**Algebra and Computation** Course Instructor: V. Arvind

# Lecture 23: Shor's Algorithm for Integer Factoring

*Lecturer: V. Arvind* *Scribe: Ramprasad Saptharishi*

## 1 Overview

In this lecture we shall see Shor's algorithm for order finding, and therefore for integer factoring.

## 2 The First Steps

We are given a number $a \in \mathbb{Z}_N^\star$ and we need to find the $r = ord_N(a)$.

Pick an $L$ such that $N^2 \leq 2^L \leq 2N^2$ and let $q = 2^L$. Hence the group $\mathbb{Z}_q$ is the set of all $L$-bit binary strings. As in the Simon's problem, prepare the uniform superposition using the hadamard transform.

$$|\psi\rangle = \frac{1}{\sqrt{q}} \sum_{x=0}^{q-1} |x\rangle$$

And we shall assume that our function $f : x \mapsto a^x \pmod{n}$ is given as a unitary tranform where $U_f(x, y) = (x, y \oplus f(x))$. Thus applying this matrix to the uniform superposition padded with 0s, we get:

$$\frac{1}{\sqrt{q}} \sum_{x=0}^{q-1} |x\rangle \left|0^L\right\rangle \longrightarrow \frac{1}{\sqrt{q}} \sum_{x=0}^{q-1} |x\rangle |f(x)\rangle$$

Now measuring the second register would give us some $a^l \pmod{N}$ for some least $l$. And hence, the current state would then be:

$$\frac{1}{\sqrt{A+1}} \sum_{j=0}^{A} |l + jr\rangle \quad , \quad A = \left\lfloor \frac{q-l-1}{r} \right\rfloor$$

Our job is to retrieve the $r$ and hence we need to get rid of the $l$. This is where the fourier tranform comes in.

## 2.1 Fourier Tranform over $\mathbb{Z}_q$

The fourier transform over $\mathbb{Z}_q$ is the following map:

$$F_q : |y\rangle \rightarrow \frac{1}{\sqrt{q}} \sum_{c=0}^{q-1} e^{\frac{2\pi i}{q} yc} |c\rangle$$

Shor showed that there exists a polynomial sized quantum circuit for the fourier tranform. For the moment, let us take it for granted that there does exists a polynomial sized quantum circuit though we shall prove it later in the lecture.

The quantum state that we are in is

$$\frac{1}{\sqrt{A+1}} \sum_{j=0}^{A} |l + jr\rangle$$

Applying the fourier tranform to this, we have:

$$\frac{1}{\sqrt{q(A+1)}} \sum_{j=0}^{A} \sum_{c=0}^{q-1} e^{\frac{2\pi i}{2^L}(l+jr)c} |c\rangle = \frac{1}{\sqrt{q(A+1)}} \sum_{c=0}^{q-1} \alpha_c |c\rangle$$

### The Easy case: $r \mid q$

Since $r$ divides $q$ and we chose the least $l$ it follows that $A = \frac{q}{r} - 1$. Therefore,

$$\alpha_c = e^{\frac{2\pi i}{q} cl} \sum_{j=0}^{A} \left( e^{2\pi i \frac{cr}{q}} \right)^j$$

Suppose $q \nmid cr$ then $\omega = e^{2\pi i \frac{c}{q/r}} \neq 1$ is a $q/r$-th root of unity. Therefore,

$$\alpha_c = e^{\frac{2\pi i}{q} cl} \left( \sum_{j=0}^{\frac{q}{r}-1} \omega^j \right) = 0$$

Since the probability amplitude is 0 for the case when $q \nmid cr$, measuring $c$ will give us a number that is a multiple of $\frac{q}{r}$ with uniform probability.

Hence we now will have a fraction $\frac{\lambda q}{r}$. How do we recover $r$ from this? Suppose $\lambda$ was coprime to $r$, we know $\frac{c}{q} = \frac{\lambda}{r}$. Since $\frac{\lambda}{r}$ is in its reduced form, just take $\frac{c}{q}$ and get it to the reduced form; the denominator would then give us the $r$.

How do we make sure that we get a fraction such that $\lambda$ and $r$ are coprime? With good probability we will. The probability that this happens is $\frac{\phi(r)}{r} > \frac{1}{\log n}$. Hence we are safe.

**The general case:$r \nmid q$**

For any $c$ in the domain,

$$\begin{aligned}\Pr[c \text{ is measured}] &= \frac{1}{q(A+1)}\left|\sum_{j=0}^{A} e^{2\pi i \frac{c(l+rj)}{q}}\right|^2 \\ &= \frac{1}{q(A+1)}\left|\sum_{j=1}^{A} e^{2\pi i j(cr \bmod q)}\right|^2\end{aligned}$$

Let the event $E = \left\{c \ : \ \frac{-r}{2} \leq cr \bmod q \leq \frac{r}{2}\right\}$. To analyse the probability that this event will happen, for every $0 \leq \lambda \leq r$ look at the interval $\left[\lambda q - \frac{r}{2}, \lambda q + \frac{r}{2}\right]$. This interval will contain a multiple of $r$. There is a possibility of both end points of this interval being multiples of $r$. But the following argument shows that this is not possible.

$$\begin{aligned} cr &= \lambda q - \frac{r}{2} \\ (c+1)r &= \lambda q + \frac{r}{2} \\ \implies (2c+1)r &= 2\lambda q \end{aligned}$$

but the last line would force $r \geq 2q$ which is absurd by the choice of $q$.

