Algebra and Computation Course Instructor: V. Arvind

Lecture 22: Simon's Problem and towards Shor

Lecturer: V. Arvind *Scribe: Ramprasad Saptharishi*

1 Overview

Last lecture we saw a toy problem that showed that the quantum model could be more efficient than the turing machine model, query complexity in the last example.

In this lecture, we shall inspect one more problem, which in a way is a true separation from the classical and quantum model since it beats even randomized algorithms on classical turing machines. The techniques used here will be essential in Shor's algorithms for integer factoring and discrete logarithm, whci hwe shall see in the following lectures.

2 Simon's Problem

We are given a function $f : \{0,1\}^n \to X$ with the promise that f is a 2-to-1 function (exactly two strings in $\{0,1\}^n$ mapping to an element of X). Further, f satisfies the additional property that there exists a $\lambda \neq 0^n$ such that $f(x \oplus \lambda) = f(x)$ for all x. The goal is to find the period λ with as few probes as possible.

We could make X canonical by enforcing an order on X. We can think of strings in $\{0,1\}^n$ as ordered pairs $\{(x, x \oplus \lambda)\}$ where x is lexicographically smaller than $x \oplus \lambda$. These pairs can now be identified with $\{0,1\}^{n-1}$ by just sorting the pairs, and hence fixes the set on the right. Thus, the number of such functions is exactly equal to the number of possible λs which is $2^n - 1$.

2.1 Lower Bounds on Classical Deterministic Computation

An adversial argument showsn an exponential lower bound for the query complexity in the determistic model. As an adversary, the point is to keep giving different answers for queries as long as it is possible. If k strings have been queried, the number of λs that we eliminate by giving different answers to each is $\binom{k}{2}$. Thus the adversary can go on giving different answers till $2^{n/2}$ queries have made.

Thus, the deterministic query complexity is atleast $2^{n/2}$.

2.2 Lower Bounds on Classical Randomized Computation

Suppose there was a procedure such that with just $q(n)$ queries the procedure will find λ with error probability atmost $1/3$. We can amplify this by repeating this experiment suitably many times and ge the error probability down to less than 2^{-n}.

Now we can use the same idea as we did in showing $\mathsf{BPP} \in \mathsf{P/poly}$. There are only $2^n - 1$ possible λs and the error probability is less than 2^{-n}. The randomized algorithms makes some random choices in the case of each λ. Suppose there was atleast 1 error in every possible λ, then the overall error probability would be more than 2^{-n}. Hence there has to be a sequence of choices such that it works for all possible values of λ.

But we have just shown earlier that the determistic computation has an exponential lowerbound, which gives a contradiction. Hence there is no probabilistic algorithm in the classical model that can find λ error bounded by a constant.

2.3 A Quantum Algorithm

Again, the hadamard code plays the major role here. Start with the uniform superposition

$$\begin{aligned} H_n |x\rangle &= \frac{1}{2^{n/2}} \sum_{u \in \{0,1\}^n} (-1)^{u \cdot x} |u\rangle \\ \implies H_n |0^m\rangle &= \frac{1}{2^{n/2}} \sum_{u \in \{0,1\}^n} |u\rangle \end{aligned}$$

By giving it input x and 0^m, it would return x and $0^m \oplus f(x) = f(x)$ on the other side.

Hence, if we make x the uniform superposition, on applying f we get:

$$|\psi\rangle = \frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} |x\rangle \otimes |0^m\rangle \longrightarrow \frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} |x\rangle \otimes |f(x)\rangle$$

Now on just measuring the qubit $|f(x)\rangle$, it would collapse to a uniformly random element in the image say $f(x_0)$. And on a measurement, since every other coordinate dies, the state would now collapse to:

$$|\psi\rangle = \frac{1}{\sqrt{2}} \left(|x_0\rangle + |x_0 + \lambda\rangle \right)$$

On applying H_n to this, we get

$$\begin{aligned}
|\psi\rangle &\rightarrow c\left(\sum_{u\in\{0,1\}^n}(-1)^{x_0\cdot u}|u\rangle+\sum_{u\in\{0,1\}^n}(-1)^{(x_0\oplus\lambda)\cdot u}|u\rangle\right)\\
&= c\sum_{u\in\{0,1\}^n}\left((-1)^{x_0\cdot u}+(-1)^{x_0\cdot u\oplus\lambda\cdot u}\right)|u\rangle\\
&= \sum_{u\cdot\lambda=0}\alpha_u|u\rangle \quad , \quad \alpha_u=\frac{1}{2^{(n-1)/2}}
\end{aligned}$$

But this is just a uniform superposition on the orthogonal space of the span of α, and thus measuring u would now give a random sampling in the orthogonal space. We will now show that once we have a random sampling of the space, we can get a basis.

