Quantum Computation **CMU 15-859BB, Fall 2015**

HOMEWORK 3

Due: Tuesday, Oct. 6, 11:59pm, email the pdf to pgarriso@andrew.cmu.edu

---

Solve any 5 out of 6

1. **[Swap test.]** The CSWAP (controlled-swap) linear operator acts on three registers ("wires"), the first being a qubit and the second and third being d-dimensional qudits. It is defined by $|0\rangle|\psi\rangle|\varphi\rangle \mapsto |0\rangle|\psi\rangle|\varphi\rangle$ and $|1\rangle|\psi\rangle|\varphi\rangle \mapsto |1\rangle|\varphi\rangle|\psi\rangle$.

   (a) Verify that CSWAP is unitary. Sketch the 18×18 matrix representing it in case $d = 3$. Show that if $d = 2^c$ and the qudits are represented by c qubits each, then CSWAP can be implemented with $3c$ CCNOT gates. (You may appeal to an earlier homework problem, even if you didn't solve it.)

   (b) To perform the "swap test" on (unentangled) qudits $|\psi\rangle$ and $|\varphi\rangle$ means to do the following:

   i. Initialize the state $|0\rangle|\psi\rangle|\varphi\rangle$.
   ii. Apply a Hadamard gate to the first register.
   iii. Apply CSWAP to the three registers.
   iv. Apply a Hadamard gate to the first register.
   v. Measure the first register.
   vi. "Accept" if the outcome is $|0\rangle$ and "reject" if the outcome is $|1\rangle$.

   Show that the probability of accepting is $\frac{1}{2} + \frac{1}{2}|\langle\varphi|\psi\rangle|^2$. In particular, the probability is 1 if the qudits are identical and is $\frac{1}{2}$ if they are orthogonal.

2. **[Reflections.]** A *reflection (through a subspace)* in $\mathbb{R}^N$ is a linear transformation R satisfying one of the following equivalent conditions:

   (i) $R^2 = I$, where I is the identity transformation. (Math-nerd terminology: "R is an involution".)

   (ii) There is a subspace, V, such that $R = 2\Pi_V - I$, where Π_V denotes projection onto V (i.e., $\sum_{i=1}^{d} |v_d\rangle\langle v_d|$, where $|v_1\rangle, \ldots, |v_d\rangle$ are an orthonormal basis of V). Equivalently, $R = I - 2\Pi_{V^\perp}$, where $V^\perp$ is the orthogonal complement of V.

   (iii) R is an orthogonal (=real unitary) transformation with all its eigenvalues ± 1. (Eigenvalues 1 on the subspace V, eigenvalues -1 on $V^\perp$.)

   (iv) You know, R is a reflection through a subspace V. You know what that means.

   You do not have to prove the above equivalences.[1]

   (a) Grover's algorithm, when applied to $f : \{0,1\}^n \to \{0,1\}$, begins by preparing the uniform superposition $|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} |x\rangle \in \mathbb{R}^N$ (where, as usual, $N = 2^n$). It then repeatedly applies the oracle $O_f^{\pm}$ (which negates all amplitudes on $|x\rangle$'s with $f(x) = 1$) and the "diffusion" operator D (which flips all amplitudes across the average amplitude). Show that both $O_f^{\pm}$ and D are reflections. What subspace does D reflect across? What subspace's orthogonal complement does $O_f^{\pm}$ reflect across?

[1] The only real trick is showing $R^2 = I$ implies R is diagonalizable. The cute trick for this: any vector u satisfies $u = \frac{1}{2}(u + Ru) + \frac{1}{2}(u - Ru)$, and the two summands on the right are easily seen to be eigenvectors...

