papergrid®

Reflects You.

You may have to fight a battle more than once to win it.

soorajs51729@gmail.com +91-9400635788

I N D E X

8

Name SOORAJ·S· Std____ Sec____

Roll No.____ Subject____ School/College____

Sl. No.	Date	Title	Page No.	Teacher Sign/ Remarks
		QUANTUM COMPUTATION		
		& QUANTUM INFORMATION		
		- Nielsen & Chuang		

Application : order-finding

$$x = kN + b.$$

$$\begin{array}{r|l} & \;k \\ \hline N & \;x \\ & \overline{\;b\;} \end{array}$$

$$x = b \pmod N$$

L.A (15)

$$b = x \text{ div } N$$

$$b = x \bmod N$$

* Suppose N is a +ve integer, and x is coprime to N, $1 \le x < N$.

The order of x modulo N is defined to be the least positive integer r such that $x^r = 1 \pmod N$. The order finding problem is to determine r, given x and N.

Ex: 5.10 Show that the order of $x = 5$ modulo $N = 21$ is 6.

Ans: $x = 5$ modulo 21 = 5 div 21

$5^r = 1 \pmod{21}$ $\quad 21\,\overline{)\,5\,}$ x

$5^1 = 5 \pmod{21}$

$5^2 = 4 \pmod{21}$

$5^3 = 19 \pmod{21}$

$5^4 = 5 \pmod{21}$

$5^5 = 17 \pmod{21}$

$5^6 = 1 \pmod{21}$

Modular exponentiation

Recall the exponential function, a^x

The modular exponential function is obtained by taking this function and calculating the remainder on division by N.

i.e., $F_N(x) = a^x \bmod N$

The order of the modular exponential, referred to as the order of $a \bmod N$ or $\mathrm{ord}(a)$, is the smallest positive integer r such that $a^r \bmod N = 1$

Equivalently, we can say that r is the period of this function $F_N(x) = a^x \bmod N$.

$a^r \bmod N = 1 \Longrightarrow \quad N \,\overline{)\, a^r\,}$ quotient k, remainder 1

$a^r = kN + 1$

$\Rightarrow a^{r+1} = kNa + a \quad N \,\overline{)\, a^{r+1}\,}$ remainder x

$\Rightarrow a^{r+1} \bmod N = a \bmod N$

$18 \bmod 5 = 23 \bmod 5$
$= 3 \bmod 5$

where $k \in \mathbb{Z}$.

$\therefore \quad F_N(x+r) = F_N(x)$

where $F_N(x) = a^x \bmod N$

$\Longrightarrow r$ is the period of $F_N(x)$

Note: $r \le N$

1

Order-finding is believed to be a hard problem on a classical computer, in the sense that no algorithm is known to solve the problem using resources polynomial in the $O(L)$ bits needed to specify the problem, where

$L \approx \log(N)$ is the # of bits needed to specify N

The quantum algorithm for order-finding is just the phase estimation algorithm applied to the unitary operator,

$$U_x|y\rangle = |xy \pmod N\rangle$$

with $y \in \{0,1\}^L$.

$\{0,1\}^L$: vectors of length L where each component is 0 or 1.

The effect of U_x on the basis states $|y\rangle$ where $N \le y \le 2^L-1$ is not specified.
Hence, it does not matter where these states are being mapped to by U_x, as long as U_x is a unitary operation.

One could choose to map these states to themselves, i.e., $U_x|y\rangle = |y\rangle$ for $N \le y \le 2^L-1$.
But for our purpose, it suffices to assume that the mapping is unitary.

The operation U_x is invertible as x has a multiplicative inverse modulo N since x and N are coprime, which is $x^{r-1} \pmod N$.
Applying the operation $|y\rangle \to |x^{r-1}y \pmod N\rangle$ for any $0 \le y \le N-1$ would lead to an inverse operation of U.

As U_x only cycles computational basis states, the length of any state vector remains unaltered, hence the operation U_x is indeed unitary, so it can be implemented in a quantum circuit.

* Given x is coprime to N, the operator U_x which is a $2^L-1 \times 2^L-1$ matrix defined as

$$U_x|y\rangle = |xy \pmod N\rangle \quad \text{when } 0 \le y \le N-1$$

$$U_x|y\rangle = |y\rangle \quad \text{when } N \le y \le 2^L-1$$

is unitary.

If the order of $x \pmod N$ is r,

i.e., $x^r = 1 \pmod N$ and $U_x^r = I$

then all eigenvalues of U_x are r^{th} roots of unity:

$$\lambda_s = e^{\frac{2\pi i s}{r}}, \quad s \in \{0, 1, 2, \dots, r-1\}$$

and the corresponding eigenstates are:

$$|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-\frac{2\pi i s k}{r}} |x^k \pmod N\rangle$$

Proof

① $U_x|y\rangle = |xy \pmod N\rangle$

$U_x|z\rangle = |xz \pmod N\rangle$, $x, y \in \{0, 1, \cdots, N-1\}$

and $x \neq y$

If U_x map $|y\rangle$ and $|z\rangle$ to the same vector, then

$$xy = xz \pmod N$$

x is coprime to N $\Big\}$ x has a multiplicative
$\gcd(x, N) = 1$ $\Big\}$ modulo N.

$\therefore \quad y = z \pmod N$

$\longrightarrow y = z$ since $y, z \in \{0, 1, \cdots, N-1\}$.

This is a contradiction.

$\therefore$ When $0 \leq y \leq N-1$, U_x will map bijectively to another vector in that range.

When $N \leq y \leq 2^L - 1$, we require that U_x map to y so that the function is invertible for all y.

If $0 \le y \le N-1$, then,

$$\langle y|U^\dagger U|z\rangle = \langle xy \pmod N | xz \pmod N\rangle$$

$$= \delta_{(xy \pmod N),\, xz \pmod N)}$$

$$= \delta_{yz}$$

If $N \le y \le 2^L - 1$ then,

$$\langle y|U^\dagger U|z\rangle = \langle y|z\rangle = \delta_{yz}$$

If y is in one range, and z is in another, then $\langle y|U^\dagger U|z\rangle = 0$ because each will get mapped to a different part of the image.

