CS 294 2/11/09 Spring 2009

Fourier Transform, Period Finding and Factoring in BQP

Lecture 4

1 Recap: Simon's Algorithm

Recall that in the Simon's problem, we are given a function $f : \mathbf{Z}_2^n \to \mathbf{Z}_2^n$ (i.e. from n-bit strings to n-bit strings), with the promise that there is a non-zero string $s \in \mathbf{Z}_2^n \setminus \{0\}$ such that

$$\text{for all } x \neq y,\ f(x) = f(y) \text{ if and only if } x \oplus y = s.$$

The challenge is to determine s. As we saw last time, the problem can be solved with the following circuit.

[thin,scale=1.5] [] (0.25,-0.25) – ++(5.5,0); [] (0.25,-0.5) – ++(5.5,0); [] (0.25,-0.75) – ++(5.5,0); [] (0.25,-1.25) – ++(5.5,0); [] (0.25,-1.5) – ++(5.5,0); [] (5.75,0) – ++(0,-1) arc (-90:90:18pt and 14pt) ++(0.1,-0.25) arc (90:0:10pt) ++(-0.4,0) [-¿] – ++(0.4,0.4); [left] at (0,-0.5) $\big|0\big\rangle$; [left] at (0,-1.375) $\big|0\big\rangle$; [right] at (6,-1.375) $\big|f(x)\big\rangle$; [rectangle,draw,fill=white,minimum height=40pt,minimum width=40pt,below] at (1.5,0) H; [rectangle,draw,fill=white,minimum height=70pt,minimum width=40pt,below] at (3,0) U_f; [rectangle,draw,fill=white,minimum height=40pt,minimum width=40pt,below] at (4.5,0) H;

Figure 1: Circuit for Simon's Algorithm

The above circuit corresponds to the following sequence of transformations.

$$\begin{aligned}
\big|0\big\rangle\big|0\big\rangle &\xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{2^n}}\sum_x \big|x\big\rangle\big|0\big\rangle \\
&\xrightarrow{f} \frac{1}{\sqrt{2^n}}\sum_x \big|x\big\rangle\big|f(x)\big\rangle \\
&\xrightarrow{\text{measure}} \frac{1}{\sqrt{2}}(\big|x_0\big\rangle + \big|x_0 \oplus s\big\rangle) \otimes \big|a\big\rangle \\
&\qquad \text{(measuring the 2nd register, we observe } a \in \mathbf{Z}_2^n \text{ such that } a = f(x_0) = f(x_0 \oplus s)) \\
&\xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{2^n}}\sum_y \alpha_y \big|y\big\rangle\big|a\big\rangle
\end{aligned}$$

for some numbers α_y.

Recall that the Hadamard transform of a general state $\big|x\big\rangle$ is

$$H^{\otimes n}\big|x\big\rangle = \frac{1}{\sqrt{2^n}}\sum_y (-1)^{x \cdot y}\big|y\big\rangle,$$

so

$$\alpha_y = \frac{1}{\sqrt{2}}((-1)^{x_0 \cdot y} + (-1)^{(x_0 \oplus s)\cdot y}).$$

There are now two cases. For each y, if $s \cdot y = 1$, then $\alpha_y = 0$, whereas if $s \cdot y = 0$, then $\alpha_y = (-1)^{x_0 \cdot y}\sqrt{2}$.

When we observe the first register, we get a uniformly random y such that $s \cdot y = s_1 y_1 + \cdots + s_n y_n = 0$. We repeat to collect more and more equations, and recover s from n linearly independent equations.

There is another way to view the final Hadamard transform. When we have $|x_0\rangle + |x_0 \oplus s\rangle$, measuring immediately would destroy the state. That's why we transform it to another basis (the Hadamard basis) before measuring.

2 Fourier Transform on $\mathbf{Z}_M$ for Integer M

Let $f : \mathbf{Z}_M \to \mathscr{C}$ be a complex-valued function on $\mathbf{Z}_M$. Its Fourier transform $\hat{f} : \mathbf{Z}_M \to \mathscr{C}$ is given by

$$\hat{f}(t) = \frac{1}{\sqrt{M}} \sum_{x \in \mathbf{Z}_M} f(x) \omega^{xt}$$

where $\omega = \exp(2\pi i/M)$ is a primitive M-th root of unity. If we write f as the vector

$$\vec{f} = \begin{pmatrix} f(0) \\ f(1) \\ \vdots \\ f(M-1) \end{pmatrix} \in \mathscr{C}^M,$$