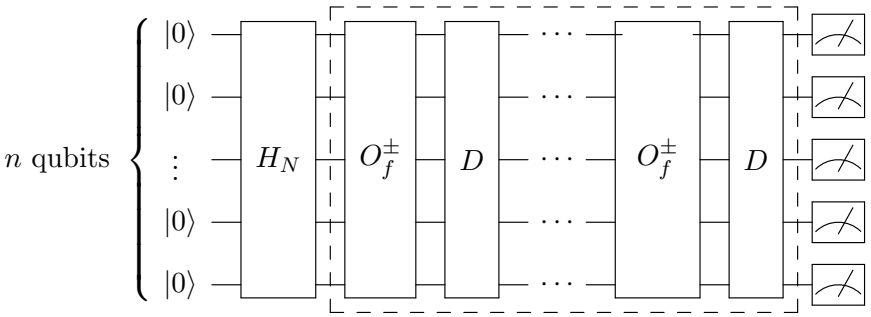
(b) Suppose R_1 is a reflection across the one-dimensional space spanned by the unit vector $|v_1\rangle$, and similarly for R_2 and $|v_2\rangle$. Let $S = R_2R_1$. Show that if $v \in V := \text{span}(|v_1\rangle, |v_2\rangle)$ then $Sv \in V$, and that if $v \in V^\perp$ then $Sv = v$; i.e., S essentially acts only in the two-dimensional space V. Draw a picture illustrating the action of S within V, letting θ denote the angle from $|v_1\rangle$ to $|v_2\rangle$. Geometrically, what kind of operation is S? Give an algebraic proof of this by multiplying two 2×2 matrices and using a trig identity.

3. **[Grover search with multiple satisfying inputs.]** In this problem we will see an alternate analysis of the Grover search algorithm, and also generalize it to the case when f has more than one satisfying input. The alternate analysis is more geometric, and follows the "product of two reflections" ideas from Problem 2.

So suppose we are given an oracle O_f for $f : \{0,1\}^n \to \{0,1\}$. Write $A = \{x : f(x) = 1\}$ and $k = |A|$. We will assume $k \geq 1$. Also write $B = \{x : f(x) = 0\}$, so $|B| = N - k$, where $N = 2^n$.

(a) Explain how we can use the oracle to check if a given string $y \in \{0,1\}^n$ has $f(y) = 1$. Explain how we can use this to find an $x \in A$ with high probability in $O(1)$ queries whenever $k \geq N/2$. (We henceforth assume $k < N/2$.)

(b) Recall Grover's algorithm, with t repetitions:



t repetitions of $O_f^\pm$ followed by D

Letting $|\psi^{(t)}\rangle$ denotes the state of the circuit after t repetitions, show that we can write it as

$$|\psi^{(t)}\rangle = \alpha_t \frac{1}{\sqrt{k}} \sum_{x \in A} |x\rangle + \beta_t \frac{1}{\sqrt{N-k}} \sum_{x \in B} |x\rangle ,$$

where $\alpha_t, \beta_t \in \mathbb{R}$ satisfy $\alpha_t^2 + \beta_t^2 = 1$. What are α_0 and β_0?

(c) Thinking of (β_t, α_t) as a point on the unit circle in $\mathbb{R}^2$, let us write θ_t for its angle from the horizontal axis (so that $\alpha_t = \sin\theta_t$, $\beta_t = \cos\theta_t$). Show that the transformation $(\beta_t, \alpha_t) \mapsto (\beta_{t+1}, \alpha_{t+1})$ is precisely rotation around the circle by an angle of $2\theta_0$.

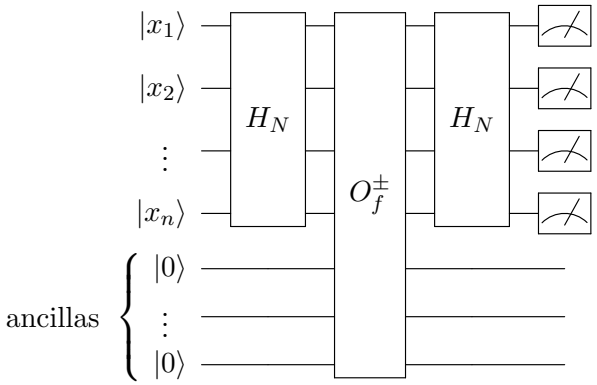
(d) Assume that the algorithm *knows* k. Briefly, why would the algorithm like to choose $t = \frac{1}{2}\left(\frac{\pi}{2\theta_0} - 1\right)$ if it could? Show that if it takes t to be the closest integer to this value, the circuit has the property that it outputs an element of A with probability at least $\frac{1}{2}$.

(e) Again, assuming the algorithm knows k, show that it can find an element of A with high probability using $O(\sqrt{N/k})$ queries to the oracle. (You may want to use that $\sin\theta \leq \theta$ for all $\theta \geq 0$.)

