CPEN 400Q Lecture 15
Order finding and Shor's algorithm

Wednesday 5 March 2025

Announcements

- Technical assignment 3 available later this week
- Quiz 7 Monday
- Midterm checkpoint due next Friday

Last time

We dug into the details of **quantum phase estimation**, which estimates the eigenvalues of unitary matrices.

$$U|k\rangle = \lambda_k |k\rangle = e^{2\pi i \theta_k} |k\rangle$$

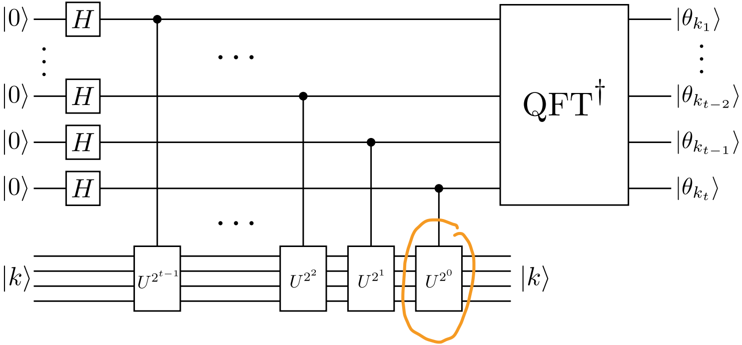


Image credit: Xanadu Quantum Codebook node P.2

Reminder: where are we going?

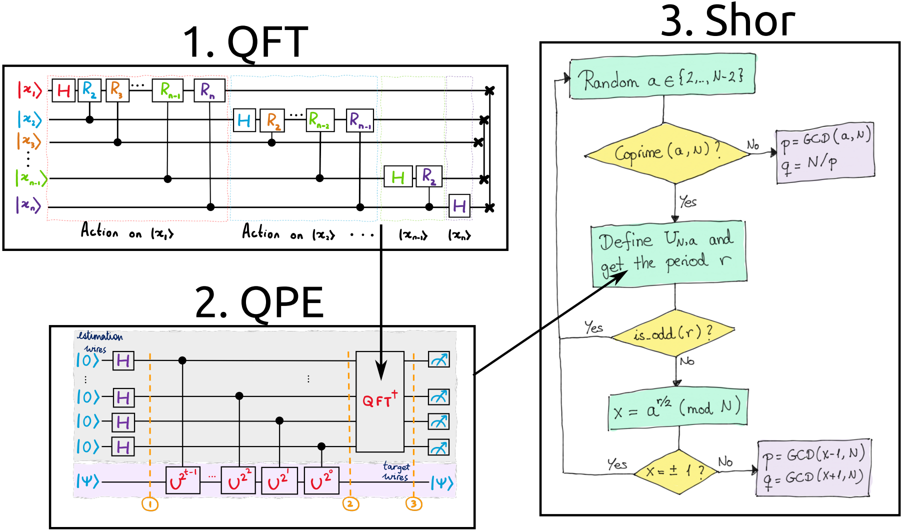


Image credit: Xanadu Quantum Codebook nodes F.3, P.3, S.4

Learning outcomes

- Use QPE to implement the order finding algorithm
- Implement Shor's algorithm in PennyLane

Order finding on a quantum computer

Given a function
$$f_{N,a}(x) = a^x \bmod N$$
we want to factor this

The *order* of a mod N is the smallest (non-zero integer) r s.t.
$$f_{N,a}(r) = a^r \bmod N = 1 \bmod N$$

$$U_{5,3}|4\rangle = |2\rangle$$

Define a unitary operation that performs
$$U_{N,a}|k\rangle = |a \cdot k \bmod N\rangle$$

If r is the order of a, and we apply $U_{N,a}$ r times,
$$(U_{N,a})^r|k\rangle = (U_{N,a})^{r-1}|a\cdot k \bmod N\rangle = \dots = |a^r \cdot k \bmod N\rangle = |k\rangle$$

So r is also the order of $U_{N,a}$! We can find it efficiently using a quantum computer.