and similarly write $\hat{f}$ as $\vec{\hat{f}} \in \mathscr{C}^M$, then the vectors $\vec{f}$ and $\vec{\hat{f}}$ are related by a change of basis $\vec{\hat{f}} = F_M \vec{f}$, where the matrix F_M takes the form

$$F_M = \frac{1}{\sqrt{M}} \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{M-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2M-2} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \cdots & \omega^{3M-3} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{M-1} & \omega^{2M-2} & \omega^{3M-3} & \cdots & \omega^{(M-1)(M-1)} \end{pmatrix},$$

that is, (i,j)-th entry of F_M is ω^{ij} (if we ignore the normalization factor $1/\sqrt{M}$).

3 Classical Fast Fourier Transform

Straightforward multiplication of the vector $\vec{f}$ by F_M would take $\Omega(M^2)$ steps because multiplication of $\vec{f}$ by each row requires M multiplications. Exploiting the symmetry of F_M, it is possible to perform Fourier transform in $O(M \log M)$ operations when M is a power of two, i.e. $M = 2^m$. This algorithm is known as fast Fourier transform (FFT).

The idea is to rewrite the Fourier coefficients $\hat{f}(j)$ as

$$\begin{aligned}
\hat{f}(j) &= \sum_{i=0}^{M-1} \omega^{ij} f(i) \qquad \text{(where for simplicity we ignore the normalization factor } 1/\sqrt{M}\text{)} \\
&= \sum_{i \text{ even}} \omega^{ij} f(i) + \sum_{i \text{ odd}} \omega^{ij} f(i) \qquad \text{(splitting into odd and even terms)} \\
&= \sum_{i'=0}^{M/2-1} (\omega^{2i'})^j f(2i') + \omega^j \sum_{i'=0}^{M/2-1} (\omega^{2i'})^j f(2i'+1) \qquad \text{(write even } i \text{ as } 2i'\text{, odd as } 2i'+1\text{)} \\
&= \left(F_{M/2} \overrightarrow{f_{\text{even}}}\right)(j) + \omega^j \left(F_{M/2} \overrightarrow{f_{\text{odd}}}\right)(j).
\end{aligned}$$

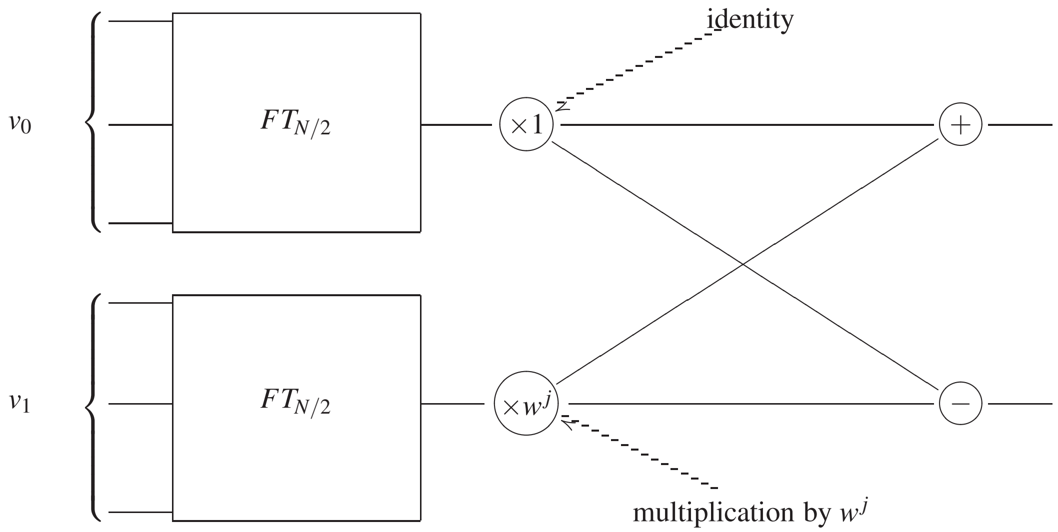

Figure 2: A circuit for classical fast Fourier transform

The above idea is summarized in the diagram below.

This representation gives a recursive algorithm for computing the Fourier transform in time $T(M) = 2T(M/2) + O(M) = O(M \log M)$.

