CPEN 400Q Lecture 15
Quantum phase estimation; order finding

Monday 4 March 2024

Announcements

- Quiz 6 today
- Technical assignment 3 available later this week
- Midterm checkpoint due Wednesday; meetings on Thurs/Fri

We implemented the quantum Fourier transform using a *polynomial* number of gates:

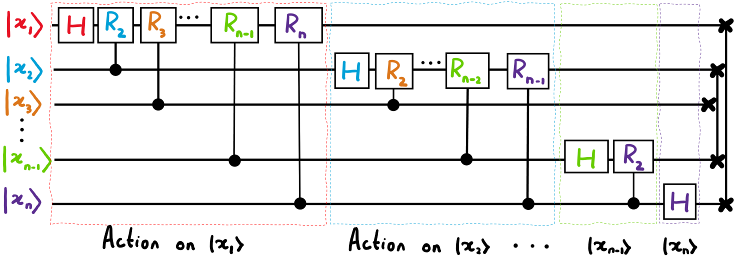


Image credit: Xanadu Quantum Codebook node F.3

Last time

We started learning about the quantum phase estimation subroutine which estimates the eigenvalues of unitary matrices.

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

$\lambda_k \downarrow$ mag. 1

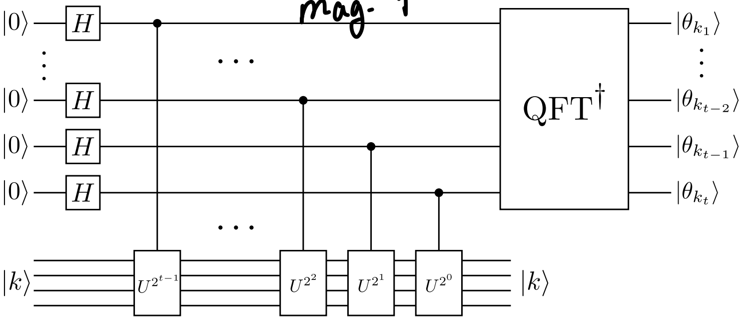
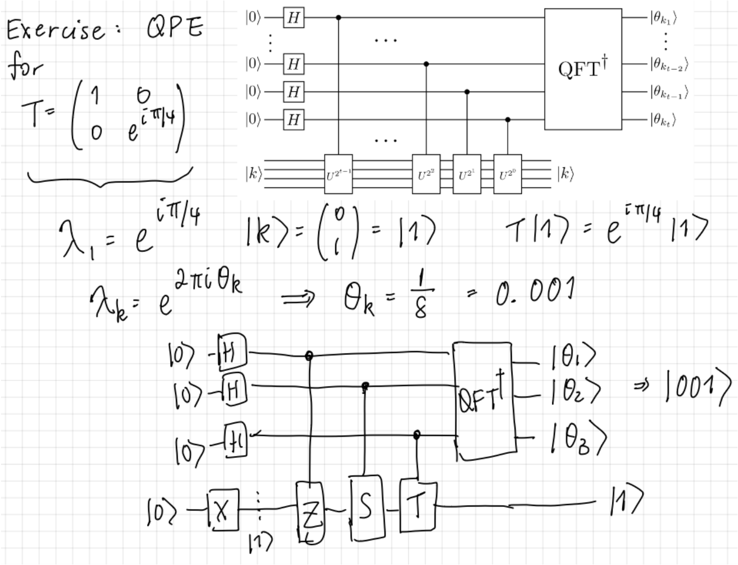


Image credit: Xanadu Quantum Codebook node P.2

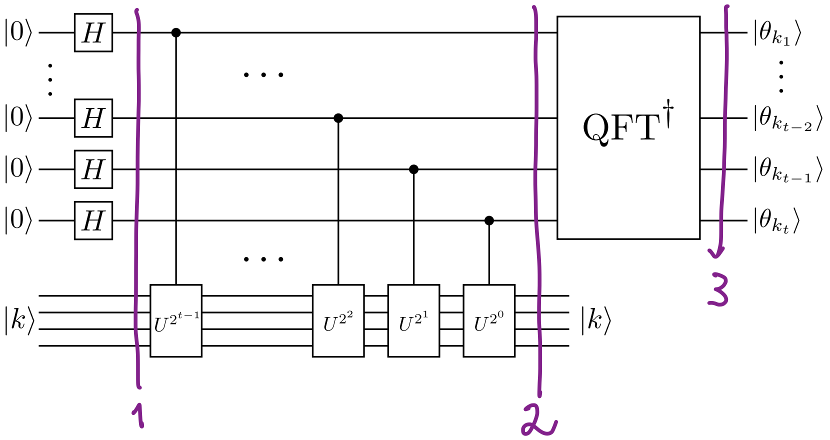
We walked through an example using T.