② If $|u\rangle$ is an eigenvector of U associated with eigenvalue λ then, $U|u\rangle = \lambda|u\rangle$

$$U^k|u\rangle = \lambda^k|u\rangle$$

$$U|y\rangle = |xy \pmod N\rangle \Longrightarrow U^k|y\rangle = |x^k y \pmod N\rangle$$

for a computational basis state $|y\rangle$ encoding a y coprime to N.

Let r be the order of x modulo N, i.e., $x^r = 1 \pmod N$. Then,

$$U^r|u\rangle = \lambda^r|u\rangle = |u\rangle$$

$\Longrightarrow \lambda$ is a # such that $\lambda^r = 1$.

$\therefore$ The eigenvalues of U_x are the rth roots of unity.

$\therefore \underline{\lambda_s = e^{2\pi i s/r}}$ for $s = 0, 1, \dots, r-1$

③ We begin with the observation that U_x permutes the states $|x^0 \pmod N\rangle, \ldots, |x^{r-1} \pmod N\rangle$.

Consequently, the uniform superposition

$$|v_0\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} |x^k \pmod N\rangle$$

is an eigenvector associated with eigenvalue 1.

Hoping that this might be a special case of a more general pattern, let's try

$$|v_{a_0,\ldots,a_{r-1}}\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} a_k |x^k \pmod N\rangle,$$

$$a_k \in \mathbb{C}$$

$$U|v_{a_0,\ldots,a_{r-1}}\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} a_k |x^{k+1} \pmod N\rangle$$

By definition of eigenvector, we need this reshuffling to allow us to pull a constant out in front of the sum: So we need $a_k = a^k$ for some constant a, i.e.,

$$|v_a\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} a^k \, |x^k \pmod N\rangle$$

$$U|v_a\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} a^k \, |x^{k+1} \pmod N\rangle$$

$$= a^{-1} \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} a^{k+1} \, |x^{k+1} \pmod N\rangle$$

If $a^r = 1$ then

$$U|v_a\rangle = \frac{a^{-1}}{\sqrt{r}} \sum_{k=0}^{r-1} a^k \, |x^k \pmod N\rangle$$

$$= a^{-1} |v_a\rangle$$

and $|v_a\rangle$ is the eigenvector associated with eigenvalue $\lambda = a^{-1}$.

$$\Rightarrow a = e^{-2\pi i s/r}$$

$$\therefore \quad |u_s\rangle = |v_{e^{-2\pi i s/r}}\rangle = \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} e^{-2\pi i ks/r}\,|x^k (\text{mod } N)\rangle$$

for integers $0 \le s \le r-1$ are eigenstates of U_x associated with eigenvalue $e^{2\pi i s/r}$.

The phase Estimation algorithm, given U_x and $|u_s\rangle$, with high accuracy, would find s/r from which we can obtain the order r of of x modulo N.

There are some obstacles to use the phase estimation procedure for this:

- We must have efficient procedures to implement a controlled-$U_x^{2^j}$ operation for any integer j (modular exponentiation).
- We must be able to efficiently prepare an eigenstate $|u_s\rangle$ with a non-trivial eigenvalue, or atleast a superposition of such eigenstates.
- Once we have some n-bit approximation of (s/r), how do we get r?

Modular Exponentiation

How can we compute the sequence of controlled-U^{2^j} operations used by the phase estimation procedure as part of the order-finding algorithm ?

That is, we wish to compute the transformation

$$|z\rangle|y\rangle \longrightarrow |z\rangle U^{z_1 2^{t-1}} \cdots U^{z_t 2^0}|y\rangle$$

$$= |z\rangle\left|x^{z_1 2^{t-1}} \times \cdots \times x^{z_t 2^0} y \pmod N\right\rangle$$

$$= |z\rangle|x^z y \pmod N\rangle$$

$\therefore$ The sequence of controlled-U^{2^j} operations used in phase estimation is equivalent to multiplying the contents of the 2^{nd} register by the modular exponential $x^z \pmod N$, where z is the contents of the 1^{st} register.

This operation may be accomplished easily using the techniques of reversible computation. The basic idea is to reversibly compute the function $x^2 \pmod N$ of z in a 3^{rd} register, and then to reversibly multiply the contents of the 2^{nd} register by $x^2 \pmod N$, using the trick of uncomputation to erase the contents of the 3^{rd} register upon completion

The algorithm for computing the modular exponential has 2 stages:

Step 1:

The 1^{st} stage uses modular multiplication to compute $x^2 \pmod N$, by squaring x modulo N, then compute $x^4 \pmod N$ by squaring $x^2 \pmod N$, and continue in this way, computing $x^{2^j} \pmod N$ for all j upto $t-1$.

Step 2:

The 2nd stage of the algorithm is based upon

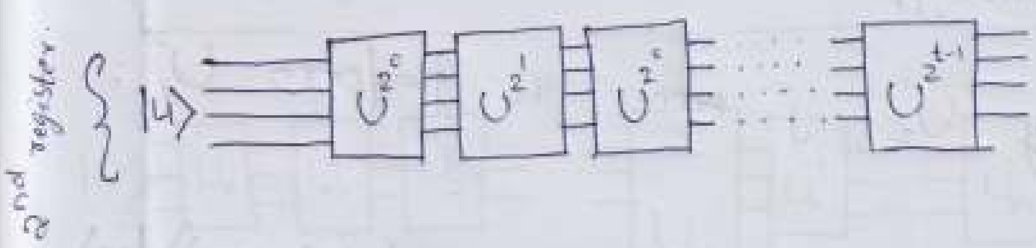
$$x^{z}(\text{mod } N) = \left(x^{z_t 2^{t-1}}(\text{mod } N)\right)\left(x^{z_{t-1} 2^{t-2}}(\text{mod } N)\right)\cdots\left(x^{z_1 2^{0}}(\text{mod } N)\right)$$

. U_x operator : $U_x|y\rangle = |xy \pmod N\rangle$

Modular multiplication : $U_m(|x\rangle|y\rangle) = |x\rangle|xy \pmod N\rangle$

Modular squaring : $U_2|x\rangle = |x^2 \pmod N\rangle$

We can use the modular multiplication and the modular squaring above to build the modular exponentiation below:



We start with some ancilla bits $|1\rangle$ and the eigenvector $|u\rangle$

$$|1\rangle|u\rangle$$

Applying the U_x operator,

$$U_x|1\rangle \longrightarrow |x\rangle$$

$$|1\rangle|u\rangle \longrightarrow |x\rangle|u\rangle$$

Then we apply the modular squaring repetitively,

Apply U_x j times to perform $U_x^{2^j}$:

$$|x\rangle|u\rangle \longrightarrow |x^2 \,(\text{mod } N)\rangle|u\rangle \longrightarrow |x^4 \,(\text{mod } N)\rangle|u\rangle$$
$$\longrightarrow |x^8 \,(\text{mod } N)\rangle|u\rangle \longrightarrow \cdots \rightarrow |x^{2^j} (\text{mod } N)\rangle|u\rangle$$

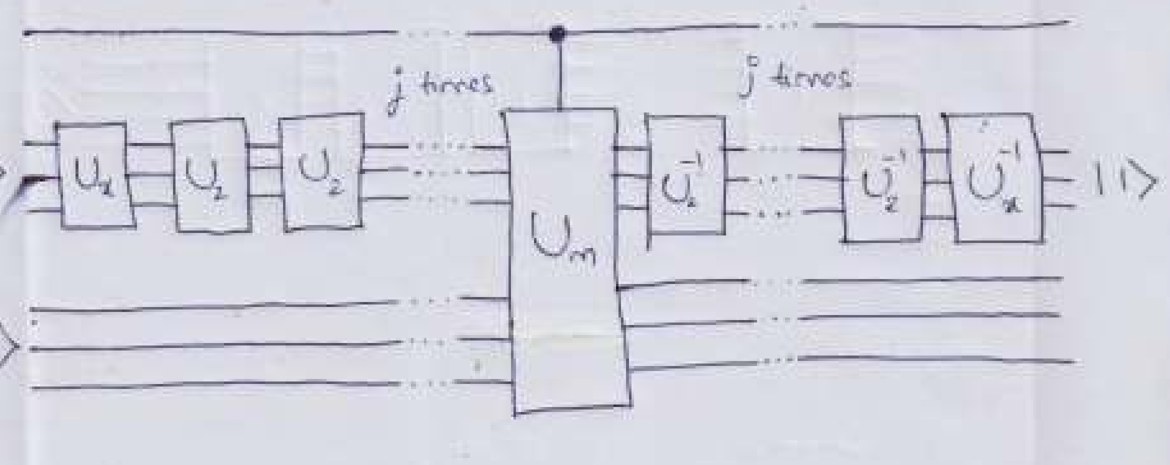
Finally, apply the modular multiplication U_m

$$|x^{2^j}(\text{mod } N)\rangle|u\rangle \longrightarrow |x^{2^j}(\text{mod } N)\rangle|x^{2^j}u\,(\text{mod } N)\rangle$$

$$\downarrow$$

$$U^{2^j}$$

* The final circuit using those modulos in producing the modular multiplication. The 2nd half of the circuit simply reverse the operation to recover the ancilla qubits.



2 Eigenstate Preparation

Given x is coprime to N,

$U_x|y\rangle = |xy \pmod N\rangle$ when $0 \le y \le N-1$

$U_x|y\rangle = |y\rangle$ when $N \le y \le 2^L - 1$

such that U_x is unitary.

The eigenstates of U_x are

$$|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-\frac{2\pi i s k}{r}} |x^k \pmod N\rangle$$

and the corresponding eigenvalues are:

$$\lambda_s = e^{\frac{2\pi i s}{r}} \quad , \quad s = \{0, \ldots, r-1\}$$

which are the r^{th} roots of unity..

Preparing $|u_s\rangle$ requires that we know r beforehand, so this is out of the question.

There is a clever observation which allows us to circumvent the problem of preparing $|u_s\rangle$, which is that

$$\boxed{\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|u_s\rangle = |1\rangle}$$

Proof

$$\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|u_s\rangle = \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}\left(\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} e^{-\frac{2\pi i s k}{r}}|x^k \pmod N\rangle\right)$$

$$= \frac{1}{r}\sum_{s=0}^{r-1}\sum_{k=0}^{r-1} e^{-\frac{2\pi i s k}{r}}|x^k \pmod N\rangle$$

$$= \frac{1}{r}\sum_{k=0}^{r-1}\left(\sum_{s=0}^{r-1} e^{-\frac{2\pi i s k}{r}}\right)|x^k \pmod N\rangle$$

$$\sum_{s=0}^{r-1} e^{\frac{2\pi i s}{r}} = 0$$

$$= \frac{1}{r}\cdot r\,|x^0 \pmod N\rangle = |1\rangle$$

In fact, this is a particular case of a more general observation that,

$$\boxed{\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1} e^{\frac{2\pi i s k}{r}}|u_s\rangle = |x^k \pmod N\rangle}$$

<u>Proof</u>

$$\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1} e^{\frac{2\pi i s k}{r}}|u_s\rangle = \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1} e^{\frac{2\pi i s k}{r}}\left(\frac{1}{\sqrt{r}}\sum_{k'=0}^{r-1} e^{\frac{-2\pi i s k'}{r}}|x^{k'} \pmod N\rangle\right)$$

$$= \frac{1}{r}\sum_{k'=0}^{r-1}|x^{k'} \pmod N\rangle \sum_{s=0}^{r-1} e^{\frac{2\pi i s(k-k')}{r}}$$

$$= \frac{1}{r}\sum_{k'=0}^{r-1}|x^{k'} \pmod N\rangle \left(r\,\delta_{kk'}\right)$$

$$= |x^k \pmod N\rangle$$

Set $k=0$, $\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|u_s\rangle = |1\rangle$

→ The state $|1\rangle$ is an equal superposition of the eigenstates of the operator U_x.

In performing the phase estimation procedure, if we use $\boxed{t = 2L+1+\log_2\left(2+\frac{1}{2\epsilon}\right)}$ qubits in the 1st register, and prepare the 2nd register in the state $|1\rangle$ – which is trivial to construct, it follows that for each s in the range 0 through $r-1$, we'll obtain an estimate of the phase $\phi \approx s/r$ accurate to $2L+1$ bits i.e., to an accuracy 2^{-2L-1}, with probability atleast $|c_s|^2(1-\epsilon) = \frac{1}{r}(1-\epsilon)$.

