Shor's Algorithm

Feb 14, 2024

amakihc

Contents

1. Introduction **2**

2. Theory **2**

2.1. Elementary Number Theory 2

2.2. Quantum Computation 3

2.3. Order Finding Problem 4

3. Simulation **6**

Bibliography **8**

1. Introduction

高性能な量子コンピュータが開発された場合、古典コンピュータでは解くことが困難な問題が解けるようになる可能性がある。 それに伴い、現在使用されている RSA 暗号という暗号が解読されてしまう可能性がある。

RSA 暗号とは、桁数が大きい合成数の素因数分解が現実的な範囲内で困難であることを安全性の根拠とした公開鍵暗号の一つである。現在使われている RSA 暗号はおよそ 600 桁（2048 ビット）である。

Shor のアルゴリズムでは、量子コンピュータを用いてこの素因数分解を高速に解くことができる。

2. Theory

2.1. Elementary Number Theory

Theorem 2.1.1 (ユークリッドの互除法): 自然数 a, b $(a > b)$ について、a を b で割ったときの商を q、余りを r $(\neq 0)$ とすると、以下が成り立つ。

$$\gcd(a, b) = \gcd(b, r) \tag{2.1}$$

Theorem 2.1.1 から、最大公約数を求める計算は古典コンピュータでも十分高速に計算できることがわかる。

Theorem 2.1.2 (中国剰余定理): n_1, n_2 を互いに素である自然数とし、p, q を任意の整数とする。 このとき、

$$\begin{aligned} s &\equiv p \pmod{n_1} \\ s &\equiv q \pmod{n_2} \end{aligned} \tag{2.2}$$

を満たす $0 \leq s < n_1 n_2$ がただ一つ存在する。

因数分解したい数を $N = n_1 n_2$ とする。このとき、Theorem 2.1.2 より、

$$\begin{aligned} s &\equiv 1 \pmod{n_1} \\ s &\equiv -1 \pmod{n_2} \end{aligned} \tag{2.3}$$

を満たす $1 < s < N - 1$ がただ一つ存在する。[1] これらの式を 2 乗すると、

$$\begin{aligned} s^2 &\equiv 1 \pmod{n_1} \\ s^2 &\equiv 1 \pmod{n_2} \end{aligned} \tag{2.4}$$

となるので、

$$s^2 \equiv 1 \pmod{N} \tag{2.5}$$

が成り立つ。[2] このとき、

[1] n_1, n_2 はともに 3 以上とする。$s = 0$ のとき、$0 \equiv 1 \pmod{n_1}$ となり、これを満たすのは $n_1 = 1$ なので不適。$s = 1$ のとき、$1 \equiv -1 \pmod{n_2}$ となり、これを満たすのは $n_2 = 2$ なので不適。 $s = N - 1$ のとき、$n_1 n_2 - 1 = k n_1 - 1$ より、$n_1(n_2 - k) = 2$ となり、これを満たすのは $n_1 = 1, 2$ なので不適。

[2] $s^2 - 1 = k_1 n_1,\ s^2 - 1 = k_2 n_2$ が成り立つので、$k_1 n_1 = k_2 n_2$ である。 n_1, n_2 は互いに素だから、k_1 は n_2 の倍数である。 したがって、$s^2 - 1 = k_1' n_1 n_2$ であり、N の倍数である。

$$(s+1)(s-1) \equiv 0 \pmod{N} \tag{2.6}$$

が成り立つ。したがって、$\gcd(N, s+1), \gcd(N, s-1)$ を考えることによって、因数を見つけることができる。 すなわち、素因数分解問題は Equation (2.5) を満たす s を見つける問題に帰着する。

$1 < a < N-1$ を満たす整数 a をランダムに選ぶ。 $\gcd(N, a) = 1$ であるとき、

$$a^r \equiv 1 \pmod{N} \tag{2.7}$$

を満たす r を見つければよい。[3] 解は、Equation (2.5) と比較して、

$$s = a^{\frac{r}{2}} \tag{2.8}$$

と求まる。 失敗した場合は a を選びなおしてやり直す。 この r を量子計算で求めることになる。 Table 1 は、21 の素因数分解における r, s の値と因数分解の結果である。

