Shor's algorithm for factoring [94]

Problem: Given a number N (not prime). Find $1 \leq q < N$ such that $q | N$.

There's a reduction from this problem to Period finding over $\mathbb{Z}$.

Problem: Given $f: \mathbb{Z} \to \mathbb{Z}$ and integer N, with the promise that there exists an $a < N$ such that $\forall x, y \quad f(x) = f(y) \iff y \in \{x, x \pm a, x \pm 2a, \dots\}$

[Note the resemblance to the problem solved by Simon's algorithm]

[The solution is also similar, but it uses a so-called Quantum Fourier Transform over $\mathbb{Z}_m$ - QFT]

Quantum Fourier Transform - QFT

Discrete FT over $\mathbb{Z}_M$: $\mathbb{Z}_M = \{0, 1, \dots, M-1\}$

For each function $f: \mathbb{Z}_M \to \mathbb{C}$ define the FT of f:

$$\hat{f}(y) \equiv \frac{1}{\sqrt{M}} \sum_{x=0}^{M-1} e^{\frac{2\pi i x y}{M}} f(x)$$

The inverse Fourier transform $\hat{f} \to f$:

$$f(x) = \frac{1}{\sqrt{M}} \sum_{y=0}^{M-1} e^{-\frac{2\pi i x y}{M}} \hat{f}(y).$$

For $\omega = e^{\frac{2\pi i}{M}}$ — M^{th} root of 1, we get a simpler description:

$$\hat{f}(y) = \frac{1}{\sqrt{M}} \sum_{x=0}^{M-1} \omega^{xy} f(x) \qquad f(x) = \frac{1}{\sqrt{M}} \sum_{x=0}^{M-1} \omega^{-xy} \hat{f}(y)$$

In matrix notation :

for $f: \mathbb{Z}_M \to \mathbb{C}$ we define a vector $(f) = \begin{pmatrix} f(0) \\ \vdots \\ f(M-1) \end{pmatrix}$

Then : $$\left(\hat{f}\right) = \underbrace{\frac{1}{\sqrt{M}} \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \omega^{M-1} \\ \vdots & \omega^2 & \omega^4 & \vdots \\ 1 & \omega^{M-1} & \dots & \omega^{(M-1)(M-1)} \end{pmatrix}} (f)$$

$$\left(\frac{1}{\sqrt{M}} a_{xy}\right)_{x,y=0}^{M-1} = U_{FT} \quad (a_{xy})_{x,y=0}^{M-1}, \quad a_{xy} = \omega^{xy}$$

U_{FT} is unitary - it's easy to check by verifying that $U_{FT}^{\dagger} U_{FT} = \mathbb{1}$

Quantum : (using Dirac's notation) $U_{FT} |x\rangle = \frac{1}{\sqrt{M}} \sum_{y=0}^{M-1} \omega^{xy} |y\rangle$

Recall that: $H^{\otimes n} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} (-1)^{x \cdot y} |y\rangle. \Rightarrow H^{\otimes n}$ is QFT over $\mathbb{Z}_2^n$.

We now need to show that we can implement U_{FT} on a quantum computer, using only a "small" number of simple 2-3-qubit gates.

Fast FT algorithm (classical) [used for fast number multiplication]

Allows computing the FT in $O(M\log M)$, instead of naive $O(M^2)$ operation.

Assume $M=2^n$.

Write: $k = k_{n-1}\cdot 2^{n-1} + k_{n-2}\cdot 2^{n-2} + \ldots + k_0$; $L = L_{n-1}\cdot 2^{n-1} + L_{n-2}\cdot 2^{n-2} + \ldots + L_0$

Our goal is to compute for all k: $\sum_L e^{2\pi i kL/M} \cdot x_L =$

$$= \sum_{L_0,\ldots,L_{n-1}} \underset{\substack{\downarrow \\ \text{binary number} \\ k_0/2}}{e^{2\pi i \cdot L_{n-1}\cdot 0.k_0}} \cdot e^{2\pi i L_{n-2}\cdot 0.k_1k_0} \cdots e^{2\pi i \cdot L_0 \cdot (0.k_{n-1}k_{n-2}\ldots k_0)} \cdot x_L =$$

$$= \sum_{L_0} e^{2\pi i L_0 (0.k_{n-1}\ldots k_0)} \sum_{L_1} e^{2\pi i L_1\cdot(0.k_{n-2}\ldots k_0)} \cdots \underbrace{\sum_{L_{n-1}} e^{2\pi i L_{n-1}\cdot(0.k_0)} \cdot x_L}_{\substack{\text{Compute for all } k_0\in\{0,1\} \\ \text{and } L_0,\ldots,L_{n-2}\in\{0,1\}. \\ \text{Total } M \text{ operations.}}}$$