Lemma 1. *Let G be a finite group. The probability that a uniform sample of size $4\log|G|$ generates G is at least $\frac{1}{3}$.*

Proof. Let $g_1, g_2, \cdots, g_m$ be the sample. Define G_i to be the group generated by $\{g_j\}_{j\le i}$. Let X be the indicator random variable that something bad happens:

$$X_i=\begin{cases}0 & \text{if } G_{i-1}=G \text{ or } g_i\notin G_{i-1}\\ 1 & \text{otherwise}\end{cases}$$

and let $X=\sum_{i=1}^m X_i$ then

$$\begin{aligned}
\Pr[X_i=0] &= \Pr[G_{i-1}=G]+\Pr[G_i\neq G]\Pr[g_i\notin G_{i-1}|G_{i-1}\neq G]\\
&= p_i+(1-p_i)\frac{1}{2}\\
&= \frac{1}{2}+\frac{p_i}{2}\geq\frac{1}{2}\\
\implies E[X_i] &= \Pr[X_i=1]\leq\frac{1}{2}\\
\implies E[X] &= \sum_{i=1}^m E[X_i]\leq\frac{m}{2}
\end{aligned}$$

By Markov's inequality,

$$\Pr[X\geq a]\leq\frac{E[x]}{a}\leq\frac{m}{2a}$$

Choosing $a=\frac{3m}{4}$ would get give $\Pr[X\geq\frac{3m}{4}]\leq\frac{2}{3}$.

But notice that if we haven't got a generating set already, $X_i = 0$ can happen atmost $\log |G|$ many times. Thus if we were to choose $m = 4 \log |G|$, at most $\log |G|$ of the X_i can be zero and hence will force $G_m = G$ with probability atleast $\frac{1}{3}$. □

Now that we have a generating set for the orthogonal set, all we need to do right now is solve a system of linear equations to get the orthogonal space of this, which is the space of λ. Thus we would find λ with just polynomially many queries since $\log |G| \leq n$.

The same idea can be used to solve a general problem as well. We are given a function f such that there is a subspace such that $f(x \oplus H) = f(x)$. The goal is to find a subspace. The same ideas can be used to do this as well. (exercise to the reader)

3 Towards Shor's Algorithms

The techniques used in the quantum solution to Simon's problem is essential for Shor's algorithms for integer factoring and the discrete logarithm problem. Shor's algorithm is a quantum algorithm for order finding (given a numbers a and n, find $ord_n(a)$). But the following lemma would show that this is good enough.

Lemma 2. *Order finding is as hard as integer factoring.*

Proof. We will show a probabilistic reduction from factoring to order finding. Without loss of generality, we can assume that n is odd. Pick an x at random from $\mathbb{Z}_n \setminus \{0\}$. We would be extremely lucky if $\gcd(x, n) \neq 1$, we then immediately have a non-trivial factor of n. Hence, we can assume that x is randomly picked from $\mathbb{Z}_n^\star$.

Suppose $ord_n(x)$ is even, and it further satisfies the property that $x^{ord_n(x)/2} \neq -1 \pmod{n}$, then we have non-trivial square root of unity and hence $\gcd(n, x^{ord_n(x)/2} - 1)$ or $\gcd(n, x^{ord_n(x)/2} + 1)$ will be non-trivial. This is the reduction. Once we show that we find a good x with high probability, we are done.

Assume that $n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k}$. By the chinese remaindering theorem,

$$\mathbb{Z}_n^\star = \mathbb{Z}_{p_1^{\alpha_1}}^\star \times \mathbb{Z}_{p_2^{\alpha_2}}^\star \times \cdots \mathbb{Z}_{p_k^{\alpha_k}}^\star$$

This map would take $x \mapsto (x_1, x_2, \cdots, x_k)$. Let $r = s2^t = ord_n(x)$ and $r_i = s_i 2^{t_i} = ord_{p_i^{\alpha_i}}(x_i)$. Clearly, $s = \text{lcm } s_i$ and $t = \max t_i$. Thus if r is odd, then $t = 0$ which means $t_1 = t_2 = \cdots = t_k = 0$.

Suppose r was even, we shall analyze the probability that $x^{r/2} = -1 \pmod{n}$. By the chinese remaindering theorem, $x^{r/2} = -1 \mapsto (-1, -1, \cdots, -1)$. But suppose some $t_i \neq t$, then $t_i \leq t - 1$ and hence $r_i \mid r/2$ and therefore the i-th coordinate in the tuple will have to be a 1 instead of a -1.

Therefore, the probability that r is odd and $x^{r/2} = -1 \pmod{n}$ is bounded above by $\Pr[t_1 = t_2 = \cdots = t_k] \leq 2^{k-1}$ since each t_i is independent.

Therefore $\Pr[r \text{ is even and } x^{r/2} \neq -1 \pmod{n}] \geq 1 - 2^{k-1}$. □

Over the next few lectures, we will see how we can find the order of an element using a quantum algorithm, and this would solve the integer factoring problem.