Table 1: 21 の因数分解

a	r	$(s \mod 21)$	結果
2	6	8	(3,7)
4	3		r が奇数
5	6	20	(21,1)
8	2	8	(3,7)
10	6	13	(7,3)
11	6	8	(3,7)
13	2	13	(7,3)
16	3		r が奇数
17	6	20	(21,1)
19	6	13	(7,3)

2.2. Quantum Computation

1 量子ビットの状態は以下のように表される。

$$|\psi\rangle = c_0|0\rangle + c_1|1\rangle \tag{2.9}$$

同様に、N 量子ビットの状態は以下のように表される。

$$|\psi\rangle = \sum_{i=0}^{2^N-1} c_i|i\rangle \tag{2.10}$$

量子ビットにユニタリ演算子を作用させることによって、計算を行う。

Hadamard ゲート

$$\begin{aligned} H|0\rangle &= \frac{|0\rangle + |1\rangle}{\sqrt{2}} \\ H|1\rangle &= \frac{|0\rangle - |1\rangle}{\sqrt{2}} \end{aligned} \tag{2.11}$$

各量子ビットにそれぞれ Hadamard ゲートを作用させると、すべての状態が均等に重ね合わさった状態になる。

[3] $\gcd(N, a) = 1$ でなければ、それが因数なので、この問題はすでに解決している。

$$
\begin{aligned}
|0\rangle|0\rangle|0\rangle \to \frac{|0\rangle+|1\rangle}{\sqrt{2}}\frac{|0\rangle+|1\rangle}{\sqrt{2}}\frac{|0\rangle+|1\rangle}{\sqrt{2}} \\
&= \frac{1}{\sqrt{2^3}}(|000\rangle+|001\rangle+...+|111\rangle) \\
&= \frac{1}{\sqrt{8}}(|0\rangle+|1\rangle+...+|7\rangle)
\end{aligned}
\tag{2.12}
$$

controlled-U ゲート

$$
\begin{aligned}
U_{\mathrm{CU}}|10\rangle &= |1\rangle\otimes U|0\rangle \\
U_{\mathrm{CU}}|11\rangle &= |1\rangle\otimes U|1\rangle
\end{aligned}
\tag{2.13}
$$

量子フーリエ変換（QFT）

$$
U_{\mathrm{QFT}}|j\rangle = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} e^{2\pi ijk/N}|k\rangle \tag{2.14}
$$

逆の操作は、逆量子フーリエ変換（IQFT）である。 QFT がユニタリであることは以下のように示せる。 QFT は演算子で表せば、

$$
U = \frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1} e^{2\pi ijk/N}|k\rangle\langle j| \tag{2.15}
$$

である。

$$
\begin{aligned}
\langle x|U^\dagger U|y\rangle &= \left(\frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} e^{-2\pi ixk/N}\langle k|\right)\left(\frac{1}{\sqrt{N}}\sum_{l=0}^{N-1} e^{2\pi iyl/N}|l\rangle\right) \\
&= \frac{1}{N}\sum_{k=0}^{N-1} e^{-2\pi ik(x-y)/N} = \frac{1}{N}N\delta_{xy} = \langle x|y\rangle
\end{aligned}
\tag{2.16}
$$

より、$U^\dagger U = I$ が示される。

2.3. Order Finding Problem

$a^r \equiv 1 \pmod N$ を満たす r を a の位数という。[4] x と N は互いに素であるとする。

$$
U|y\rangle = |xy \bmod N\rangle \tag{2.17}
$$

となる U を考える。 U はユニタリ演算子である。

proof: $\langle y_1|U^\dagger U|y_2\rangle = \langle xy_1 \bmod N|xy_2 \bmod N\rangle$ の値は、$xy_1 \equiv xy_2$ のとき 1、それ以外は 0 である。 x, N は互いに素なので、$y_1 \equiv y_2$ と同値である。 したがってこの式は $\langle y_1|y_2\rangle$ となり、$U^\dagger U = I$ が示される。 □

このとき、

$$
U^j|y\rangle = \left|x^j y \bmod N\right\rangle \tag{2.18}
$$

である。 U の固有状態は、$x^r \equiv 1 \pmod N$ のとき、

$$
|u_s\rangle = \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} e^{-2\pi isk/r}\left|x^k \bmod N\right\rangle \tag{2.19}
$$

