CS333 - Problem Set 7

1. Let $f : \{0,1\}^2 \to \{0,1\}$ be a black-box function whose input is two bits, where f takes the value 1 on exactly one input value (and is zero on the other three input values). For example we could have $f(00) = 0$, $f(01) = 1$, $f(10) = 0$ and $f(11) = 0$. We would like to create an algorithm that identifies which input to f gives output 1. So in our example, the algorithm should output 01. This is a small example of a "search problem."

   (a) Fill in the following truth table with the 4 possible black box functions: The possible functions are $f_{00}, f_{01}, f_{10}, f_{11}$ where the subscript indicates which

x_1	x_2	$f_{00}(x_1, x_2)$	$f_{01}(x_1, x_2)$	$f_{10}(x_1, x_2)$	$f_{11}(x_1, x_2)$

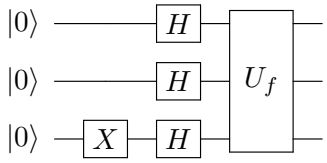
   (1)

   (b) How many classical queries are needed to solve one-out-of-four search in the worst case? In our example above, it requires 1 query to learn that $f(10) = 0$.

   (c) Suppose $|x_1\rangle$, $|x_2\rangle$, and $|y\rangle$ are single qubit standard basis states. Suppose we can query f using a quantum unitary U_f that acts as

   $$U_f|x_1, x_2\rangle|y\rangle = |x_1, x_2\rangle|y \oplus f(x_1, x_2)\rangle, \tag{2}$$

   where $\oplus$ is XOR, that is, addition mod 2. Now consider the following circuit:

   (3)

   i. What is the state of the system immediately before the unitary U_f?

   ii. What is the state of the system immediately after the unitary U_f? Show that we get a phase kickback like in Deutsch's algorithm

   iii. Show that the four possible outputs obtained in the previous part are all orthogonal to each other. Using question 4 from Pset 6, explain why this is enough to argue that there is a quantum algorithm that only uses U_f once.

   iv. Describe at a high level (using words like phase kickback and superposition) how this quantum algorithm works.

2. In this problem, we will investigate a many-qubit version of Deutsch's Algorithm called the Deutsch-Josza Algorithm. This problem involves using the unitary operation $H^{\otimes n}$, so we will first investigate its properties. Note that $H^{\otimes n}$ is n copies of H acting simultaneously on n

qubits. We describe how $H^{\otimes n}$ acts by describing how it acts on standard basis states. Let $|x\rangle$ be a standard basis state on n qubits. This means $|x\rangle$ can be written as a tensor product of n standard single qubit basis states. Then

$$H^{\otimes n}|x\rangle = \frac{1}{2^{n/2}} \sum_{y \in \{0,1\}^n} (-1)^{x \cdot y} |y\rangle. \tag{4}$$

where the summation is over all 2^n standard basis states y, and

$$x \cdot y = \sum_{i=1}^{n} x_i y_i \tag{5}$$

where x_i is the ith bit of x and y_i is the ith bit of y.

So for example:

$$\begin{aligned} H^{\otimes 2}|01\rangle &= \frac{1}{2} \sum_{y \in \{0,1\}^n} (-1)^{01 \cdot y} |y\rangle \\ &= \frac{1}{2} \left((-1)^{01 \cdot 00} |00\rangle + (-1)^{01 \cdot 01} |01\rangle + (-1)^{01 \cdot 10} |10\rangle + (-1)^{01 \cdot 11} |11\rangle \right) \\ &= \frac{1}{2} \left(|00\rangle - |01\rangle + |10\rangle - |11\rangle \right). \end{aligned} \tag{6}$$

(a) Use the formula in Eq. (4) to determine the resulting state of

$$H^{\otimes n}|0\rangle^{\otimes n} \tag{7}$$

and

$$H^{\otimes n}|1\rangle^{\otimes n}. \tag{8}$$

(b) (Challenge question) Let $|x\rangle$ be a standard basis state that is not $|0\rangle^{\otimes n}$. Explain why

$$H^{\otimes n}|x\rangle \tag{9}$$

produces a superposition of all standard basis states with exactly half of the amplitudes positive, and half of the amplitudes negative.

(c) Consider a function $f : \{0,1\}^n \to \{0,1\}$ (f takes as input an n-bit string and outputs 1 bit) that has the promise that it is either even or balanced. If it is even, $f(x) = 0$ for all inputs, or $f(x) = 1$ for all inputs. If it is balanced, then for exactly half of the inputs, $f(x) = 0$ and for half of the inputs $f(x) = 1$. We would like to determine which case we are in.

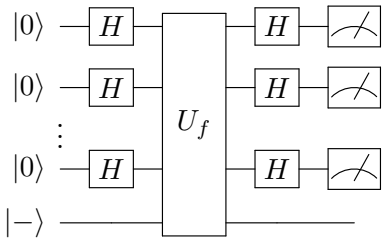
i. What is the worst case deterministic classical query complexity of deciding with certainty if the function is even or balanced? You should assume that you have to make public your algorithm ahead of time, before getting access to f. Then an adversary gets to design f in such a way to try to make your algorithm make the maximum number of queries possible. The query complexity is the maximum number of queries in this setting.

ii. What is the worst case probabilistic classical query complexity of determining if the function is even or balanced? (This means you have a probabilistic algorithm, and your answer should be correct with probability at least 2/3. You don't have to prove, just sketch the idea.) You should assume that you have to make public your algorithm ahead of time, before getting access to f. Then an adversary gets to design f in such a way to try to make your algorithm make the maximum number of queries possible. The query complexity is the maximum number of queries in this setting. Why is this case different from the deterministic setting?

(d) If $|x\rangle$ is a standard basis state on n qubits and $|y\rangle$ is a standard basis state on 1 qubit, then U_f acts as follows:

$$U_f|x\rangle_A|y\rangle_B = |x\rangle_A|y \oplus f(x)\rangle_B, \tag{10}$$

where A is an n qubit system and B is a 1 qubit system. Consider the following circuit



(11)

i. What is the state of the system after U_f acts? (Use phase kickback - your expression should have terms $f(y)$ that you do not need to simplify.)

ii. What is the state of the system after the second $H^{\otimes n}$ acts? (You can distribute the $H^{\otimes n}$ inside a summation symbol. Also, if you have a summation inside another summation, you can move both summation terms to the beginning, and then switch their order.)

iii. If the function is even, what is the probability of getting outcome $|0\rangle^{\otimes n}$ (i.e. $|0\rangle$ on every qubit) when the measurement is made? What if the function is balanced?

iv. What is the quantum query complexity of determining if the function is even or balanced?

3. (a) Suppose you know that over the course of a quantum algorithm on n qubits, the quantum system is never in more than a superposition of T standard basis states. Suppose that the computation involves M single and two qubit unitaries. You may assume that at each time step, only one unitary acts at a time (so there is no parallel computation). Explain how you can simulate this computation using a classical computer with $O(Tn)$ bits in $O(MT^2)$ time. (You can probably do $O(M \times T)$ time with some extra cleverness...bonus points if you do!) (Hint: think about our path integral analysis of a quantum algorithm, and analyze the cost of implementing that analysis on a classical computer in this case.)

(b) (Big picture) Based on the previous part, what property of a quantum algorithm is necessary (although not sufficient) in order to have an exponential speed-up versus any classical algorithm?