**CS 221: Computational Complexity** **Prof. Salil Vadhan**

**Lecture Notes 27**

**May 5, 2010** **Scribe: James Williamson**

# 1 Recap of Quantum Computation

- the state of an $n$-qubit register is given by:

$$\phi = \sum_{s \in \{0,1\}^n} \alpha_s |s\rangle \in \mathbb{C}^{2^n}, \qquad \sum_s |\alpha_s|^2 = 1$$

- starts in the state $|x\rangle|0^k\rangle|0^m\rangle$
- apply a sequence of local unitary operators, each on $O(1)$ qubits.
- measure the final state $\sum_s \alpha_s|s\rangle$ and get $s \in \{0,1\}^{n+k+m}$ with probability $|\alpha_s|^2$.
- output the last $m$ bits of $s$.

# 2 Quantum Fourier Transform

## 2.1 Discrete Fourier Transform

Take $f : \mathbb{Z}_M \to \mathbb{C}$ and map to $\hat{f} : \mathbb{Z}_M \to \mathbb{C}$ where below we assume that $M = 2^m$.

The transform takes the form:

$$\hat{f}(x) = \frac{1}{\sqrt{M}} \sum_{y \in \mathbb{Z}_M} f(y)\omega^{xy}, \qquad \omega = e^{2\pi i/M}$$

Now taking $x \in \mathbb{Z}_{M/2}$ define $f_{even} = f(x0), f_{odd} = f(x1)$ where we are fixing the least significant bit to separate even and odd inputs. As derived in the previous lecture it is possible to write $\hat{f}$ recursively in terms of the odd and even parts as follows:

$$\begin{aligned} \hat{f}(0x) &= \widehat{f_{\text{even}}}(x) + \omega^x \widehat{f_{\text{odd}}}(x) & (1) \\ \hat{f}(1x) &= \widehat{f_{\text{even}}}(x) - \omega^x \widehat{f_{\text{odd}}}(x) & (2) \end{aligned}$$

Using this recursive definition we can give the well known Fast Fourier Transform algorithm or FFT:

1. Compute $\widehat{f_{\text{even}}}$ and $\widehat{f_{\text{odd}}}$ recursively.
2. Recombine according to the recurrence stated above.

If the whole algorithm has (algebraic) complexity $T(M)$, then the first step has complexity $2T(M/2)$ and the second is $O(M)$ since we need to do vector operations over vectors of length up to $M$. Therefore $T(M) = 2T(M/2) + O(M)$ which when solved gives us a complexity of $O(M \log M)$. In the classical case this is provably optimal for algebraic circuits over $\mathbb{C}$ with bounded coefficients (i.e. all constants have magnitude $O(1)$, such as the roots of unity used above).

## 2.2 Quantum Fourier Transfrom

The Quantum Fourier Transform operates on an $m = \log M$ qubit state taking

$$\sum_{x \in \mathbb{Z}_M} f(x)|x\rangle \mapsto \sum_{x \in \mathbb{Z}_M} \hat{f}(x)|x\rangle$$

Note that we do not get the values of $\hat{f}(x)$ explicitly. However, we can measure the transformed state and get $x$ with probability $\left|\hat{f}(x)\right|^2$, so this enables us to sample the frequencies of the function $f$.

Note that the recurrences above imply that the QFT can be split into even and odd parts as follows:

$$\sum_{x \in \mathbb{Z}_M} \hat{f}(x)|x\rangle = \frac{1}{\sqrt{2}} \sum_{x \in \mathbb{Z}_{M/2}} \left[ \left( \widehat{f_{\text{even}}}(x) + \omega^x \widehat{f_{\text{odd}}}(x) \right) |0x\rangle + \left( \widehat{f_{\text{even}}}(x) - \omega^x \widehat{f_{\text{odd}}}(x) \right) |1x\rangle \right]$$

Now similarly to the classical DFT we give a recursive algorithm for $\text{QFT}_M$. We can use quantum mechanics to enable us to use only one recursive call and hence to lower the complexity from $O(M \log M)$ to $O(\log^2 M) = O(m^2)$.

The algorithm is as follows:

1. Start with $\sum_{x \in \mathbb{Z}_M} f(x)|x\rangle$ and rewrite as:

$$\sum_{x \in \mathbb{Z}_{M/2}} \left( f_{even}(x)|x0\rangle + f_{odd}(x)|x1\rangle \right)$$

2. Apply $\text{QFT}_{M/2}$ to the first $m-1$ qubits to obtain

$$\sum_{x \in \mathbb{Z}_{M/2}} \left( \widehat{f_{\text{even}}}(x)|x0\rangle + \widehat{f_{\text{odd}}}(x)|x1\rangle \right)$$

3. For $j = 0, \ldots, m-2$ with $|xb\rangle = |x_{m-2} \cdots x_0 b\rangle$, apply the following 2-qubit operation:

$$|x_j\rangle|b\rangle \rightarrow \begin{cases} \omega^{2^j}|x_j\rangle|b\rangle & x_j = b = 1 \\ |x_j\rangle|b\rangle & \text{otherwise} \end{cases}$$

Observe the effect of these $m-1$ operations is the following:

$$\begin{aligned} |x0\rangle &\mapsto |x0\rangle \\ |x1\rangle &\mapsto \left( \prod_{j : x_j = 1} \omega^{2^j} \right) |x1\rangle = \omega^x |x1\rangle \end{aligned}$$

So our state now is:
$$\sum_{x \in \mathbb{Z}_{M/2}} \left[ \widehat{f_{\text{even}}}(x)|x0\rangle + \omega^x \widehat{f_{\text{odd}}}(x)|x1\rangle \right].$$

4. Apply the Hadamard gate to the last qubit to obtain state:
$$\frac{1}{\sqrt{2}} \sum_{x \in \mathbb{Z}_{M/2}} \left[ \left( \widehat{f_{\text{even}}}(x) + \omega^x \widehat{f_{\text{odd}}}(x) \right) |x0\rangle + \left( \widehat{f_{\text{even}}}(x) - \omega^x \widehat{f_{\text{odd}}}(x) \right) |x1\rangle \right]$$

5. Swap the least significant qubit and the most significant qubit to obtain state.
$$\frac{1}{\sqrt{2}} \sum_{x \in \mathbb{Z}_{M/2}} \left[ \left( \widehat{f_{\text{even}}}(x) + \omega^x \widehat{f_{\text{odd}}}(x) \right) |0x\rangle + \left( \widehat{f_{\text{even}}}(x) - \omega^x \widehat{f_{\text{odd}}}(x) \right) |1x\rangle \right] = \sum_{x \in \mathbb{Z}_M} \hat{f}(x),$$
as desired

For the complexity of this algorithm 1 and 5 are free operations, 4 takes one gate, 3 takes $m-1$ and 2 takes $T(M/2)$ with total complexity given by $T(M) = T(M/2) + \log M$. This expands to $O(\log^2 M)$ as desired, making this algorithm polynomial in the number of bits.

# 3 Factoring on a Quantum Computer

We shall use without proof the known result that there is a classical, randomized reduction from factoring to finding the order of a number modulo $N$. To define this problem more formally, consider $N$ and $A \in \mathbb{Z}_N^* = \{b \in \{0, \dots, N-1\} \mid \gcd(b, N) = 1\}$ Then we want to find $\text{ord}_N(A)$ which is the least $0 < x < N-1$ such that $A^x \equiv 1 \mod N$.

Now we give a quantum algorithm for order finding given $N, A$. Let $m = \lceil 5 \log N \rceil, M = 2^m = \Theta(N^5)$.

1. Generate the uniform superposition over $\mathbb{Z}_M$
$$\frac{1}{\sqrt{M}} \sum_{x \in \mathbb{Z}_M} |x\rangle$$
by applying the Hadamard gate $m$ times.

