CSE 190 / Math 152 - Introduction to Quantum Computing
Homework 6
Due **Friday, June 6, 11:59pm**

Instructions: There are essentially 4 parts to this homework:

1. Concept Check : Must be completed individually
2. Survey of student learning in quantum computing : See details on Canvas. Must be completed individually
3. Normal written homework : Can be completed individually or in a team of 2 people. There are only two questions: 3(a) and 3(b)
4. Optional written homework : The remainder of problem 3 is optional and will not be graded. It uses a technique call phase estimation which we have not covered in class (nor will it appear on the final).

Problems:

1. **Concept check**

   Complete the assignment "Homework 6 - Concept Check" on Gradescope.

   Must be completed individually even if working in a team for the rest of the assignment.

2. **Survey of student learning in quantum computing**

   Survey linked on Canvas as an announcement (the link is private to this class).

   Must be completed individually even if working in a team for the rest of the assignment.

3. **Counting solutions with Grover's algorithm and Phase Estimation**

   In class, we gave a quantum algorithm for the following "search" problem:

   Input: Query access to a function $f\colon \{0,1\}^n \to \{0,1\}$
   Output: Bit string $x \in \{0,1\}^n$ such that $f(x) = 1$ (or report none exists)

   In particular, we showed that Grover's algorithm uses roughly $\sqrt{2^n}$ queries to the oracle O_f whereas any classical algorithm requires 2^n queries to f. In this problem, we will explore the "counting" version of this problem:

   Input: Query access to a function $f\colon \{0,1\}^n \to \{0,1\}$
   Output: Approximate number of $x \in \{0,1\}^n$ such that $f(x) = 1$

   This problem seems to be a lot harder than the search problem since intuitively you need find all inputs for which f evaluates to 1, not just a single input. Nevertheless, we will show that there is a quantum algorithm that combines the ideas in Grover's algorithm with phase estimation to solve this problem. It may be helpful to review the analysis of Grover's algorithm before working on this problem, but we will reintroduce some of the major components.

Let $\mathcal{M}$ and $\mathcal{U}$ be the sets of "marked" and "unmarked" inputs, respectively. That is,

$$\mathcal{M} = \{x \in \{0,1\}^n \mid f(x) = 1\} \qquad \text{and} \qquad \mathcal{U} = \{x \in \{0,1\}^n \mid f(x) = 0\}.$$

As in the analysis of Grover's algorithm, we will consider a 2-dimensional space spanned by the uniform superposition of marked and unmarked items:

$$|s\rangle := \frac{1}{\sqrt{|\mathcal{M}|}} \sum_{x \in \mathcal{M}} |x\rangle \qquad \text{and,} \qquad |\Psi\rangle := \frac{1}{\sqrt{|\mathcal{U}|}} \sum_{x \in \mathcal{U}} |x\rangle$$

Once again, we will start the quantum algorithm in the state

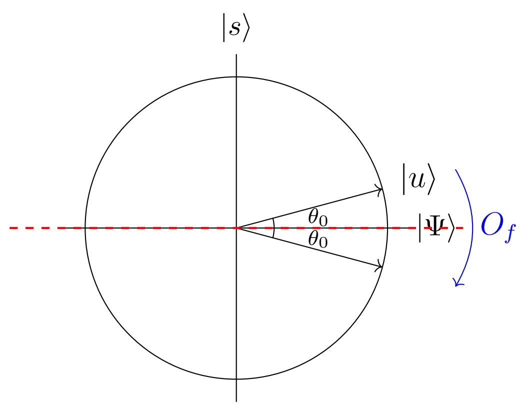
$$|u\rangle := H^{\otimes n} |0^n\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle \,.$$

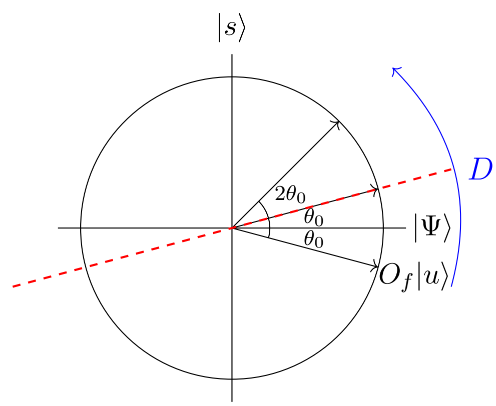
(a) Show that $|u\rangle = \sqrt{\frac{|\mathcal{M}|}{2^n}}\,|s\rangle + \sqrt{1 - \frac{|\mathcal{M}|}{2^n}}\,|\Psi\rangle$.

