + \cdots + \frac{k\theta_t}{2^t} = \underbrace{2^{t-2}\theta_1 + \cdots + \theta_{t-1}}_{\text{integer}} + \frac{\theta_t}{2} \ .$$

Therefore

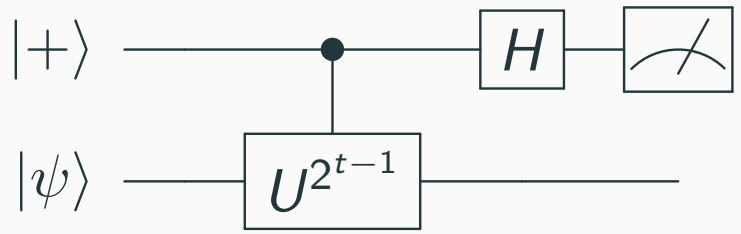
$$e^{2\pi ik\theta} = e^{2\pi i\frac{\theta_t}{2}} \in \{+1, -1\} \ .$$

If we run this circuit



the control qubit will be $|\theta_t\rangle$. We have learned one bit about θ!

If we run this circuit



the control qubit will be $|\theta_t\rangle$. We have learned one bit about θ!

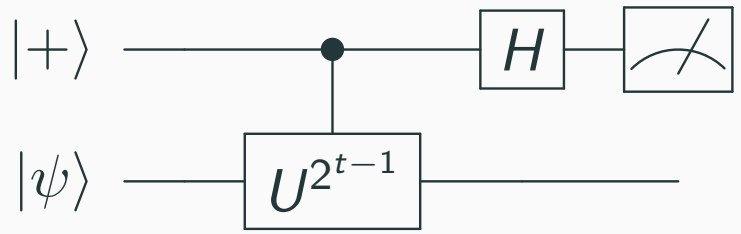
Consider the unitary

$$V = e^{-2\pi i \frac{\theta_t}{2^t}} U \ .$$

which satisfies

$$V|\psi\rangle = e^{-2\pi i \frac{\theta_t}{2^t}} U|\psi\rangle = e^{2\pi i(\theta - \frac{\theta_t}{2^t})}|\psi\rangle \ .$$

If we run this circuit



the control qubit will be $|\theta_t\rangle$. We have learned one bit about θ!

Consider the unitary

$$V = e^{-2\pi i \frac{\theta_t}{2^t}} U \ .$$

which satisfies

$$V|\psi\rangle = e^{-2\pi i \frac{\theta_t}{2^t}} U|\psi\rangle = e^{2\pi i(\theta - \frac{\theta_t}{2^t})}|\psi\rangle \ .$$

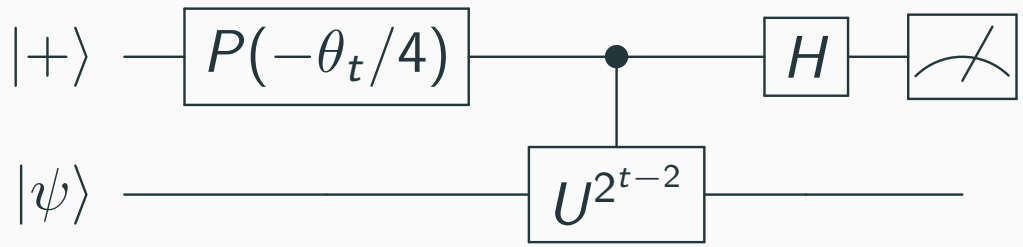
Notice that

$$\theta - \frac{\theta_t}{2^t} = \frac{\theta_1}{2} + \frac{\theta_2}{4} + \cdots + \frac{\theta_{t-1}}{2^{t-1}} \ .$$

We can try to learn θ_{t-1} next by doing phase estimation on

$$V^{2^{t-2}} = e^{-2\pi i \frac{\theta_t}{4}} U^{2^{t-2}}.$$

using the following circuit:



where

$$P(\alpha) = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i \alpha} \end{pmatrix}.$$

We can continue in this manner until we learn all the bits of θ.

The number of iterations is t, which is the number of bits in the exact binary representation of θ.

The number of iterations is t, which is the number of bits in the exact binary representation of θ.

There is one call to each of controlled $U^{2^{t-1}}, \ldots, U^2, U$. The same control qubit can be reset and reused.

The number of iterations is t, which is the number of bits in the exact binary representation of θ.

There is one call to each of controlled $U^{2^{t-1}}, \ldots, U^2, U$. The same control qubit can be reset and reused.

This counts calls to **powers** of U, not individual applications of U. Implementing every power by repetition would require

$$1 + 2 + \cdots + 2^{t-1} = 2^t - 1$$

applications of controlled U.