Ex. 5.14 The quantum state produced in the order-finding algorithm, before the IQFT is,

(a) $$\boxed{|\psi\rangle = \sum_{j=0}^{2^t-1} |j\rangle U^j |1\rangle = \sum_{j=0}^{2^t-1} |j\rangle |x^j (\mathrm{mod}\ N)\rangle}$$

if we initialize the 2^{nd} register as $|1\rangle$.

Ans. After ~~ap~~ the application of the 1^{st} unitary $U_x^{2^0}$ and the tensor product of the last qubit and the 2^{nd} register will be,

$$\frac{1}{\sqrt{2}}\left(|0\rangle|1\rangle + |1\rangle U_x|1\rangle\right) =$$

$$= \frac{1}{\sqrt{2}}\left(|0\rangle|1\rangle + |1\rangle \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{2\pi i s/r} |u_s\rangle\right)$$

$$= \frac{1}{\sqrt{2}}\left(|0\rangle|1\rangle + |1\rangle|x^1 (\mathrm{mod}\ N)\rangle\right)$$

Consider the tensor product of the last two qubits and the 2^{nd} register after the application of the 2^{nd} unitary, U_x^{2}.

$$\frac{1}{2}\left(|0\rangle|0\rangle|1\rangle + |0\rangle|1\rangle U_x|1\rangle + |1\rangle|0\rangle U_x^2|1\rangle + |1\rangle|1\rangle U_x^3|1\rangle\right)$$

$$= \frac{1}{2}\left(|0\rangle|0\rangle|1\rangle + |0\rangle|1\rangle|x^1(\mathrm{mod}\ N)\rangle + |1\rangle|0\rangle|x^2(\mathrm{mod}\ N)\rangle + |1\rangle|1\rangle|x^3(\mathrm{mod}\ N)\rangle\right)$$

The tensor product of the last 3 qubits and the 2nd register after the application of the 3rd unitary $U_x^{2^2}$

$$\frac{1}{2^{3/2}}\Big(|0\rangle|0\rangle|0\rangle|1\rangle + |0\rangle|0\rangle|1\rangle U_x|1\rangle + |0\rangle|1\rangle|0\rangle U_x^2|1\rangle + |0\rangle|1\rangle|1\rangle U_x^3|1\rangle$$
$$+ |1\rangle|0\rangle|0\rangle U_x^4|1\rangle + |1\rangle|0\rangle|1\rangle U_x^5|1\rangle + |1\rangle|1\rangle|0\rangle U_x^6|1\rangle$$
$$+ |1\rangle|1\rangle|1\rangle U_x^7|1\rangle\Big)$$

$$= \frac{1}{2^{3/2}}\Big(|0\rangle|0\rangle|0\rangle|1\rangle + |0\rangle|0\rangle|1\rangle|x^1 (\mathrm{mod}\ N)\rangle + |0\rangle|1\rangle|0\rangle|x^2(\mathrm{mod}\ N)\rangle$$
$$+ |0\rangle|1\rangle|1\rangle|x^3(\mathrm{mod}\ N)\rangle + |1\rangle|0\rangle|0\rangle|x^4(\mathrm{mod}\ N)\rangle$$
$$+ |1\rangle|0\rangle|1\rangle|x^5(\mathrm{mod}\ N)\rangle + |1\rangle|1\rangle|0\rangle|x^6(\mathrm{mod}\ N)\rangle$$
$$+ |1\rangle|1\rangle|1\rangle|x^7(\mathrm{mod}\ N)\rangle\Big)$$

:

Leaving out the factor of $\frac{1}{\sqrt{2^t}}$, the result of the phase estimation algorithm before the inverse QFT is:

$$|\psi\rangle = \sum_{j=0}^{2^t-1} |j\rangle U^j|1\rangle = \sum_{j=0}^{2^t-1} |j\rangle|x^j(\mathrm{mod}\ N)\rangle$$

$$|\psi\rangle = \sum_{j=0}^{2^t-1} |j\rangle |x^j \pmod N\rangle = \sum_{j=0}^{2^t-1} |j\rangle \left(\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{2\pi i s j/r} |u_s\rangle \right)$$

$$= \sum_{j=0}^{2^t-1} |j\rangle \frac{1}{\sqrt{r}} \left(e^{2\pi i j (0/r)} |u_0\rangle + e^{2\pi i j (1/r)} |u_1\rangle + e^{2\pi i j (2/r)} |u_2\rangle + \cdots + e^{2\pi i j \left(\frac{r-1}{r}\right)} |u_{r-1}\rangle \right)$$

$$= \frac{1}{\sqrt{r}} \sum_{j=0}^{2^t-1} e^{2\pi i j (0/r)} |j\rangle |u_0\rangle + \frac{1}{\sqrt{r}} \sum_{j=0}^{2^t-1} e^{2\pi i j (1/r)} |j\rangle |u_1\rangle + \cdots + \frac{1}{\sqrt{r}} \sum_{j=0}^{2^t-1} e^{2\pi i j \left(\frac{r-1}{r}\right)} |j\rangle |u_{r-1}\rangle$$

$$QFT|j\rangle = \frac{1}{2^{t/2}} \sum_{k=0}^{2^t-1} e^{2\pi i j k/2^t} |k\rangle$$

The inverse QFT acts on each sum linearly

The inverse QFT will give an equal superposition of all the possible phases in the 1st register

$$(QFT)^{\dagger}|\psi\rangle = \frac{1}{\sqrt{r}}\left(\left|\frac{2^t 0}{r}\right\rangle|u_0\rangle + \left|\frac{2^t 1}{r}\right\rangle|u_1\rangle + \left|\frac{2^t 2}{r}\right\rangle|u_2\rangle + \cdots + \left|\frac{2^t(r-1)}{r}\right\rangle|u_{r-1}\rangle\right)$$

$$= \frac{1}{\sqrt{r}}\left(\left|\frac{\tilde{0}}{r}\right\rangle|u_0\rangle + \left|\frac{\tilde{1}}{r}\right\rangle|u_1\rangle + \left|\frac{\tilde{2}}{r}\right\rangle|u_2\rangle + \cdots + \left|\frac{\widetilde{r-1}}{r}\right\rangle|u_{r-1}\rangle\right)$$

→ A measurement of the 1st register will give an approximation to s/r accurate to $2L+1$ bits, with a probability of error less than ϵ. Increasing the # of bits in the 1st register can put a very low upper-bound on the error; so at the end of the phase estimation algorithm, we can expect to have an approximation to s/r for some s ranging from 0 to $r-1$.

