18.405J/6.841J Advanced Complexity Theory 20 March 2001

Lecture 12

Lecturer: Daniel Spielman *Scribe: Jonathan Herzog*

This particular lecture was structured into four parts:

0. The Fourier transform (FT)
1. Computing the discrete logarithm problem with arbitrary quantum FTs
2. Which quantum FT's can we compute?
3. Showing that it suffices to use the quantum FTs that can be computed

0 Fourier Transform

The Fourier transform is a linear transform, and so can be represented with an $n \times n$ matrix. Let $\omega = e^{\frac{2\pi i}{n}}$. Then M is a matrix where

$$M_{x,y} = \frac{1}{\sqrt{n}} \omega^{xy}$$

The quantum fourier transform (QFT) is the transformation on states:

$$QFT_N : |x\rangle \longrightarrow \frac{1}{\sqrt{n}} \sum_{y=0}^{N-1} \omega^{xy} |y\rangle$$

where both x and y are basis vectors of $n = \log N$ bits, interpreted as a number in binary. If N is a power of 2, then we can compute QFT_N in $O(n^2)$ time.

1 Discrete Log Problem

The discrete log problem is a basic number-theoretic problem whose hardness serves as the foundation of many modern cryptographic techniques and protocols. Given as inputs a prime p, a number g which generates $\mathcal{Z}_p^\star$ (i.e., the sequence 1, g, g^2, g^3... $g^{p-1} \bmod p$ contains all the numbers 1, 2, 3... $p-1$ in some order) and some number x, the discrete log problem is to compute an r such that $g^r = x \bmod p$. [**Scribe note**: The best known (classical) algorithms for this problem are the Pohlig-Hellman algorithm, which is fast if $p-1$ is smooth, and the number field sieve algorithm, which runs in $O\left(e^{(1.923+o(1))(\log p)^{\frac{1}{3}}(\log\log p)^{\frac{2}{3}}}\right)$ time.]

The discrete log problem can be done in QP, with Shor's algorithm:

1. Begin in the state $|0,0\rangle \otimes |0\rangle$, where the 0's are actually $\log(p-1)$ bit registers. (This is another way of writing a $3\log(p-1)$ bit register, where we treat each third of the register as a distinct unit.)
2. Apply QFT_{p-1} to each of the first two registers, where $\omega = e^{\frac{2\pi i}{p-1}}$. This moves the system into the state:

$$\frac{1}{p-1} \sum_{a \in \mathcal{Z}/(p-1)} \sum_{b \in \mathcal{Z}/(p-1)} |a,b\rangle \otimes |0\rangle$$

3. Replace the final $|0\rangle$ with $|g^a x^{-b} \bmod p\rangle$. (Since a and b are in the $|a,b\rangle$ register, this substitution is a reversible computation.[1]) This puts us in the state:

$$\frac{1}{p-1} \sum_{a \in \mathcal{Z}/(p-1)} \sum_{b \in \mathcal{Z}/(p-1)} |a,b\rangle \otimes |g^a x^{-b} \bmod p\rangle$$

4. Then we apply QFT_{p-1} again to each of the first two registers, putting us in state:

$$\frac{1}{p-1} \sum_{a \in \mathcal{Z}/(p-1)} \sum_{b \in \mathcal{Z}/(p-1)} \sum_{c \in \mathcal{Z}/(p-1)} \sum_{d \in \mathcal{Z}/(p-1)} \omega^{ac+bd} |c,d\rangle \otimes |g^a x^{-b} \bmod p\rangle$$

5. Measure the entire register.

Okay, why does this work?

Claim 1 *If you actually observe c, d, and $g^k \bmod p$, then c and d are uniformly chosen from pairs such that $cr = -d \bmod p$, and k is chosen uniformly from $[0 \ldots p-1]$.*

Proof

For a given c, d, and $g^k \bmod p$, what is the magnitude of $|c,d,g^k \bmod p\rangle$? Well, how many ways can we end up with those values? Look at the last register. In step 2, we replaced $|0\rangle$ with $|g^a x^{-b} \bmod p\rangle$, and $x = g^r \bmod p$. So if we observe $g^k \bmod p$ it must be that $k = a - br \bmod (p-1)$. Or, rewritten, that $a = br - k \bmod (p-1)$.

So, given a k, a is fixed by the value of b. So write the final state as:

$$\sum_c \sum_d \sum_k \beta^k_{c,d} |c,d,g^k\rangle$$

Through algebraic manipulation, we can find that

$$\beta^k_{c,d} = \frac{1}{(p-1)^2} \sum_{b \in \mathcal{Z}/(p-1)} \omega^{c(br-k)+db} \quad (1)$$

$$= \frac{1}{(p-1)^2} \omega^{ck} \sum_{b \in \mathcal{Z}/(p-1)} \omega^{b(cr+d)} \quad (2)$$

If $cr + d = 0 \bmod (p-1)$, then $\omega^{b(cr+d)} = 1$. In that case, $\left|\beta^k_{c,d}\right|^2 = \frac{1}{(p-1)^2}$. So, if $cr = -d \bmod p$, then the probability of observing the state $|c,d,g^k \bmod p\rangle$ is exactly $\frac{1}{(p-1)^2}$. But there are exactly $(p-1)^2$ such pairs of (c,d), so the probability of observing one such pair is exactly 1. And since k is free, it can range over all values in $[0 \ldots p-1]$. ■

Great. Now, if c is relatively prime to $p-1$, we can solve the equation $cr - d = 0 \bmod (p-1)$ to find the value of r. (What is the probability that c is relatively prime to $p-1$? It turns out to be at least $\frac{1}{\log p}$.)

