COL863: Quantum Computation and Information

Ragesh Jaiswal, CSE, IIT Delhi

Quantum Computation: Factoring

Factoring

Given a positive composite integer N, output a non-trivial factor of N.

- We will solve the factoring problem by reduction to the order finding problem.
- Theorem 1: Suppose N is an L bit composite number, and x is a non-trivial solution to the equation $x^2 = 1 \ (mod \ N)$ in the range $1 \leq x \leq N$, that is, neither $x = 1 \ (mod \ N)$ nor $x = -1 \ (mod \ N)$. Then at least one of $gcd(x-1, N)$ and $gcd(x+1, N)$ is a non-trivial factor of N that can be computed using $O(L^3)$ operations.
- Theorem 2: Suppose $N = p_1^{\alpha_1} ... p_m^{\alpha_m}$ is the prime factorisation of an odd composite positive integer. Let x be an integer chosen uniformly at random, subject to the requirement that $1 \leq x \leq N-1$ and x is co-prime to N. Let r be the order of x modulo N. Then

$$\mathbf{Pr}[r \text{ is even and } x^{r/2} \neq -1 \ (mod \ N)] \geq 1 - \frac{1}{2^m}.$$

Factoring

Given a positive composite integer N, output a non-trivial factor of N.

Quantum Factoring Algorithm

1. If N is even, return 2 as a factor.
2. Determine if $N = a^b$ for integers $a, b \geq 2$ and if so, return a.
3. Randomly choose $1 \leq x \leq N - 1$. If $gcd(x, N) > 1$, then return $gcd(x, N)$.
4. Use the Quantum order-finding algorithm to find the order r of x modulo N.
5. If r is even and $x^{r/2} \neq -1 \ (mod \ N)$, then compute $p = gcd(x^{r/2} - 1, N)$ and $q = gcd(x^{r/2} + 1, N)$. If either p or q is a non-trivial factor of N, then return that factor else return "Failure".

Quantum Computation: Period finding

Quantum Computation

Phase estimation → Period finding

Period finding problem

Given a boolean function f such that $f(x) = f(x + r)$ for some unknown $0 < r < 2^L$, where $x, r = \{0, 1, 2, ...\}$ and given a unitary transform U_f that performs the transformation $U|x\rangle|y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle$, determine the least such $r > 0$.

Period-finding algorithm

1. $|0\rangle|0\rangle$ (Initial state)
2. $\rightarrow \frac{1}{2^{t/2}}\sum_{x=0}^{2^t-1}|x\rangle|0\rangle$ (Create superposition)
3. $\rightarrow \frac{1}{2^{t/2}}\sum_{x=0}^{2^t-1}|x\rangle|f(x)\rangle$ (Apply U)

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

Quantum Computation

Phase estimation → Period finding

Period finding problem

Given a boolean function f such that $f(x) = f(x + r)$ for some unknown $0 < r < 2^L$, where $x, r = \{0, 1, 2, ...\}$ and given a unitary transform U_f that performs the transformation $U\left|x\right\rangle\left|y\right\rangle \rightarrow \left|x\right\rangle\left|y \oplus f(x)\right\rangle$, determine the least such $r > 0$.

Period-finding algorithm

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

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

- Claim 1: Let $\left|\hat{f}(\ell)\right\rangle \equiv \frac{1}{\sqrt{r}}\sum_{x=0}^{r-1} e^{-(2\pi i)\frac{\ell x}{r}}\left|f(x)\right\rangle$. Then $\left|f(x)\right\rangle = \frac{1}{\sqrt{r}}\sum_{\ell=0}^{r-1} e^{(2\pi i)\frac{\ell x}{r}}\left|\hat{f}(\ell)\right\rangle$.

- The basic ideas involved in order finding and period finding seems to be the same.
- Question: *Can we generalise the core ideas and design a canonical algorithm for a very general problem so that order finding, factoring, period finding etc. are just special cases of this general problem?*
  - Yes. The general problem is called the Hidden Subgroup Problem.
- Before we see the hidden subgroup problem, we will see another special case: Discrete Logarithm.

Quantum Computation: Discrete logarithm

Discrete logarithm problem

Given positive integers a, b, N such that $b = a^s \ (mod\ N)$ for some unknown s, find s.

- Question: What is the running time of the naive classical algorithm?

Discrete logarithm problem

Given positive integers a, b, N such that $b = a^s \ (mod\ N)$ for some unknown s, find s.

- Question: What is the running time of the naive classical algorithm? $\Omega(N)$

Discrete logarithm problem

Given positive integers a, b, N such that $b = a^s \ (mod \ N)$ for some unknown s, find s.

- Consider a bi-variate function $f(x_1, x_2) = a^{sx_1 + x_2} \ (mod \ N)$.
- Claim 1: f is a periodic function with period $(\ell, -\ell s)$ for any integer ℓ.
    - So it may be possible for us to pull out s using some of the previous ideas developed.
- Question: How does discovering s for the above function help us in solving the discrete logarithm problem?

Discrete logarithm problem

Given positive integers a, b, N such that $b = a^s \ (mod \ N)$ for some unknown s, find s.

- Consider a bi-variate function $f(x_1, x_2) = a^{sx_1 + x_2} \ (mod \ N)$.
- Claim 1: f is a periodic function with period $(\ell, -\ell s)$ for any integer ℓ.
    - So it may be possible for us to pull out s using some of the previous ideas developed.
- Question: How does discovering s for the above function help us in solving the discrete logarithm problem?
    - Main idea: $f(x_1, x_2) \equiv b^{x_1} a^{x_2} \ (mod \ N)$.

End