Now compute $\sum_{L_{n-2}} e^{2\pi i L_{n-2}\cdot(0.k_0k_1)} \cdot \sum_{L_{n-1}} \cdot e^{2\pi i \cdot L_{n-1}\cdot(0.k_0)}$ over all $k_0, k_1, L_0, \ldots L_{n-3} \in \{0,1\}$, total of M operations. Continue computing larger parts, a total of $\underset{n}{O(M\log M)}$ operations.

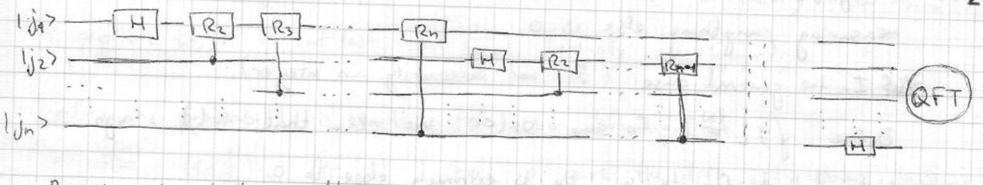
Remark: Finding QFT is possible in $O((\log N)^2)$ gates.

In quantum notation, for $M=2^n$, the FT is: (where $|j\rangle$ is a basis state)

$$|j\rangle = |j_1\ldots j_n\rangle \longmapsto \frac{1}{2^{n/2}} (|0\rangle + e^{2\pi i (0.j_n)}|1\rangle)(|0\rangle + e^{2\pi i (0.j_{n-1}j_n)}|1\rangle)\cdot\ldots$$
$$\ldots\cdot(|0\rangle + e^{2\pi i (0.j_1j_2\ldots j_n)}|1\rangle)$$

Also, QFT is linear, so we can compute it for all $|x\rangle$.

Using this description, we can construct a quantum circuit. Define $R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{pmatrix}$



Remark: We get the result in reversed order.

Finding a period (over the integers)

We are given a function f from $\mathbb{Z}$ to $\{0,1\}^k$.

More precisely, f is on $\{0,1,\dots,2^n-1\}$.

We know that f has some period r, i.e. $\forall x,y \;\; f(x)=f(y) \iff y = l\cdot r + x$ for some $l \in \mathbb{Z}$. We assume that $r < M = 2^m$.

Classically, it is possible to find r with an $O(\sqrt{r})$ probabilistic algorithm.

Quantumly, we will se an algorithm in time poly (n).

First, choose n to be large enough integer, and $N = 2^n$.

Create the state $\frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle |f(x)\rangle$ (see Simon's algorithm), and measure the 2^{nd} register. We obtain $f(x_0)$ for some $0 \le x_0 < r$, and state collapses to $\frac{1}{\sqrt{A}} \sum_{j=0}^{A-1} |x_0 + jr\rangle$, where $A = \left\lceil \frac{N - x_0}{r} \right\rceil$

We now apply the QFT on the state and obtain:

$$\frac{1}{\sqrt{NA}} \cdot \sum_{y=0}^{N-1} \sum_{j=0}^{A-1} e^{2\pi i (x_0 + jr) y / N} |y\rangle$$

We measure, and the probability of getting y is:

$$\Pr(y) = \frac{1}{NA} \cdot \left| \sum_{j=0}^{A-1} e^{2\pi i \cdot (x_0 + jr) \cdot y / N} \right|^2 = \frac{1}{NA} \cdot \left| \sum_{j=0}^{A-1} e^{2\pi i j r y / N} \right|^2$$

If $\frac{N}{r}$ happens to be an integer then $A = \frac{N}{r}$. Define $\theta = \frac{2\pi y r}{N} \pmod{2\pi}$

So: $\Pr(y) = \frac{1}{NA} \cdot \left| \sum_{j=0}^{A-1} e^{i\theta j} \right|^2$

If $\frac{N}{r}$ is an integer, then for $y = 0, \frac{N}{r}, \frac{2N}{r}, \dots, \frac{(r-1)N}{r}$ the probability of seeing y is $\frac{A}{N}$. Those probabilities sum to 1, so probability of measuring anything else is 0.

