Algorithms Analysis and Design

Week 12 - Diary

Ayan Agrawal (2020101034)

Lecture 18 : Shor's Algorithm

Problems for the class :

We discussed about the Shor's algorithm which shows that a quantum computer could be used to factor a number n in polynomial time, thus effectively breaking RSA.

An efficient Quantum Algorithm for Integer Factorization :

We have already studied FFT where we learnt how to factorise a number. Here we will discuss a quantum algorithm to do the same.

Step 1 : Factoring and finding a nontrivial square root of $1\%N$.

Lemma: *If x is a non-trivial square root of $1\%N$, then $gcd(N, x+1)$ is a non-trivial factor.*

<u>Proof</u> :

Since, $x^2\%N = 1 \implies x^2 - 1$ is multiple of N.

$\implies (x+1)(x-1) \equiv 0\%N$, but since x is a non-trivial sqrt of $1\%N$.

$\implies x \neq \pm 1 mod N$

$\therefore N$ must have a non-trivial factor common with $x-1, x+1$. These factors can be given by $gcd(N, x+1)$ and $gcd(N, x-1)$. Though, for the reduction to go through, we only need one of these, say $x+1$.

Step 2 : Reducing non-trivial square root of 1 to computing the order mod N

Order of a number x is defined as the smallest positive number k such that $x^k\%N = 1$.

Step 3 : The order of an integer is precisely the period of a particular periodic superposition.

First, we need to find a periodic function $f(a)$ whose period is equal to the degree of x, $f(a) = x^a(\%N)$. This function is periodic with a period of r, r is the degree of x.

Quantum superpositions that are periodically different from 0 can be established only if the period is an integer same as the period of the function for all periodic functions.

Step 4 : Quantum Fourier Transform (QFT)

In polynomial time complexity, a quantum computer may apply the unitarised fourier transform matrix of a state vector :

$$|\alpha\rangle = \sum_{j=0}^{M/k-1} \sqrt{\tfrac{k}{M}}\, |jk\rangle$$

Now, If β is the fourier transform of α, such that $|\beta>= (\beta_0, \beta_1 \ldots \beta_{M-1})$, then

$$|\beta\rangle = \frac{1}{\sqrt{k}} \sum_{j=0}^{k-1} \left|\frac{jM}{k}\right\rangle$$

Now if s independent samples are drawn uniformly from $0, M/k, 2M/k, \ldots, (k-1)M/k$, then the G.C.D of these samples is M/k with probability of atleast $1 - k/2^s$.

In an efficient QFT, we take help of FFT. In FFT, we found the factors in $O(nlog(n))$. To find the factors in $O(logn)$, we superimpose the 2 DFT steps and do a H-transform on the last bit. Then we apply some unitary transformations and shift by ω^j by using a unitary matrix. Finally, in $O(log^2 M)$ quantum operations, we can perform the QFT.

Complete Code :

- **Input : an odd composite number N**
  Output : a factor of N

1. First we randomly choose a number x, such that $x \in [1, N)$.
2. Let M be a power of 2 near N.
3. Repeat this step $t = 2 \times log(N)$ times:

> Start with two quantum registers, both initially 0, the first large enough to store a number modulo M and the second modulo N.
> Compute $f(a) = x^a \bmod N$ using a quantum circuit, to get the superposition $\sum_{a=0}^{M-1} \frac{1}{\sqrt{M}} |a, x^a \bmod N\rangle$.
> Measure the second register. Now the first register contains the periodic superposition $|\alpha\rangle = \sum_{j=0}^{M/r-1} \sqrt{\frac{r}{M}} |jr + k\rangle$ where k is a random offset between 0 and $r - 1$ (recall that r is the order of x modulo N).

After completing given step, take $g = gcd(j_1, j_2, j_3, \ldots j_t)$ where j_i is the index obtained in the i^{th} iteration of above step.

4. If M/g is even, we compute $gcd(N, x^{M/2g} + 1)$ and output it if it is a non-trivial factor of N, else we start again.