である。

[4] a の位数 r は存在するならば $r \le N$ を満たす。

proof:

$$
\begin{aligned}
U|u_s\rangle &= U\left(\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} e^{-2\pi isk/r}\left|x^k \bmod N\right\rangle\right) \\
&= \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} e^{-2\pi isk/r}\left|x^{k+1} \bmod N\right\rangle \\
&= e^{2\pi is/r}\frac{1}{\sqrt{r}}\sum_{m=1}^{r} e^{-2\pi ism/r}|x^m \bmod N\rangle
\end{aligned}
\tag{2.20}
$$

ここで、

$$
e^{-2\pi ism/r}|x^m \bmod N\rangle|_{m=0} = e^{-2\pi ism/r}|x^m \bmod N\rangle|_{m=r} \tag{2.21}
$$

より、$U|u_s\rangle = e^{2\pi is/r}|u_s\rangle$ である。 □

Equation (2.19) より、以下の式が成り立つ。

$$
\frac{1}{\sqrt{r}}\sum_{s=0}^{N-1} e^{2\pi isk/r}|u_s\rangle = \left|x^k \bmod N\right\rangle \tag{2.22}
$$

proof:

$$
\begin{aligned}
\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1} e^{2\pi isk/r}|u_s\rangle &= \frac{1}{r}\sum_{m=0}^{r-1}\sum_{s=0}^{r-1} e^{2\pi is(k-m)/r}|x^m \bmod N\rangle \\
&= \frac{1}{r}\sum_{m=0}^{r-1} r\delta_{km}|x^m \bmod N\rangle \\
&= \left|x^k \bmod N\right\rangle
\end{aligned}
\tag{2.23}
$$

□

以下に位数発見問題の手順を示す。t は第 1 ビット列のビット数である。[5]

$$
\begin{aligned}
& |0\rangle|1\rangle && \text{(初期状態)} \\
\to\ & \frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}|j\rangle|1\rangle && \text{(Hadamard ゲートを適用)} \\
\to\ & \frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}|j\rangle\left|x^j \bmod N\right\rangle && \text{(U を適用)} \\
=\ & \frac{1}{\sqrt{r2^t}}\sum_{s=0}^{r-1}\sum_{j=0}^{2^t-1} e^{2\pi isj/r}|j\rangle|u_s\rangle && \\
\to\ & \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}\left|2^t\frac{s}{r}\right\rangle|u_s\rangle && \text{(IQFT を適用)} \\
\to\ & 2^t\frac{s}{r} && \text{(測定)} \\
\to\ & r && \text{(連分数展開)}
\end{aligned}
\tag{2.24}
$$

連分数展開とは、小数をある精度で分数に直す方法である。例えば、0.69231 を簡単な分数で表す過程は以下の通り。

[5] t は、N のビット数 L の 2 倍より少し大きい数をとるとよいとされている。

$$
\begin{aligned}
0.69231 \simeq \frac{1}{1.44443} &= \frac{1}{1+0.44443} \\
&\simeq \frac{1}{1+\frac{1}{2.25007}} = \frac{1}{1+\frac{1}{2+0.25007}} \\
&\simeq \frac{1}{1+\frac{1}{2+\frac{1}{3.99888}}} \simeq \frac{1}{1+\frac{1}{2+\frac{1}{4}}} = \frac{9}{13}
\end{aligned}
\tag{2.25}
$$

量子計算で r が求められることがわかったが、高速に計算できるのか、計算量を吟味する必要がある。古典計算では、素因数分解は $O\left(\sqrt{N}\right) = O\left(2^{\frac{L}{2}}\right)$ であり、指数時間で計算される。一方、量子計算では、Hadamard ゲートで $O(L)$、QFT で $O(L^2)$、U の適用で $O(L^3)$ なので、合わせて $O(L^3)$ であり、多項式時間で計算される。[1]

3. Simulation

$N = 21$ のとき、この方法を実際に試す。 ビット数は $L = 5$ であり、$t = 13$ とする。 まず、ランダムに数を選ぶ。 今回は $x = 19$ とする。 $x^j \bmod N$ の値は Figure 1 のように周期的である。

