

Shor's algorithm

A very short introduction

Jinghong Yang

June 3, 2021

Problem to Solve

$N = pq$, where p, q are prime.

- ▶ You know N
- ▶ Want to find p and q .

Table of Contents

Preliminary

Quantum Fourier Transform

Shor's algorithm

Table of Contents

Preliminary

Quantum Fourier Transform

Shor's algorithm

How to factor N

Suppose we somehow find a number M such that

$$\gcd(M, N) \neq 1$$

$$\gcd(M, N) \neq N$$

Then $\gcd(M, N)$ is just a non-trivial factor of N .

We can find $\gcd(M, N)$ using Euclid's algorithm.

Review: Congruence

$$a \equiv b \pmod{N}$$

Basic properties:

If $a \equiv b \pmod{N}$ and $c \equiv d \pmod{N}$, then

► $a \pm c \equiv b \pm d \pmod{N}$

► $ac \equiv bd \pmod{N}$

A bit of number theory

Find order r given a , N :

$$a^r \equiv 1 \pmod{N}$$

Define function $f_{a,N}$:

$$f_{a,N}(j) := a^j \pmod{N}$$

Clearly, we have

$$\begin{aligned} f_{a,N}(j+r) &= a^{j+r} \pmod{N} \\ &= (a^j \pmod{N}) * (a^r \pmod{N}) \\ &= (a^j \pmod{N}) * 1 \\ &= f_{a,N}(j) \end{aligned} \tag{1}$$

A bit of number theory

$$\begin{aligned}a^r &\equiv 1 \pmod{N} \\a^r - 1 &\equiv 0 \pmod{N} \\(a^{r/2} - 1)(a^{r/2} + 1) &\equiv 0 \pmod{N}\end{aligned}\tag{2}$$

Unless $a^{r/2} \equiv -1 \pmod{N}$, we have

$$\gcd(a^{r/2} - 1, N) = p, \quad \gcd(a^{r/2} + 1, N) = q\tag{3}$$

Table of Contents

Preliminary

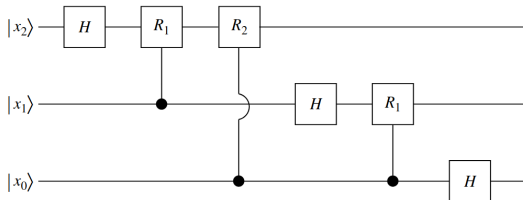
Quantum Fourier Transform

Shor's algorithm

Quantum Fourier Transform

n : the number of qubits. Definition:

$$U_{FT} |x\rangle = \sum_y \frac{1}{2^{n/2}} e^{2i\pi xy/2^n} |y\rangle \quad (4)$$



(b)

Figure 5.9 (a) The box U_{FT} . (b) A circuit constructing U_{FT} in the case $n = 3$.

Figure: Quantum Fourier Transform illustration from Bellac (2006).

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad R_d = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/2^d} \end{pmatrix} \quad (5)$$

Some notations

$$|x\rangle = |x_{n-1} \cdots x_1 x_0\rangle \quad (6)$$

$$\begin{aligned} x &= x_0 + 2x_1 + \cdots + 2^{n-1}x_{n-1} \\ y &= y_0 + 2y_1 + \cdots + 2^{n-1}y_{n-1} \end{aligned} \quad (7)$$

Denote

$$.x_p x_{p-1} \cdots x_1 x_0 = \frac{x_p}{2} + \frac{x_{p-1}}{2^2} + \cdots + \frac{x_0}{2^p} \quad (8)$$

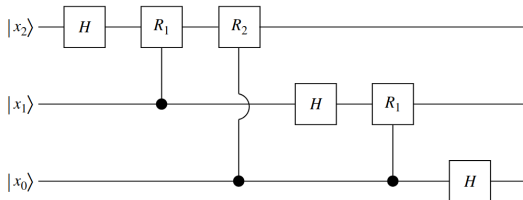
E.g. $n = 3$,

$$\begin{aligned} \exp\left(2\pi i \frac{xy}{8}\right) &= \exp\left(2\pi i \frac{1}{8} (x_0 + 2x_1 + 4x_2) (y_0 + 2y_1 + 4y_2)\right) \\ &= \exp\left(2\pi i \left[y_0 \left(\frac{x_2}{2} + \frac{x_1}{4} + \frac{x_0}{8}\right) + y_1 \left(\frac{x_1}{2} + \frac{x_0}{4}\right) + y_2 \frac{x_0}{2}\right]\right) \\ &= \exp(2\pi i (y_0 \cdot x_2 x_1 x_0 + y_1 \cdot x_1 x_0 + y_2 \cdot x_0)) \end{aligned} \quad (9)$$

