Quantum Algorithms, Spring 2022: Lecture 8 Scribe

Pahulpreet Singh   Shreyas Pradhan

February 04, 2022

1 Recap

1.1 Simon's Algorithm

Given a blackbox U_f, for a function f from n bit-strings to n bit-strings i.e $f : \{0,1\}^n \rightarrow \{0,1\}^n$, with the promise that f is $2-1$ function defined as follows:$\forall$ input $(x, y) \in \{0,1\}^n, f(x) = f(y), \iff x = y \oplus s$.

Here, $s \in (0,1)^n$ is secret string. How many queries to U_f are needed to find s?

1.1.1 Classical Algorithm

We proved that the complexity is $O(\sqrt{2^n})$. [Matching lower bound exists.]

1.1.2 Quantum Algorithm

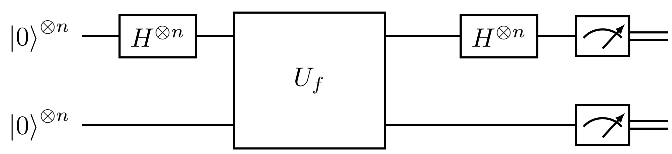

Figure 1: Quantum Implementation of Simon's Algorithm

After running the circuit operations we have:

- When $x \oplus y = 0^n$ (i.e. $s = 0^n$), the measurement results in each string $y \in \{0,1\}^n$ with probability $p_y = \frac{1}{2^n}$
- in the cas $x \oplus y = s$ (where $s \neq 0^n$), the probability to obtain each string $y \in \{0,1\}^n$ is given by

$$p_y = \begin{cases} 1/2^{n-1} & \text{if } y \cdot s \text{ is even} \\ 0 & \text{if } y \cdot s \text{ is odd} \end{cases}$$

Thus, in both cases, the measurement results is some string $y \in \{0,1\}^n$ that satisfies $s \cdot y = 0$, and the distribution is uniform over all of the strings that satisfy this constraint and this is enough information to determine s.

We repeat the above process $n-1$ times to get $n-1$ strings $y_1, y_2, \ldots, y_{n-1} \in \{0,1\}^n$, such that

$$\begin{cases} y_1 \cdot s & = 0 \\ y_2 \cdot s & = 0 \\ & \vdots \\ y_{n-1} \cdot s & = 0 \end{cases}$$

and we can use this set of linear equations to find s. We only get a unique non-zero solution s if $y_1, y_2, \ldots, y_{n-1} \in \{0,1\}^n$ are linearly independent (P $\geq 1/4$).

- Use Gaussian elimination to obtain s. $\big[$ Needs $O(n^3)$ time, but no additional queries to U_f $\big]$
- Check if $f(0\cdots0) = f(s)$. If yes, we are done. If not, repeat this whole procedure a few times.
- Repeating the process $O(1)$ times, i.e. $O(n)$ number of queries to U_f, are enough to guarantee a high success probability.

  For k runs, $\quad P(\text{success}) = 1 - (\frac{3}{4})^k = 1 - \epsilon \quad$ where $k \approxeq \dfrac{log(1/\epsilon)}{log(4/3)}$
- Query complexity: $O(n)$ $\quad$ Quantum lower bound: $\Omega(n)$

For an exact version of Simon's Algorithm, refer to **?**].

2 Q.F.T. and its applications

2.1 Elementary Concepts

2.1.1 Complex Roots of Unity

$$\omega^N = 1 \qquad\qquad \omega = e^{2\pi\iota/N}$$
$$1 + \omega + \omega^2 + \cdots + \omega^{N-1} = 0$$

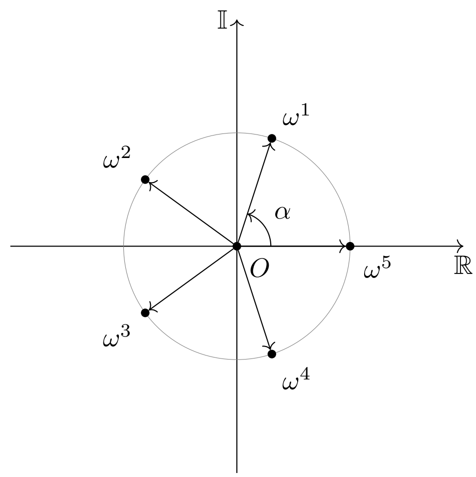

Figure 2: Roots of Unity for N=5

2.1.2 Discrete Fourier Transform

A complex vector $\big(x_0, x_1, \cdots, x_{N-1}\big)$ is transformed to another complex vector $\big(y_0, y_1, \cdots, y_{N-1}\big)$ such that:

$$y_k = \sum_{j=0}^{N-1} x_j \cdot \omega^{jk} \tag{1}$$

2.2 Quantum Fourier Transform

For Q.F.T. (Quantum Fourier Transform), the same transformation is carried out, except that it maps quantum states to quantum states.

Example Say $N = 2^n$ and $\{|0\rangle, |1\rangle, \cdots, |N-1\rangle\}$ are computational basis states. Then QFT on a state $|j\rangle$ is defined as:

$$|j\rangle \xmapsto{QFT} \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega^{jk} |k\rangle$$

where $\frac{1}{\sqrt{N}}$ is the normalization factor. Generalizing this to an ensemble of states,

$$\sum_{j=0}^{N-1} x_j |j\rangle \xmapsto{QFT} \sum_{k=0}^{N-1} y_k |k\rangle \tag{2}$$

$$\Big[\text{where } y_k = \sum_{j=0}^{N-1} x_j \frac{\omega^{jk}}{\sqrt{N}}\Big]$$

We define F_N as the unitary operator for QFT modulo N.

$$F_N |j\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega^{jk} |k\rangle$$

And the $(j,k)^{th}$ entry of the matrix is given by:

$$(F_N)_{j,k} = \langle k|F_N|j\rangle = \frac{\omega^{jk}}{\sqrt{N}}$$

From this, we can compute F_N as:

$$F_N = \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(N-1)} \\ \vdots & & & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \cdots & \omega^{(N-1)^2} \end{pmatrix}$$

1. Classically, for D.F.T., we need $O(N \log N)$ steps / operations to F.F.T. (Fast Fourier Transform).
2. For QFT, we shall require $O(\log^2 N)$ elementary gates.
3. Instead of a vector, QFT outputs a quantum state!
4. Classically, we have access to all the y_k in the vector. This is not the case for QFT.
5. At the end of QFT, we need to make a measurement. We observe some $|k\rangle$ with the probability $|y_k|^2$.
6. When we sample from the output state, we do so according to the Fourier Transform coefficients. (This is called Fourier Sampling)
7. So, we need to make clever use of QFT to harness its power for algorithmic applications!