(b) Show that the same state is obtained if we replace U^j with a different unitary transform V, which computes

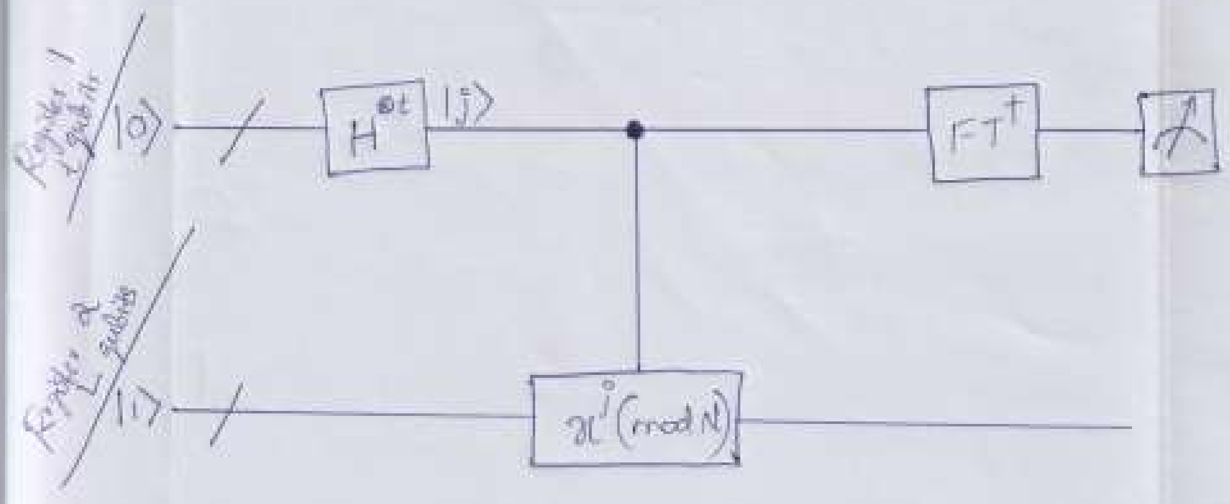
$$V|j\rangle|k\rangle = |j\rangle|k + x^j \pmod{N}\rangle$$

and start the 2^{nd} register in the state $|0\rangle$. Also show how to construct V using

7. $O(L^3)$ gates.

Ans. $V|j\rangle|0\rangle = |j\rangle|0 + x^j \pmod{N}\rangle = |j\rangle|x^j \pmod{N}\rangle$

* Quantum circuit for the order-finding algorithm.
— The 2^{nd} register is shown as being initialized to the $|1\rangle$ state, but if the method of Ex: 5.14 is used, it can be initialized to $|0\rangle$ instead. This circuit can also be used for factoring.



3 The Continued fraction expansion

How to obtain the desired answer, r, from the result of the phase estimation algorithm, $\phi \approx s/r$?

We only know ϕ to $2L+1$ bits, but we also know a priori that it is a rational number, and if we could compute the nearest such fraction to ϕ we might obtain r.

Theorem 5.1: Suppose s/r is a rational number such that,

$$\left|\frac{s}{r}-\phi\right| \le \frac{1}{2r^2}.$$

PROOF
Appendix 4.3
Number Theory

Then s/r is a convergent of the continued fraction for ϕ, and thus can be computed in $O(L^3)$ operations using the continued fraction algorithm.

QC Book
13/4/2022

Since ϕ is an approximation of s/r accurate to $2L+1$ bits, i.e., to an accuracy of 2^{-2L-1}, it follows that

$$\left|\phi - s/r\right| \leq \frac{1}{2^n} = \frac{1}{2^{2L+1}}$$

$$2r^2 \leq 2\phi^2(N) < 2N^2 \leq 2(2^L)^2 = 2^{2L+1}$$

$$\Rightarrow \frac{1}{2^{2L+1}} < \frac{1}{2r^2}$$

QC Book
13/4/2022

$\therefore$

$$\left|s/r - \phi\right| \leq \frac{1}{2^{2L+1}} < \frac{1}{2r^2}$$

$\longrightarrow$ The theorem 5.1 applies.

Given ϕ, the continued fraction algorithm efficiently produces numbers s' and r' with no common factor, such that $s'/r' = s/r$.

The number r' is our candidate for the order. We can check to see whether it is the order by calculating $x^{r'}$ mod N, and seeing if the result is 1. If so then r' is the order of x modulo N, and we are done.

Performance of order-finding algorithm

How can the order-finding algorithm fail? There are 2 possibilities.

1st The phase estimation procedure might produce a bad estimate to s/r. This occurs with probability at most ϵ, and can be made small with a negligible increase in the size of the circuit since

$$\epsilon = \frac{1}{2(e-1)} = \frac{1}{2(2^{t-n}-2)}.$$

2nd More serios!

It might be that s and r have a common factor, in which case the number r' returned by the continuous fractions algorithm be a factor of r, and not r itself.

There are at least 3 ways around this problem.

①

For a randomly chosen s in the range 0 through r-1, it's pretty likely that s and r are coprime, in which case the continued fractions algorithm must return r.

Prime Number theorem $\Rightarrow \pi(2n) \geq \frac{n}{\ln(2n)}$

Number theory 3

$\therefore$ The # of prime numbers less than r is atleast $\frac{r}{2\ln(r)}$.

i.e., $\pi(r) \geq \frac{r/2}{\ln(r)} = \frac{r}{2\ln(r)}$

$\therefore$ The chance that s is prime (and therefore coprime to r) is at least

$$\frac{\pi(r)}{r} = \frac{1}{2\ln(r)} > \frac{1}{2\ln(N)}.$$

Thus,

repeating the algorithm $2\ln(N)$ times we'll, with high probability, observe a phase s/r such that s and r are coprime, and therefore the continued fraction algorithm produces r, as desired.

(2) If $r' \neq r$, then r' is guaranteed to be a factor of r, unless $s=0$ which possibility occurs with $\frac{1}{r} \leq \frac{1}{2}$, and which can be discounted further by a few repetitions.

s is chosen uniformly at random from 0 through $r-1$.