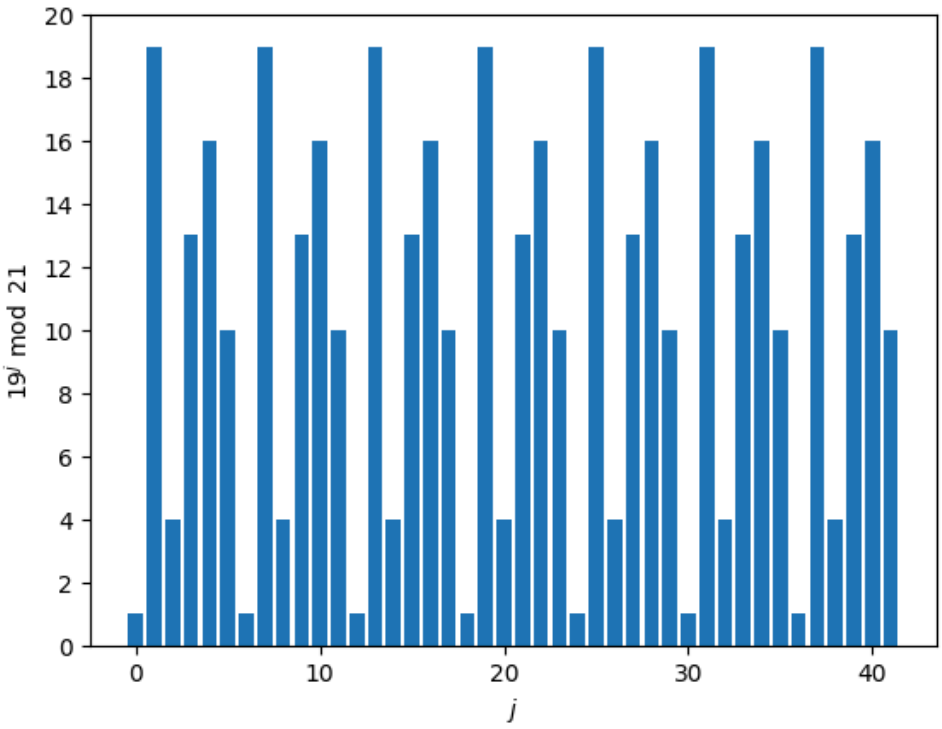

Figure 1: $x^j \bmod N$ の規則性 $(N = 21, x = 19)$

IQFT を作用させた後は、Figure 2 のような確率分布になる。

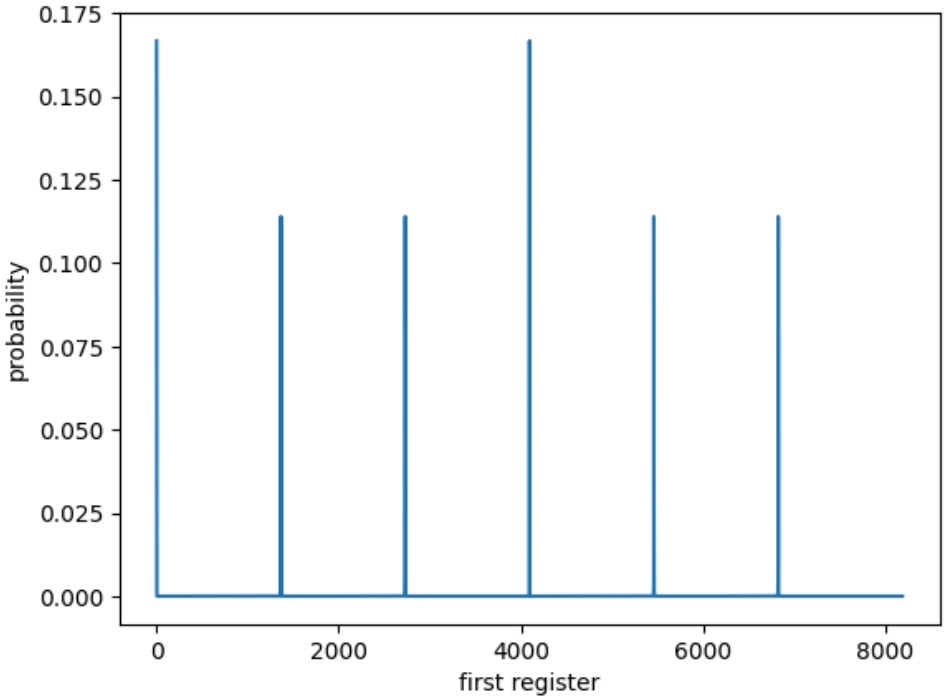

Figure 2: IQFT ($N = 21, t = 13, x = 19$)