Phase Estimation Algorithm Analysis

Question: What if the phase θ cannot be exactly expressed as t bits?

Phase Estimation Algorithm Analysis

Question: What if the phase θ cannot be exactly expressed as t bits?

Answer: For failure probability $0 < \delta < 1$, run the same bit-by-bit procedure for

$$m = t + \lceil \log_2(1/\delta) \rceil + 2$$

rounds, using powers up to $U^{2^{m-1}}$.

Phase Estimation Algorithm Analysis

Question: What if the phase θ cannot be exactly expressed as t bits?

Answer: For failure probability $0 < \delta < 1$, run the same bit-by-bit procedure for

$$m = t + \lceil \log_2(1/\delta) \rceil + 2$$

rounds, using powers up to $U^{2^{m-1}}$.

Fact (without proof): The estimate $\widetilde{\theta}$ satisfies

$$\Pr\left[\min_{\ell \in \mathbb{Z}} |\widetilde{\theta} - \theta - \ell| \leq 2^{-t}\right] \geq 1 - \delta.$$

The distance is measured modulo 1, since phases θ and $\theta + 1$ are the same.

Phase Estimation Algorithm Analysis

Question: What happens if $|\psi\rangle$ is not an eigenvector of U?

Phase Estimation Algorithm Analysis

Question: What happens if $|\psi\rangle$ is not an eigenvector of U?

Answer: We can choose an orthonormal basis $\{|\phi_j\rangle\}$ of eigenvectors of U for $\mathbb{C}^{2^n}$ (if U is n-qubit unitary). We can write $|\psi\rangle$ as

$$|\psi\rangle = \sum_j \alpha_j |\phi_j\rangle$$

where $U|\phi_j\rangle = e^{2\pi i\theta_j}|\phi_j\rangle$.

For input $|\psi\rangle = \sum_j \alpha_j |\phi_j\rangle$, the output distribution is

$$\Pr[z \mid \psi] = \sum_j |\alpha_j|^2 \Pr[z \mid \phi_j],$$

where z is the string of measured bits.

For input $|\psi\rangle = \sum_j \alpha_j |\phi_j\rangle$, the output distribution is

$$\Pr[z \mid \psi] = \sum_j |\alpha_j|^2 \Pr[z \mid \phi_j],$$

where z is the string of measured bits.

Equivalently: choose j with probability $|\alpha_j|^2$, then estimate its eigenphase θ_j.

We get a sampled eigenphase, **not the average eigenphase**.

Measuring one bit at a time

Prepare the input state once, at the beginning.

After measuring each bit, reset the control qubit to $|0\rangle$.

Continue with the other qubits in the state left by the measurement—don't prepare the input again.

Measuring one bit at a time

Prepare the input state once, at the beginning.

After measuring each bit, reset the control qubit to $|0\rangle$.

Continue with the other qubits in the state left by the measurement—don't prepare the input again.

Why? Starting over for each bit could give you bits from different eigenphases, rather than several bits of the same eigenphase.

Factoring and order finding

RSA Cryptosystem

- Invented by Rivest, Shamir, and Adleman in 1977
- Enables public-key encryption as well as digital signatures

RSA Cryptosystem

- Invented by Rivest, Shamir, and Adleman in 1977
- Enables public-key encryption as well as digital signatures

An efficient algorithm for factoring would break RSA.

We won't cover how RSA cryptosystem works here, but you can look it up on Wikipedia.

Factoring problem

Input: Integer $N > 1$ written in binary.

Output: Prime factorization of N as $p_1^{a_1} p_2^{a_2} \cdots$.

Factoring problem

Input: Integer $N > 1$ written in binary.

Output: Prime factorization of N as $p_1^{a_1} p_2^{a_2} \cdots$.

The prime factorization of N is unique by the **Fundamental Theorem of Arithmetic**.

To find a factorization of N, it suffices to be able to find *some* nontrivial divisor of N.

Factoring problem

A simple way to factor N is to iterate from $2, 3, \ldots$ and try to find a number that divides N.

Factoring problem

A simple way to factor N is to iterate from $2, 3, \ldots$ and try to find a number that divides N.

In the worst case, if N is not prime, then one only has to iterate up to $\sqrt{N}$. (Why?)

Factoring problem

A simple way to factor N is to iterate from $2, 3, \ldots$ and try to find a number that divides N.

In the worst case, if N is not prime, then one only has to iterate up to $\sqrt{N}$. (Why?)