For now, let's consider the same operations—the phase oracle O_f, and the Grover diffusion operator D—that we used in Grover's algorithm:

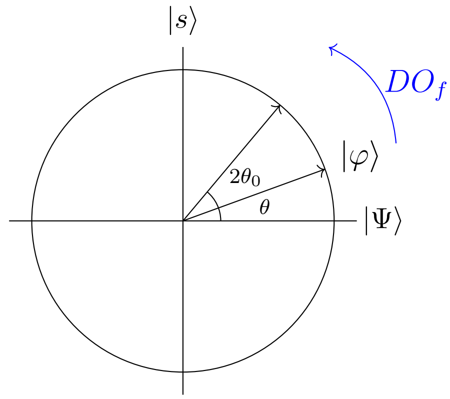
$$O_f |x\rangle = (-1)^{f(x)} |x\rangle \,, \qquad \text{and} \qquad D|x\rangle = 2\langle u|x\rangle |u\rangle - |x\rangle$$

for all $x \in \{0,1\}^n$. It will be useful to once again think of O_f and D as reflections in the space spanned by $|s\rangle$ and $|\Psi\rangle$. The axis of reflection for each operation is shown in red in the figures below. The first figure shows O_f applied to $|u\rangle$ (the initial state of our algorithm), and the second figure shows D applied to $O_f|u\rangle$:





If we compose the two operations (i.e., DO_f) and apply them to any arbitrary state $|\varphi\rangle$, we simply get a rotation in this space of $2\theta_0$, where θ_0 is the initial angle between $|u\rangle$ and $|\Psi\rangle$:



Let's get some concrete practice with this rotation.

(b) Compute the amplitude on $|s\rangle$ after one Grover iteration. In other words, compute $\langle s| DO_f |u\rangle$. *Make sure to check that your answer makes sense. The initial amplitude on $|s\rangle$ is $\sqrt{|\mathcal{M}|/2^n}$. Using the small-angle approximation, this implies that $\theta_0 \approx \sqrt{|\mathcal{M}|/2^n}$.*

** THE REMAINDER OF THIS HOMEWORK IS ENTIRELY OPTIONAL AND WILL NOT BE GRADED **

We've concluded that DO_f is a rotation by $2\theta_0$. Since $\theta_0 \approx \sqrt{|\mathcal{M}|/2^n}$, we can approximate $|\mathcal{M}|$ if we know the value of θ_0: $|\mathcal{M}| \approx 2^n\theta_0^2$. Therefore, to solve the counting problem, our plan will be to determine θ_0. We will use phase estimation to do this, so let's review the setting of that algorithm. First, recall that for any unitary U, we define $\Lambda_m(U)$ to be the unitary that has the following behavior:

$$\Lambda_m(U)(|k\rangle \otimes |x\rangle) = |k\rangle \otimes U^k |x\rangle$$

for all $x \in \{0,1\}^n$ and all integers k represented in binary using m bits. The phase estimation algorithm outputs an approximation of the eigenvalue of a unitary U corresponding to a given eigenstate:

Input: Unitary $\Lambda_m(U)$, eigenstate $|\psi\rangle$ such that $U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$
Output: Approximate value of $\theta \in [0,1)$

Our goal will be to use phase estimation on DO_f. Here, we can see that to use phase estimation, we must depart from the original Grover setting where we only required query access to O_f. For now, let's assume we also have access to $\Lambda_m(DO_f)$.

