Notes for Phase Estimation and Factoring

Ansh Nagda

Quantum Fourier Transform

For this section, we will think of the classical states as integers between $0$ and $2^n - 1$. $\mathcal{F}$ (which is called the Quantum Fourier Transform) is the unitary operation taking a classical state $|j\rangle$ to

$$\mathcal{F}|j\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i j/k} |k\rangle$$

So given a state $|x\rangle = \sum_{k=0}^{2^n-1} a_k |k\rangle$, the coefficient of $|j\rangle$ in $\mathcal{F}|x\rangle$ is $\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i j/k} a_k$, which is reminiscent of the $j^{th}$ value of the Discrete Fourier Transform of the sequence $\{a_k\}$. We will assume that $\mathcal{F}$ (and $\mathcal{F}^\dagger$) can be implemented efficiently as a quantum circuit.

Phase Estimation

Problem Statement:

Given a unitary operation $V$ on $n$ qubits and an eigenvector $|\psi\rangle$ that has eigenvalue $e^{2\pi i \phi/2^n}$ for some integer $\phi$, find $\phi$.

Algorithm:

1. Prepare an initial state of $2n$ qubits where the first $n$ qubits are in an equal superposition of all possible $2^n$ classical states, and the last $n$ qubits are initialized as $|\psi\rangle$.
2. For $i$ from 0 to $n - 1$, perform a controlled $V^{2^i}$ operation on the last $n$ qubits with the $i + 1^{th}$ qubit as the control qubit.
3. Perform $\mathcal{F}^\dagger$ (Inverse QFT) on the first $n$ bits.
4. Measure the first $n$ bits to obtain an integer between $0$ and $2^n - 1$. Return that integer.

Explanation of Algorithm:

The main idea of this algorithm is that it entangles the last $n$ qubits with the first $n$ qubits with a phase value that differs based on the binary value of the first $n$ bits. Here is the way that I found most helpful to think about what the algorithm is doing: say that the initial state of the first $n$ qubits was a classical state equal to $|x\rangle$ for some bit string $x$. The for loop in step 2 applies $V^{2^i}$ to the last $n$ qubits
exactly when the $i^{th}$ bit of $x$ is 1, so the final state of the last $n$ qubits would be $V^x |v\rangle = e^{2\pi i x/2^n} |v\rangle$. Notice that the sum in the exponent is exactly equal to the value of $x$ as a binary integer, so the final state of the last $n$ qubits is $V^x |v\rangle = e^{2\pi i x/2^n} |v\rangle$.

But the initial state we prepare in the algorithm is not a classical state, it is instead a superposition of all classical states. So the state of the $n + 1$ qubits after step 2 will be a superposition of all these states. In particular, it equals

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} e^{2\pi i x/2^n} |x, v\rangle = \left( \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} e^{2\pi i x/2^n} |x\rangle \right) \otimes |v\rangle$$

The first part of this tensor product looks like the QFT of the state $|\phi\rangle$, so performing an inverse QFT on the first $n$ qubits will result in the classical state of $|\phi\rangle$.

One detail to note is that the eigenvector must not be exactly $e^{2\pi i x/2^n}$ for some integer $x$: the algorithm will return an integer $x$ that we can assume gives the closest approximation to the eigenvector.

**Reduction from Order Finding to Phase Estimation**

During the last lecture, we reduced factoring to the Order Finding problem. It remains to show how to solve Order Finding using this algorithm. As a reminder, **Order Finding** is a problem with inputs $y < N$, and the goal is to find the smallest $r$ such that $y^r = 1 \mod N$. Choose $n$ to be larger than $\log N$. Let $V$ be the unitary operation that sends $|x\rangle$ to $|x y^{-1} \mod N\rangle$ if $x$ is an integer less than $N$ that is coprime to $N$, and sends $|x\rangle$ to itself otherwise. Since $V$ is a permutation matrix, this is a unitary operation. $V$ has eigenvectors of the following form for $s$ from 0 to $k - 1$:

$$|v_s\rangle = \frac{1}{\sqrt{k}} \sum_{k=0}^{r-1} e^{2\pi i k s/r} |y^k \mod N\rangle$$

Multiplying $V$ by $|v_s\rangle$ essentially has the effect of shifting the coefficients of the states $|y^k \mod N\rangle$ cyclically, which corresponds to multiplying each state by $e^{2\pi i s/r}$, which is the eigenvalue corresponding to $|v_s\rangle$. The issue is that we cannot prepare any of the states $|v_s\rangle$ for use in the phase estimation algorithm without knowing the value of $r$. But it turns out that we can easily prepare their superposition:

$$\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |v_s\rangle = \frac{1}{r} \sum_{k=0}^{r-1} \left( \sum_{s=0}^{r-1} e^{2\pi i k s/r} \right) |y^k \mod N\rangle = \frac{1}{r} \sum_{k=0}^{r-1} \frac{e^{2\pi i k} - 1}{e^{2\pi i k/r} - 1} |y^k \mod N\rangle$$

Where the last equality follows by the formula for geometric sum. For all values $k$ not equal to 0, the numerator is 0 and the denominator is nonzero, so they all evaluate to 0. For $k = 0$, the geometric sum equals exactly $r$ since all terms in the sum are 0. Therefore this superposition equals $\frac{1}{r} \cdot r |y^0\rangle = |1\rangle$, which can be prepared. So if we run the phase estimation algorithm using $V$ and this
superposition of eigenvectors, the result of the algorithm will be the phase of an eigenvalue of one of the $v_s$ picked uniformly at random. That is, the result is $2^n s/\pi$ with $s$ uniformly picked at random that is unknown to us. The next step was not completely explained during lecture, but it turns out that this is enough to find the true $\pi$ using some classical techniques.