4 Quantum Fourier Transform

We continue to assume $M = 2^m$. Suppose a quantum state $|f\rangle$ on m qubits is given as $|f\rangle = \sum_{x=0}^{M-1} f(x)|x\rangle$. Quantum Fourier transform (QFT) is the operation that maps $|f\rangle$ to $|\hat{f}\rangle$, where $|\hat{f}\rangle = \sum_{x=0}^{M-1} \hat{f}(x)|x\rangle$ (and $\hat{f}(x)$ are the Fourier coefficients of f).

As we shall see, QFT can be implemented by circuit of size $O(\log^2 M)$. However, this does not constitute an exponential speed-up over the classical algorithm because the result of quantum Fourier transform is a superposition of states which can be observed, and any measurement can extract at most $m = \log M$ bits of information.

We now describe a circuit that implements quantum Fourier transform.

Step 1: $QFT_{M/2}$ on the first $m-1$ qubits

Similar to the classical fast Fourier transform, we will split $\sum_x f(x)|x\rangle$ into odd and even terms. Hence

$$\sum_x f(x)|x\rangle = \sum_{i=0}^{M/2-1} f(2i)|i\rangle|0\rangle + \sum_{i=0}^{M/2-1} f(2i+1)|i\rangle|1\rangle.$$

where $|2i\rangle$ is written as $|i\rangle|0\rangle$ because appending a zero to a binary number is the same as doubling the number (e.g. if i is 101100 in binary, then $2i$ is 1011000). In a quantum circuit for uantum Fourier transform, we will first apply $QFT_{M/2}$ on the first register (i.e. the first $m-1$ qubits), obtaining

$$\sum_{i=0}^{M/2-1} \alpha_i|i\rangle|0\rangle + \sum_{i=0}^{M/2-1} \beta_i|i\rangle|1\rangle$$

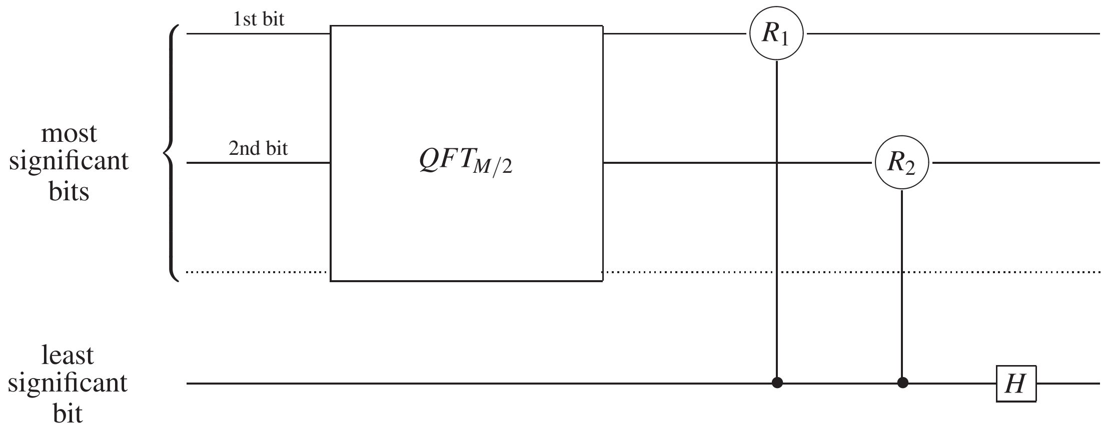

Figure 3: Circuit for quantum Fourier transform

for certain amplitudes α_i and β_i.

Step 2: Controlled phase shifts

Next, for each of the first m-1 qubits k $(1 \leq k \leq m)$, if both the k-th qubit and the last qubit are 1, then we need to multiply the phase by $\omega^{2^{m-k}}$, and otherwise leave the phase unchanged. Thus, we apply the following transformations R_k:

$$\begin{aligned}
R_k\big|\overbrace{1}^{k\text{-th qubit}}\;\overbrace{1}^{\text{last qubit}}\big\rangle &= \omega^{2^{m-k}}\big|11\big\rangle \\
R_k\big|01\big\rangle &= \big|01\big\rangle \\
R_k\big|10\big\rangle &= \big|10\big\rangle \\
R_k\big|00\big\rangle &= \big|00\big\rangle
\end{aligned}$$

Hence, R_k is just a controlled phase shift (with angle $2\pi/2^k$). After the controlled phase shift, we get the state

$$\sum_{i=0}^{M/2-1} \alpha_i\big|i\big\rangle\,\big|0\big\rangle + \sum_{i=0}^{M/2-1} \omega^i\beta_i\big|i\big\rangle\,\big|1\big\rangle\,.$$