The number of iterations is

$$\sqrt{N} = \sqrt{2^{\log N}} = 2^{(\log N)/2}$$

which is **exponential** in the binary representation of N. For a 2048-bit integer, this is about

$$2^{1024}$$

iterations.

Shor's algorithm

A quantum algorithm to solve Factoring in $\mathrm{poly}(\log N)$ steps.

Shor's algorithm

A quantum algorithm to solve Factoring in $\mathrm{poly}(\log N)$ steps.

Shor's Algorithm is also a hybrid classical-quantum algorithm.

1. **Classical part**: reduce the factoring problem to **order finding**. This means that if you can solve order finding, you can factor numbers.

Shor's algorithm

A quantum algorithm to solve Factoring in $\mathrm{poly}(\log N)$ steps.

Shor's Algorithm is also a hybrid classical-quantum algorithm.

1. **Classical part**: reduce the factoring problem to **order finding**. This means that if you can solve order finding, you can factor numbers.
2. **Quantum part**: solve order finding.

Order Finding

Input: given positive integers N, x such that

1. $1 \leq x < N$
2. $\gcd(N, x) = 1$ (i.e. they do not have any nontrivial factors in common; we also say that x is **coprime** with N)

Order Finding

Input: given positive integers N, x such that

1. $1 \leq x < N$
2. $\gcd(N, x) = 1$ (i.e. they do not have any nontrivial factors in common; we also say that x is **coprime** with N)

Output: find smallest positive integer r such that $x^r = 1 \bmod N$ (called the **order** of x mod N).

Order finding example

Find order of $x = 7 \mod N = 15$.

Order finding example

Find order of $x = 7 \bmod N = 15$.

$$7^1 = 7, \qquad 7^2 = 4, \qquad 7^3 = 13, \qquad 7^4 = 1 \qquad (\mathrm{mod}\ 15)\ .$$

Therefore order of x mod N is $r = 4$.

Order finding example

Find order of $x = 7 \mod N = 15$.

$$7^1 = 7, \qquad 7^2 = 4, \qquad 7^3 = 13, \qquad 7^4 = 1 \qquad (\mathrm{mod}\ 15)\ .$$

Therefore order of x mod N is $r = 4$.

Note that the function $f(k) = x^k \mod N$ is **periodic**.

Order finding example

Find order of $x = 7 \mod N = 15$.

$$7^1 = 7, \qquad 7^2 = 4, \qquad 7^3 = 13, \qquad 7^4 = 1 \qquad (\mathrm{mod}\ 15)\ .$$

Therefore order of $x \mod N$ is $r = 4$.

Note that the function $f(k) = x^k \mod N$ is **periodic**. This is because

$$x^{k+r} = x^k x^r = x^k \mod N$$

whenever r is the order of $x \mod N$. So $f(k) = f(k + r)$.

Order finding example

The condition that $\gcd(x, N) = 1$ is necessary for the order finding problem to be well-defined.

Order finding example

The condition that $\gcd(x, N) = 1$ is necessary for the order finding problem to be well-defined.

Consider $x = 5, N = 15$. This does **not** satisfy $\gcd(x, N) = 1$. It also does not have an order:

$$5^1 = 5, \qquad 5^2 = 10, \qquad 5^3 = 5, \qquad \cdots \qquad (\mathrm{mod}\ 15)\ .$$

Quantum algorithm for Order Finding

Quantum Algorithm for Order Finding

The Order Finding problem

Input: Integers N and x such that $\gcd(x, N) = 1$.

Goal: Find smallest positive integer r such that $x^r = 1 \mod N$.

Quantum Algorithm for Order Finding

The Order Finding problem

Input: Integers N and x such that $\gcd(x, N) = 1$.

Goal: Find smallest positive integer r such that $x^r = 1 \mod N$.

To efficiently solve this on a quantum computer, we will use the Phase Estimation Algorithm with a carefully constructed unitary that depends on x, N. The phases will yield valuable information about the order of x.

Modular multiplication unitary

Fix x, N with $\gcd(x, N) = 1$.

Consider the Hilbert space $\mathbb{C}^N$ with standard basis $\{|0\rangle, |1\rangle, \ldots, |N-1\rangle\}$. Define the **modular multiplication unitary** on this space

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

Modular multiplication unitary

Fix x, N with $\gcd(x, N) = 1$.

Consider the Hilbert space $\mathbb{C}^N$ with standard basis $\{|0\rangle, |1\rangle, \ldots, |N-1\rangle\}$. Define the **modular multiplication unitary** on this space

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

Example: $x = 7, N = 15$

$$U|1\rangle = |7\rangle, \qquad U|2\rangle = |14\rangle, \qquad U|3\rangle = |6\rangle, \qquad \cdots$$

$$U|13\rangle = |1\rangle, \qquad U|14\rangle = |8\rangle$$

Modular multiplication unitary

Fix x, N with $\gcd(x, N) = 1$.