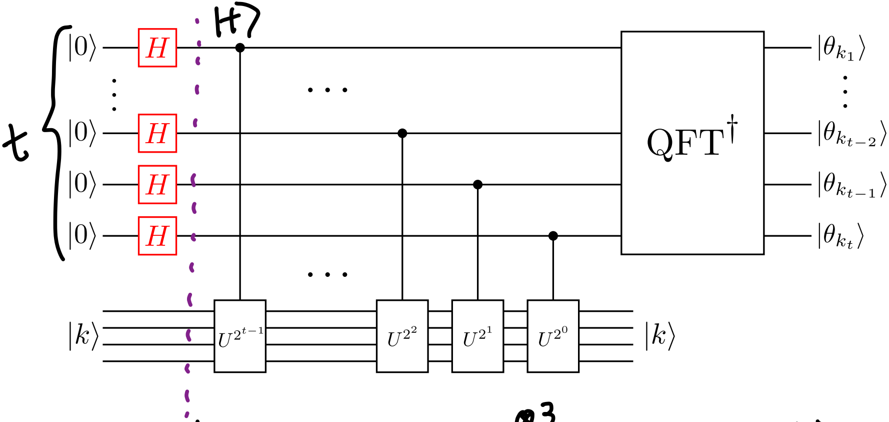
Learning outcomes

- Outline the steps of the quantum phase estimation (QPE) subroutine
- Use QPE to implement the order finding algorithm
- Implement Shor's algorithm in PennyLane

Quantum phase estimation

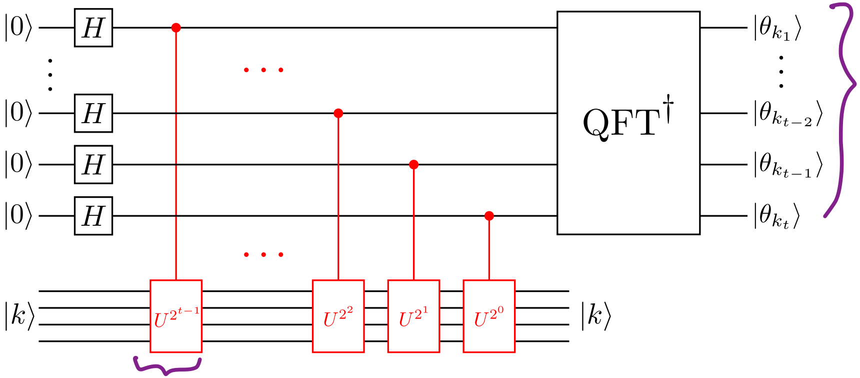


Quantum phase estimation: step 1



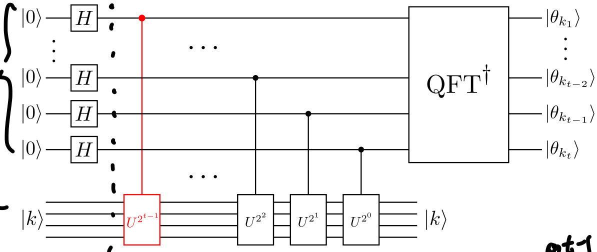
$\underbrace{|+\rangle^{\otimes t} \; |k\rangle} \rightarrow |+\rangle^{\otimes 3} = |+\rangle \otimes |+\rangle \otimes |+\rangle$

Quantum phase estimation: step 1



Quantum phase estimation: step 2

est. register



target register

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

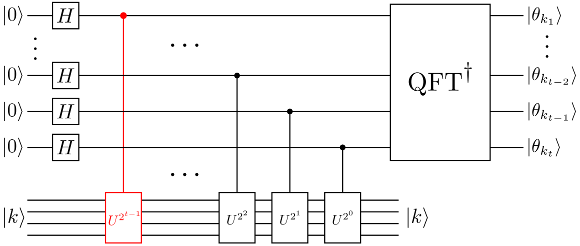
$$|+\rangle^{\otimes t}|k\rangle : \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)\otimes|+\rangle^{\otimes t-1}|k\rangle = \frac{1}{\sqrt{2}}|0\rangle|+\rangle^{\otimes t-1}|k\rangle + \frac{1}{\sqrt{2}}|1\rangle|+\rangle^{\otimes t-1}|k\rangle$$