Let U be an operator and $|\phi\rangle$ any state. How do we find the minimum r such that

$$U^r|\phi\rangle = |\phi\rangle$$

QPE does the trick if we set things up in a clever way:

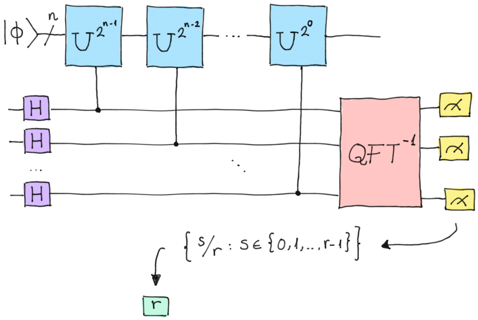


Image credit: Xanadu Quantum Codebook node S.3

Order finding on a quantum computer

$$|\phi\rangle \to U^r|\phi\rangle = |\phi\rangle$$

Consider the state

$$|\psi_0\rangle = \frac{1}{\sqrt{r}}\left(|\phi\rangle + U|\phi\rangle + U^2|\phi\rangle + \dots + U^{r-1}|\phi\rangle\right)$$

If we apply U to this:

$$U|\psi_0\rangle = \frac{1}{\sqrt{r}}\left(U|\phi\rangle + U^2|\phi\rangle + U^3|\phi\rangle + \dots + \underbrace{U^r|\phi\rangle}_{=|\phi\rangle}\right)$$

$$= |\psi_0\rangle$$

$\Rightarrow$ eigenstate w/ eigval 1

Order finding on a quantum computer

Now consider the state

$$|\psi_1\rangle = \frac{1}{\sqrt{r}}\left(|\phi\rangle + e^{-\frac{2\pi i}{r}} U|\phi\rangle + e^{-2\cdot\frac{2\pi i}{r}} U^2|\phi\rangle + \dots + e^{-(r-1)\frac{2\pi i}{r}} U^{r-1}|\phi\rangle\right)$$

$\times e^{\frac{2\pi i}{r}}$

$e^{\frac{2\pi i}{r}}$

If we apply U to this:

$$U|\psi_1\rangle = \frac{1}{\sqrt{r}}\left(U|\phi\rangle + e^{-\frac{2\pi i}{r}} U^2|\phi\rangle + e^{-2\cdot\frac{2\pi i}{r}} U^3|\phi\rangle + \dots + e^{-(r-1)\frac{2\pi i}{r}} U^r|\phi\rangle\right)$$

$|\phi\rangle$

$$= e^{\frac{2\pi i}{r}} \cdot |\psi_1\rangle$$

This generalizes to $|\Psi_s\rangle$

$$|\Psi_s\rangle = \frac{1}{\sqrt{r}}\left(|\phi\rangle + e^{-s\frac{2\pi i}{r}} U|\phi\rangle + e^{-2s\cdot\frac{2\pi i}{r}} U^2|\phi\rangle + \dots + e^{-(r-1)s\frac{2\pi i}{r}} U^{r-1}|\phi\rangle \right)$$

It has eigenvalue

$$U|\Psi_s\rangle = e^{\frac{2\pi i s}{r}} |\Psi_s\rangle$$

Idea: if we can create *any* one of these $|\Psi_s\rangle$, we could run QPE and get an estimate for s/r, and then recover r.

$\{|b_j\rangle\}$ $\{|a_k\rangle\}$ $|b_k\rangle = \sum_j c_j |a_j\rangle$

$U \uparrow$

est phase

$U|b_k\rangle = e^{2\pi i \theta_k}|b_k\rangle$

Problem: to construct any $|\Psi_s\rangle$, we would need to know r in advance!

Solution: construct the uniform superposition of all of them.

$$|\Psi\rangle = \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |\Psi_s\rangle$$

But what does this equal?

