CSE 291 / Math 277A - Quantum Complexity Theory	October 25, 2022
Lecture 9	
Lecturer: Daniel Grier	*Scribe: Guru Kalyan Jayasingh*

1 Warm Up

Let $N = 2^n$ and suppose you a have a bit string $x = x_1x_2 \dots x_N \in \{0,1\}^N$ and you want to determine if one of the bits is 1. How do you phrase this as a query complexity question for a function?

Answer. Define the function $f\colon \{0,1\}^n \to \{0,1\}$ where $f(i) = x_i$. Here, x_i is just the bit of x indexed by i where we think of i as a single number written in binary. Therefore, the query problem with an unknown function translates to one of an unknown bit string. It can sometimes be useful to switch between the two settings. For instance, the problem defined is the OR function, which is simple to express as $x_1 \vee x_2 \vee \cdots \vee x_N$.

2 Total vs. Promise Problems

So far, this class has focused on promise problems, where the output is undefined for input that does not satisfy some property. As we've seen with Simon's Problem and Forrelation, we can get huge quantum speedups (as measured by query complexity) for promise problems. Today, we move from promise problems to problems about total functions, which must be defined over all inputs.

One might wonder why we cannot just take any promise problem for which a quantum computer had some kind of advantage and extend it to inputs for which it wasn't previously defined. Unfortunately, the issue is that we cannot easily detect the inputs for which the original promise held. Since we *must* be able to detect those inputs to answer consistently on all inputs, it's unclear how to make such a strategy work. If fact, such a strategy provably cannot work:

Theorem 1 (Aaronson, Ben-David, Kothari, and Tal [ABDKT20])**.** *The deterministic query complexity of a total function is at most the quantum query complexity of that function to the fourth power.*

In other words, total functions can only yield polynomial query speedups. It is currently unknown whether or not this bound is tight.

Today, we will focus on a particularly important total function, the OR function from the intro. While we know that the quantum query algorithm cannot obtain an exponential speedup, we will see that it can still get a quadratic speedup. Let's first look at its classical query complexity. Consider the case where there is at most one bit that is 1. Any classical deterministic algorithm will need to make N queries since it might get unlucky and query $N-1$ zeroes. Similarly, any classical randomized algorithm will need $N/2$ queries in expectation.

Quantumly, however, Grover's algorithm requires only $O(\sqrt{N})$ queries. Although it doesn't provide an exponential speedup, the fact that the algorithm can be used for any unstructured search problem makes it useful for a wide variety of applications.

3 Grover's Algorithm

Let's start with a search version of the OR problem: given oracle access to the function $f\colon \{0,1\}^n \to \{0,1\}$, find $x \in \{0,1\}^n$ such that $f(x) = 1$ or report that none exists. Since there is no assumption made on f, this problem is often called *unstructured search*. Any x such that $f(x) = 1$ is called a *marked item*. It will turn out that the simplest version of Grover's algorithm depends on the number of marked items, so let us assume for now that there is only a single marked item. To be clear, the most general version of Grover's algorithm will work regardless of the number of marked items.

The entirety of Grover's algorithm is simply alternating between the following two operators:

- Phase Oracle: $O_f |x\rangle = (-1)^{f(x)} |x\rangle$.
- Diffusion Operator: $D := 2(|u\rangle\langle u|) - I$, where $|u\rangle := H^{\otimes n} |0^n\rangle$ is the uniform superposition.

The phase oracle O_f comes from the query model, so let's just check that we can efficiently execute the diffusion operator:

Claim 1. *The diffusion operator $D := 2(|u\rangle\langle u|) - I$ is a unitary operation that reflects[1] about $|u\rangle$. Furthermore, D can be constructed with linearly-many gates in log depth.*

Proof. To verify that D is unitary, we can simply compute

$$DD^\dagger = (2(|u\rangle\langle u|) - I) \cdot (2(|u\rangle\langle u|) - I)^\dagger = 4(|u\rangle\langle u|) - 2(|u\rangle\langle u|) - 2(|u\rangle\langle u|) + I = I.$$

To see why D is a reflection about $|u\rangle$, first notice that we can decompose an arbitrary state $|\psi\rangle$ as its component aligned with $|u\rangle$ and its component orthogonal to $|u\rangle$.

$$|\psi\rangle = \alpha |u\rangle + \beta |v\rangle \ ,$$

where $\langle u|v\rangle = 0$ and $|\alpha|^2 + |\beta|^2 = 1$. Then, we can verify

$$D|\psi\rangle = \alpha(2(|u\rangle\langle u|) - I)|u\rangle + \beta(2(|u\rangle\langle u|) - I)|v\rangle = \alpha|u\rangle - \beta|v\rangle \ ,$$

where we use the fact that $\langle u|u\rangle = 1$ and $\langle u|v\rangle = 0$.