Apply ctrl $U^{2^{t-1}}$

$$\frac{1}{\sqrt{2}}|0\rangle|+\rangle^{\otimes t-1}|k\rangle + \frac{1}{\sqrt{2}}|1\rangle|+\rangle^{\otimes t-1}\left(e^{2\pi i\theta_k}\right)^{2^{t-1}}|k\rangle$$

$$= \frac{1}{\sqrt{2}}|0\rangle|+\rangle^{\otimes t-1}|k\rangle + \frac{1}{\sqrt{2}}\left(e^{2\pi i\theta_k}\right)^{2^{t-1}}|1\rangle|+\rangle^{\otimes t-1}|k\rangle$$

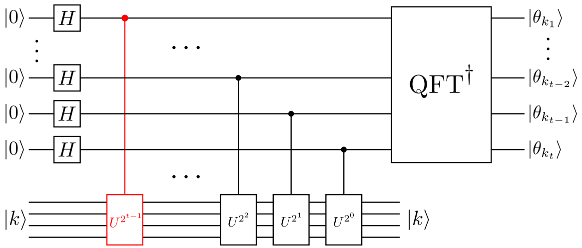
Quantum phase estimation: step 2



Use phase kickback

$$\frac{1}{\sqrt{2}}|0\rangle\underbrace{|+\rangle^{\otimes t-1}}\underbrace{|k\rangle} + \frac{1}{\sqrt{2}}\left(e^{2\pi i\theta_k}\right)^{2^{t-1}}|1\rangle\underbrace{|+\rangle^{\otimes t-1}}\underbrace{|k\rangle}$$

$$= \frac{1}{\sqrt{2}}\left(|0\rangle + \underline{\left(e^{2\pi i\theta_k}\right)}^{2^{t-1}}|1\rangle\right) \otimes |+\rangle^{\otimes t-1} \otimes |k\rangle$$



$\theta_k = 0.\theta_{k_1}\theta_{k_2}\dots$

Exercise:

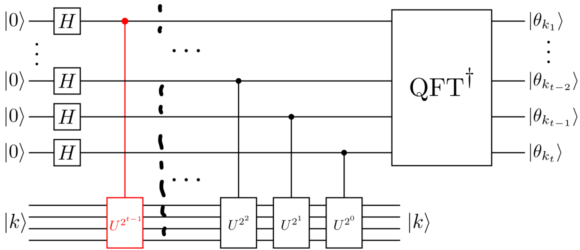
What is happening in the exponent?

$$\left(e^{2\pi i\theta_k}\right)^{2^{t-1}} = e^{2\pi i\left(\frac{\theta_{k_1}}{2}+\frac{\theta_{k_2}}{4}+\dots\frac{\theta_{k_t}}{2^t}\right)2^{t-1}}$$

$$= e^{2\pi i\cdot\frac{\theta_{k_t}}{2}}$$

$$= e^{2\pi i\, 0.\theta_{k_t}}$$

$$= e$$



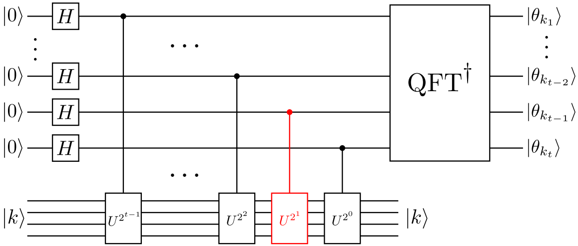
$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\, 0.\theta_{k_t}}|1\rangle\right)|+\rangle^{\otimes t-1}\,|k\rangle$$

Next qubit:

$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\, 0.\theta_{k_t}}|1\rangle\right)|+\rangle|+\rangle^{\otimes t-2}\,|k\rangle$$

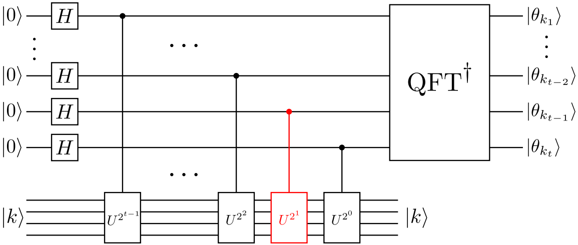
after ctrl $U^{2^{t-2}}$ $\Longrightarrow$ $\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\, 0.\theta_{k_t}}|1\rangle\right)\otimes\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\, 0.\theta_{k_{t-1}}\theta_{k_t}}|1\rangle\right)|+\rangle^{\otimes t-2}\,|k\rangle$