$\therefore \quad P(s=0) = \frac{1}{r}$

We replace 'a' by $a' \equiv a^{r'} \pmod N$. Then the order of a' is r/r'. We can repeat the algorithm, and try to compute the order of a', which if we succeed, allows us to compute the order of a, since $r = r' \times \frac{r}{r'}$.

If we fail, then we obtain r'' which is a factor of r/r', and we now try to compute the order of $a'' \equiv (a')^{r''} \pmod N$.

We iterate this procedure until we determine the order of 'a'.

$r \le N \le 2^L \longrightarrow \log_2(r) \le \log_2(N) \le L$

$\therefore$

At most $\log_2(r) = O(L)$ iterations are required, since each repetition reduces the order of the current candidate $a''\cdots$ by a factor of atleast 2.

③

The 3^{rd} method is better than the 1^{st} two methods, in that it requires only a constant # of trials, rather than $O(L)$ repetitions.

The idea is to repeat the phase-estimation-continued fractions procedure twice, obtaining r_1', s_1' the 1^{st} time, and r_2', s_2' the second time. Provided s_1 and s_2 have no common factors, r may be extracted by taking the least common multiple of r_1 and r_2.

Ex:- $r = 60$

the phase estimation step gives a good approximate to $\frac{s_1}{r} = \frac{40}{60}$ (i.e., $s_1 = 40$) with $\frac{s_1'}{r_1'} = \frac{2}{3}$ (i.e., $s_1' = 2$, $r_1' = 3$) as the convergent.

Case 1

Performing the phase estimation again might give you a good approximate to $\frac{s_2}{r} = \frac{21}{60}$ (i.e., $s_2 = 21$), and you'll get $\frac{s_2'}{r_2'} = \frac{7}{20}$ (i.e., $s_2' = 7$, $r_2' = 20$) as a convergent.

$$\gcd(s_1, s_2) = \gcd(40, 21) = 1 \ \& \ \gcd(s_1', s_2') = \gcd(2, 7) = 1$$

$$\therefore \ \text{lcm}(r_1', r_2') = \text{lcm}(3, 20) = 60 = r$$

Case 2

If we got $\frac{40}{60} = \frac{2}{3}$ and $\frac{28}{60} = \frac{7}{15}$ in our two attempts, i.e., $s_1 = 40$ & $s_2 = 28$.

$s_1' = 2$, $r_1' = 3$; $s_2' = 7$, $r_2' = 15$.

$gcd(s_1, s_2) = gcd(40, 28) = 4$ &

$gcd(s_1', s_2') = gcd(2, 7) = 1.$

$\therefore lcm(r_1', r_2') = lcm(3, 15) = 15 \neq 60 = r$

Proof

Lemma: $\gcd(a, mn) = 1$ iff $\gcd(a, m) = 1$ &
$\gcd(a, n) = 1$

Suppose that $\gcd(s_1, s_2) = 1$,

$$\frac{s_1'}{r_1'} = \frac{s_1' a}{r_1' a} = \frac{s_1}{r} \quad \& \quad \frac{s_2'}{r_2'} = \frac{s_2' b}{r_2' b} = \frac{s_2}{r}$$

with $\gcd(r_1', s_1') = \gcd(r_2', s_2') = 1$.

$\gcd(s_1, s_2) = 1 \Longrightarrow \gcd(s_1' a, s_2' b) = 1$

Lemma implies, $\gcd(a, b) = 1$

$r = r_1' a = r_2' b$ & a, b have no common factors.

$a \mid r_2' b$ & $b \mid r_1' a$

$\Longrightarrow a \mid r_2'$ & $b \mid r_1'$

$\therefore r_1' = k_1 b$ & $r_2' = k_2 a$ for some integers $k_1, k_2 \in \mathbb{Z}$

Then,

$$a k_1 b = r = a k_2 b$$

$$\Longrightarrow k_1 = k_2 = k \text{ (say)}$$

Hence, $\gcd(r_1', r_2') = k$

$$\Longrightarrow \operatorname{lcm}(r_1', r_2') = \frac{r_1' r_2'}{\gcd(r_1', r_2')} = \frac{k_1 b \times k_2 a}{k}$$

$$= \frac{kb \times ka}{k} = akb = r$$

The probability that s_1 and s_2 have no common factors is given by,

$$1 - P(s_1 \ \& \ s_2 \text{ have common factors}) =$$

$$= 1 - \sum_q P(q|s_1)\, P(q|s_2)$$

where the sum is over all prime numbers q, and $P(x|y)$ means the probability of x dividing y.

s_1, s_2 are chosen uniformly at random from 0 through $r-1$.

$s=0$ case is eliminated.

Let n be the largest integer such that $nq \le r$, where q is the required prime number. Then there are n elements in $\{1, 2, \ldots, r-1\}$ which are divisible by q.

Start 2/6/22

$$P(\text{chosen } s \text{ is divisible by } q) = P(q|s)$$
$$= \frac{n}{r}$$

$$nq \le r \Longrightarrow \frac{n}{r} \le \frac{1}{q} \quad \Big\} \; P(q|s) \le \frac{1}{q}$$

$\therefore \quad P(q|s_1) \le \frac{1}{q} \quad \& \quad P(q|s_2) \le \frac{1}{q}$

$$P\left(s_1 \,\&\, s_2 \text{ have no common factors}\right) = 1 - \sum_q P(q|s_1)\,P(q|s_2)$$

$$\ge 1 - \sum_q \frac{1}{q^2}$$

Ex: 5.16 For all $x \ge 2$ prove that $\int_x^{x+1} \frac{dy}{y^2} \ge \frac{2}{3x^2}$.

Show that $\sum_q \frac{1}{q^2} \le \frac{3}{2}\int_2^{\infty} \frac{dy}{y^2} = \frac{3}{4}$.