また、測定結果と得られた小数、連分数展開の結果が Table 2 である。

Table 2: 測定結果と s/r の値 ($N = 21, t = 13, x = 19$)

Measure	Phase	Fraction
0	0.000000	$0/1$
1365	0.166626	$1/6$
1366	0.166748	$1/6$
2730	0.333252	$1/3$
2731	0.333374	$1/3$
4096	0.500000	$1/2$
5461	0.666626	$2/3$
5462	0.666748	$2/3$
6826	0.833252	$5/6$
6827	0.833374	$5/6$

よって、何回も測定を行うことによって、$r = 2, 3, 6$ を得ることができるので、$r = 6$ であるとわかる。 したがって、$s = 19^{\frac{6}{2}} = 6859$ である。 $\gcd(6859 + 1, 21) = 7,\ \gcd(6859 - 1, 21) = 3$ より、因数分解をすることができた。[2]

では、第 1 ビット列のビット数 t を $t = L = 5$ としてみる。 このときの IQFT の結果は Figure 3 となる。

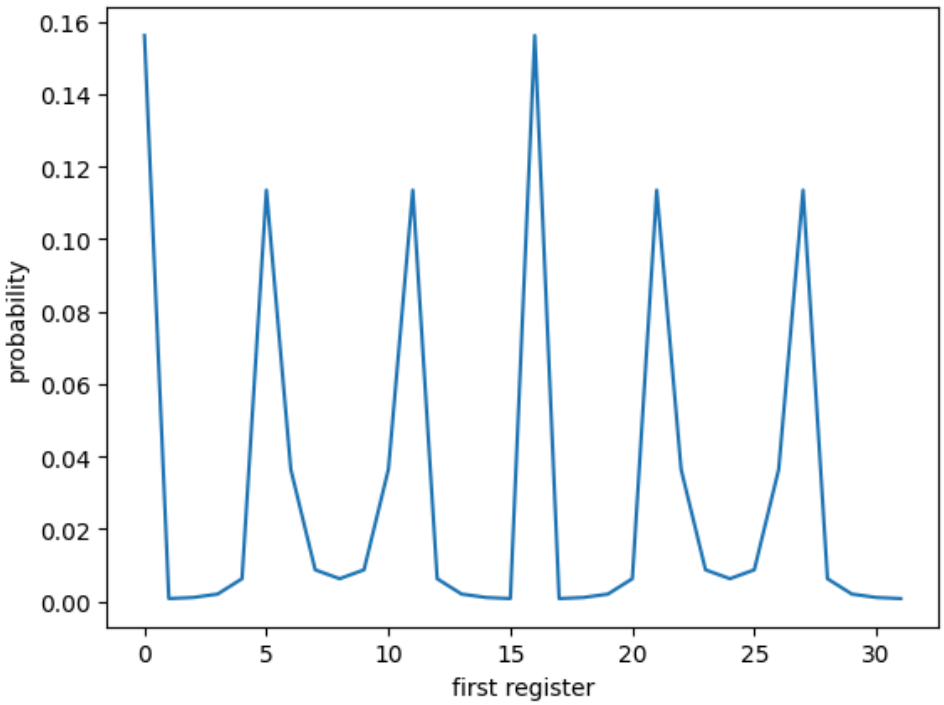

Figure 3: IQFT ($N = 21, t = 5, x = 19$)

また、測定結果は Table 3 の通り。

Table 3: 測定結果と s/r の値 ($N = 21, t = 5, x = 19$)

Measure	Phase	Fraction
0	0.00000	$0/1$
5	0.15625	$3/19$
11	0.34375	$7/20$
16	0.50000	$1/2$
21	0.65625	$13/20$
27	0.84375	$16/19$

このように、第 1 ビット列のビット数が小さいと正しい r の値が得られないことがわかる。

では、正しく r が与えられたとして、それが正しく因数を与える確率はどの程度だろうか。1000 までの、偶数と平方数を除いた半素数で成功確率をけいさんすると、Table 4 が得られる。

Table 4: 成功確率

最大値	100.00
平均値	76.52
最小値	53.35

したがって、何回か測定をすることによって、ほぼ確実に因数を与えられることがわかる。

Bibliography

[1] M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information. Cambridge University Press, 2010.

[2] Qiskit, “Shor's Algorithm”.