Lecture 4: Shor's Algorithm II: Factoring to period-finding

Abe Asfaw

Quick review

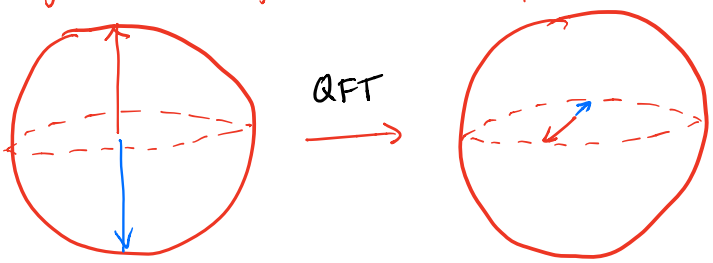
Yesterday, covered QFT and QPE.

(1) QFT – basis change from computational basis to Fourier basis

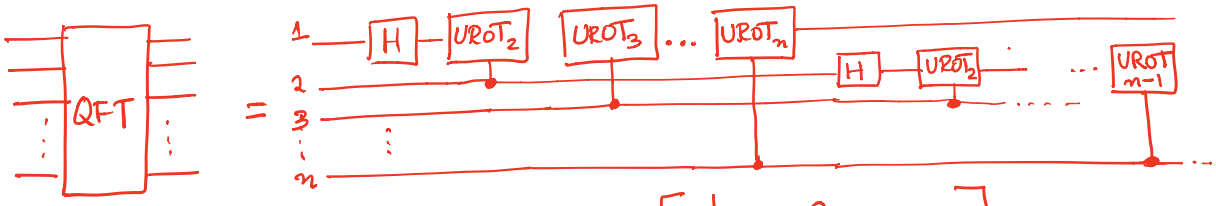
- one-qubit QFT is simply H gate

computational basis		Fourier basis
$\|0\rangle$	QFT →	$\|+\rangle$
$\|1\rangle$		$\|-\rangle$

- go from two poles of Bloch sphere to equatorial plane



- quick demo
- circuit implementation



$$UROT_k = \begin{bmatrix} 1 & 0 \\ 0 & \exp\left(\frac{2\pi i}{2^k}\right) \end{bmatrix}$$

control, target

$$CROT_k |1 x_j\rangle = e^{\frac{2\pi i}{2^k} x_j} |1 x_j\rangle$$

$$H|x_j\rangle = \frac{|0\rangle + e^{2\pi i x_j}|1\rangle}{\sqrt{2}}$$

circuit implements QFT in reverse order of qubits

Form of QFT: (n qubits, $N \equiv 2^n$); $|x\rangle = |x_1 \cdots x_n\rangle$

$$QFT_N |x\rangle = \frac{1}{\sqrt{N}} \left(|0\rangle + e^{\frac{2\pi i}{2} x} |1\rangle\right) \otimes$$

$$\left(|0\rangle + e^{\frac{2\pi i}{2^2} x'} |1\rangle\right) \otimes \qquad x' = [0 x_2 \cdots x_n]$$

$$\vdots \qquad x'' = [0 0 x_3 \cdots x_n]$$

$$\left(|0\rangle + e^{\frac{2\pi i}{2^n} \overbrace{x''\cdots'}^{n-1}} |1\rangle\right) \equiv |\tilde{x}\rangle \qquad x^{\overbrace{\prime\cdots\prime}^{n-1}} = [0 0 \cdots 0 x_n]$$

Circuit implements (from $UROT_n$, from $UROT_2$, from H gate)

$$|x_1\rangle \longrightarrow \left(|0\rangle + \exp\left[\frac{2\pi i}{2^n} x_n + \frac{2\pi i}{2^{n-1}} x_{n-1} + \cdots + \frac{2\pi i}{2^2} x_2 + \frac{2\pi i}{2^1} x_1\right] |1\rangle\right)$$

Recall: $x = [x_1 x_2 \cdots x_n]$

$$= 2^{n-1} x_1 + 2^{n-2} x_2 + \cdots + 2^0 x_n$$

$$\Rightarrow |x_1\rangle \longrightarrow |0\rangle + \exp\left[\frac{2\pi i}{2^n} x\right] |1\rangle$$

(from $UROT_{n-1}$, from H gate)

$$|x_2\rangle \longrightarrow \left(|0\rangle + \exp\left[\frac{2\pi i}{2^{n-1}} x_n + \frac{2\pi i}{2^{n-2}} x_{n-1} + \cdots + \frac{2\pi i}{2^1} x_2\right] |1\rangle\right)$$

$\downarrow$

define $x' = [0 x_2 x_3 \cdots x_n]$

$$= 2^{n-1} 0 + 2^{n-2} x_2 + \cdots + 2^0 x_n$$

then, above is $\frac{x'}{2^{n-1}}$ and we have

$$|x_2\rangle \rightarrow |0\rangle + \exp\left[\frac{2\pi i}{2^{n-1}} x'\right] |1\rangle$$

$$\vdots$$

$$|x_n\rangle \longrightarrow |0\rangle + \exp\left[\frac{2\pi i}{2^1} \overbrace{x'\cdots'}^{n-1}\right] |1\rangle$$

$= [0 0 0 \cdots 0 x_n]$

(2) Quantum Phase Estimation

Objective: given a unitary U and its eigenstate/eigenvector $|\psi\rangle$

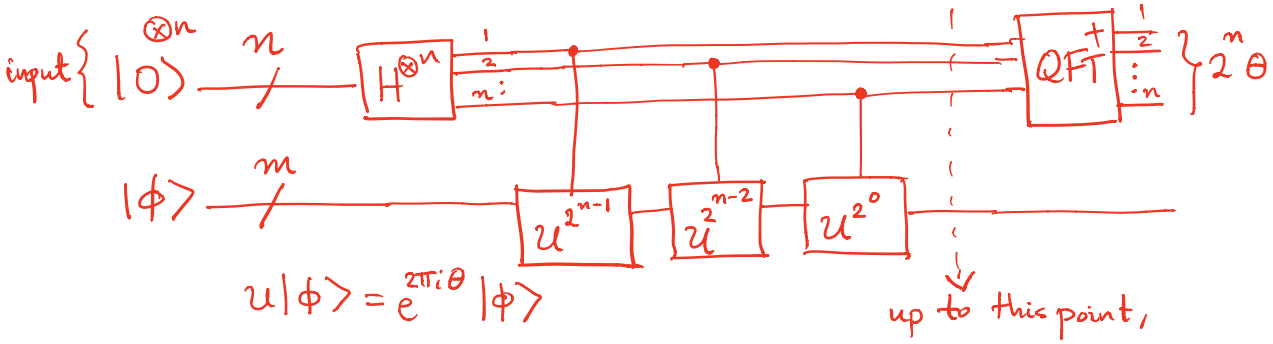
$$U|\psi\rangle = e^{i\theta_\psi}|\psi\rangle$$

- can we find θ_ψ? [Assume can prepare $|\psi\rangle$]

Phase estimation: allows us to convert phase information into amplitudes that we can measure.

QPE protocol:

given $U|\phi\rangle = e^{2\pi i\theta}|\phi\rangle$, QPE gives $2^n\theta$, where n is the number of qubits used to estimate θ.



subtleties: (1) notice the $2\pi\theta$ in the QPE protocol

(2) notice that we are doing $QFT^\dagger$, not QFT.

⇒ need to carefully build. Recall:

$$(ABC)^\dagger = C^\dagger B^\dagger A^\dagger$$

(3) yesterday's lab solution: used $U_1(\lambda)$ gate in Qiskit and did QPE to find λ.

↓ can see Qiskit textbook Shor's algo chapter for $QFT^\dagger$ implementation

(4) reference Quantum counting

Shor's Algorithm

(1) Problem: already discussed that factoring a number

$$N = pq$$

where p, q are prime and large is classically difficult

$O(\exp[c \cdot n^{1/3}(\log n)^{2/3}])$ using best known methods.

(2) Quick primer on modular arithmetic

$5 \div 3 =$ quotient 1, remainder 2

$$3\overline{)5}\quad 1,\ \text{remainder } 2$$

$$5 \equiv 2 \pmod 3$$

x	1	2	3	4	5	6	7	8	9	
$x \equiv$	1	2	0	1	2	0	1	2	0	$\pmod 3$

notice: $x \equiv 0 \pmod 3 \Rightarrow x$ is a multiple of 3

$x \equiv 1 \pmod 3 \Rightarrow x$ is 1 + some multiple of 3

generally,

$$x \equiv y \pmod 3 \Rightarrow x = 3k + y \text{ for some } k \in \mathbb{Z}$$

also notice the periodicity of modular arithmetic

$x \equiv y \pmod N$ means $y \in \{0, 1, 2, \ldots, N\}$

eg: $x \equiv y \pmod 3$ means $y \in \{0, 1, 2\}$

(2) Solution:

begin algorithm: $\equiv \gcd(a, N) = 1$

(1) pick "a" coprime with $N = pq$

- if unsure, check quickly if coprime

(2) find the "order" r of the function $a^r \pmod N$

$\equiv$ smallest r such that $a^r \equiv 1 \pmod N$

(3) if r is even,

$$x \equiv a^{r/2} \pmod N$$

if $x+1 \not\equiv 0 \pmod N$ then

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

else: find another "a".

(3) Concrete example: factoring 15

$15 = [1111] = 4$ bits.

coprime: pick $a = 13$.

$$13^x \pmod{15} = \underset{x=}{\overset{0}{1}}, \overset{1}{13}, \overset{2}{4}, \overset{3}{7}, \overset{4}{1}, \overset{5}{13}, \overset{6}{4}, \overset{7}{7}$$

Smallest $r > 0$ s.t. $13^r \equiv 1 \pmod{15}$ is $r = 4$.

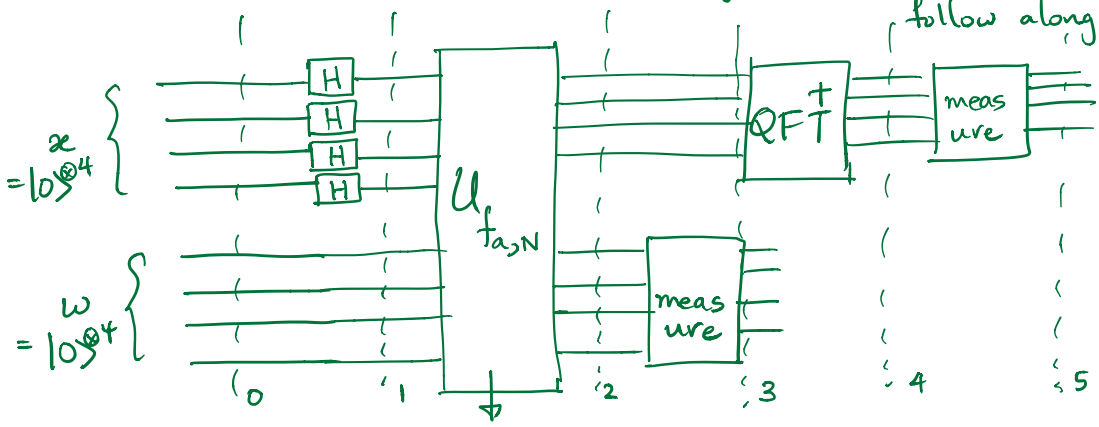
given $r = 4$,

$$x \equiv 13^{4/2} \pmod{15} \equiv 4 \pmod{15}$$

$$x + 1 = 5 \not\equiv 0 \pmod{15}$$

$$\{p, q\} = \{4-1, 4+1\} = \{3, 5\}$$

(4) Quantum circuit for factoring 15 (subtleties later, for now follow along)



$$|x\rangle|w\rangle \rightarrow |x\rangle|w \oplus f_{a,N}(x)\rangle$$

step 0: $|0\rangle^{\otimes 4}|0\rangle^{\otimes 4}$ (x, w) $\qquad f_{a,N}(x) \equiv a^x \pmod N$

step 1: $= \left[H^{\otimes 4}|0\rangle^{\otimes 4}\right]|0\rangle^{\otimes 4}$

$= \frac{1}{4}\left[|0\rangle_4 + |1\rangle_4 + |2\rangle_4 + \cdots + |15\rangle_4\right]|0\rangle_4$

step 2: $\frac{1}{4}\left[|0\rangle_4|0 \oplus 13^0 \pmod{15}\rangle_4 + |1\rangle_4|0 \oplus 13^1 \pmod{15}\rangle_4 + \cdots\right]$

since $0 \oplus z = z$, we have

$$= \frac{1}{4}\Big[\underset{x}{|0\rangle_4}\underset{w}{|1\rangle_4} + |1\rangle_4|13\rangle_4 + |2\rangle_4|4\rangle_4 + |3\rangle_4|7\rangle_4$$
$$+ |4\rangle_4|1\rangle_4 + |5\rangle_4|13\rangle_4 + |6\rangle_4|4\rangle_4 + |7\rangle_4|7\rangle_4$$
$$+ |8\rangle_4|1\rangle_4 + |9\rangle_4|13\rangle_4 + |10\rangle_4|4\rangle_4 + |11\rangle_4|7\rangle_4$$
$$+ |12\rangle_4|1\rangle_4 + |13\rangle_4|13\rangle_4 + |14\rangle_4|4\rangle_4 + |15\rangle_4|7\rangle_4\Big]$$

step 3: after measuring w register, let's say we measured 7:

$$\frac{1}{2}\left[|3\rangle_4 + |7\rangle_4 + |11\rangle_4 + |15\rangle_4\right] \otimes |7\rangle_4$$

↳ note the normalization

step 4: apply $QFT^\dagger$ on the x register

Recall:

$$QFT\,|x\rangle = |\tilde{x}\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{\frac{2\pi i}{N} xy} |y\rangle$$

$$QFT^\dagger |\tilde{x}\rangle = |x\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{\frac{-2\pi i}{N} xy} |y\rangle$$

$$QFT^\dagger |3\rangle_4 = \frac{1}{\sqrt{16}} \sum_{y=0}^{15} e^{\frac{-2\pi i \cdot 3y}{16}} |y\rangle$$

$$= \frac{1}{4} \sum_{y=0}^{15} e^{\frac{-\pi i}{8} 3y} |y\rangle$$

$$QFT^\dagger |7\rangle_4 = \frac{1}{4} \sum_{y=0}^{15} e^{\frac{-\pi i}{8} 7y} |y\rangle$$

$$QFT^\dagger |11\rangle_4 = \frac{1}{4} \sum_{y=0}^{15} e^{\frac{-\pi i}{8} 11y} |y\rangle$$

$$QFT^\dagger |15\rangle_4 = \frac{1}{4} \sum_{y=0}^{15} e^{\frac{-\pi i}{8} 15y} |y\rangle$$

$$QFT^\dagger |x\rangle = \frac{1}{8} \sum_{y=0}^{15} \left[\exp\left(-i \cdot \frac{3\pi}{8} y\right) + \exp\left(-i \frac{7\pi}{8} y\right) + \exp\left(-i \frac{11\pi}{8} y\right) + \exp\left(-i \frac{15\pi}{8} y\right) \right] |y\rangle$$

show computer demo

$$= \frac{1}{8} \cdot \left[4\,|0\rangle_4 + 4i\,|4\rangle_4 - 4\,|8\rangle_4 - 4i\,|12\rangle_4 \right]$$

step 5: measure the $|x\rangle$ register: get $|0\rangle$ or $|4\rangle$ or $|8\rangle$ or $|12\rangle$ with equal probability of $\frac{1}{4}$.