Quantum phase estimation: step 2



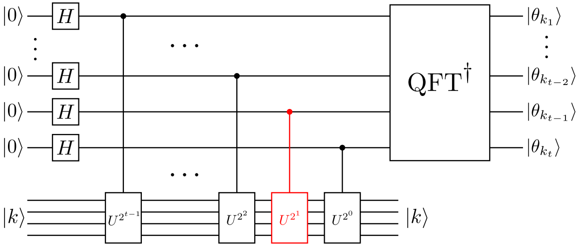
Check second-last qubit (ignore the others)

Quantum phase estimation: step 2

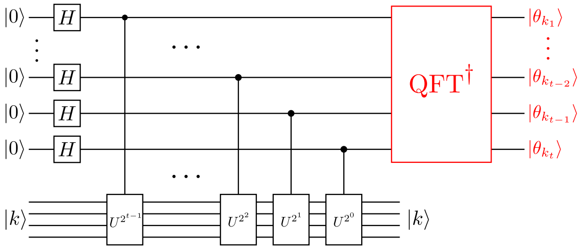


Again check the exponent...

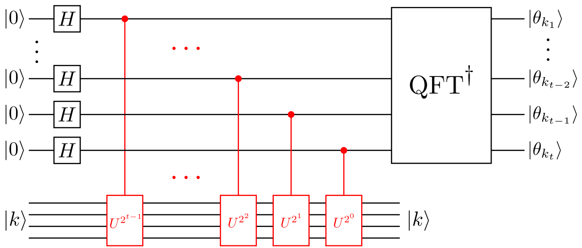
Quantum phase estimation: step 2



Quantum phase estimation: step 2



Can show in the same way for the last qubit (ignore others)



After step 2, we have the state

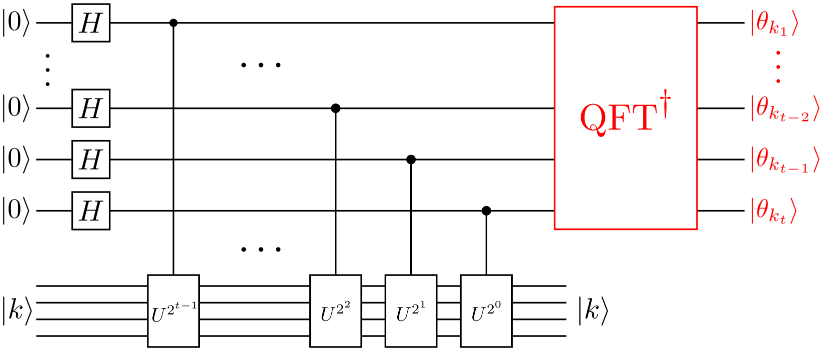
$$\frac{1}{\sqrt{2}}(|0\rangle+e^{2\pi i 0.\theta_{k_t}}|1\rangle)\cdots\frac{1}{\sqrt{2}}(|0\rangle+e^{2\pi i 0.\theta_{k_2}\cdots\theta_{k_t}}|1\rangle)\frac{1}{\sqrt{2}}(|0\rangle+e^{2\pi i 0.\theta_{k_1}\cdots\theta_{k_t}}|1\rangle)|k\rangle$$

Should look familiar!

$$= (\mathrm{QFT}\,|\theta_k\rangle)\,|k\rangle$$
$$= (\mathrm{QFT}\,|\theta_{k_1}\theta_{k_2}\cdots\theta_{k_t}\rangle)\,|k\rangle$$

Measure to learn the bits of θ_k.

$$\lambda_k = e^{2\pi i \theta_k}$$



$$T \otimes T \otimes T = \begin{pmatrix} \cdot & & & & & \\ & \cdot & & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ & & & & & \cdot \end{pmatrix}$$

Let's implement it.

Reminder: where are we going?

