Assignment 3 (updated October 7, 9:15am)
Due date: 11:59pm, October 11, 2024

1. **Function $f : \{00, 01, 10\} \to \{0, 1\}$ with one or three 1s [15 points].** Suppose that you are given a black-box that computes a function $f : \{00, 01, 10\} \to \{0, 1\}$ such that the function takes the value 1 for a single input, or the function is 1 for all three inputs. The function takes two bits as input and if it is queried at the "out of range" point 11 then it takes the value 0. In summary, there are four possible functions with this property:

x	$f_{100}(x)$
00	1
01	0
10	0
11	0

x	$f_{010}(x)$
00	0
01	1
10	0
11	0

x	$f_{001}(x)$
00	0
01	0
10	1
11	0

x	$f_{111}(x)$
00	1
01	1
10	1
11	0

Your goal is to determine which of the four functions your black-box is, where you are allowed to query f at any point in $\{00, 01, 10, 11\}$.

(a) How many classical queries are there needed for this problem? Justify your answer.

(b) Give a quantum algorithm that solves this problem making one query to the function and explain why it works.

2. **Modulo 3 version of constant vs. balanced [15 points].** Let $f : (\mathbb{Z}_3)^n \to \mathbb{Z}_3$ have the property that, *either* f is "constant" (meaning that there exists an $c \in \mathbb{Z}_3$ such that, for all $a \in (\mathbb{Z}_3)^n$, it holds that $f(a) = c$), *or* f is "balanced," in the sense that it takes on all three values in an equal number of places. In other words, balanced means:

$$\big|\{a \in (\mathbb{Z}_3)^n : f(a) = 0\}\big| = \big|\{a \in (\mathbb{Z}_3)^n : f(a) = 1\}\big| = \big|\{a \in (\mathbb{Z}_3)^n : f(a) = 2\}\big| = 3^{n-1}.$$

You are primised that f is either constant or balanced and your goal is to distinguish between the two cases.

(a) [6 points] How many classical f-queries are there needed for this problem? Justify your answer.

(b) [9] A quantum f-query is the unitary operation U_f acting on $n+1$ qutrits such that, for all $a \in (\mathbb{Z}_3)^n$ and $b \in \mathbb{Z}_3$, $U_f|a_1, \ldots, a_n\rangle|b\rangle = |a_1, \ldots, a_n\rangle|b + f(a_1, \ldots, a_n) \bmod 3\rangle$.

Give a quantum algorithm that solves this problem making one f-query and explain in detail why it works. You may assume that, in addition to an f-query gate, the algorithm can apply Fourier transform gates, F_3, inverse Fourier transform gates, F_3^*, and measurements in the computational basis.

3. **Variant of the Simon mod p problem [15 points].** Let p be a prime number and consider the vector space $(\mathbb{Z}_p)^3$ over $\mathbb{Z}_p$ (i.e., where the scalars are the elements of $\mathbb{Z}_p$). Let $S \subset (\mathbb{Z}_p)^3$ be a 2-dimensional subspace, which means there are linearly independent $v_1, v_2 \in (\mathbb{Z}_p)^3$ such that $S = \{c_1 v_1 + c_2 v_2 : \text{such that } c_1, c_2 \in \mathbb{Z}_p\} = \text{span}(v_1, v_2)$.

   Let $f : (\mathbb{Z}_p)^3 \to (\mathbb{Z}_p)^3$ have the property that $f(a) = f(b)$ if and only if $a - b \in S$.

   Our goal is to determine S with a quantum algorithm that makes a single f-query. We do this in two steps:

   (a) [9 points] Define $S^\perp = \{w \in (\mathbb{Z}_p)^3 : \text{such that } w \cdot v = 0, \text{ for all } v \in S\}$. Describe and analyze a quantum algorithm that makes a single f-query and produces a uniformly distributed random element of $S^\perp$. Include a proof that the output of the algorithm is as above.

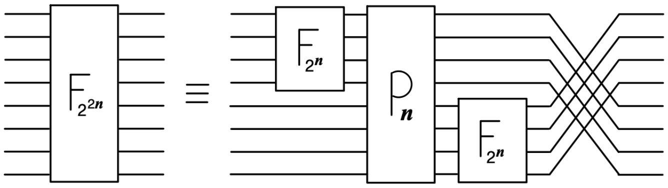
   Hint: This is similar to the query algorithm for Simon mod m (section 9.3 in the *Algorithms* notes; but the details are not identical).

   (b) [6] Assume that you have a uniformly distributed random $w \in S^\perp$. What is the probability that $w \neq (0,0,0)$? If $w \in S^\perp$ and $w \neq (0,0,0)$ then show how to use w to construct a basis for S.

4. **Another construction of the quantum Fourier transform [15 points].** Here we consider another, recursive, construction of F_{2^n}, the quantum Fourier transform on n qubits. This construction works if n is a power of 2. The base case of the recursion is where $n = 1$, in which case $F_{2^n} = H$. Suppose that we have an implementation of F_{2^n} and want to implement $F_{2^{2n}}$. Divide the $2n$ qubits into two registers: the first n qubits, and the last n qubits. Define the unitary P_n on two registers such that

$$P_n|a\rangle|b\rangle = \left(e^{2\pi i/2^{2n}}\right)^{m(a,b)}|a\rangle|b\rangle, \tag{1}$$

   for all $a, b \in \{0,1\}^n$, and where $m(a,b)$ denotes the product of a and b as n-bit integers (e.g., 1101 denotes 13, and 0100 denotes 4, so $m(1101, 0100) = 13 \times 4 = 52$). Then the following circuit computes $F_{2^{2n}}$ in terms of F_{2^n}, P_n, and SWAP gates:



   (a) [5 points] Show that this recursive construction holds for the $n = 1$ case, where F_4 is computed in terms of F_2 $(= H)$, P_1, and a SWAP gate.

   (b) [5 points] Show that this recursive construction holds for the $n = 2$ case, where F_{16} is computed in terms of F_4, P_2, and SWAP gates.

   (c) [5 points] Show that this recursive construction holds for all $n \geq 1$, where $F_{2^{2n}}$ is computed in terms of F_{2^n}, P_n and SWAP gates. (If you answer part (c) correctly, you receive full marks for question 4.)

Note: This approach leads to a construction of F_{2^n} consisting of $O(n \log n)$ gates.

5. **(This is an optional question for bonus credit)**
Function $f : \{00, 01, 10\} \to \{0, 1\}$ with an odd number of 1s revisited [8 points]. As in question 1, you are given a black-box computing a function $f : \{00, 01, 10\} \to \{0, 1\}$ such that the number of inputs where the function has value 1 is odd. However, now the value at the "out of range" point 11 is *arbitrary.* Now there are these four possibilities

x	$f_{100}(x)$
00	1
01	0
10	0
11	$*$

x	$f_{010}(x)$
00	0
01	1
10	0
11	$*$

x	$f_{001}(x)$
00	0
01	0
10	1
11	$*$

x	$f_{111}(x)$
00	1
01	1
10	1
11	$*$

where $*$ means that the bit could be *either* 0 *or* 1 *and you don't know which it is.* Is there a quantum algorithm that solves *this* version of the problem (determining which of the four cases) with one single query? If your answer is yes then you must give the quantum algorithm. If your answer is no then you must prove that there is no one-query quantum algorithm.

There is a solution that can be explained in less than one page. If you submit a solution to this question, then please do not exceed two pages.