Order finding on a quantum computer

The superposition of all $|\Psi_s\rangle$ is just our original state $|\phi\rangle$!

$$|\Psi\rangle = \frac{1}{\sqrt{r}} \left(\begin{array}{l} |\Psi_0\rangle + \\ + |\Psi_1\rangle + \\ \quad \vdots \\ + |\Psi_{r-1}\rangle \end{array} \right) = \frac{1}{\sqrt{r}} \cdot \underbrace{\begin{array}{l} 1/\sqrt{r}\left(|\phi\rangle + 1 \cdot U|\phi\rangle + \dots + 1 \cdot U^{r-1}|\phi\rangle\right) \\ + 1/\sqrt{r}\left(|\phi\rangle + e^{\frac{-2\pi i}{r}} U|\phi\rangle + \dots + e^{\frac{-2\pi i (r-1)}{r}} U^{r-1}|\phi\rangle\right) \\ \quad \vdots \\ + 1/\sqrt{r}\left(|\phi\rangle + e^{\frac{-2\pi i (r-1)}{r}} U|\phi\rangle + \dots + e^{\frac{-2\pi i (r-1)^2}{r}} U^{r-1}|\phi\rangle\right) \\ \quad \downarrow \qquad\qquad \downarrow \qquad\qquad \downarrow \qquad \downarrow \\ \quad r \qquad\qquad\;\; 0 \quad \cdots \quad 0 \;\; \cdots \;\; 0 \end{array}}_{= 1/\sqrt{r} \cdot 1/\sqrt{r} \cdot r|\phi\rangle = \boxed{|\phi\rangle}}$$

$U^r|\phi\rangle = |\phi\rangle$

If we run QPE, the output will be s/r for one of these states.

$|\phi\rangle = ???$

$U_{N,a}|k\rangle = |ak \bmod N\rangle$ $\boxed{|\phi\rangle = |1\rangle}$

Image credit: Xanadu Quantum Codebook node S.3

Shor's algorithm

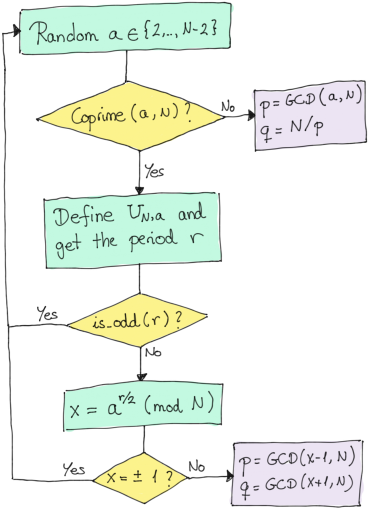


Image credit: Xanadu Quantum Codebook node S.4

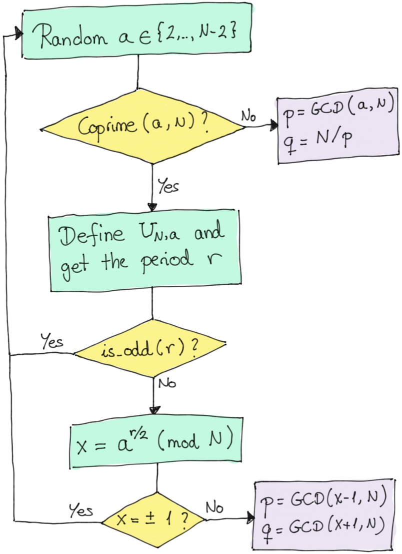
Shor's algorithm can factor a number N like

$$N = pq$$

where p, q are prime.

A quantum computer runs order finding to obtain p and q.

Everything else is number theory.



Idea: find a *non-trivial square root* of N, i.e., some $x \neq \pm 1$ s.t.

$$x^2 = 1 \bmod N$$

If we find such an x,

$$x^2 - 1 = 0 \bmod N$$

$$(x-1)(x+1) = 0 \bmod N$$

$$\Downarrow$$

Then

$$(x-1)(x+1) = kN$$

for some integer k.