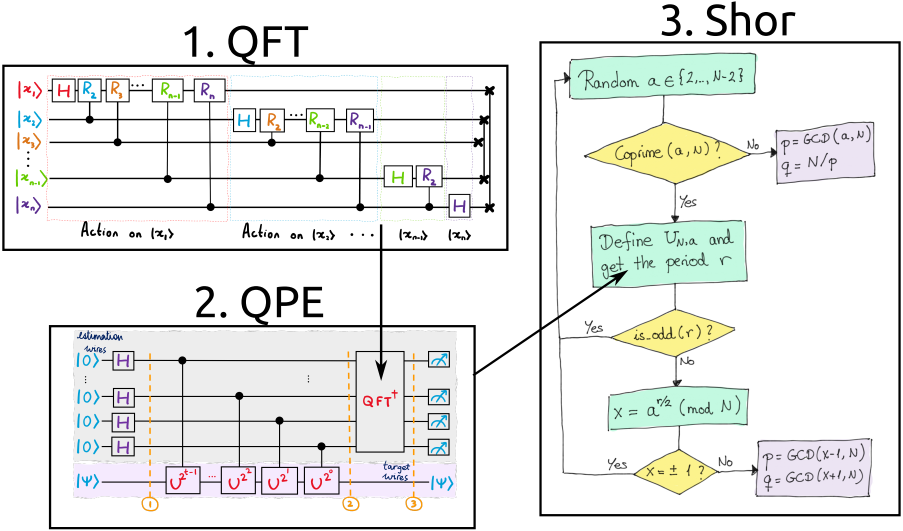


Image credit: Xanadu Quantum Codebook nodes F.3, P.3, S.4

Suppose we have a function

$$f(x)$$

over the integers modulo N.

If there exists $r \in \mathbb{Z}$ s.t.

$$f(x+r) = f(x) \quad \forall x$$

$f(x)$ is periodic with period r.

Suppose

$$f(x) = a^x \bmod N \qquad a \in \mathbb{Z}$$

The *order* of *a* is the smallest *m* such that

$$f(m) = a^m \bmod N \equiv 1 \bmod N$$

Note that this is also the period:

$$f(x+m) = a^{x+m} \bmod N = a^x a^m \bmod N$$
$$= a^x \bmod N$$

$$f(m) = a^m \bmod N \equiv 1 \bmod N$$

Exercise: find the order of $a = 5$ for $N = 7$.

$$m = 6$$

$$5^6 = 15625 \,\%\, 7 = 1$$

Order finding on a quantum computer

More formally, define

$$f_{N,a}(m) = a^m \equiv 1 \bmod N$$

This Wed: short

Monday: hands-on quiz

Wed: zoom

Define a unitary operation that performs

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

$k=2 \quad a=3 \quad N=7 \Rightarrow U_{N,a}|2\rangle = |6 \bmod 7\rangle$

If m is the order of a, and we apply $U_{N,a}$ m times, $U_{N,a}|010\rangle = |110\rangle$

$$U_{N,a}^m|k\rangle = |a^m k \bmod N\rangle = |k\rangle$$

$\hookrightarrow \mathbb{1}$

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

Order finding on a quantum computer

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

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

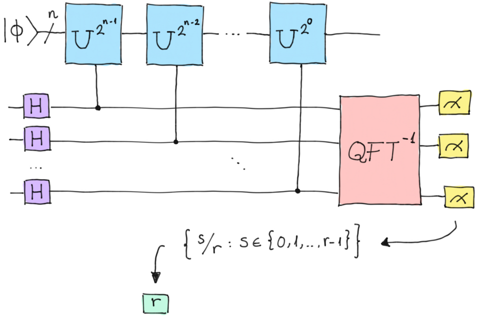


Image credit: Xanadu Quantum Codebook node S.3

Consider the state

If we apply U to this:

Order finding on a quantum computer

Now consider the state

If we apply U to this:

This generalizes to $|\Psi_s\rangle$

It has eigenvalue

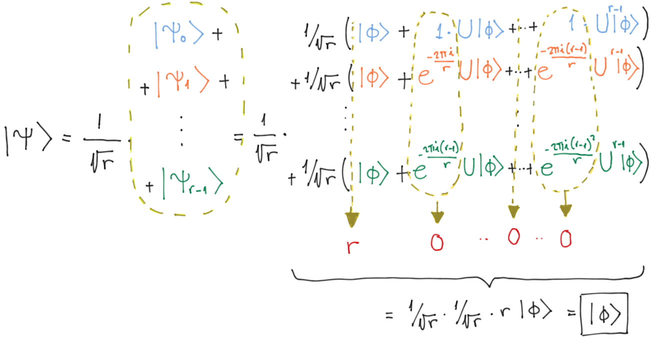
Idea: if we can create *any* one of these $|\Psi_s\rangle$, we could run QPE and get an estimate for s/r, and then recover r.