To finish the algorithm, we must show how the eigenvalues of DO_f are related to θ_0. For this, we once again appeal to the geometric interpretation of DO_f as a rotation by $2\theta_0$ in the space spanned by $|s\rangle$ and $|\Psi\rangle$. We will use that rotations in this space correspond to the rotation matrix

$$R(\phi) = \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix}.$$

For DO_f, specifically, we have shown that $\phi = 2\theta_0$. Since the eigenvalues of $R(\phi)$ are $e^{\pm i\phi}$, the non-zero eigenvalues of DO_f should be $e^{\pm 2i\theta_0}$. Let's check this directly. To do this, we will need the following expressions:

$$\sin\theta_0 = \langle u|s\rangle = \sqrt{|\mathcal{M}|/2^n}$$
$$\cos\theta_0 = \langle u|\Psi\rangle = \sqrt{1 - |\mathcal{M}|/2^n}$$

which can be derived by referring back to the figures and using some standard trigonometric identities.

(c) Show that DO_f has eigenstate $\frac{|s\rangle + i|\Psi\rangle}{\sqrt{2}}$ with eigenvalue $e^{2i\theta_0}$.
This is an involved calculation. Here is a rough outline if you get stuck: start with the state $(|s\rangle + i\,|\Psi\rangle)/\sqrt{2}$ *and apply the operators* O_f *and* D*. In particular, the application of* D *will lead to a state which has a component of* $|u\rangle$*. But notice by part (a), we can expand* $|u\rangle$ *back in the* $\{|s\rangle, |\Psi\rangle\}$ *basis. Furthermore, the coefficients of that expansion can be represented as* $\sin\theta_0$ *and* $\cos\theta_0$ *using the equations above. To complete the calculation, you will have to reason about how the resulting expression simplifies using some standard trig identities—for example, depending on how you do the calculation you may need the double angle formulas*

$$\sin(2x) = 2\sin x\cos x,$$
$$\cos(2x) = 1 - 2\sin^2 x = 2\cos^2 x - 1$$

as well as Euler's formula: $e^{ix} = \cos x + i\sin x$.

A nearly identical calculation will show that $\frac{|s\rangle - i|\Psi\rangle}{\sqrt{2}}$ is an eigenstate of DO_f with eigenvalue $e^{-2i\theta_0}$. To recap, we've shown that DO_f has eigenvalues $e^{\pm 2i\theta_0}$, which can be written as

$$e^{2\pi i(\frac{\theta_0}{\pi})} \quad \text{or} \quad e^{2\pi i(1-\frac{\theta_0}{\pi})}$$

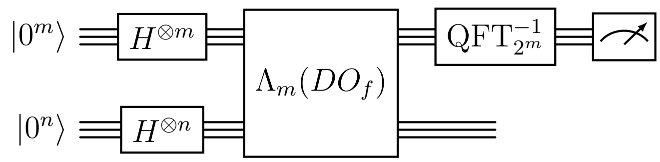
so using phase estimation we should be able to obtain an estimate to θ_0/π or $(1 - \theta_0/\pi)$, which we can use to estimate θ_0 and ultimately $|\mathcal{M}|$. There's one catch—we don't actually have access to the eigenstates of DO_f. We computed the eigenstates in part (c) from the states $|s\rangle$ and $|\Psi\rangle$, but we don't actually know what the states are (we're only assuming that we know them in the analysis). As it turns out, we don't need to!

Our plan will be to run phase estimation on the state $|u\rangle$, which can be written as a superposition of eigenstates. In this way, the phase estimation algorithm will learn either θ_0/π or $(1 - \theta_0/\pi)$. We can't control which one we learn, but it also doesn't matter—either one suffices to estimate θ_0. Let's do the analysis, first giving some names to the eigenstates:

$$|\varphi_+\rangle := \frac{|s\rangle + i\,|\Psi\rangle}{\sqrt{2}} \quad \text{and} \quad |\varphi_-\rangle := \frac{|s\rangle - i\,|\Psi\rangle}{\sqrt{2}}.$$

(d) Claim: $|u\rangle = \alpha\,|\varphi_+\rangle + \beta\,|\varphi_-\rangle$ for complex amplitudes α, β. Compute α and β.