Consider the Hilbert space $\mathbb{C}^N$ with standard basis $\{|0\rangle, |1\rangle, \ldots, |N-1\rangle\}$. Define the **modular multiplication unitary** on this space

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

Example: $x = 7, N = 15$

$$U|1\rangle = |7\rangle, \qquad U|2\rangle = |14\rangle, \qquad U|3\rangle = |6\rangle, \qquad \cdots$$

$$U|13\rangle = |1\rangle, \qquad U|14\rangle = |8\rangle$$

Since $\gcd(x, N) = 1$, multiplication by x permutes the residues. Therefore U is unitary.

Modular multiplication unitary

Fix x, N. What are some eigenvectors and eigenvalues of U?

Modular multiplication unitary

Fix x, N. What are some eigenvectors and eigenvalues of U?

Let r be the order of x modulo N. Then for all $0 \leq s < r$, the vector

$$|v_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \exp\left(-2\pi i \frac{k}{r} s\right) |x^k \bmod N\rangle$$

Modular multiplication unitary

Fix x, N. What are some eigenvectors and eigenvalues of U?

Let r be the order of x modulo N. Then for all $0 \leq s < r$, the vector

$$|v_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \exp\left(-2\pi i \frac{k}{r} s \right) |x^k \bmod N\rangle$$

is an eigenvector satisfying

$$U|v_s\rangle = \exp\left(2\pi i \frac{s}{r}\right) |v_s\rangle$$

Modular multiplication unitary

Fix x, N. What are some eigenvectors and eigenvalues of U?

Let r be the order of x modulo N. Then for all $0 \leq s < r$, the vector

$$|v_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \exp\left(-2\pi i \frac{k}{r} s\right) |x^k \bmod N\rangle$$

is an eigenvector satisfying

$$U|v_s\rangle = \exp\left(2\pi i \frac{s}{r}\right) |v_s\rangle$$

Phase estimation on $|v_s\rangle$ estimates s/r: the unknown order appears in the eigenphase!

Modular multiplication unitary

Consider the example $x = 7$, $N = 15$, $r = 4$ again. One of the eigenvectors is

$$|v_1\rangle = \frac{1}{2}\big(\,|1\rangle - i\,|7\rangle - |4\rangle + i\,|13\rangle\,\big).$$

Modular multiplication unitary

Consider the example $x = 7$, $N = 15$, $r = 4$ again. One of the eigenvectors is

$$|v_1\rangle = \frac{1}{2}\big(\,|1\rangle - i\,|7\rangle - |4\rangle + i\,|13\rangle\,\big).$$

Applying the modular multiplication unitary in superposition:

$$U\,|v_1\rangle = \frac{1}{2}\big(U\,|1\rangle - iU\,|7\rangle - U\,|4\rangle + iU\,|13\rangle\,\big)$$

Modular multiplication unitary

Consider the example $x = 7$, $N = 15$, $r = 4$ again. One of the eigenvectors is

$$|v_1\rangle = \frac{1}{2}\big(\,|1\rangle - i\,|7\rangle - |4\rangle + i\,|13\rangle\,\big).$$

Applying the modular multiplication unitary in superposition:

$$\begin{aligned} U\,|v_1\rangle &= \frac{1}{2}\big(U\,|1\rangle - iU\,|7\rangle - U\,|4\rangle + iU\,|13\rangle\,\big) \\ &= \frac{1}{2}\big(\,|7\rangle - i\,|4\rangle - |13\rangle + i\,|1\rangle\,\big) \end{aligned}$$

Modular multiplication unitary

Consider the example $x = 7$, $N = 15$, $r = 4$ again. One of the eigenvectors is

$$|v_1\rangle = \frac{1}{2}\big(\,|1\rangle - i\,|7\rangle - |4\rangle + i\,|13\rangle\,\big).$$

Applying the modular multiplication unitary in superposition:

$$\begin{aligned} U\,|v_1\rangle &= \frac{1}{2}\big(U\,|1\rangle - iU\,|7\rangle - U\,|4\rangle + iU\,|13\rangle\,\big) \\ &= \frac{1}{2}\big(\,|7\rangle - i\,|4\rangle - |13\rangle + i\,|1\rangle\,\big) \\ &= i\,|v_1\rangle = e^{2\pi i/4}\,|v_1\rangle\,. \end{aligned}$$

Phase estimation on this eigenvector returns $\theta = 1/4$.