Shor Algorithm
A course work of Quantum Computation and Quantum Information

田博宇刘予希
ACM Class, 2016
2018 年 8 月 7 日

目录

1 Traditional Part with C# : Driver.cs

1.1 Main : the input interface

Input the number N to be factored.$(1 \leq N \leq 31)$. Use the function **factor(N)** to calculate the answer.

1.2 gcd(a, b) : the function to calculate greatest common divisor

Euclidean Algorithm.

1.3 Frac : a class to implement fraction

Implement a fraction class with addition, inversion and normalization.

1.4 quickPower(x, y, N) : the function to calculate $x^y \mod N$

1.5 factor(N) : calculate a non-trivial factor of N

The main factorization algorithm.

1. If N is even, return the factor 2.

2. Randomly choose x in the range from 1 to $N-1$. If $(gcd(x, N) \geq 1)$ then return the factor $gcd(x, N)$.

3. Use the order-finding subroutine to find the order r of x modulo N.

4. If r is even and $x^{\frac{r}{2}} \neq -1 (\mod N)$ then compute $gcd(x^{\frac{r}{2}} - 1, N)$ and $gcd(x^{\frac{r}{2}} + 1, N)$, and test to see if one of these is a non-trivial factor, returning that factor if so. Otherwise, the algorithm fails.

1.6 qOrderFinding(x, N) : the function to calculate the order of x to N

The main function of order finding.

1. Find phase estimation $\frac{s}{r}$ using quantum algorithms(see getPhaseEstimation)
2. Generate continued fraction of $\frac{s}{r}$.
3. Restore sr into a group of fractions.
4. Check the dominator of each fraction.

1.7 getPhaseEstimation(a, N) : the function to calculate the phase estimation

Use quantum algorithm to get phase estimation(see quantumOrderFinding in Shor.qs)

2 Nontraditional Part with Q# : Shor.qs

2.1 U_xN : the black box which performs the transformation $|j\rangle|k\rangle \rightarrow |j\rangle|x^j k \mod N\rangle$

Use the method **ModularMultiplyByConstantLE** from **Microsoft.Quantum.Canon**

2.2 quickPower(x, y) : the function to calculate x^y

2.3 quickPowerWithModule(x, y, N) : the function to calculate $x^y \mod N$

2.4 measure(qubits, t) : the method to measure the t qubits

Use the **M** door for each qubits.

2.5 quantumFourierTransform(qubits) : the circuit to implement quantum fourier trasformation

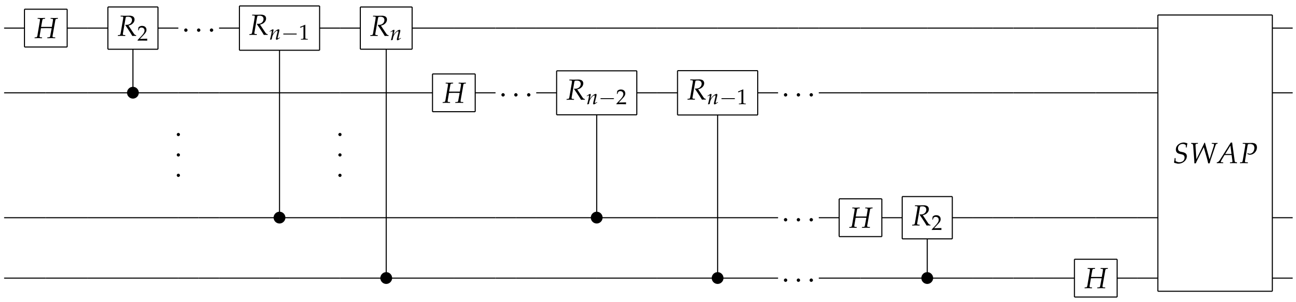
Get the details in the circuit section.

2.6 quantumOrderFinding

Get the details in the circuit section.

3 Quantum Circuit

3.1 Quantum Fourier Transformation



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

3.2 Adjoint Quantum Fourier Transformation

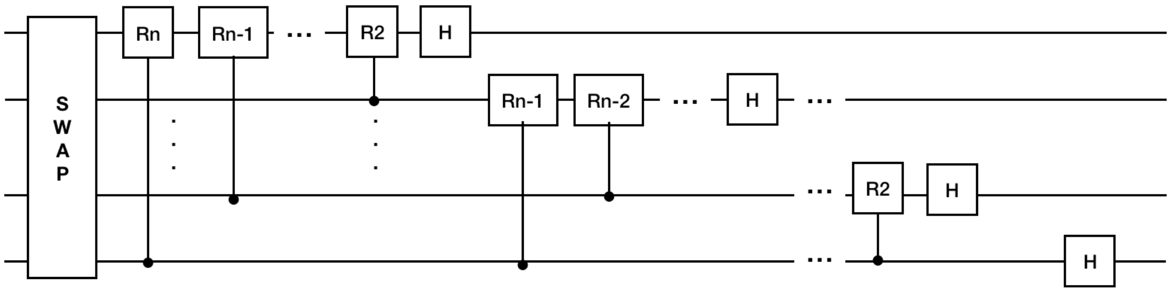


图 1:

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

3.3 Quantum Order Finding

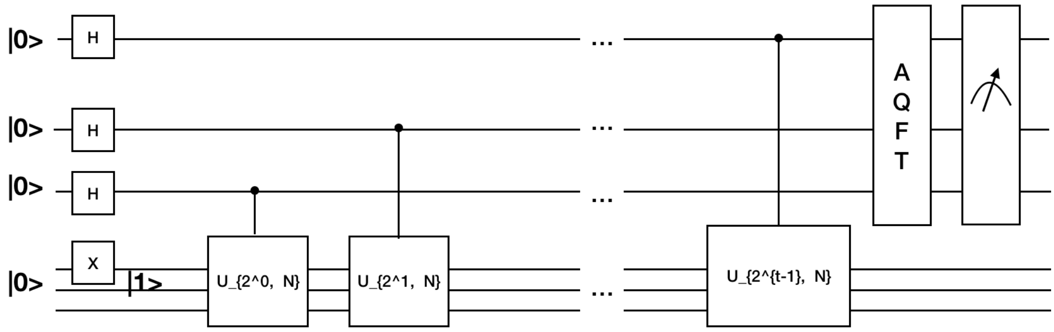


图 2: