COL863: Quantum Computation and Information

Ragesh Jaiswal, CSE, IIT Delhi

Quantum Computation: Order finding

- Given integers $N > x > 0$ such that x and N have no common factors, the order of x modulo N is defined to be the least positive integer r such that $x^r = 1 \ (mod \ N)$.
- Exercise: What is the order of 5 modulo 21?

- Given integers $N > x > 0$ such that x and N have no common factors, the order of x modulo N is defined to be the least positive integer r such that $x^r = 1 \ (mod \ N)$.
- Exercise: What is the order of 5 modulo 21? 6

Order finding

Given co-prime integers $N > x > 0$, compute the order of x modulo N.

- Exercise: Is there an algorithm that computes the order of x modulo N in time that is polynomial in N?

- Given integers $N > x > 0$ such that x and N have no common factors, the order of x modulo N is defined to be the least positive integer r such that $x^r = 1 \ (mod \ N)$.
- Exercise: What is the order of 5 modulo 21? 6

Order finding

Given co-prime integers $N > x > 0$, compute the order of x modulo N.

- Exercise: Is there an algorithm that computes the order of x modulo N in time that is polynomial in N? Yes
- Exercise: Is it an efficient algorithm?

- Given integers $N > x > 0$ such that x and N have no common factors, the order of x modulo N is defined to be the least positive integer r such that $x^r = 1 \ (mod \ N)$.
- Exercise: What is the order of 5 modulo 21? 6

Order finding

Given co-prime integers $N > x > 0$, compute the order of x modulo N.

- Exercise: Is there an algorithm that computes the order of x modulo N in time that is polynomial in N? Yes
- Exercise: Is it an efficient algorithm?
- Let $L = \lceil \log n \rceil$. The number of bits needed to specify the problem is $O(L)$. So, an efficient algorithm should have running time that is polynomial in L.

Order finding

Given co-prime integers $N > x > 0$, compute the order of x modulo N.

- Consider the operator U that has the following behaviour:

$$U|y\rangle \equiv \begin{cases} |xy \ (mod \ N)\rangle & \text{if } 0 \leq y \leq N-1 \\ |y\rangle & \text{if } N \leq y \leq 2^L - 1 \end{cases}$$

- Exercise: Show that U is unitary.

Order finding

Given co-prime integers $N > x > 0$, compute the order of x modulo N.

- Consider the operator U that has the following behaviour:

$$U|y\rangle \equiv \begin{cases} |xy \ (mod \ N)\rangle & \text{if } 0 \leq y \leq N-1 \\ |y\rangle & \text{if } N \leq y \leq 2^L - 1 \end{cases}$$

- Exercise: Show that U is unitary.
- Exercise: Show that the states defined by

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

are the eigenstates of U. Find the corresponding eigenvalues.

Order finding

Given co-prime integers $N > x > 0$, compute the order of x modulo N.

- Consider the operator U that has the following behaviour:

$$U\left|y\right\rangle \equiv \begin{cases} \left|xy \ (mod \ N)\right\rangle & \text{if } 0 \leq y \leq N-1 \\ \left|y\right\rangle & \text{if } N \leq y \leq 2^L - 1 \end{cases}$$

- Exercise summary: Let $\left|u_s\right\rangle \equiv \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-(2\pi i)\frac{sk}{r}} \left|x^k \ (mod \ N)\right\rangle$ be an eigenstate of U. Then $U\left|u_s\right\rangle = e^{(2\pi i)\frac{s}{r}}\left|u_s\right\rangle$

Order finding

Given co-prime integers $N > x > 0$, compute the order of x modulo N.

- Consider the operator U that has the following behaviour:

$$U|y\rangle \equiv \begin{cases} |xy \ (mod \ N)\rangle & \text{if } 0 \leq y \leq N-1 \\ |y\rangle & \text{if } N \leq y \leq 2^L - 1 \end{cases}$$

- Exercise summary: Let $|u_s\rangle \equiv \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-(2\pi i)\frac{sk}{r}} \left|x^k \ (mod \ N)\right\rangle$ be an eigenstate of U. Then $U|u_s\rangle = e^{(2\pi i)\frac{s}{r}}|u_s\rangle$
- Main idea for determining r: We will use phase estimation to get an estimate on $\frac{s}{r}$ and then obtain r from it.