Step 3: Hadamard gate

Finally, we apply a Hadamard gate to the last qubit, and end up with the state

$$\begin{aligned}
&\frac{1}{\sqrt{2}}\left[\sum_{i=0}^{M/2-1} \alpha_i\big|i\big\rangle\,(\big|0\big\rangle+\big|1\big\rangle\,)+\sum_{i=0}^{M/2-1} \omega^i\beta_i\big|i\big\rangle\,(\big|0\big\rangle-\big|1\big\rangle\,)\right] \\
&= \frac{1}{\sqrt{2}}\left[\sum_i(\alpha_i+\omega^i\beta_i)\underbrace{\big|i\big\rangle\,\big|0\big\rangle}_{i\text{th output}} + \sum_i(\alpha_i-\omega^i\beta_i)\underbrace{\big|i\big\rangle\,\big|1\big\rangle}_{(i+M/2)\text{th output}}\right].
\end{aligned}$$

Putting together, in the circuity above the quantum Fourier transform on $m-1$ qubits corresponds to two Fourier transforms on $m-1$ bits in the figure **??**. The controlled phase shifts correspond to multiplications by ω^j in classical circuit. Finally, the Hadamard gate at the very end corresponds to the summation.

The number of gates $T(M)$ satisfies the recurrence relation $T(M) = T(M/2) + \log M$. Thus $T(M) = O(\log^2 M)$.

5 Period Finding

Period finding is the problem in which we are given a function $f : \mathbf{Z}_M \to \mathscr{C}$, with the promise that f is periodic with period r, i.e.

> there is a r such that for all $x \neq y$, $f(x) = f(y)$ if and only if $x \equiv y \bmod r$.

The challenge is the find the period r.

This problem can be solved efficiently using the following circuit.

[thin,scale=1.5] [] (0.25,-0.25) – ++(5.5,0); [] (0.25,-0.5) – ++(5.5,0); [] (0.25,-0.75) – ++(5.5,0); [] (0.25,-1.25) – ++(5.5,0); [] (0.25,-1.5) – ++(5.5,0); [] (5.75,0) – ++(0,-1) arc (-90:90:18pt and 14pt) ++(0.1,-0.25) arc (90:0:10pt) ++(-0.4,0) [-¿] – ++(0.4,0.4); [left] at (0,-0.5) $|0\rangle$; [left] at (0,-1.375) $|0\rangle$; [right] at (6,-1.375) $|f(x)\rangle$; [rectangle,draw,fill=white,minimum height=40pt,minimum width=40pt,below] at (1.5,0) QFT_M; [rectangle,draw,fill=white,minimum height=70pt,minimum width=40pt,below] at (3,0) U_f; [rectangle,draw,fill=white,minimum height=40pt,minimum width=40pt,below] at (4.5,0) QFT_M;

Figure 4: Circuit for period finding

The above circuit corresponds to the following sequence of transformations.

$$|0\rangle|0\rangle \xrightarrow{QFT_M} \frac{1}{\sqrt{M}} \sum_{x \in \mathbf{Z}_M} |x\rangle|0\rangle$$

$$\xrightarrow{f} \frac{1}{\sqrt{M}} \sum_x |x\rangle|f(x)\rangle$$

$$\xrightarrow{\text{measure 2nd register}} \frac{1}{\sqrt{M/r}} \sum_{k=0}^{M/r-1} |\ell + kr\rangle|f(\ell)\rangle$$

(Here we assume r divides M to simplify the analysis. We will remove this restriction later.)

$$\xrightarrow{QFT_M} \sqrt{\frac{r}{M}} \frac{1}{\sqrt{M}} \sum_y \alpha_y |y\rangle,$$

where $\alpha_y = \sum_{k=0}^{M/r-1} \omega^{(\ell+kn)y} = \omega^{\ell y} \sum_k \omega^{kry}$.

There are two cases for y:

1. Case 1: y is a multiple of $\frac{M}{r}$.

   In this case, then $\omega^{kry} = e^{2\pi i r y/M} = 1$. So $\alpha_y = \frac{\sqrt{r}}{M} \frac{M}{r} = \frac{1}{\sqrt{r}}$.

   Note that there are r multiples of M/r. The sum of the magnitudes squared for these values of y is 1. This implies that for any other y, $\alpha_y = 0$.

2. Case 2: y is not a multiple of $\frac{M}{r}$.

   We already showed that α_y must be 0 from the previous case. But we can also give an intuition for why this is the case. Note that $\omega^{ry}, \omega^{2ry}, \ldots$ are evenly spaced vectors of unit length around the origin. Being the sum of these complex numbers, α_y is 0.

In other words, if we measure the output from the second quantum Fourier transform, we get a uniformly random multiple of M/r. If we repeat the whole process t times, getting t random multiples $y_1,\ldots,y_t$ of M/r, the greatest common divisor of the y_j's is likely to be M/r. Since we know M, we can recover r from $M/\gcd(y_1,\ldots,y_t)$.

Let us compute the chance of finding the correct period with t samples. Suppose after repeating t times, we have not found the desired period M/r, but instead a multiple, say $\lambda M/r$. This means that each of the t samples must be a multiple of $\lambda M/r$. There are $M/(\lambda M/r) = r/\lambda$ multiples of $\lambda M/r$, and since there are r multiples in total, the probability of getting a multiple of $\lambda M/r$ is $1/\lambda$. Therefore,

$$\Pr[\text{gcd is a multiple of } \lambda M/r] = \left(\frac{1}{\lambda}\right)^t \le \left(\frac{1}{2}\right)^t,$$

and we err with probability

$$\Pr[\text{gcd} > M/r \text{ after } t \text{ samples}] \le M\left(\frac{1}{2}\right)^t.$$

So $t = O(\log M)$ measurements suffice to guarantee a solution. A more careful analysis shows that a constant number of samples is sufficient.

6 Period Finding: The General Case

For the general case where M is not a multiple of r, we will fix $M = 2^m$ to be a power of two that is at least r^2.

The change to the above analysis is that, after measuring the 2nd register, we get

$$\xrightarrow{\text{measure 2nd register}} \frac{1}{\sqrt{s}}\sum_{k=0}^{s-1}\big|\ell+kr\big\rangle\big|f(\ell)\big\rangle$$

where $s = \lfloor M/r\rfloor$ or $s = \lfloor M/r\rfloor + 1$. If we now take QFT on the first register, we get $\sum_y \alpha_y\big|y\big\rangle$ with

$$\alpha_y = \frac{1}{\sqrt{sM}}\omega^{\ell y}\sum_{k=0}^{s-1}\omega^{kry}.$$

Case 1 now becomes:

1. Case 1: $|ry \bmod M| \le \frac{r}{2}$ (in this section the remainder mod M is allowed to be negative):

   Intuitively, in this case the amplitudes ω^{kry} “almost line up” in the complex plane. Previously, when the period r divides M exactly, all the amplitudes for multiples of M/r “line up” at 1.

   Claim: If $|ry \bmod M| \le \frac{r}{2}$, then $|\alpha_y| \ge Cs$ for some constant C.

   This claim implies that we have substantial probability of observing a y that falls into case 1. How many y belong to this case? When r is coprime to M (which is the usual case when we run period finding as a subroutine in factoring), the set $\{ry \mid y\in\mathbf{Z}_M\}$ is just $\mathbf{Z}_M$. Put differently, as y runs through $\mathbf{Z}_M$, the product ry also runs through $\mathbf{Z}_M$. Hence there are about $2\frac{r}{2} = r$ such y. Then

$$\Pr[\text{Observing such a } y] \ge r\cdot C^2\cdot s^2\frac{1}{Ms} \ge C^2\cdot s\cdot\frac{r}{M} \ge \text{const}.$$

For the rest of the discussion, assume that we measure an y satisfying $|ry \bmod M| \leq r/2$. How does this help us compute the period r? By assumption $|ry - cM| \leq r/2$ for some integer c, and hence

$$\left|\frac{y}{M} - \frac{c}{r}\right| \leq \frac{1}{2M}.$$

Here y and M are both known. We shall show, assuming that $M > r^2$, how to recover c/r by continued fraction.

Here is the idea: c/r is a close approximation to y/M. Is it possible to get a better rational approximation with denominator at most r? We will show it is impossible. Suppose c'/r' is a better rational approximation with denominator $r' \leq r$. Then

$$\left|\frac{c}{r} - \frac{c'}{r'}\right| = \left|\frac{cr' - c'r}{rr'}\right| \geq \frac{1}{r^2}$$

But now it follows that

$$\left|\frac{c'}{r'} - \frac{y}{M}\right| \geq \frac{1}{r^2} - \frac{1}{2M} > \frac{1}{2M}.$$

So if we compute the continued fraction expansion of y/M and look at the successive approximations to y/M, one of these must be c/r, thus yielding r. (See the section on continued fractions below.)