Putting everything together, we have the following circuit for quantum counting:



(e) Assume that $\theta_0/\pi = j/2^m$ for some positive integer $j < 2^m$. Using the α and β you computed in part (d), show that the measurement returns j with probability $|\alpha|^2$ and returns $2^m - j$ with probability $|\beta|^2$. *Hint: It might be helpful to first revisit the standard analysis of the phase estimation algorithm.*

There's one final aspect of this algorithm we need to consider. Grover's algorithm used roughly $2^{n/2}$ queries to O_f, so how many queries did our algorithm use? In some sense, it feels like the answer is just "1" since the quantum phase estimation circuit we built only has a single call to the unitary $\Lambda_m(DO_f)$. Since the counting problem is harder than the search problem, $\Lambda_m(DO_f)$ is clearly *very* powerful. Unfortunately, $\Lambda_m(DO_f)$ does not properly capture the complexity of implementing phase estimation in practice—if we have an efficient circuit for O_f we might not have an efficient circuit for $\Lambda_m(DO_f)$.

To more accurately capture the difficulty of constructing the unitary $\Lambda_m(DO_f)$, we will count how many controlled-O_f gates we need in order to implement it. For most practical problems, the difference in cost of implementing controlled-O_f and O_f is small, so this is a reasonable gate to allow ourselves. One can show how to implement $\Lambda_m(DO_f)$ using roughly 2^m controlled-O_f gates (*Bonus exercise*: prove this).

Therefore, to determine the query complexity of counting, we need to determine how large m needs to be to accurately estimate θ_0. In part (e), we were able to compute the phase exactly (and therefore θ_0 exactly); however, in general, phase estimation may have some $1/2^m$ error. In other words, if the phase estimation procedure returns some approximation $E \in \mathbb{R}$ for the phase θ/π, then the only guarantee is that

$$|E - \theta_0/\pi| \le \frac{1}{2^m}.$$

Let's investigate how this error affects the number of queries needed for counting. We've argued before that $\theta_0 \approx \sqrt{|\mathcal{M}|/2^n}$, but for simplicity, let's just assume that they are exactly equal: $\theta_0 = \sqrt{|\mathcal{M}|/2^n}$. Furthermore, let's make a simplifying assumption that the phase estimation algorithm returns an estimate E to θ_0/π (the analysis will be identical if the phase estimation procedure returns an estimate to $1 - \theta_0/\pi$). Since E is an approximation of θ_0/π, then $2^n\pi^2E^2$ should be an approximation of $|\mathcal{M}|$.

(f) Using the error bound on E, show the following error bound on $2^n\pi^2E^2$:

$$|2^n\pi^2E^2 - |\mathcal{M}|| \le 2\pi\frac{\sqrt{2^n|\mathcal{M}|}}{2^m} + \pi^2\frac{2^n}{2^{2m}}$$

(g) Using the bound from part (f), determine what value of m is required to obtain a multiplicative ϵ-approximation of $|\mathcal{M}|$. In other words, we want

$$|2^n\pi^2E^2 - |\mathcal{M}|| \le \epsilon|\mathcal{M}|$$

Express the value of m in terms of n, ϵ, and $|\mathcal{M}|$, and show that it gives the desired inequality above. Show that your answer implies that the total number of quantum queries used in the algorithm is (up to a constant factor) $\frac{1}{\epsilon}\sqrt{2^n/|\mathcal{M}|}$. *It may help to assume that* $\epsilon = 1/2^a$ *for some positive integer* a. *This can affect the final solution by at most a factor of* 2.

As a final remark, notice that your bound on m in part (g) depended on $|\mathcal{M}|$. This looks somewhat weird—we are trying to estimate $|\mathcal{M}|$, so we can't know it ahead of time. However, there is a trick where we can guess that $|\mathcal{M}| = 2^n, 2^{n-1}, 2^{n-2}, \ldots$ to sneak up on the right answer. The analysis of that trick is slightly involved, so we won't go into it.