Order finding

Given co-prime integers $N > x > 0$, compute the order of x modulo N.

- Consider the operator U that has the following behaviour:

$$U|y\rangle \equiv \begin{cases} |xy \ (mod \ N)\rangle & \text{if } 0 \leq y \leq N-1 \\ |y\rangle & \text{if } N \leq y \leq 2^L - 1 \end{cases}$$

- Exercise summary: Let $|u_s\rangle \equiv \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-(2\pi i)\frac{sk}{r}} \left|x^k \ (mod \ N)\right\rangle$ be an eigenstate of U. Then $U|u_s\rangle = e^{(2\pi i)\frac{s}{r}} |u_s\rangle$
- Main idea for determining r: We will use phase estimation to get an estimate on $\frac{s}{r}$ and then obtain r from it.
  - How do we implement controlled U^{2^j}?
  - How do we prepare an eigenstate $|u_s\rangle$?

Order finding

Given co-prime integers $N > x > 0$, compute the order of x modulo N.

- Consider the operator U that has the following behaviour:

$$U|y\rangle \equiv \begin{cases} |xy \ (mod \ N)\rangle & \text{if } 0 \leq y \leq N-1 \\ |y\rangle & \text{if } N \leq y \leq 2^L - 1 \end{cases}$$

- Exercise summary: Let $|u_s\rangle \equiv \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-(2\pi i)\frac{sk}{r}} \left|x^k \ (mod \ N)\right\rangle$ be an eigenstate of U. Then $U|u_s\rangle = e^{(2\pi i)\frac{s}{r}}|u_s\rangle$
- Main idea for determining r: We will use phase estimation to get an estimate on $\frac{s}{r}$ and then obtain r from it.
  - How do we implement controlled U^{2^j}? Modular exponentiation
  - How do we prepare an eigenstate $|u_s\rangle$?

Modular exponentiation

Given $|z\rangle |y\rangle$, design a circuit that ends in the state $|z\rangle |x^z y \ (mod \ N)\rangle$.

- What we wanted to do was $|z\rangle |y\rangle \rightarrow |z\rangle U^{z_t 2^{t-1}} ... U^{z_1 2^0} |y\rangle$ but then this is the same as $|z\rangle |x^z y \ (mod \ N)\rangle$.
- Question: Suppose we work with the first register being of size $t = 2L + 1 + \lceil \log \left(2 + \frac{1}{2\varepsilon}\right) \rceil = O(L)$. What would be the size of the circuit?

Modular exponentiation

Given $|z\rangle|y\rangle$, design a circuit that ends in the state $|z\rangle|x^z y \ (mod \ N)\rangle$.

- What we wanted to do was $|z\rangle|y\rangle \rightarrow |z\rangle U^{z_t 2^{t-1}} \dots U^{z_1 2^0}|y\rangle$ but then this is the same as $|z\rangle|x^z y \ (mod \ N)\rangle$.
- Question: Suppose we work with the first register being of size $t = 2L + 1 + \lceil \log(2 + \frac{1}{2\varepsilon}) \rceil = O(L)$. What would be the size of the circuit? $O(L^3)$

Order finding

Given co-prime integers $N > x > 0$, compute the order of x modulo N.

- Consider the operator U that has the following behaviour:

$$U|y\rangle \equiv \begin{cases} |xy \ (mod \ N)\rangle & \text{if } 0 \leq y \leq N-1 \\ |y\rangle & \text{if } N \leq y \leq 2^L - 1 \end{cases}$$

- Exercise summary: Let $|u_s\rangle \equiv \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-(2\pi i)\frac{sk}{r}} \left|x^k \ (mod \ N)\right\rangle$ be an eigenstate of U. Then $U|u_s\rangle = e^{(2\pi i)\frac{s}{r}} |u_s\rangle$
- Main idea for determining r: We will use phase estimation to get an estimate on $\frac{s}{r}$ and then obtain r from it.
  - How do we implement controlled U^{2^j}? Modular exponentiation
  - How do we prepare an eigenstate $|u_s\rangle$?
    - We work with $|1\rangle$ as the first register since $\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle = |1\rangle$.

Quantum Computation

Phase estimation → Order-finding

Order finding

Given co-prime integers $N > x > 0$, compute the order of x modulo N.

- Consider the operator U that has the following behaviour:

$$U\left|y\right\rangle \equiv \begin{cases} \left|xy \ (mod \ N)\right\rangle & \text{if } 0 \leq y \leq N-1 \\ \left|y\right\rangle & \text{if } N \leq y \leq 2^L - 1 \end{cases}$$

- Exercise summary: Let $\left|u_s\right\rangle \equiv \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-(2\pi i)\frac{sk}{r}} \left|x^k \ (mod \ N)\right\rangle$ be an eigenstate of U. Then $U\left|u_s\right\rangle = e^{(2\pi i)\frac{s}{r}} \left|u_s\right\rangle$
- Main idea for determining r: We will use phase estimation to get an estimate on $\frac{s}{r}$ and then obtain r from it.
  - How do we implement controlled U^{2^j}? Modular exponentiation
  - How do we prepare an eigenstate $\left|u_s\right\rangle$?
    - We work with $\left|1\right\rangle$ as the first register since $\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \left|u_s\right\rangle = \left|1\right\rangle$.
- So, we will argue that for each $0 \leq s \leq r-1$, we will obtain an estimate of $\varphi \approx \frac{s}{r}$ accurate to $2L+1$ bits with probability at least $\frac{(1-\varepsilon)}{r}$.

Order finding

Given co-prime integers $N > x > 0$, compute the order of x modulo N.

- Consider the operator U that has the following behaviour:

$$U\left|y\right\rangle \equiv \begin{cases} \left|xy \ (mod \ N)\right\rangle & \text{if } 0 \leq y \leq N-1 \\ \left|y\right\rangle & \text{if } N \leq y \leq 2^L - 1 \end{cases}$$

- Exercise summary: Let $\left|u_s\right\rangle \equiv \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-(2\pi i)\frac{sk}{r}} \left|x^k \ (mod \ N)\right\rangle$ be an eigenstate of U. Then $U\left|u_s\right\rangle = e^{(2\pi i)\frac{s}{r}} \left|u_s\right\rangle$
- Main idea for determining r: We will use phase estimation to get an estimate on $\frac{s}{r}$ and then obtain r from it.
  - How do we implement controlled U^{2^j}? Modular exponentiation
  - How do we prepare an eigenstate $\left|u_s\right\rangle$?
    - We work with $\left|1\right\rangle$ as the first register since $\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \left|u_s\right\rangle = \left|1\right\rangle$.
- So, we will argue that for each $0 \leq s \leq r-1$, we will obtain an estimate of $\varphi \approx \frac{s}{r}$ accurate to $2L+1$ bits with probability at least $\frac{(1-\varepsilon)}{r}$.
  - Question: How do we extract r from this? Continued fractions

Continued fraction

A finite simple continued fraction is defined by a collection of positive integers $a_0, \ldots, a_N$:

$$[a_0, \ldots, a_N] \equiv a_0 + \cfrac{1}{a_1 + \frac{1}{a_2 + \frac{1}{\ldots + \frac{1}{a_N}}}}$$

The n^{th} convergent $(0 \leq n \leq N)$ of this continued fraction is defined to be $[a_0, \ldots, a_n]$.

- Theorem: Suppose $x \geq 1$ is a rational number. Then x has a representation as a continued fraction, $x = [a_0, \ldots, a_N]$. This may be found by the continued fraction algorithm.
- Exercise: Find the continued fraction expansion of $\frac{31}{13}$.
- Question: What is the running time for the continued fractions algorithm for any given rational number $\frac{p}{q} \geq 1$?

Continued fraction

A finite simple continued fraction is defined by a collection of positive integers $a_0, ..., a_N$:

$$[a_0, ..., a_N] \equiv a_0 + \cfrac{1}{a_1 + \frac{1}{a_2 + \frac{1}{... + \frac{1}{a_N}}}}$$

The n^{th} convergent ($0 \leq n \leq N$) of this continued fraction is defined to be $[a_0, ..., a_n]$.

- Question: What is the running time for the continued fractions algorithm for any given rational number $\frac{p}{q} \geq 1$?
- Theorem: Let $a_0, ..., a_N$ be a sequence of positive numbers. Then $[a_0, ..., a_n] = \frac{p_n}{q_n}$, where p_n and q_n are real numbers defined inductively by $p_0 \equiv 0$, $q_0 \equiv 1$, $p_1 \equiv 1 + a_0 a_1$, $q_1 \equiv a_1$, and for $2 \leq n \leq N$,

$$\begin{aligned} p_n &\equiv a_n p_{n-1} + p_{n-2} \\ q_n &\equiv a_n q_{n-1} + q_{n-2} \end{aligned}$$

  In the case when a_j are positive integers, so too are p_j and q_j and moreover $q_n p_{n-1} - p_n q_{n-1} = (-1)^n$ for $n \geq 1$ which implies that $gcd(p_n, q_n) = 1$.

Continued fraction

A finite simple continued fraction is defined by a collection of positive integers $a_0, ..., a_N$:

$$[a_0, ..., a_N] \equiv a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{... + \frac{1}{a_N}}}}$$

The n^{th} convergent ($0 \leq n \leq N$) of this continued fraction is defined to be $[a_0, ..., a_n]$.

- Question: What is the running time for the continued fractions algorithm for any given rational number $\frac{p}{q} \geq 1$?
  - Let $[a_0, ..., a_N] = \frac{p}{q} \geq 1$ with $L = \lceil \log p \rceil$ and let p_n, q_n be as defined in the theorem.
  - Observation: p_n, q_n are increasing with $p_n \geq 2p_{n-2}, q_n \geq 2q_{n-2}$.
- Theorem: Let $a_0, ..., a_N$ be a sequence of positive numbers. Then $[a_0, ..., a_n] = \frac{p_n}{q_n}$, where p_n and q_n are real numbers defined inductively by $p_0 \equiv 0$, $q_0 \equiv 1$, $p_1 \equiv 1 + a_0 a_1$, $q_1 \equiv a_1$, and for $2 \leq n \leq N$,

$$\begin{aligned} p_n &\equiv a_n p_{n-1} + p_{n-2} \\ q_n &\equiv a_n q_{n-1} + q_{n-2} \end{aligned}$$

In the case when a_j are positive integers, so too are p_j and q_j and moreover $q_n p_{n-1} - p_n q_{n-1} = (-1)^n$ for $n \geq 1$ which implies that $gcd(p_n, q_n) = 1$.

Quantum Computation
Digression: Continued fractions

Continued fraction

A finite simple continued fraction is defined by a collection of positive integers $a_0, ..., a_N$:

$$[a_0, ..., a_N] \equiv a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{... + \frac{1}{a_N}}}}$$

The n^{th} convergent $(0 \leq n \leq N)$ of this continued fraction is defined to be $[a_0, ..., a_n]$.

- Question: What is the running time for the continued fractions algorithm for any given rational number $\frac{p}{q} \geq 1$?
  - Let $[a_0, ..., a_N] = \frac{p}{q} \geq 1$ with $L = \lceil \log p \rceil$ and let p_n, q_n be as defined in the theorem.
  - Observation: p_n, q_n are increasing with $p_n \geq 2p_{n-2}, q_n \geq 2q_{n-2}$.
  - This implies that $2^{\lfloor N/2 \rfloor} \leq q \leq p$. So, $N = O(L)$ and the running time of algorithm is $O(L^3)$.
- Theorem: Let $a_0, ..., a_N$ be a sequence of positive numbers. Then $[a_0, ..., a_n] = \frac{p_n}{q_n}$, where p_n and q_n are real numbers defined inductively by $p_0 \equiv 0$, $q_0 \equiv 1$, $p_1 \equiv 1 + a_0 a_1$, $q_1 \equiv a_1$, and for $2 \leq n \leq N$,

$$p_n \equiv a_n p_{n-1} + p_{n-2}; \quad q_n \equiv a_n q_{n-1} + q_{n-2}$$

In the case when a_j are positive integers, so too are p_j and q_j and moreover $q_n p_{n-1} - p_n q_{n-1} = (-1)^n$ for $n \geq 1$ which implies that $gcd(p_n, q_n) = 1$.

Continued fraction

A finite simple continued fraction is defined by a collection of positive integers $a_0, ..., a_N$:

$$[a_0, ..., a_N] \equiv a_0 + \cfrac{1}{a_1 + \frac{1}{a_2 + \frac{1}{... + \frac{1}{a_N}}}}$$

The n^{th} convergent ($0 \leq n \leq N$) of this continued fraction is defined to be $[a_0, ..., a_n]$.

- Theorem: Let x be a rational number and suppose $\frac{p}{q}$ is a rational number such that $|\frac{p}{q} - x| \leq \frac{1}{2q^2}$. Then $\frac{p}{q}$ is a convergent of the continued fraction for x.

Quantum Computation

Digression: Continued fractions

Theorem

Let x be a rational number and suppose $\frac{p}{q}$ is a rational number such that $|\frac{p}{q} - x| \leq \frac{1}{2q^2}$. Then $\frac{p}{q}$ is a convergent of the continued fraction for x.

Proof sketch

- Let $\frac{p}{q} = [a_0, ..., a_n]$ and let p_j, q_j as defined in the previous theorem so that $\frac{p}{q} = \frac{p_n}{q_n}$.
- Define δ by the equation:

$$x \equiv \frac{p_n}{q_n} + \frac{\delta}{2q_n^2}, \text{so that } |\delta| \leq 1.$$

- Define λ by

$$\lambda \equiv 2\left(\frac{q_n p_{n-1} - p_n q_{n-1}}{\delta}\right) - \frac{q_{n-1}}{q_n}$$

Quantum Computation

Digression: Continued fractions

Theorem

Let x be a rational number and suppose $\frac{p}{q}$ is a rational number such that $|\frac{p}{q} - x| \leq \frac{1}{2q^2}$. Then $\frac{p}{q}$ is a convergent of the continued fraction for x.

Proof sketch

- Let $\frac{p}{q} = [a_0, ..., a_n]$ and let p_j, q_j as defined in the previous theorem so that $\frac{p}{q} = \frac{p_n}{q_n}$.
- Define δ by the equation: $x \equiv \frac{p_n}{q_n} + \frac{\delta}{2q_n^2}, \text{so that } |\delta| \leq 1.$
- Define λ by $\lambda \equiv 2\left(\frac{q_n p_{n-1} - p_n q_{n-1}}{\delta}\right) - \frac{q_{n-1}}{q_n}$
- Claim 1: $x = \frac{\lambda p_n + p_{n-1}}{\lambda q_n + q_{n-1}}$ and therefore $x = [a_0, ..., a_n, \lambda]$.

Theorem

Let x be a rational number and suppose $\frac{p}{q}$ is a rational number such that $|\frac{p}{q} - x| \leq \frac{1}{2q^2}$. Then $\frac{p}{q}$ is a convergent of the continued fraction for x.

Proof sketch

- Let $\frac{p}{q} = [a_0, ..., a_n]$ and let p_j, q_j as defined in the previous theorem so that $\frac{p}{q} = \frac{p_n}{q_n}$.
- Define δ by the equation: $x \equiv \frac{p_n}{q_n} + \frac{\delta}{2q_n^2}, \text{so that } |\delta| \leq 1.$
- Define λ by $\lambda \equiv 2\left(\frac{q_n p_{n-1} - p_n q_{n-1}}{\delta}\right) - \frac{q_{n-1}}{q_n}$
- Claim 1: $x = \frac{\lambda p_n + p_{n-1}}{\lambda q_n + q_{n-1}}$ and therefore $x = [a_0, ..., a_n, \lambda]$.
- Claim 2: $\lambda = \frac{2}{\delta} - \frac{q_{n-1}}{q_n} > 2 - 1 > 1$ which further implies that $\lambda = [b_0, ..., b_m]$ and $x = [a_0, ..., a_n, b_0, ..., b_m]$.
- This completes the proof of the theorem. ☐

Quantum Computation

Phase estimation → Order-finding

Order finding

Given co-prime integers $N > x > 0$, compute the order of x modulo N.

- Consider the operator U that has the following behaviour:

$$U|y\rangle \equiv \begin{cases} |xy \ (mod \ N)\rangle & \text{if } 0 \leq y \leq N-1 \\ |y\rangle & \text{if } N \leq y \leq 2^L - 1 \end{cases}$$

- Exercise summary: Let $|u_s\rangle \equiv \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-(2\pi i)\frac{sk}{r}} \left|x^k \ (mod \ N)\right\rangle$ be an eigenstate of U. Then $U|u_s\rangle = e^{(2\pi i)\frac{s}{r}} |u_s\rangle$
- Main idea for determining r: We will use phase estimation to get an estimate on $\frac{s}{r}$ and then obtain r from it.
  - How do we implement controlled U^{2^j}? Modular exponentiation
  - How do we prepare an eigenstate $|u_s\rangle$?
    - We work with $|1\rangle$ as the first register since $\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle = |1\rangle$.
- So, we will argue that for each $0 \leq s \leq r-1$, we will obtain an estimate of $\varphi \approx \frac{s}{r}$ accurate to $2L+1$ bits with probability at least $\frac{(1-\varepsilon)}{r}$.
  - Question: How do we extract r from this? Continued fractions
  - Question: Are we guaranteed to get r using continued fractions? What could go wrong?