To see that D can be constructed with linearly-many gates in log depth, notice that if we conjugate D by Hadamard, we get the reflection about the all-zeros state: $D_0 = 2|0^n\rangle\langle 0^n| - I$. Therefore, we just need a circuit for D_0. On the computational basis states, we have $D_0|x\rangle = (-1)^{x_1 \vee \cdots \vee x_n}|x\rangle$ so we just need to be able to detect if any of the qubits are 1 (which can be done with a linear-size, log-depth reversible circuit) and apply a phase gate depending on the answer. □

Algorithm 1 Grover's algorithm

Input: 2^n unknown input bits accessed through the oracle O_f.
Output: $s \in \{0,1\}^n$ such that $f(s) = 1$, or `null` if none exists.

1: $|\psi_0\rangle = H^{\otimes n}|0^n\rangle$
2: **for** $i \in \{1, \ldots, T\}$ **do**
3: $\quad |\psi_i\rangle \leftarrow DO_f |\psi\rangle_{i-1}$
4: **end for**
5: $s^* \leftarrow$ measurement of $|\psi_T\rangle$
6: **return** s^* if $f(s^*) = 1$; otherwise, `null`

Examining Grover's algorithm, we see that the final state before we measure is given by

$$DO_f \cdots DO_f DO_f |u\rangle$$

To understand why this algorithm works, it will be extremely useful to take a geometric perspective. To start, notice that our initial state $|u\rangle$ lies in a particular 2-dimensional subspace that is spanned by $|s\rangle$ (our marked item) and $|\Psi\rangle = \frac{1}{\sqrt{2^n - 1}} \sum_{x \neq s} |x\rangle$ (the uniform superposition over all unmarked states):

$$|u\rangle = \frac{1}{\sqrt{2^n}} \sum_x |x\rangle = \frac{1}{\sqrt{2^n}} |s\rangle + \frac{1}{\sqrt{2^n}} \sum_{x \neq s} |x\rangle = \frac{1}{\sqrt{2^n}} |s\rangle + \sqrt{1 - \frac{1}{2^n}} |\Psi\rangle \, .$$

First, we make the following intriguing observation:

Observation 2. *Each Grover iteration keeps the state in the span of $|s\rangle$ and $|\Psi\rangle$.*

[1] By "reflect" about $|u\rangle$, we mean that D flips the sign of every vector in the subspace orthogonal to $|u\rangle$.

Proof. This is easy to see for the phase oracle: $\alpha |s\rangle + \beta |\Psi\rangle \xrightarrow{O_f} -\alpha |s\rangle + \beta |\Psi\rangle$. For the diffusion operator, we have

$$\alpha |s\rangle + \beta |\Psi\rangle \xrightarrow{D} (2|u\rangle\langle u| - I)(\alpha |s\rangle + \beta |\Psi\rangle) = 2(\alpha \langle u|s\rangle + \beta \langle u|\Psi\rangle)|u\rangle - \alpha |s\rangle - \beta |\Psi\rangle$$

but we've already seen above that $|u\rangle$ can be expressed a linear combination of $|s\rangle$ and $|\Psi\rangle$. □

In other words, each Grover operation is a rotation in the plane spanned by $|s\rangle$ and $|\Psi\rangle$. We have that O_f reflects about $|\Psi\rangle$, and the diffusion operation reflects about $|u\rangle$. See Figure 1 and Figure 2 for a visualization of these rotations.

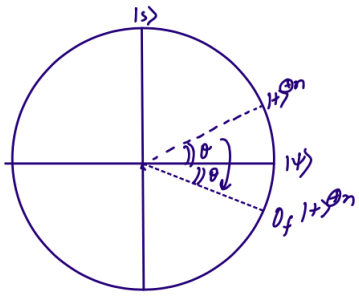

Figure 1: Action under phase oracle.

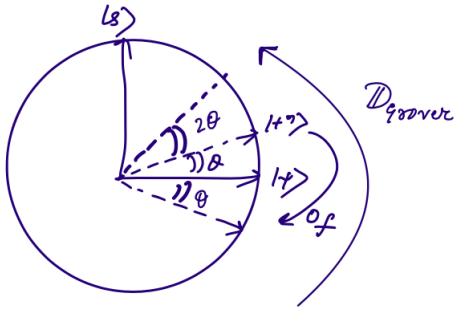

Figure 2: Action under diffusion operator.