7 Continued Fraction

Definition 4.1 (Continued Fraction): A real number α can be approximated by an iterated fraction

$$CF_n(\alpha) = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots + \cfrac{1}{a_n}}}} = \frac{P_n}{Q_n},$$

where $a_0, \ldots, a_n$ (and hence P_n and Q_n) are integers.

2 **Example 1**: Let us approximate π to two decimal places with a rational number. We know that

$$\begin{aligned}\pi &= 3.14\ldots \\ &= 3+\frac{14}{100} \\ &= 3+\frac{1}{\frac{100}{14}} \\ &= 3+\frac{1}{7+\frac{2}{14}} \\ &\approx 3+\frac{1}{7} \\ &= \frac{22}{7}\end{aligned}$$

Example 2: If we decide to approximate π to four decimal places, we would have

$$\begin{aligned}\pi &= 3.1415\ldots \\ &= 3+\frac{1415}{10000} \\ &= 3+\frac{1}{\frac{10000}{1415}} \\ &= 3+\frac{1}{7+\frac{95}{1415}} \\ &= 3+\frac{1}{7+\frac{1}{\frac{1415}{95}}} \\ &= 3+\frac{1}{7+\frac{1}{14+\frac{85}{95}}} \\ &\approx 3+\frac{1}{7+\frac{1}{14}} \\ &= \frac{311}{99}\end{aligned}$$

The following lemmas are well known facts about continued fraction that we state without proof.

Lemma 4.1: *$CF_n(\alpha)$ is the best rational approximation of α with denominator $\leq Q_n$.*

Lemma 4.2: *If α is rational then it occurs as one of the approximations $CF_n(\alpha)$.*

Moreover, it is easy to see that the continued fraction is easy to compute for any rational number.

8 Shor's Quantum Factoring Algorithm

Below we give a quantum algorithm that factors N in $\text{polylog}(N)$ time (factoring in $\text{poly}(N)$ time is trivial and is too slow). It turns out the problem of factoring reduces to finding a nontrivial square root.

Claim: If we can find u such that $u^2 \equiv 1 \pmod N$ and $u \not\equiv \pm 1 \pmod N$, then we can factor N. (Such a number u is called a nontrivial square root of $1 \pmod N$.)

Proof: The condition on u is equivalent to $N \mid u^2 - 1 = (u+1)(u-1)$ but $N \nmid u+1$ or $N \nmid u-1$. So $\gcd(N, u+1)$ and $\gcd(N, u-1)$ are nontrivial factors of N. □

To find a non-trivial square root, we simply pick a random number $x \pmod N$ and compute its order, where the order of x is the least positive r such that $x^r \equiv 1 \pmod N$.

Claim: If N is odd, with probability at least $1/2$ over a random $x \in \mathbf{Z}_N$, the order r of x is even and $x^{r/2} \not\equiv \pm 1 \pmod N$.

Example: Let $N = 15$. Then let's suppose we picked $x = 7$. Then $x = 7, x^2 = 4, x^3 = 13, x^4 = 1$, so x has order 4. Now, taking $y = x^{r/2} = 4$, notice that $y - 1 = 3$ and $y + 1 = 5$ are both factors of 15.

To compute the order, we can use a quantum circuit to compute $f(a) = x^a \pmod N$. The function maps element from $\mathbf{Z}_M$ to $\mathbf{Z}_M$ with $M > N^2$. This function can be implemented efficiently with $\tilde{O}(\log^2 N)$ gates if we do modular exponentiation with repeated squaring.

Below we give the algorithm that, given an odd integer N, outputs a nontrivial factor of N with constant probability.

[1] Pick a random $x \in Z_N$ $\gcd(x, N) > 1$ Output $\gcd(x, N)$ Run period finding on $f(a) = x^a \pmod N$ to get the order r of $x \pmod N$ Compute $x^{r/2} \pmod N$ $x^{r/2} \not\equiv \pm 1 \pmod N$ Output $\gcd(N, x^{r/2} \pm 1)$ Abort

Note that GCD can be computed quickly using Euclid's algorithm.

Because of randomized nature of part of the algorithm, we may need to repeat this procedure many times. Since the algorithm succeeds with constant probability, a constant number of repitiions suffice to split N into two non-trivial factors. We can then repeat the procedure on each factor until we are down to the prime factors.