Analyze measured result: meas. results peak near $j\frac{N}{r}$ for some $j \in \mathbb{Z}$

$|0\rangle$ is trivial. If we measure $|0\rangle$, restart. ✗

$|4\rangle$: $j \cdot \frac{16}{r} = 4 \Rightarrow r = 4$ if $j=1$. $r=4$ is even = good

$x \equiv a^{r/2} \pmod N = 13^{4/2} \pmod{15} = 4$

$x+1 = 5$; $\gcd(x+1, N) = 5$

$x-1 = 3$; $\gcd(x-1, N) = 3$ ✓

$|8\rangle$: $j \cdot \frac{16}{r} = 8 \Rightarrow$ $r=2$ and $j=1$ or $r=4$ and $j=2$ (works like above ✓)

$x \equiv 13^1 \pmod{15} = 2$

$x+1 = 3$; $\gcd(x+1, N) = 3$

$x-1 = 1$ $\gcd(x-1, N) = 1$ } partial

$|12\rangle$: $j \cdot \frac{16}{r} = 12 \Rightarrow r = 4$ and $j=3$, works like above ✓

Looks like 3/4 results work, and we are able to extract the factors successfully.

- demo of Vandersypen, Steffen et al paper showing how to factor 15 on a quantum computer

Caveats: need $2n$ qubits at the input register for $|x\rangle$ instead of just n.

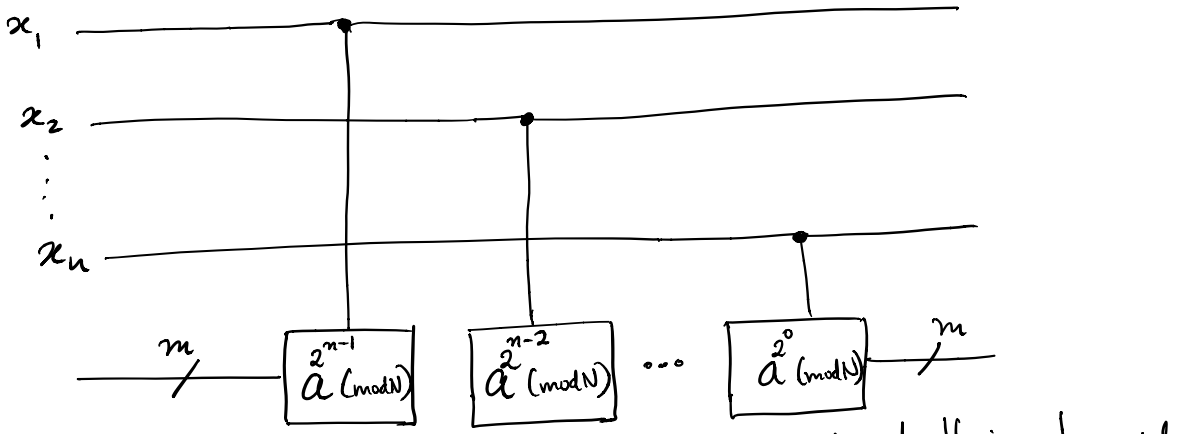
- with probability $> \frac{1}{2}$, r will be even and $a^{r/2} + 1 \not\equiv 0 \pmod N$

How to implement $U_{f_{a,N}}$

Recall: $f_{a,N}(x) \equiv a^x \pmod N$

$$x = [x_1 x_2 \cdots x_n] = 2^{n-1}x_1 + 2^{n-2}x_2 + \cdots + 2^0 x_n$$

$$\begin{aligned} f_{a,N}(x) &\equiv a^x \pmod N \\ &= a^{2^{n-1}x_1 + 2^{n-2}x_2 + \cdots + 2^0 x_n} \pmod N \\ &= a^{2^{n-1}x_1}\, a^{2^{n-2}x_2} \cdots a^{2^0 x_n} \pmod N \end{aligned}$$



↳ depth is polynomial in n.

can see: Shor's algo is QPE in disguise

- Also see Qiskit textbook chapter for an example of modular exponentiation implemented in Qiskit.