Let's see how a sequence of these reflections act on our starting state $|u\rangle$. Intuitively, our starting state is extremely close to $|\Psi\rangle$ and we would like to rotate it towards $|s\rangle$. Let θ_0 be the initial angle between $|u\rangle$ and $|\Psi\rangle$. We have[2]

$$\cos(\theta_0) = \langle u|\Psi\rangle = \sqrt{1 - \frac{1}{2^n}} \implies \sin(\theta_0) = \frac{1}{\sqrt{2^n}} \implies \theta_0 \approx \frac{1}{\sqrt{2^n}}$$

where we have used that $\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \ldots$ is approximately x for small x. We will see that the phase oracle and diffusion operator together will rotate our state slightly closer to $|s\rangle$.

Let's now walk through the action of these two Grover operations. Suppose our state is θ radians above the $|\Psi\rangle$ axis as shown in Figure 1. When we apply the phase oracle, we reflect about $|\Psi\rangle$, and so our state is now θ radians below the $|\Psi\rangle$ axis. When we apply the diffusion operation we reflect back about the $|u\rangle$ state. Since $|u\rangle$ is θ_0 radians above the $|\Psi\rangle$ axis and state started θ radians below the axis, our reflected state is $2\theta_0 + \theta$ radians above the axis. Figure 2 shows this reflection when $\theta = \theta_0$. As you can see, while our state started θ_0 radians above the axis, it is now $3\theta_0$ radians above the axis.

Repeating this process, the evolution of the angle is given by $\theta_0, 3\theta_0, 5\theta_0, \ldots, (2T+1)\theta_0$. Notice that we want to reach the angle $\pi/2$, so we get that we need $T \approx \pi/(4\theta_0) = O(\sqrt{2^n})$ steps. Of course, this analysis only holds if there was indeed a marked element. However, after we've done this procedure, we measure to obtain to some candidate marked item s^*. We can use one more query to our oracle to check that $f(s^*) = 1$. This completes the analysis of Grover's algorithm for a single marked element.

What happens if there are more than 1 marked items? In this case, let $|s\rangle$ be the uniform superposition over all marked items. If we have m marked elements, then initial angle is $\langle s|u\rangle \approx \sqrt{m/2^n}$ at least when there aren't too many marked items (if there are so many marked items, we can just randomly sample until we find one). Therefore, with the same analysis, the number of queries required to rotate our state to $|s\rangle$ is $O(\sqrt{2^n/m})$. When we measure, we get a uniformly random marked item. This speedup follows our intuition that if there are more marked elements, it should be easier to find one of them.

There is one final question to address. Namely, the above analysis only works when we know the number marked elements. Indeed, if we continue to do more Grover iterations, then then our state continues to rotate around the unit circle. If the number of marked items is unknown, how do we know when to stop and

[2]Recall that in Euclidean space the angle θ between two complex vectors u and v is given by $\cos\theta = (u \cdot v)/(\|u\|\|v\|)$.

measure? The trick is something called "exponential search." We make the following sequence of guesses for m: $2^n, 2^{n-1}, 2^{n-2}, \ldots, 4, 2$. Notice that if we make all n possible guesses, then we are at most a factor of 2 off from the true answer. One can check that this does not dramatically affect the analysis. The reason that the we search in decreasing order is because we want to obtain a speedup in the case that there are actually many marked items. If at any point we find a marked item, then we stop.

4 Bennett-Bernstein-Brassard-Vazirani (BBBV) lower bound

Grover's algorithm gives us an efficient $O(\sqrt{2^n})$ quantum query algorithm for the unstructured search problem. A natural question is whether or not we can do better. In fact, what if only care about the number of queries our algorithm makes, not necessarily the number of gates in our quantum circuit? Unfortunately, even under this relaxation we cannot improve upon Grover's algorithm:

Theorem 3 (Bennett, Bernstein, Brassard, Vazirani [BBBV97])**.** *The quantum query complexity of the OR problem is* $\Omega(\sqrt{2^n})$.

As we will see later in class, there are actually many possible ways to prove this lower bound, but the BBBV lower bound was the first and perhaps most intuitive lower bound technique, so let's start with that. First, notice that a generic quantum query algorithm alternates between applying some unitary and applying the oracle. In other words, after t queries, the state of our system looks like

$$U_t O_f U_{t-1} \cdots O_f U_1 O_f U_0 \left|0^n\right\rangle.$$

To be fully rigorous here, we would also need to specify a set of ancillary workspace qubits, but this will not change the analysis and only make the notation more cumbersome, so we will drop these extra qubits.

A key point about this decomposition is that the unitaries $U_0, U_1, \ldots, U_t$ are fixed and are independent of what the oracle does. When there are few oracle queries, our goal will be to show that for every choice of unitaries, there is some state $|y\rangle$ that always has small amplitude when queried by the oracle. Because of this, it will be very difficult for the algorithm to "see" whether or not this item is marked. Therefore, we can fool the algorithm into accepting/rejecting when it shouldn't.