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

Solution: construct the uniform superposition of all of them.

But what does this equal?

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



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

Image credit: Xanadu Quantum Codebook node S.3

Shor's algorithm

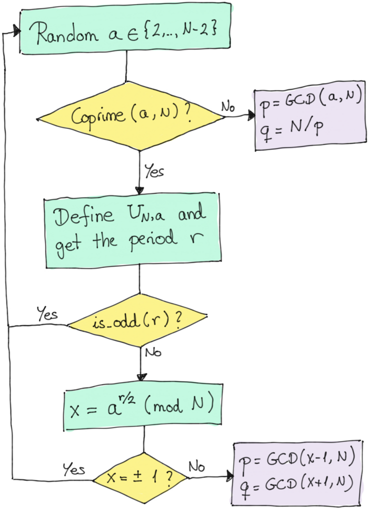


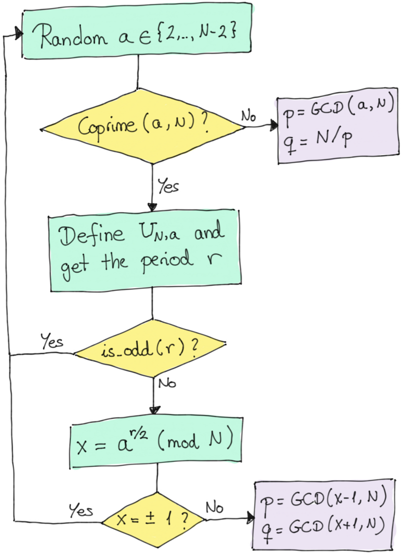
Image credit: Xanadu Quantum Codebook node S.4

Shor's algorithm is used to factor some number N into

where p and q are prime.

A quantum computer runs order finding, and the result is used to obtain p and q.

The rest of the algorithm is classical.



Non-trivial square roots

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

If we find such an x, then we know

This means that

for some integer k.

Non-trivial square roots

If

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

we can compute the values of p and q by finding their *gcd* with N:

But... how do we find such an x?

Non-trivial square roots and factoring

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

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

Shor's algorithm

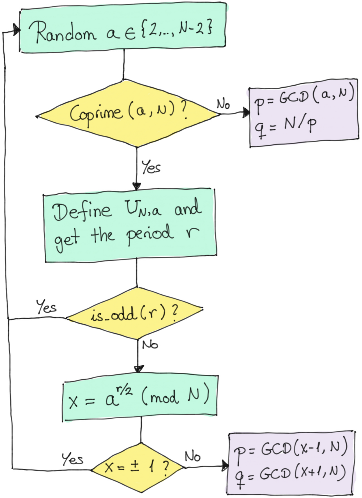
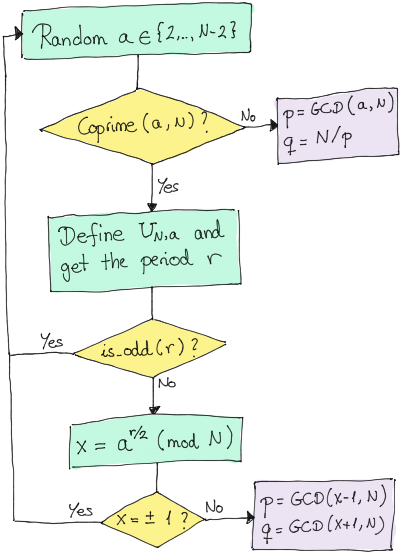


Image credit: Xanadu Quantum Codebook node S.4

Is this really efficient?

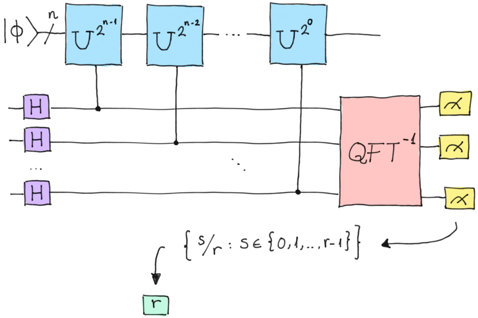
GCD: polynomial w/Euclid's algorithm

Modular exponentiation: can use exponentiation by squaring, other methods to reduce number of 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.

Next time

Content:

- Hands-on with quantum key distribution

Action items:

1. Midterm checkpoint submission

Recommended reading:

- Codebook modules F, P, and S
- Nielsen & Chuang 5.3, Appendix A.5