If

$$(x-1)(x+1) = kN = k \cdot pq$$

then $x - 1$ is a multiple of one of p or q, and $x + 1$ is a multiple of the other.

$$x - 1 = s \cdot p$$
$$x + 1 = t \cdot q$$

We can compute p and q by finding their *gcd* with N:

$$p = \gcd(x-1, N)$$
$$q = \gcd(x+1, N)$$

Non-trivial square roots and factoring

It's actually okay to find any *even* power of x for which this holds:

$$x^{r} = x^{2r'} = \left(x^{r'}\right)^{2}$$

We can use order finding to find such an r. If it is even, we can obtain x and factor N.

$$\underset{\underset{1 \bmod N}{\parallel}}{a^{r}} = \left(a^{\frac{r}{2}}\right)^{2} =$$

$a^{\frac{r}{2}}$ is nontrivial square root!

can factor N

Shor's algorithm

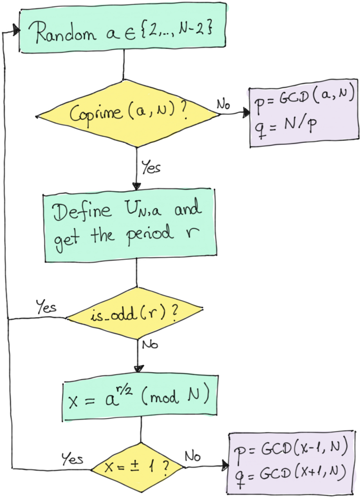
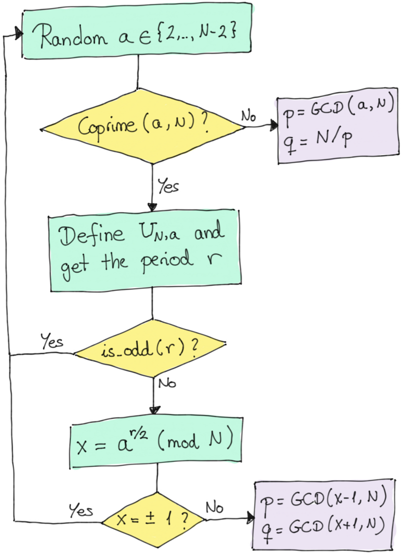


Image credit: Xanadu Quantum Codebook node S.4

Is this really efficient?

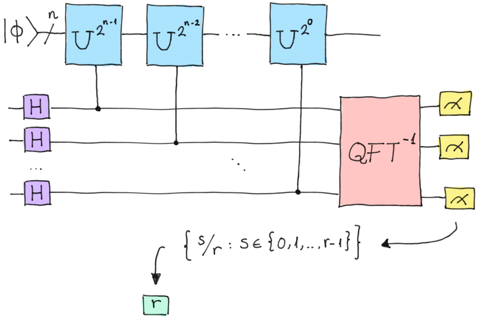
GCD: polynomial w/Euclid's algorithm

Modular exponentiation: can use exponentiation by squaring, other methods to reduce operations and memory required



Is this really efficient?

Quantum part: let $L = \lceil \log_2 N \rceil$.



QFT: polynomial in number of qubits $O(L^2)$

Controlled-U gates: implemented using something called *modular exponentiation* in $O(L^3)$ gates.

Discussion

Form groups of 3-4, and consider the following questions:

1. Shor's algorithm was developed in 1994. Estimate the fraction of today's world population that can actually implement it.
2. Shor's algo can be used to break cryptosystems like RSA. Estimate the proportion of the world that would be affected if someone actually deployed it at scale.
3. Is it ethical to develop such an algorithm? Is it ethical to *teach* such an algorithm?
4. Look up some resource estimates; how long would it actually take to break 2048-bit RSA? How many qubits are needed?
5. Think critically about (a) who knows how to implement the algorithm, and (b) who will potentially have access to quantum hardware in the future. What issues can you foresee?
6. What are ways we can keep our cryptographic infrastructure secure in the future?

Next time

Content:

- Module 4: quantum channels, and noise in quantum systems

Action items:

1. Assignment 3
2. Work on project and midterm checkpoint report

Recommended reading:

- From this class: Codebook QFT, QPE, SH; N&C 5.3, A.5
- For next class: Codebook NT; N&C 8.1-8.3