Thus $|E| \geq r$ since there are $r$ possible $\lambda$'s. We now have

$$\begin{aligned} |cr - \lambda q| &\leq \frac{r}{2} \\ \left|\frac{c}{q} - \frac{\lambda}{r}\right| &\leq \frac{1}{2q} \leq \frac{1}{2N^2} \end{aligned}$$

The following theorem then show how to recover $r$ from this.

**Theorem 1.** *For any $\sigma \in \mathbb{Q}$, if $\left|\sigma - \frac{a}{b}\right| \leq \frac{1}{2b^2}$ then $\frac{a}{b}$ is one of the convergents[1] of $\sigma$.*

Thus all we need to do is get $\frac{c}{q}$, look at all its convergents and one of them will be $\frac{\lambda}{r}$. As in the earlier case, the probability that $\frac{\lambda}{r}$ will be in its reduced form will happen with good probability and thus the denominator of the fraction is the order of $a$.

Hence we just need to show that event $E$ will happen with good probability.

[1] truncation of continued fractions

### 2.2 Bounding probability of event $E$

*Needs to be done, didn't take good notes here.*

## 3 Quantum Circuit for Fourier Transform

We want a circuit that transforms $|y\rangle$ to $\frac{1}{\sqrt{q}}\sum_{c=0}^{q-1} e^{\frac{2\pi i}{q}cy}|c\rangle$. Fix a $c$ and lets its binary representation be $c_0c_1\cdots c_{L-1}$ and that of $y$ be $y_0y_1\cdots y_{L-1}$.

$$\begin{aligned} e^{\frac{2\pi i}{2^L}cy}|c\rangle &= e^{\frac{2\pi i}{2^L}y(2^{L-1}c_0+\cdots+c_{L-1})}|c_0\rangle|c_1\rangle\cdots|c_{L-1}\rangle \\ &= \left(e^{2\pi i\frac{y}{2^L}(2^{L-1}c_0)}|c_0\rangle\right)\otimes\left(e^{2\pi i\frac{y}{2^L}(2^{L-2}c_1)}|c_1\rangle\right)\otimes\cdots\otimes\left(e^{2\pi i\frac{y}{2^L}(c_{L-1})}|c_{L-1}\rangle\right) \\ &= \left(e^{2\pi i(0.y_{L-1})c_0}|c_0\rangle\right)\otimes\left(e^{2\pi i(0.y_{L-2}y_{L-1})c_1}|c_1\rangle\right)\otimes\cdots\otimes\left(e^{2\pi i(0.y)c_{L-1}}|c_{L-1}\rangle\right) \end{aligned}$$

Thus, summing over the possible bits for each $c_i$ we get the following tensor product.

$$\sum_{c=0}^{q-1} e^{\frac{2\pi i}{q}yc} = \left(\frac{|0\rangle+e^{2\pi i(0.y_{L-1})}|1\rangle}{\sqrt{2}}\right)\otimes\left(\frac{|0\rangle+e^{2\pi i(0.y_{L-2}y_{L-1})}|1\rangle}{\sqrt{2}}\right)\otimes\cdots\left(\frac{|0\rangle+e^{2\pi i(0.y)}|1\rangle}{\sqrt{2}}\right)$$

Define the following rotation gates:

$$R_k = \begin{pmatrix} 1 & 0 \\ 1 & e^{\frac{2\pi i}{2^k}} \end{pmatrix}$$

and its controlled version $CR_k = I\otimes R_k$, that takes in an extra bit and does the rotation only if that bit was 1. Our circuit will use the hadamard gates and these controlled rotation gates. For sake of notation $CR_k(x,y)$ will apply the rotation on $x$ with $y$ as the control bit.

Let us look at $\left(\frac{|0\rangle+e^{2\pi i(0.y_my_{m+1}\cdots y_{L-1})}|1\rangle}{\sqrt{2}}\right)$. Applying a hadamard transform on $|y_m\rangle$ would give us $\left(\frac{|0\rangle+e^{2\pi i(0.y_m)}}{\sqrt{2}}\right)$. Now there is a smaller rotation created by $y_{m+1}$ only if it is equal to one. But this just amounts to rotating the present state by $2^{-(m+1)}$ controlled by $y_{m+1}$. And so on.
Writing it as an algorithm:

1: **for** $i = 0$ to $L-1$ **do**
2: $\quad y_i = H_1(y_i)$
3: $\quad j = 2$
4: $\quad$ **while** $(i+j-1 \le L-1)$ **do**

5: $(y_i, y_{i+j-1}) = CR_j(y_i, y_{i+j-1})$
6: **end while**
7: **end for**

(picture here would be much better)