As a sanity check, let's make sure that all the other amplitudes go to zero: Let $cr + d = f \bmod (p-1)$. Then if $f \neq 0$, look at the value of $\beta^k_{c,d}$ we derived in equation (2) above:

$$\beta^k_{c,d} = \frac{1}{(p-1)^2} \sum_{b \in \mathcal{Z}/(p-1)} \omega^{bf}$$

[1] That is, if g and x are considered to be constants, or hardwired into the circuit.

But we defined ω to be a $(p-1)$st root of unity, so the sum in the above equation is the sum of a number of roots of unity, all evenly spaced around 0. Hence, they all cancel each other out in the sum.

Now, as a digression, how can we use this algorithm to factor? The trick is to use a number theoretic fact: if m is a composite number, and $x^2 = y^2 \bmod m$, but $x \neq \pm y \bmod m$, then $x - y$ is a non-trivial factor of m. We also know that $(-1)^2 = 1 \bmod m$, so all we need to do is find another square root of unity.

So, choose a random $x < m$. Then find the order of x in $\mathcal{Z}_m^\star$.[2] In other words, find the least r such that $x^r = 1 \bmod m$. Then $x^{\frac{r}{2}}$ will be a square root of 1 mod m, and with probability .5 it will not be ± 1.

2 Calculable Quantum Fourier Transforms

So, which quantum transforms can we actually compute? When $N = 2^n$, i.e. the size of the matrix is a power of 2, it is easy to compute the function:

$$x \longrightarrow \frac{1}{\sqrt{N}} \sum_{y=0}^{2^n-1} e^{\frac{2\pi i x y}{N}} |y\rangle$$

Here's how it's done. Write x and y in binary notation:

$$\begin{aligned} x &= 2^{n-1}x_1 + 2^{n-2}x_2 + \ldots 2x_{n-1} + x_n \\ y &= 2^{n-1}y_1 + 2^{n-2}y_2 + \ldots 2y_{n-1} + y_n \end{aligned}$$

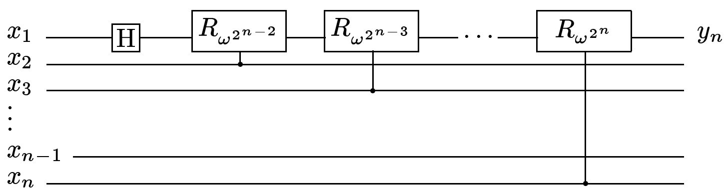
Then

$$\begin{aligned} xy \bmod 2^n = \; & y_1\left(2^{n-1}x_n\right) \\ & + y_2\left(2^{n-2}x_n + 2^{n-1}x_{n-1}\right) \\ & \vdots \\ & + y_n\left(x_n + 2x_{n-1} + \ldots 2^{n-1}x_1\right) \end{aligned}$$

So, letting $\omega = e^{\frac{2i\pi}{N}}$, we can write the transform as:

$$\begin{aligned} (x_1, \ldots x_n) \longrightarrow \; & \tfrac{1}{\sqrt{N}} \left(|0\rangle + \omega^{x_n 2^{n-1}} |1\rangle\right) \\ \otimes \; & \left(|0\rangle + \omega^{x_{n-1}2^{n-1} + x_n 2^{n-2}} |1\rangle\right) \\ \vdots \; & \\ \otimes \; & \left(|0\rangle + \omega^{x_n + 2x_{n-1} + \ldots + x_1 2^{n-1}} |1\rangle\right) \end{aligned}$$

So, for bit y_n of the output, we can construct the sequence of gates:



[2]Note that if $x < m$ and $x \notin \mathcal{Z}_m^\star$, then x is itself a factor of m and you're done.

where R_{ω^i} is the gate represented by the unary transform

$$\begin{pmatrix} 1 & 0 \\ 0 & \omega^i \end{pmatrix}$$

We then obtain bit y_{n-1} in a similar way from bits $x_2, \ldots, x_n$, and so on.

3 Sufficiency of these Transforms

[**Scribe note:** Due to time constraints, the lecturer gave only a proof sketch of this.] Since we can only use transforms where the size is a power of 2, that's exactly what we do when actually performing Shor's algorithm. Replace QFT_{p-1} with QFT_{2^n}, where $30(p-1)\log p < 2^n < 60(p-1)\log p$. Then, if we get (c, d) out of the first algorithm, there is a non-negligible chance that we get:

$$\left(\left\lceil \frac{2^n}{p-1} c \right\rceil , \left\lceil \frac{2^n}{p-1} d \right\rceil \right)$$

How much of a chance? We will see the pair on the right with probability at least $\frac{1}{10 \log p} \Pr[(c,d)]$. So, by using these larger transformations, we map each possible output of the original algorithm to something in a larger domain.

Another way to see this is to look at the results of

$$QFT_{2^n} \left(QFT_{p-1}^{-1} \left(|c\rangle \right) \right)$$

Before we measure, we get out some superposition

$$\sum_{c'=0}^{2^n} \beta_{c'} \left| c' \right\rangle$$

where $\beta_{c'}$ is distributed nicely. If you graph the distribution[3] of $\beta_{c'}$, you'll see that it is close to zero almost everywhere, with all the non-zero amplitudes in a nice bell-like distribution centered at $\left\lceil \frac{2^n c}{p-1} \right\rceil$, with a width of $\frac{2^n}{p-1}$.

In other words, using the calculable transformations instead of the exact transformations will give you results "close" to those from the exact algorithm.

[3]Which I can't do easily in LaTeX...