Lecture 18

Complexity of quantum ordering finding part. Comment: Will cover main ideas today, may go into more details later (TBD). Note that I chose to cover Shor's algorithm as described in his original paper, which is somewhat different from Kitaev's phase-estimation-based variant that is more often seen in textbooks. The latter is perhaps easier to teach but makes the similarity with Simon's algorithm that originally motivated Shor harder to see.

Recall the definition of quantum Fourier transform.

Definition 19. Let M be a positive integer. The QFT on $\mathbb{C}^M$ is the unitary defined by

$$\mathrm{QFT}_M \left|j\right\rangle = \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} \omega_M^{kj} \left|k\right\rangle, \quad \text{for all } j \in \{0,1,\dots,M-1\} \tag{93}$$

where $\omega_M := \exp(2\pi i/M)$.

Choose M to be a large positive integer; anything bigger than N^2 works[8]
Define

$$f\colon \{0,1,\dots,M-1\} \to \{0,1\dots,N-1\}; \quad x \mapsto a^x \mod N \tag{94}$$

We will construct a quantum circuit using the quantum oracle for f.
We will make the following

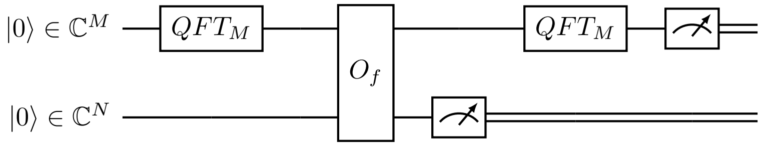
Simplifying assumption: $r \mid M$.

Remark 5. This assumption *significantly* simplifies the technical analysis but there's not much lost at a *conceptual* level. Under the assumption, the table below is perfectly rectangular, all the rows are "full". Without this assumption, the last row won't be full. But if you imagine M is large, then there would be so many full rows that their combined effect "washes out" the effect of the non-full ("corrupted") row. How large do you need M to be for this to happen? Answer: it turns out $M > N^2$ suffices and this is precisely why we imposed this condition at the start.

Then write $M = \lambda r$ for some positive integer λ.
Then can visualize the domain of f as the following table

$$\begin{array}{cccc}
0 & 1 & \cdots & r-1 \\
r & r+1 & \cdots & 2r-1 \\
\vdots & \vdots & & \vdots \\
(\lambda-1)r & (\lambda-1)r+1 & \cdots & \lambda r - 1
\end{array}$$

The value of f on each column is constant and it's distinct across columns. Comment: similar to Simon's problem where there are two rows; in fact, there's a direct correspondence at a more abstract level: here, each column is a *coset* of the subgroup $\langle r\rangle$ of $\mathbb{Z}_M$ generated by r; in Simon's problem, for the $f \in D_1$ case, each column is a coset of the subgroup $\langle s\rangle$ of $\mathbb{F}_2^n$ generated by the period s of f.



$$\frac{1}{\sqrt{M}} \sum_{j=0}^{M-1} \left|j\right\rangle \left|0\right\rangle \mapsto \frac{1}{\sqrt{M}} \sum_{j=0}^{M-1} \left|j\right\rangle \left|f(j)\right\rangle \tag{95}$$

Measure the second register: get $f(k)$ for some $k \in \{0,1,\dots,r-1\}$. Then the state of the first register collapses

$$\frac{1}{\sqrt{\lambda}} \sum_{l=0}^{\lambda-1} \left|lr+k\right\rangle. \tag{96}$$

Comment: you may wonder why we don't just measure here to save the next QFT, the reason is we don't know[9] k and so can't easily recover r from a measurement outcome of $lr+k$; we also can't easily "cancel out k" be doing repeats because the ks at each repeat are most likely different.

[8] This won't come into play in our analysis since we will make the simplifying assumption that $r \mid M$, but it will in the general case, see later.

[9] There is in fact a related quantum algorithm for finding k from $f(k)$ — we do know $f(k)$ — called the quantum discrete logarithm algorithm, also due to Shor. But that also involves a QFT so we're not saving anything.

Taking the QFT gives

$$\frac{1}{\sqrt{\lambda M}}\sum_{l=0}^{\lambda-1}\sum_{j=0}^{M-1}\omega_M^{(lr+k)\cdot j}\left|j\right\rangle = \frac{1}{\sqrt{\lambda M}}\sum_{j=0}^{M-1}\omega_M^{kj}\Big(\sum_{l=0}^{\lambda-1}\omega_M^{lrj}\Big)\left|j\right\rangle . \tag{97}$$

We have

$$\sum_{l=0}^{\lambda-1}\omega_M^{lrj} = \sum_{l=0}^{\lambda-1}(\omega_M^{rj})^l = \begin{cases}\frac{\omega_M^{\lambda rj}-1}{\omega_M^{rj}-1} = 0 & \text{if } rj \text{ is not a multiple of } M\\ \lambda & \text{if } rj \text{ is a multiple of } M = \lambda r\end{cases} \tag{98}$$

The second case corresponds to $j = 0, \lambda, 2\lambda, 3\lambda, \ldots, (r-1)\lambda$.

So the state can be written as

$$\sqrt{\frac{\lambda}{M}}\sum_{n=0}^{r-1}\omega_M^{kn}\left|n\lambda\right\rangle = \frac{1}{\sqrt{r}}\sum_{n=0}^{r-1}\omega_M^{kn}\left|n\frac{M}{r}\right\rangle . \tag{99}$$

If we measure this state, we get

$$n\frac{M}{r} \tag{100}$$

for some n chosen uniformly at random from $\{0, 1, \ldots, r-1\}$.

Finally, divide by M to get $f \coloneqq n/r$, which is a fixed decimal number, and

1. express f as a fraction in its simplest terms and read off the denominator r'.
2. check if $a^{r'} = 1 \mod N$, if so output r'. Else output "fail".

There are two cases:

1. If n happens to be coprime to r, then there can be no cancellations in n/r, so we can read r off from the denominator, That is, r' will be equal to r, the check passes, and the algorithm is definitely correct.
2. If n is not coprime to r, then there will be cancellations in n/r, so $r' < r$ and the check fails.

Suppose the first case occurs with probability at least γ. Then doing $O(1/\gamma)$ repeats of the algorithm ensures that we almost certainly land in the first case at some repeat, so we will output r as desired. The overall complexity of the algorithm is the complexity of each repeat times $O(1/\gamma)$.

Fact 8. The complexity of each repeat is $O(\log^3 N)$ including all costs (not just O_f; we're no longer in the query model), even if we drop the $r \mid M$ assumption. Moreover, γ can be set to $\Omega(1/\ln\ln(N))$.

Therefore, the overall complexity is $O(\log^3(N)\log\log(N))$, i.e., roughly $O(\log^3(N))$ since $\log\log(N)$ grows *very* slowly.