CPEN 400Q Lecture 12
The quantum Fourier transform (QFT)

Monday 24 February 2025

Announcements

- Quiz 5 today
- Literacy Assignment 2 due tomorrow at 23:59
- Assignment 2 due Thursday at 23:59
- Tutorial tomorrow: intro to variational algorithms
  - helpful for many project groups
- First project peer assessment survey this week
  - Qualtrics link will be posted in Piazza

Where have we been?

Where are we going?

```python
def shors_algorithm(N):
    p, q = 0, 0

    while p * q != N:
        a = np.random.choice(list(range(2, N - 1)))

        if np.gcd(a, N) != 1:
            p = np.gcd(a, N)
            q = N // p
            return p, q

        sample = get_sample(a, N)
        phase = fractional_binary_to_float(sample)
        candidate_order = phase_to_order(phase, N)

        if candidate_order % 2 == 0:
            square_root = (a ** (candidate_order // 2)) % N

            if square_root not in [1, N - 1]:
                p = np.gcd(square_root - 1, N)
                q = np.gcd(square_root + 1, N)

    return p, q
```

Module 3 learning outcomes

Learning outcomes:

- define, and state the scaling of, the quantum Fourier transform
- use quantum phase estimation to determine the eigenvalues of a unitary matrix
- use the QFT and QPE as subroutines to implement order finding, and simulate Shor's factoring algorithm
- identify cryptographic schemes susceptible to quantum attack
- describe the societal and ethical implications of quantum technology

Today

Learning outcomes:

- Express floating-point values in fractional binary representation
- Describe the behaviour of the quantum Fourier transform
- Construct a circuit for the quantum Fourier transform and analyze its resource usage

The DFT and FFT (which implements it efficiently) convert between time and frequency domains in digital signal processing.

$$DFT = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \bar{\omega} & \bar{\omega}^2 & \cdots & \bar{\omega}^{N-1} \\ 1 & \bar{\omega}^2 & \bar{\omega}^4 & \cdots & \bar{\omega}^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \bar{\omega}^{N-1} & \bar{\omega}^{2(N-1)} & \cdots & \bar{\omega}^{(N-1)(N-1)} \end{pmatrix}$$

where $\bar{\omega} = e^{-2\pi i/N}$.

Given a signal $x[n]$, the DFT computes

The inverse DFT computes

where $\omega = e^{2\pi i/N} = \bar{\omega}^{-1}$

The quantum Fourier transform (QFT) is the quantum analog of the **inverse DFT**.

Exercise: Apply the QFT to an n-qubit basis state $|x\rangle$

As a matrix, it looks a lot like the DFT:

$$QFT = \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \cdots & \omega^{(N-1)(N-1)} \end{pmatrix}$$

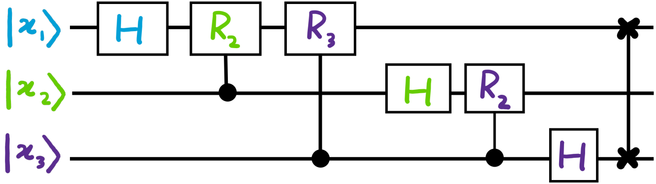
How do we *synthesize* a circuit for it?

Exercise: Start with special case $n = 1$ ($N = 2$).

Quantum Fourier transform

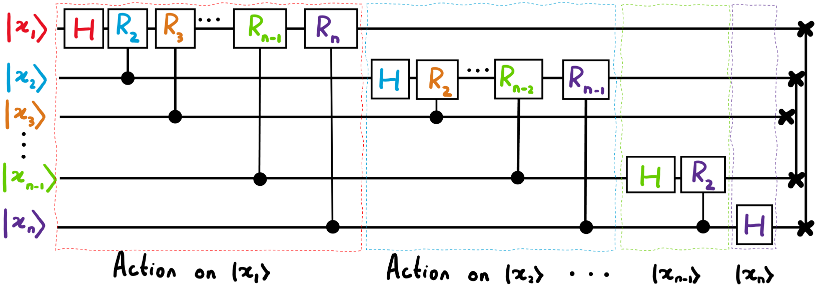
Next case: $n = 2$ $(N = 4)$

Circuit for $n = 3$ ($N = 8$):



Here, $R_2 = S$ and $R_3 = T$.

Image credit: Xanadu Quantum Codebook node F.3



We will derive this by reverse-engineering the analytical definition,

Image credit: Xanadu Quantum Codebook node F.3

A circuit for the QFT

Here x and k are integers, which have binary equivalents $|x\rangle = |x_1 \cdots x_n\rangle$, $|k\rangle = |k_1 \cdots k_n\rangle$:

and similarly for k.

A circuit for the QFT

We are working with

$$\omega^{xk} = e^{2\pi i x(k/N)}$$

with $N = 2^n$.

We can write a fraction $k/2^n$ in a ‘decimal version’ of binary:

Exercise: let $k = 0.11010$. What is the numerical value of k?