Order finding

Given co-prime integers $N > x > 0$, compute the order of x modulo N.

- We obtain $\varphi \approx \frac{s}{r}$ for some $0 \leq s \leq r-1$ and then we use continued fractions to obtain s', r' such that $s'/r' = s/r$.
- The problem is r' may not equal r. One such case is when $s = 0$. This, however, is a small probability event.
- Claim: Suppose we repeat twice and obtain s'_1, r'_1 and s'_2, r'_2. If s_1 and s_2 are co-prime, then $r = lcm(r'_1, r'_2)$.

Order finding

Given co-prime integers $N > x > 0$, compute the order of x modulo N.

- We obtain $\varphi \approx \frac{s}{r}$ for some $0 \leq s \leq r-1$ and then we use continued fractions to obtain s', r' such that $s'/r' = s/r$.
- The problem is r' may not equal r. One such case is when $s = 0$. This, however, is a small probability event.
- Claim: Suppose we repeat twice and obtain r'_1 and r'_2 corresponding to s_1, s_2. If s_1 and s_2 are co-prime, then $r = lcm(r'_1, r'_2)$.
- Claim: $\mathbf{Pr}[s_1 \text{ and } s_2 \text{ are co-prime}] \geq 1/4$.

Quantum Computation

Phase estimation → Order-finding

Order finding

Given co-prime integers $N > x > 0$, compute the order of x modulo N.

Quantum Order-finding

1. $|0\rangle |1\rangle$ (Initial state)
2. $\rightarrow \frac{1}{2^{t/2}} \sum_{j=0}^{2^t-1} |j\rangle |1\rangle$ (Create superposition)
3. $\rightarrow \frac{1}{2^{t/2}} \sum_{j=0}^{2^t-1} |j\rangle \left|x^j \ (mod \ N)\right\rangle$ (Apply $U_{x,N}$)

   $\approx \frac{1}{\sqrt{r}2^{t/2}} \sum_{s=0}^{r-1} \sum_{j=0}^{2^t-1} e^{(2\pi i)\frac{sj}{r}} |j\rangle |u_s\rangle$
4. $\rightarrow \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \left|\widetilde{(s/r)}\right\rangle |u_s\rangle$ (Apply inverse FT to 1^{st} register)
5. $\rightarrow \widetilde{(s/r)}$ (Measure first register)
6. $\rightarrow r$ (Use continued fractions algorithm)

- What is the size of the circuit that computes the order with high probability?

Quantum Computation

Phase estimation → Order-finding

Order finding

Given co-prime integers $N > x > 0$, compute the order of x modulo N.

Quantum Order-finding

1. $|0\rangle |1\rangle$ (Initial state)
2. $\rightarrow \frac{1}{2^{t/2}} \sum_{j=0}^{2^t-1} |j\rangle |1\rangle$ (Create superposition)
3. $\rightarrow \frac{1}{2^{t/2}} \sum_{j=0}^{2^t-1} |j\rangle \left|x^j \ (mod \ N)\right\rangle$ (Apply $U_{x,N}$)
   $\approx \frac{1}{\sqrt{r}2^{t/2}} \sum_{s=0}^{r-1} \sum_{j=0}^{2^t-1} e^{(2\pi i)\frac{sj}{r}} |j\rangle |u_s\rangle$
4. $\rightarrow \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \left|\widetilde{(s/r)}\right\rangle |u_s\rangle$ (Apply inverse FT to 1^{st} register)
5. $\rightarrow \widetilde{(s/r)}$ (Measure first register)
6. $\rightarrow r$ (Use continued fractions algorithm)

- What is the size of the circuit that computes the order with high probability? $O(L^3)$

End