2. Use classical modular arithmetic to send each $|x\rangle|0\rangle \mapsto |x\rangle|A^x \mod N\rangle$

3. Measure to obtain $y_0 \in \mathbb{Z}_N^*$ from each $A^x \mod N$ leaving the state as follows:
$$\frac{1}{\sqrt{K}} \sum_{x \in \mathbb{Z}_M : A^x \mod N = y_0} |x\rangle|y_0\rangle$$
where $K = \#\{x \,|A^x \mod N = y_0\}$. Notice that if $A^x \bmod N = y_0$, then we also have $A^{x+r} \bmod N = y_0$ and $A^{x-r} \bmod N = y_0$ where $r = \text{ord}_N(A)$. Conversely, if $A^{x_1} \bmod N = y_0$ and $A^{x_2} \bmod N = y_0$, then $A^{x_1 - x_2} \bmod N = 1$, so $r$ divides $x_1 - x_2$. This implies that the set of $x \in \{0, 1, \dots, M-1\}$ such that $A^x \bmod N = y_0$ is an arithmetic progression $\{x_0, x_0 + r, x_0 + 2r, \dots, x_0 + (K-1)r\}$, where $x_0 < r$ and $K = \lfloor (M - x_0 - 1)/r \rfloor + 1 \approx M/r$.

So our state is equal to the following sum:

$$\frac{1}{\sqrt{K}}(|x_0\rangle + |x_0 + r\rangle + cdots + |x_0 + (K-1)r\rangle).$$

Thinking of this state as a function $\sum_x f(x)|x\rangle$, the function $f$ has a periodicity of $r$ (i.e. $f(x+r) = f(x)$ for most values of $x \in \mathbb{Z}_M$ - except possibly for values close to 0 or $M$). Since the Quantum Fourier Transform allows us to sample the frequencies of a function, we should be able to use it to recover the period $r$.

4. Apply the QFT to this sum and obtain:

$$\sum_x \left( \frac{1}{\sqrt{KM}} \sum_{l=0}^{K-1} \omega^{(x_0+lr)x} \right) |x\rangle$$

This is since $f(y)$ is $1/\sqrt{K}$ for values $y$ of the form $x_0 + lr$ and 0 elsewhere.

5. Measure and obtain $x \in \mathbb{Z}_M$ with probability

$$\frac{1}{KM} \left| \sum_{l=0}^{K-1} \omega^{lrx} \right|^2$$

6. Find $a, b \in \mathbb{N}$ such that $|a/b - x/M| < 1/10M$ where $\gcd(a,b) = 1$ and $b < N$. This can be done classically with continued fractions and the pair $a, b$ is unique.

   Compute $A^b \mod N$ and check if it is congruent to 1. If yes output $b$.

For analysis we claim that $b = r$ with probability $\Omega(1/\log N)$. Thus repeating $O(\log N)$ times and taking the smallest value of $b$ obtained will yield $\mathrm{ord}_N(A)$ with high probability. We will show the simple case where $r|M$, the general case can be found in the Arora–Barak text.

In this case $K = M/r$, and we have:

$$\sum_{x=0}^{K-1} \omega^{lrx} = \begin{cases} K & x \text{ a multiple of } M/r \\ 0 & \text{otherwise} \end{cases}$$

This holds because $\omega$ is a primitive $M$'th root of unity: if $x$ is a multiple of $M/r$,then $\omega^{rx} = 1$, and otherwise $\omega^{rx}$ is an $M/r$'th root of unity other than 1, so its powers will be spread out evenly on the unit circle and cancel out.

This tells us that

$$\Pr\left[output = x\right] = \begin{cases} K^2/KM = 1/r & x \text{ a multiple of } M/r \\ 0 & \text{otherwise} \end{cases}$$

Therefore $x$ is a uniformly random multiple of $M/r$, i.e. $x/M = c/r$ where $c$ is a random number between 0 and $r-1$. Note that if $c$ and $r$ are relatively prime, then the pair $(a,b)$ will have to be $(c,r)$ and we'll output $r$. The probability that $c$ and $r$ are relatively prime is at least:

$$\frac{\#(\text{primes} < r) - \#(\text{prime divisors of } r)}{r} \geq \frac{\Omega(r/\log r) - \log r}{r} = \Omega(1/\log r)$$

This gives the desired probability of success.