We will reexpress k/N in fractional binary notation, then reshuffle and *factor* the output state to uncover the circuit structure.

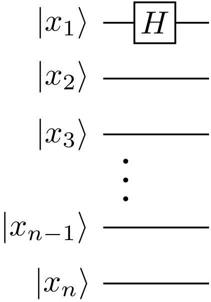
A circuit for the QFT

A circuit for the QFT

Exercise: Starting from

$$|x\rangle = |x_1 \cdots x_n\rangle,$$

apply Hadamard to qubit 1, then express the phase in terms of x_1 using fractional binary notation.



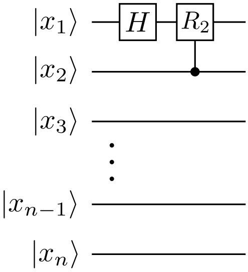
A circuit for the QFT

Recall: trying to make the state

$$|x\rangle \to \frac{\left(|0\rangle + e^{2\pi i 0.x_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.x_{n-1}x_n}|1\rangle\right)\cdots\left(|0\rangle + e^{2\pi i 0.x_1\cdots x_n}|1\rangle\right)}{\sqrt{N}}$$

Every qubit has a different *relative phase*. Define

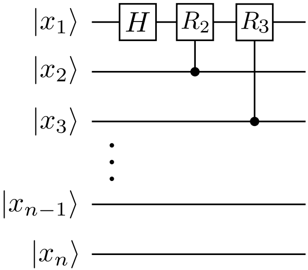
Apply controlled R_2 from $2 \to 1$



First qubit picks up a phase:

A circuit for the QFT

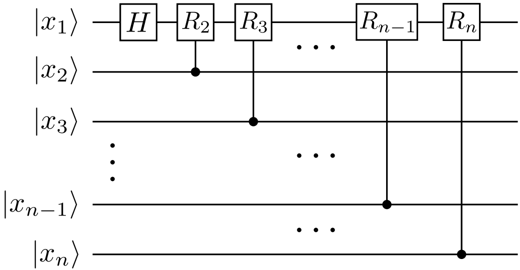
Apply controlled R_3 from $3 \to 1$



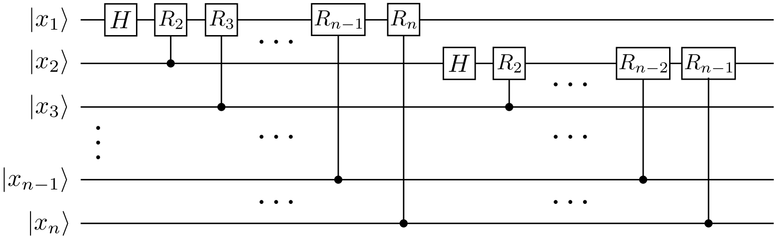
First qubit picks up another phase:

A circuit for the QFT

Apply a controlled R_4 from $4 \rightarrow 1$, etc. to get

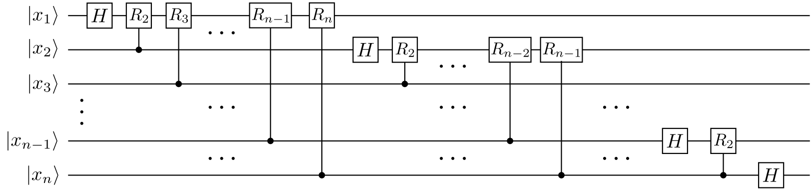


Repeat with the second qubit: apply H then controlled rotations from qubits 3 to n to get



Repeat for remaining qubits to obtain the big state from earlier:

$$|x\rangle \rightarrow \frac{\left(|0\rangle + e^{2\pi i 0.x_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.x_{n-1}x_n}|1\rangle\right)\cdots\left(|0\rangle + e^{2\pi i 0.x_1\cdots x_n}|1\rangle\right)}{\sqrt{N}}$$



The qubits are "backwards" - easily fixed with SWAP gates.

Exercise: What are the gate counts and depth of this circuit?

-
-
-
-

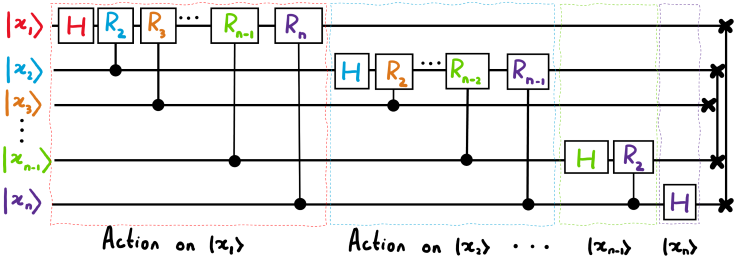


Image credit: Xanadu Quantum Codebook node F.3

Next time

Content:

- Variational algorithms (tutorial)
- Quantum phase estimation

Action items:

1. LA2 and A2
2. Work on project

Recommended reading:

- For this class: Codebook module QFT, Nielsen & Chuang 5.1
- For next class: Codebook module QPE, Nielsen & Chuang 5.2