Proof

① $\int_x^{x+1} \frac{1}{y^2}\,dy = \left[\frac{-1}{y}\right]_x^{x+1} = \frac{1}{x} - \frac{1}{x+1} = \frac{1}{x(x+1)}$

Define, $f(x) = \frac{1}{x(x+1)} - \frac{2}{3x^2}$

$f(2) = \frac{1}{6} - \frac{2}{12} = 0$

$f'(x) = \frac{-(2x+1)}{x^2(x+1)^2} + \frac{2}{3x} = \frac{-6x - 3 + 2x(x+1)^2}{3x^2(x+1)^2}$

$x \ge 2 \Longrightarrow 2x(x+1)^2 \ge 36$

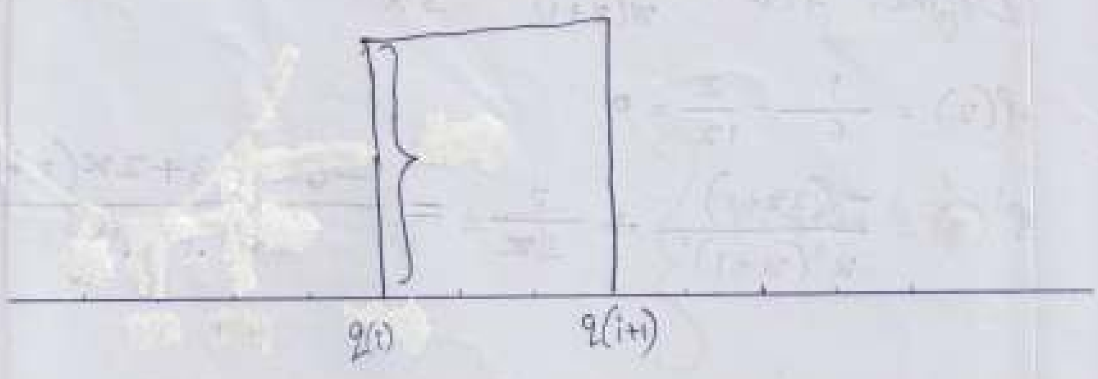
$$-(6x+3) \ge -15$$

$$-6x-3+2x(x+1)^2 \ge 21$$

$\Rightarrow f'(x)$ is +ve for all $x \ge 2$

$$\therefore \int_x^{x+1} \frac{1}{y^2}\,dy \ge \frac{2}{3x^2} \quad \forall\, x \ge 2$$

(2) $$\frac{1}{x^2} \le \frac{3}{2}\int_x^{x+1} \frac{dy}{y^2} \Longrightarrow \boxed{\frac{1}{q^2(i)} \le \frac{3}{2}\int_{q(i)}^{q(i+1)} \frac{dy}{y^2}}$$



$$\frac{1}{q^2(1)} + \frac{1}{q^2(2)} + \cdots \le \frac{3}{2}\int_{q(1)}^{q(2)} \frac{dy}{y^2} + \frac{3}{2}\int_{q(2)}^{q(3)} \frac{dy}{y^2} + \cdots$$

$$\sum_q \frac{1}{q^2} \le \frac{3}{2}\int_2^{\infty} \frac{dy}{y^2} = \frac{3}{2}\left[\frac{-1}{y}\right]_2^{\infty}$$

$$= \frac{3}{2}\left[0 - \frac{1}{2}\right] = \frac{3}{4}$$

$\therefore$

$$\sum_q \frac{1}{q^2} \le \frac{3}{2}\int_2^{\infty} \frac{dy}{y^2} = \frac{3}{4}$$

$$\Bigg\} \; 1 - \sum_q q^2 \ge 1 - \frac{3}{4} = \frac{1}{4}$$

$P(s_1 \,\&\, s_2 \text{ have no common factors}) =$

$$= 1 - P(s_1 \,\&\, s_2 \text{ have common factors})$$

$$= 1 - \sum_q P(q|s_1)\,P(q|s_2)$$

$$\ge 1 - \sum_q \frac{1}{q^2} \ge \frac{1}{4}$$

Period - finding

Suppose f is a periodic function producing a single bit as output and such that $f(x+r) = f(x)$ for some unknown $0 < r < 2^L$, where $x, r \in \{0, 1, 2, \ldots\}$.

Given a quantum black box U which performs the unitary transform

$$U|x\rangle|y\rangle \longrightarrow |x\rangle|y \oplus f(x)\rangle$$

$\oplus$: addition modulo 2.

In practice U operates on a finite domain, whose size is determined by the desired accuracy for r.

Algorithm : Period-finding

Inputs : ① A Black box which performs the operation

$U|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle$

② (b) state to store the function evaluation, initialized to $|0\rangle$

③ $\boxed{t = O\left(L + \ln(1/\epsilon)\right)}$ qubits initialized to $|0\rangle$

Outputs. The least integer $r > 0$ such that

$f(x+r) = f(x)$

Runtime : One use of U, and $O(L^2)$ operations succeeds with probability $O(1)$.

Procedure :

① $|0\rangle|0\rangle$ — initial state

② $\rightarrow \frac{1}{\sqrt{2^t}} \sum_{x=0}^{2^t-1} |x\rangle|0\rangle$ — create superposition

③ $\longrightarrow \frac{1}{\sqrt{2^t}} \sum_{x=0}^{2^t-1} |x\rangle |f(x)\rangle$ apply U

$$\approx \frac{1}{\sqrt{r 2^t}} \sum_{l=0}^{r-1} \sum_{x=0}^{2^t-1} e^{2\pi i l x/r} |x\rangle |\hat{f}(l)\rangle$$

④ $\longrightarrow \frac{1}{\sqrt{r}} \sum_{l=0}^{r-1} |\widetilde{l/r}\rangle |\hat{f}(l)\rangle$ apply inverse Fourier transform to 1st register

⑤ $\longrightarrow \widetilde{l/r}$ measure first register

⑥ $\longrightarrow r$ apply continued fraction algorithm.

The uniform superposition is given by,

$$H^{\otimes t}|0^t\rangle = \frac{1}{\sqrt{2^t}}\sum_{x=0}^{2^t-1}|x\rangle$$

– We initialize our system in the state

$$|\psi_0\rangle = |0^t\rangle|0^q\rangle$$

Next, we create a uniform superposition over $\{0, 1, 2, \ldots, 2^t - 1\}$ in the 1^{st} register

$$|0\rangle|0\rangle \xrightarrow{H^{\otimes t}} \frac{1}{\sqrt{2^t}} \sum_{x=0}^{2^t-1} |x\rangle|0\rangle = |\psi_2\rangle$$

Classically, we could solve this period finding problem by querying our function with subsequent inputs until the function repeats. This takes $O(r) = O(2^t)$ queries to the function. With a Quantum computer, we can access the function in superposition to query the function with $N = 2^t$ inputs for each t qubits at the same time.

