Shor's Algorithm P2

October 03, 2018

1 QFT

We now have two problems:

1. How do we do a Quantum Fourier Transform (QFT)?
2. How do we compute Shor's algorithm if we don't know M?

Theorem: Suppose $M = 2^m$, then there exists a recursive algorithm in terms of QFT mod 2^{m-1}

Recall: a fast Fourier transform (FFT) maps:

$$\begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \cdots \\ \alpha_{M-1} \end{pmatrix} \to \begin{pmatrix} \hat{\alpha}_0 \\ \hat{\alpha}_1 \\ \cdots \\ \hat{\alpha}_{M-1} \end{pmatrix}, \hat{\alpha}_y = \frac{1}{\sqrt{M}} \sum_{x=0}^{M-1} \alpha_x \omega_M^{xy}$$

Goal: use the classical algorithm to inspire a quantum algorithm.

Classical algorithm: Runs in $O(M \log M)$ time.

$$\hat{\alpha}_y = \frac{1}{\sqrt{\frac{M}{2}}} \left(\sum_{x=0}^{\frac{M}{2}-1} \alpha_{2x} \omega_{\frac{m}{2}}^{xy} \right) + \frac{1}{\sqrt{\frac{M}{2}}} \left(\sum_{x=0}^{\frac{M}{2}} \alpha_{2x+1} \omega_{\frac{m}{2}}^{xy} \right)$$

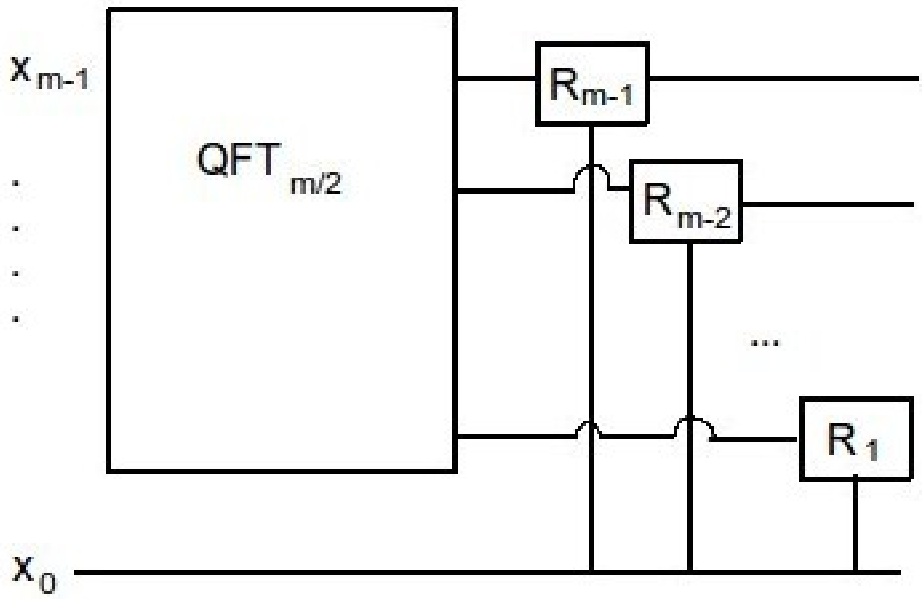
Observe that the left summation is the even terms and the right summation is the odd terms

1.1 QFT with known M

Let there be m qubits $x_{m-1}, x_{m-2}, ..., x_0$, from most significant to least significant.

We apply $\text{QFT}_{\frac{m}{2}}$ to $x_{m-1}...x_1$. Each output goes through a controlled phase gate with x_0. That is, when $x_0 = 0$, do not change the output. When $x_0 = 1$, apply $R_i = \begin{pmatrix} 1 & 0 \\ 0 & \omega_{\frac{m}{2^i}} \end{pmatrix}$. So $R_i |0\rangle = |0\rangle$ and $R_i |1\rangle = \omega_{\frac{m}{2^i}} |1\rangle$

We can illustrate this algorithm as the following circuit:

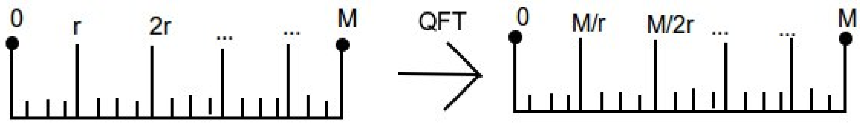


This solves problem 1!

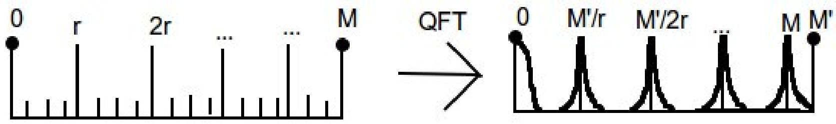
1.2 Unknown M

Now we need to solve problem 2. Solution: use M' that is a known power of 2.

Our QFT without using such an M' makes a transformation as below. The QFT takes a periodic spike and inverts the period.



After using an M' instead, the QFT still inverts the period, but with has a local region instead of single-point spikes, as below.



We need that $M' \approx M^2$ and that $M = N - 2\sqrt{N}$

The original spikes need to extend to the whole domain, $0 \to M'$, not just $0 \to M$

Problem 2 solved!

2 Applications

Shor's algorithm solves the hidden subgroup problem. That is, given $f : G \to S$, where G is an additive group, and given that there exists a subgroup $H \subset G$ where $\forall h \in H, f(x+h) = f(x)$ and $f(x) \neq f(y)$ if $x - y \notin H$, the problem is to find H.

Here, we show that many problems can be described as a hidden subgroup problem, meaning Shor's algorithm can solve them efficiently.

2.1 Simon's problem

Let $G = \mathbb{Z}_2^m$ and $H = \{0, S\}$ for $S \neq 0$. This is an instance of the hidden subgroup problem.

2.2 Factoring

Let $G = \mathbb{Z}_{\phi(N)}$ and $H = \{0, r, 2r, ...\}$, where $r|\phi(N)$. This is an instance of the hidden subgroup problem.

2.3 Discrete Logarithm

First we'll provide a recap of the discrete log problem: Fix some prime p. Given $g \in \mathbb{Z}_p^*, x \in \mathbb{Z}_{p-1}$, then computing $g^x (\text{mod } p)$ is easy, but computing x given $g, g^x (\text{mod } p)$ is hard. This works in any cyclic group.

Next we'll provide a recap of the Diffie-Hellman Key Exchange:

1. Alice chooses some x, and sends Bob $a = g^x$
2. Bob chooses some y, and sense Alice $b = g^y$
3. Alice computes a shared key $k = b^x$, and Bob computes a shared key $k' = a^y$
4. If all parties are honest, Alice and Bob compute $k = k' = g^{xy}$, and no other party can compute k

Now, using Shor's algorithm, we can solve the discrete logarithm problem.

Given $g, h = g^x$, we let $f(a,b) \to g^a h^{-b} (\text{mod } p)$. Let $G = Z_p^2$ and $H = \langle (x, 1) \rangle$. Then observe $f((a,b) + (x,1)) = g^{a+x} h^{-(b+1)} = g^a h^{-b} g^x h^{-1} = g^a h^{-b}$, so this is an instance of the hidden subgroup problem.