$$\begin{aligned}
U_{FT} |x\rangle &= \frac{1}{2^{n/2}} \sum_y e^{2\pi i y_{n-1} \cdot x_0} \dots e^{2\pi i y_0 \cdot x_{n-1} \dots x_1 x_0} |y_{n-1} \dots y_1 y_0\rangle \\
&= \frac{1}{2^{n/2}} \left(\sum_{y_{n-1}} e^{2\pi i y_{n-1} \cdot x_0} \right) \dots \left(\sum_{y_1} e^{2\pi i y_1 \cdot x_{n-2} \dots x_0} \right) \left(\sum_{y_0} e^{2\pi i y_0 \cdot x_{n-1} \dots x_0} \right) \\
&= \frac{1}{2^{n/2}} (|0\rangle + e^{2\pi i \cdot x_0} |1\rangle) \otimes \dots \otimes (|0\rangle + e^{2\pi i \cdot x_{n-2} \dots x_0} |1\rangle) \\
&\quad \otimes (|0\rangle + e^{2\pi i \cdot x_{n-1} \dots x_0} |1\rangle)
\end{aligned} \tag{10}$$

When $n = 3$,

$$\frac{1}{\sqrt{8}} (|0\rangle_2 + e^{2\pi i \cdot x_0} |1\rangle_2) (|0\rangle_1 + e^{2\pi i \cdot x_1 x_0} |1\rangle_1) (|0\rangle_0 + e^{2\pi i \cdot x_2 x_1 x_0} |1\rangle_0) \tag{11}$$



(b)

Figure 5.9 (a) The box U_{FT} . (b) A circuit constructing U_{FT} in the case $n = 3$.

Figure: Quantum Fourier Transform illustration from Bellac (2006).

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad R_d = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/2^d} \end{pmatrix} \quad (12)$$

What do we get from the circuit:

$$\frac{1}{\sqrt{8}} (|0\rangle_2 + e^{2\pi i \cdot x_2 x_1 x_0} |1\rangle_2) (|0\rangle_1 + e^{2\pi i \cdot x_1 x_0} |1\rangle_1) (|0\rangle_0 + e^{2\pi i \cdot x_0} |1\rangle_0) \quad (13)$$

What do we want:

$$\frac{1}{\sqrt{8}} (|0\rangle_2 + e^{2\pi i \cdot x_0} |1\rangle_2) (|0\rangle_1 + e^{2\pi i \cdot x_1 x_0} |1\rangle_1) (|0\rangle_0 + e^{2\pi i \cdot x_2 x_1 x_0} |1\rangle_0) \quad (14)$$

Solution, interpret the output as its bit reversal

$$|y\rangle = |y_0 y_1 \cdots y_{n-1}\rangle \quad (15)$$

instead of

$$|y_{n-1} \cdots y_1 y_0\rangle$$

Table of Contents

Preliminary

Quantum Fourier Transform

Shor's algorithm

Suppose we get a quantum computer, with $n+m$ qubits.
 Want to factor N ($2^n > N^2$). Given random number a .
 Initial state:

$$|\Phi\rangle = \frac{1}{2^{n/2}} \left(\sum_{x=0}^{2^n-1} |x\rangle_n \right) \otimes |0 \cdots 0\rangle_m \quad (16)$$

This step is easy; for the first n qubits, cast them in $\frac{1}{\sqrt{2}} (|0\rangle + |1\rangle)$.
 Next, we transform the state into

$$|\psi\rangle = \frac{1}{2^{n/2}} \left(\sum_{x=0}^{2^n-1} |x\rangle_n \right) \otimes |a^x \pmod{N}\rangle_m \quad (17)$$

Then, measure the output register (last m qubits). Suppose the result is f_0 . The solution to $f_0 = f_{aN}(x) = a^x \bmod N$ will have the form $x = x_0 + kr$.

Thus, after the measurement, the first n qubits are:

$$|\psi_0\rangle = \frac{1}{\sqrt{K}} \sum_{k=0}^{K-1} |x_0 + kr\rangle_n \quad (18)$$

where $K \approx 2^n/r$.