Let U be the unitary transformation that carries out our function, and implement it:

$$\frac{1}{\sqrt{2^t}} \sum_{x=0}^{2^t-1} |x\rangle|0\rangle \xrightarrow{U} \frac{1}{\sqrt{2^t}} \sum_{x=0}^{2^t-1} |x\rangle|f(x)\rangle = |\psi_3\rangle$$

For analysis purpose, we create a modified Fourier transform that makes sense for our periodic function. Specifically, if we restrict ourselves to the domain $x \in \{0, 1, \ldots, r-1\}$, we can use $|f(x)\rangle$ rather than $|x\rangle$ as our basis states. This is because f is not permitted to have any duplicate values each part of the domain.

The modified Fourier transform is now defined as:

$$|\hat{f}(l)\rangle = \frac{1}{\sqrt{r}} \sum_{x=0}^{r-1} e^{-2\pi i l x / r} |f(x)\rangle$$

which is defined for $0 \le l < r$.

$$\Rightarrow |f(x)\rangle = \frac{1}{\sqrt{r}} \sum_{l=0}^{r-1} e^{2\pi i l x / r} |\hat{f}(l)\rangle$$

$$\therefore |\psi_3\rangle = \frac{1}{\sqrt{2^t}} \sum_{x=0}^{2^t-1} |x\rangle \left(\frac{1}{\sqrt{r}} \sum_{l=0}^{r-1} e^{2\pi i l x / r} |\hat{f}(l)\rangle \right)$$

$$= \frac{1}{\sqrt{r 2^t}} \sum_{l=0}^{r-1} \sum_{x=0}^{2^t-1} e^{2\pi i l x / r} |x\rangle |\hat{f}(l)\rangle$$

$$= \frac{1}{\sqrt{r 2^t}} \left[\sum_{x=0}^{2^t-1} e^{2\pi i x (0/r)} |x\rangle |\hat{f}(0)\rangle + \sum_{x=0}^{2^t-1} e^{2\pi i x (1/r)} |x\rangle |\hat{f}(1)\rangle + \cdots + \sum_{x=0}^{2^t-1} e^{2\pi i x \left(\frac{r-1}{r}\right)} |x\rangle |\hat{f}(r-1)\rangle \right]$$

$$\text{QFT}\,|j\rangle = \frac{1}{2^{t/2}} \sum_{k=0}^{2^t-1} e^{2\pi i jk/2^t} |k\rangle$$

$$\longrightarrow (\text{QFT})^{\dagger} \frac{1}{\sqrt{2^t}} \sum_{x=0}^{2^t-1} e^{2\pi i x (l/r)} |x\rangle = \left|\frac{2^t l}{r}\right\rangle$$

$$(\text{QFT})^{\dagger} |\psi_3\rangle = \frac{1}{\sqrt{r}} \left(\left|\frac{2^t 0}{r}\right\rangle |\hat{f}(0)\rangle + \left|\frac{2^t 1}{r}\right\rangle |\hat{f}(1)\rangle + \cdots \cdots + \left|\frac{2^t (r-1)}{r}\right\rangle |\hat{f}(r-1)\rangle \right)$$

$$= \frac{1}{\sqrt{r}} \left(\left|\widetilde{\frac{0}{r}}\right\rangle |\hat{f}(0)\rangle + \left|\widetilde{\frac{1}{r}}\right\rangle |\hat{f}(1)\rangle + \cdots \cdots + \left|\widetilde{\frac{r-1}{r}}\right\rangle |\hat{f}(r-1)\rangle \right)$$

$\therefore$

Applying the inverse Fourier transform to the 1st register, in step 4, gives an estimate of the phase l/r, where l is chosen randomly. r can be obtained efficiently in the final step using a continued fraction expansion.

Note

Step 3:

$$\frac{1}{\sqrt{r2^t}}\sum_{l=0}^{r-1}\sum_{x=0}^{2^t-1} e^{2\pi i l x/r}|x\rangle|\hat{f}(l)\rangle =$$

$$= \frac{1}{\sqrt{r}}\sum_{l=0}^{r-1}\left(\frac{1}{\sqrt{2^t}}\sum_{x=0}^{2^t-1} e^{2\pi i x\left(\frac{l2^t}{r}\right)/2^t}|x\rangle\right)|\hat{f}(l)\rangle$$

$$FT|j\rangle = \frac{1}{\sqrt{2^t}}\sum_{x=0}^{2^t-1} e^{2\pi i j x/2^t}|x\rangle$$

j in the exponent's power is an integer, and j in $|j\rangle$ is a binary representation of an integer j by a quantum state.

$\therefore$

(i) If $\left(\frac{l2^t}{r}\right)$ is integer, i.e.,

if 2^t is an integer multiple of r

then the expression inside round brackets is exactly the QFT of the state $\left|\frac{l2^t}{r}\right\rangle$.

$$\Rightarrow \frac{1}{\sqrt{2^t}}\sum_{x=0}^{2^t-1} e^{2\pi i x\left(l2^t/r\right)/2^t}|x\rangle = FT\left(|l2^t/r\rangle\right)$$

(ii) If $\frac{l2^t}{r}$ is not an integer

i.e., if 2^t is not an integer multiple of r

$$\frac{1}{\sqrt{2^t}}\sum_{x=0}^{2^t-1} e^{2\pi i (l2^t/r)/2^t}|x\rangle \neq FT\left(|l2^t/r\rangle\right)$$

since for rational $l2^t/r$ there is no integer binary representation, and thus, no quantum state $|l2^t/r\rangle$.

In this case, what we get from $\left|\frac{l2^t}{r}\right\rangle = \left|\frac{\tilde{l}}{r}\right\rangle$ in step 4 is only an approximation.

- The order of an element 'a' in the group $(\mathbb{Z}_p^*, \cdot)$ is the smallest +ve integer r such that $a^r \equiv 1 \pmod{p}$.

The order finding can be seen as an instance of period finding for the function $f_a(s) = a^s \bmod p$, since the period of the function is exactly the order,

i.e.,

$$f_a(s+r) = a^{s+r} \bmod p = a^s a^r \bmod p$$
$$= a^s \bmod p = f(s)$$