(f) **(Bonus.)** Show that if the algorithm does *not* know k, it can nevertheless find an $x \in A$ with high probability using $O(\sqrt{N/k})$ queries. (Hint: it's not quite as easy as "try $t = 0, 1, 2, 4, 8, 16, \ldots$", since the basic Grover succeeds only with probability $\frac{1}{2}$. You'll want to show that once T is "large enough", if we pick $t \in \{0, 1, 2, \ldots, T\}$ uniformly at random then there is a constant chance Grover will succeed. Then grow T at a slow exponential rate. . . .)

4. **[The necessity of uncomputing.]** Recall the convention that the oracle gate $O_f^{\pm}$ for a Boolean function $f : \{0,1\}^n \to \{0,1\}$ denotes the unitary transformation $|x\rangle \mapsto (-1)^{f(x)} |x\rangle$. When implementing $O_f^{\pm}$ in applications (such as SAT-solving), we have seen that we might need additional ancilla/garbage bits, in which case $O_f^{\pm}$ *actually* denotes the unitary transformation $|x\rangle |0^m\rangle \to (-1)^{f(x)} |x\rangle |g(x)\rangle$, where $g(x)$ is whichever m-bit garbage string produced on input x. In class, we have insisted that all oracle circuits *uncompute* their garbage, meaning that $g(x) = 0^m$ for all $x \in \{0,1\}^n$ (which is without loss of generality by Problem 4 from Homework 1).

   In this problem, we will justify why it is okay to "pretend" the ancilla bits don't exist. Furthermore, we will show why it is important for the circuit to uncompute its garbage. We will use as our example the Deutsch–Jozsa circuit; including ancilla/garbage bits, it is drawn as follows.



   (a) Suppose $O_f^{\pm}$ uncomputes its garbage, i.e. $g(x) = 0^m$ for all x. For a given Boolean function $f : \{0,1\}^n \to \{0,1\}$ (which need not be all-0's or balanced), compute the state of the system after the $O_f^{\pm}$ gate and after the second H_N gate. Show that each of these states can be written as $|\psi_1\rangle \otimes |\psi_2\rangle$, where $|\psi_1\rangle$ is the state of the first n qubits and $|\psi_2\rangle$ is the state of the last m qubits. Finally, show that the distribution on measurement outcomes is the same as it would have been if $O_f^{\pm}$ had no ancilla bits.

   (b) Suppose $O_f^{\pm}$ does not uncompute its garbage, i.e. $g : \{0,1\}^n \to \{0,1\}^m$ is allowed to be arbitrary. For a given Boolean function $f : \{0,1\}^n \to \{0,1\}$, compute the state of the system after the $O_f^{\pm}$ gate and after the second H_N gate. Explain why in general the distribution on measurement outcomes is *not* the same as it would have been if $O_f^{\pm}$ had no ancilla bits.

5. **[Connectivity.]** Suppose we are given an oracle O_G for an undirected simple n-vertex *graph* G. We assume its action is $|i\rangle |j\rangle |b\rangle \mapsto |i\rangle |j\rangle |b \oplus G_{ij}\rangle$, where $1 \le i, j \le n$ are vertex

indices (expressed with $\lceil \log_2 n \rceil$ qubits each) and

$$G_{ij} = \begin{cases} 1 & \text{if edge } \{i,j\} \text{ is present in } G, \\ 0 & \text{if edge } \{i,j\} \text{ is absent in } G. \end{cases}$$

(a) Describe a quantum algorithm that correctly decides (with high probability) whether G is a *connected* graph, using only $O(n^{3/2})$ queries to O_G. You can and should use Problem 3f (even if you didn't solve it). You may also describe your algorithm in a mix of conventional classical pseudocode-language and circuit language; i.e., you can say things like, "The algorithm now constructs a reversible classical circuit that does [*simple classical task*] and plugs it into a quantum circuit as follows...". It would be cool if you also explained why your algorithm: i) returned a spanning tree of G, provided G is connected; ii) has overall running time $\widetilde{O}(n^{3/2})$ (assuming that evaluating O_G takes unit time).[2]

(b) Prove that any classical (even randomized) query algorithm for deciding connectivity requires $\Omega(n^2)$ queries to succeed with high probability. (Hint: imagine an adversary answering the queries who always pretends that vertices $1, \ldots, n/2$ and vertices $n/2 + 1, \ldots, n$ form cliques, but is cagey about the existence of "cross-edges"...)