In the general case ($\frac{N}{r}$ not necessarily an integer).

Define $y = \left\lfloor \frac{lN}{r} \right\rceil$ fo some $0 \le l < r$. We claim that $\Pr(y)$ is high.

Notice that for this y, θ_y is extremely close to 0.

$$\left| \sum_{j=0}^{A-1} e^{i\theta_y j} \right| \ge \left| \sum_{j=0}^{A-2} e^{i\theta_y j} \right| - 1 = \left| \frac{e^{i(A-1)\theta} - 1}{e^{i\theta} - 1} \right| - 1 \ge \frac{2 \cdot (A-1) \cdot |\theta|}{\pi |\theta|} - 1$$

$$|1 - e^{i\theta}| \le \theta$$

$$|1 - e^{i(A-1)\theta}| \ge \frac{2}{\pi}(A-1)\theta$$

So $\Pr(y) \ge \frac{4}{\pi^2} \cdot \frac{1}{r}$

Hence, with probability $\geq \frac{4}{\pi^2}$ we obtain a value $\lfloor \frac{kN}{r} \rceil$ for some αker. Dividing by N, we get an appoximation of $\frac{k}{r}$ to within $\pm \frac{1}{2N}$.
As long as $M \leq \sqrt{N}$ there is unique answer, because the distance between any 2 fractions of denominator $\leq M$ is at least $\frac{1}{M^2}$.
We can find $\frac{k}{r}$ efficiently using continued fractions.

Lemma: Suppose $\frac{s}{r}$ is a rational number and φ is such that: $\left|\frac{s}{r} - \varphi\right| \leq \frac{1}{2r^2}$.
Then we can compute $\frac{s}{r}$ from φ efficiently.

Proof sketch: Recall that is $\frac{s}{r} = [a_0, a_1, \ldots, a_n]$ – (continued fraction expansion) then $\varphi = [a_0, a_1, \ldots, a_n, \lambda]$ $(\lambda \in \mathbb{R}, \lambda > 1)$. So we need to expand the continued fraction of φ, until the denominator is too large.

—

So we are able to obtain $\frac{k}{r}$ for some random $k \in \{0, \ldots, r-1\}$.
We can repeat until k is coprime to r – then we just see r at the denominator. The probality of that is $\frac{\varphi(r)}{r}$ where it is known that:
$\liminf_{r \to \infty} \frac{\varphi(r)}{r} = \Omega\left(\frac{1}{\log \log r}\right)$.
A better solution is to take 2 samples $\frac{k_1}{r}, \frac{k_2}{r}$, and output the lcm of the denominators. This works if $\gcd(k_1, k_2) = 1$, which happens with probability $1 - \Pr(2|k_1, 2|k_2) - \Pr(3|k_1, 3|k_2) - \ldots = 1 - \sum_{p \text{ prime}} \frac{1}{p^2} \geq 1 - \sum_{n \geq 2} \frac{1}{n^2} \geq \frac{1}{4}$

—

Claim: $\forall r > 0$ the probability that a number k chosen uniformly from $\{0, \ldots, r-1\}$ is coprime to r is at last $\Omega\left(\frac{1}{\log r}\right)$ (actually $\Omega\left(\frac{1}{\log \log r}\right)$).

Proof: The probability is exactly $\frac{\varphi(r)}{r}$ (by definition of φ).
$\frac{\varphi(r)}{r} = \left(1 - \frac{1}{p_1}\right) \cdot \ldots \cdot \left(1 - \frac{1}{p_\ell}\right)$, where p_i are the different prime divisors of r. $\left(1 - \frac{1}{p_1}\right) \cdot \ldots \cdot \left(1 - \frac{1}{p_\ell}\right) \geq \left(1 - \frac{1}{2}\right) \cdot \left(1 - \frac{1}{3}\right) \cdot \ldots \cdot \left(1 - \frac{1}{\ell+1}\right) = \frac{1}{\ell+1} \geq \Omega\left(\frac{1}{\log r}\right)$ $\square$