Let's first consider what our algorithm does on the constant zero function. In this case, the oracle is just the identity, and the algorithm should reject. The state of the algorithm after t queries is

$$\left|\psi_t\right\rangle := U_t U_{t-1} \cdots U_1 U_0 \left|0^n\right\rangle = \sum_{x \in \{0,1\}^n} \alpha_{x,t} \left|x\right\rangle.$$

Supposing there are T total queries, define the quantity

$$m_x := \sum_{t=0}^{T} |\alpha_{x,t}|^2.$$

to be the sum of the squares of the magnitudes on x over all states $|\psi_t\rangle$. We have that

$$\sum_{x \in \{0,1\}^n} m_x = \sum_{x \in \{0,1\}^n} \sum_{t=0}^{T} |\alpha_{x,t}|^2 = \sum_{t=0}^{T} \left(\sum_{x \in \{0,1\}^n} |\alpha_{x,t}|^2 \right) = \sum_{t=0}^{T} 1 = T.$$

Since m_x is non-negative, this implies that there must exist some $y \in \{0,1\}^n$ such that $m_y \leq T/2^n$ (otherwise, the sum is greater than T). This y will be the element that the algorithm fails to properly consider if T is too small. The above argument gives us a bound on the sum of the squares of the magnitudes for the input y, but it will turn out that we will actually need a bound on the sum of the magnitudes themselves. Fortunately, by Cauchy-Schwarz, we have

$$\sum_{t=0}^{T} |\alpha_{y,t}| \leq \sqrt{\sum_{t=1}^{T} |\alpha_{y,t}|^2 \cdot T} = \sqrt{m_y T} \leq \frac{T}{\sqrt{2^n}}.$$

Since we can refer to the all-zeros function as the identity, let f be the function which is 1 on y and 0 elsewhere. Our goal is to distinguish the oracle for f from the oracle for the identity, but for the purposes of analysis, let's consider a set of rather strange oracles $\{O^{(t)}\}_{t=0}^{T}$. Here, $O^{(t)}$ is defined to be the identity for the first t queries and f on the remaining $T-t$ queries. In other words, the oracle is interpolates between our two function instances. Let's define the set of states arising from the application of these oracles as

$$|\varphi^{(t)}\rangle := U_T O^{(t)} U_{T-1} \cdots O^{(t)} U_1 O^{(t)} U_0 |0^n\rangle = U_T O_f U_{T-1} \cdots O_f U_{t+1} O_f |\psi_t\rangle$$

So, for example, we have that $|\varphi^{(T)}\rangle = |\psi_T\rangle$ is the state for the complete execution of the quantum algorithm for the constant 0 function, and $|\varphi^{(0)}\rangle$ is the state for the execution of the quantum algorithm for f.

If we can show that $|\varphi^{(t+1)}\rangle$ is close to $|\varphi^{(t)}\rangle$ for all t, then by the triangle inequality, we will be able to conclude that the states from the two different problem instances are also close to each other. We have the following:

$$\begin{aligned}
\||\varphi^{(t+1)}\rangle - |\varphi^{(t)}\rangle\| &= \|U_T O_f U_{T-1} \cdots O_f U_{t+2} O_f |\psi_{t+1}\rangle - U_T O_f U_{T-1} \cdots O_f U_{t+1} O_f |\psi_t\rangle\| \\
&= \|(U_T O_f U_{T-1} \cdots O_f U_{t+2} O_f U_{t+1}) |\psi_t\rangle - (U_T O_f U_{T-1} \cdots O_f U_{t+1}) O_f |\psi_t\rangle\| \\
&= \||\psi_t\rangle - O_f |\psi_t\rangle\| \\
&= 2|\alpha_{y,t}|
\end{aligned}$$

where we have used the fact that unitaries preserves the 2-norm and the fact that $O_f |\psi_t\rangle = |\psi_t\rangle - 2\alpha_{y,t} |y\rangle$. Combining everything together, we get

$$\||\varphi^{(T)}\rangle - |\varphi^{(0)}\rangle\| \leq \sum_{t=0}^{T-1} \||\varphi^{(t+1)}\rangle - |\varphi^{(t)}\rangle\| \leq 2 \sum_{t=0}^{T-1} |\alpha_{y,t}| \leq \frac{2T}{\sqrt{2^n}}.$$

Hence, we see that for $T \ll \sqrt{2^n}$, the two states are close under l_2 norm. Next time we will show this implies that the distribution over outcomes induced by measuring the two states must be statistically close, implying that the two instances cannot be distinguished by the quantum algorithm (with high probability).

References

[ABDKT20] Scott Aaronson, Shalev Ben-David, Robin Kothari, and Avishay Tal. Quantum implications of Huang's sensitivity theorem. *arXiv:2004.13231*, 2020.

[BBBV97] Charles H Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM journal on Computing*, 26(5):1510–1523, 1997.