Now, do a Quantum Fourier Transform U_{FT} on $|\psi_0\rangle$.

$$\langle y | U_{FT} | \psi_0 \rangle = \frac{1}{2^{n/2}} \frac{1}{\sqrt{K}} \sum_{k=0}^{K-1} e^{2i\pi y(x_0 + kr)/2^n} \quad (19)$$

After the Quantum Fourier Transform, suppose you do a measurement, and you get y .

We will show later that if there's an integer j such that y is close to $\frac{j2^n}{r}$, you can obtain r .

For now, let's look at the probability of getting each y .

$$\begin{aligned} p(y) &= \frac{1}{2^n K} \left| \sum_{k=0}^{K-1} e^{2i\pi y(x_0 + kr)/2^n} \right|^2 \\ &= \frac{1}{2^n K} \left| \sum_{k=0}^{K-1} e^{2i\pi kry/2^n} \right|^2 \end{aligned} \tag{20}$$

$$\sum_{k=0}^{K-1} e^{2i\pi kry/2^n} = \frac{1 - e^{2i\pi Kry/2^n}}{1 - e^{2i\pi ry/2^n}} = e^{\frac{i\pi r(K-1)}{2^n}} \frac{\sin(\pi yKr/2^n)}{\sin(\pi yr/2^n)} \quad (21)$$

Thus,

$$p(y) = \frac{1}{2^n K} \frac{\sin^2(\pi yKr/2^n)}{\sin^2(\pi yr/2^n)} \quad (22)$$

Suppose $\frac{2^n}{r}$ happens to be an integer, which means it is exactly equal to K . Then

$$p(y) = \frac{1}{2^n K} \frac{\sin^2(\pi y)}{\sin^2(\pi y/K)} = \begin{cases} \frac{1}{r} & \text{if } y = jK \\ 0 & \text{otherwise} \end{cases} \quad (23)$$

This means $y/j = K = 2^n/r$, which means

$$\frac{j}{r} = \frac{y}{2^n} \quad (24)$$

Otherwise, we write

$$y = j \frac{2^n}{r} + \delta_y$$

$$\begin{aligned} P(y) &= \frac{1}{2^n K} \frac{\sin^2 \left(\pi \left(j \frac{2^n}{r} + \delta_y \right) K r / 2^n \right)}{\sin^2 \left(\pi \left(j \frac{2^n}{r} + \delta_y \right) r / 2^n \right)} \\ &= \frac{1}{2^n K} \frac{\sin^2 \left(\pi \delta_y K r / 2^n \right)}{\sin^2 \left(\pi \delta_y r / 2^n \right)} \end{aligned} \quad (25)$$

We know $\frac{2}{\pi}x \leq \sin x \leq x$ if $0 \leq x \leq \frac{\pi}{2}$. Thus, when we have $|\delta_y| \leq \frac{1}{2}$, we will get ¹

$$p(y) \geq \frac{4}{\pi^2} \frac{K}{2^n} \approx \frac{4}{\pi^2} \frac{1}{r} \approx 0.405 \frac{1}{r} \quad (26)$$

This means we have roughly 40% chance of having y close to $j2^n/r$.

¹Recall $K \approx \frac{2^n}{r}$

Now, we have $|y - j\frac{2^n}{r}| \leq \frac{1}{2}$. Thus,

$$\left| \frac{y}{2^n} - \frac{j}{r} \right| \leq \frac{1}{2 * 2^n} \quad (27)$$

So j/r must lie in a region of size $1/2^n$ around $\frac{y}{2^n}$.


Obviously, for fractions, unless $\frac{a}{b} = \frac{c}{d}$, we always have

$$\left| \frac{a}{b} - \frac{c}{d} \right| \geq \frac{1}{bd}.$$

Suppose fractions $\frac{j_1}{r_1}$ and $\frac{j_2}{r_2}$ both lie in this region, and they are not equal, then ²

$$\left| \frac{j_1}{r_1} - \frac{j_2}{r_2} \right| \geq \frac{1}{r_1 r_2} \geq \frac{1}{N^2} \geq \frac{1}{2^n} \quad (28)$$

Thus, the value $\frac{j}{r}$ is unique.

²Recall that $2^n > N^2$; also, by number theory, we have $r \leq N$. 

- M. L. Bellac. *A short introduction to quantum information and quantum computation*. Cambridge University Press, 1 edition, 2006. ISBN 0-521-86056-3, 978-0-521-86056-7, 978-0-511-22009-8, 0-511-22009-X.
- P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5): 1484–1509, Oct 1997. ISSN 1095-7111. doi: 10.1137/S0097539795293172. URL <http://dx.doi.org/10.1137/S0097539795293172>.

Questions