6. **[Multiplication/Fourier transform, iterated.]** In this problem we think of $\{0,1\}^n$ as $\mathbb{F}_2^n$; i.e., length-n vectors of integers under the operation of addition-mod-2. We also write $N = 2^n$.

(a) Let $f, g : \mathbb{F}_2^n \to \mathbb{F}_2$. Define

$$\Phi_{f,g} = \frac{1}{\sqrt{N}^3} \sum_{x,y \in \mathbb{F}_2^n} (-1)^{f(x)} (-1)^{x \cdot y} (-1)^{g(y)},$$

where $x \cdot y$ is the dot-product, $\sum_{i=1}^n x_i y_i \pmod 2$. Show that there is a quantum query algorithm that outputs YES with probability $\frac{1}{2} + \frac{1}{2}\Phi_{f,g}$ and NO otherwise. Your algorithm should make only 1 query; to facilitate this, we assume that instead of separate oracles $O_f^{\pm}$ and $O_g^{\pm}$ we have a single oracle $O_{f,g}^{\pm}$ with the following behavior: on input $|0\rangle |x\rangle$ it outputs $(-1)^{f(x)} |0\rangle |x\rangle$ and on input $|1\rangle |x\rangle$ it outputs $(-1)^{g(x)} |1\rangle |x\rangle$. Your algorithm should use at most $O(n)$ other gates.

(b) Show how to modify your algorithm with a little classical (randomized) post-processing so that it has the following guarantee: it outputs YES with probability at least .6 if $\Phi_{f,g} \geq \frac{3}{5}$ and outputs NO with probability at least .6 if $|\Phi_{f,g}| \leq \frac{1}{15}$. (Remark: it is known that any classical (randomized) query algorithm satisfying this guarantee requires $\Omega(\sqrt{N}/\log N)$ queries. This large gap — 1 quantum query versus $\widetilde{\Omega}(\sqrt{N})$ randomized queries — is known to be essentially maximal.)

(c) More generally, let $f_0, \ldots, f_{K-1} : \mathbb{F}_2^n \to \mathbb{F}_2$ and define $\Phi_{f_0,\ldots,f_{K-1}}$ to be $\frac{1}{\sqrt{N}^{K+1}}$ times the sum, over all vectors $x_0, \ldots, x_{K-1} \in \mathbb{F}_2^n$, of

$$(-1)^{f(x_0)} (-1)^{x_0 \cdot x_1} (-1)^{f_1(x_1)} (-1)^{x_1 \cdot x_2} (-1)^{f_2(x_2)} \cdots (-1)^{x_{K-2} \cdot x_{K-1}} (-1)^{f_{K-1}(x_{K-1})}.$$

[2] The $\widetilde{O}(\cdot)$ here denotes omission of $\log n$ factors. We put that there to avoid excruciatingly boring debates about the exact cost of transforming between Turing machine code and circuits, whether or not we're using the Turing machine model or the word RAM model, etc. etc.

Show that there is a quantum query algorithm that outputs YES with probability $\frac{1}{2} + \frac{1}{2}\Phi_{f_0,\ldots,f_{K-1}}$, NO otherwise, makes just $K/2$ queries, and uses $O(Kn)$ other gates. For simplicity, you may assume K is a power of 2 and that there is one oracle $O^{\pm}$ that on input $|j\rangle\,|x\rangle$ outputs $(-1)^{f_j(x)}\,|j\rangle\,|x\rangle$, where j is a $\log_2 K$-qubit string and x is an n-qubit string.

(d) **(Bonus.)** Let "Task T" mean "output YES with probability at least .6 if $\Phi_{f_0,\ldots,f_{K-1}} \geq \frac{3}{5}$ and output NO with probability at least .6 if $|\Phi_{f_0,\ldots,f_{K-1}}| \leq \frac{1}{15}$. Show that the ability to do Task T when $K = \text{poly}(n)$ and $f_0, \ldots, f_{K-1}$ are given explicitly as polynomial-size classical circuits is "complete for quantum computation". (I.e., any computational decision problem that can be solved in $\text{poly}(n)$ size and error at most .4 by a quantum circuit can also be solved in $\text{poly}(n)$ size by a classical circuit given the ability to do Task T.)