Quantum Computation **CMU 15-859BB, Fall 2018**

Work #8: Nov. 4 — Nov. 15

16-hour biweek

Obligatory problems are marked with [**]

---

1. **[The cyclic group.]** The cyclic[1] group of size N is the set $\mathbb{Z}_N$ of integers modulo N, together with the operation of addition (modulo N). Of course, the neutral element is 0, and the inverse of $u \in \mathbb{Z}_N$ is $-u = N - u$.

   (a) [**] The *order* of an element $u \in \mathbb{Z}_N$, denoted $\text{ord}_{\mathbb{Z}_N}(u)$ or just $\text{ord}(u)$, is the least positive integer r such that

   $$\underbrace{u + u + u + \cdots + u}_{r \text{ times}} = 0. \tag{1}$$

   This is called "order" because it is the order (size) of the subgroup $H = \{0, u, u+u, u+u+u, \cdots\}$ of $\mathbb{Z}_N$ generated by u.
   Print a table showing all of the elements of $\mathbb{Z}_{96}$, together with their orders. Obviously, you will want to use a computer to help you.

   (b) [**] Prove that

   $$\text{ord}(u) = \frac{N}{\text{GCD}(u, N)} = \frac{\text{LCM}(u, N)}{u},$$

   where $\text{LCM}(u, N)$ denotes the least common multiple of u and N, with u and N being treated as natural numbers. (Nitpick: ignore the LCM formula when $u = 0$.)

   (c) [**] Let 2^K be the largest power of 2 dividing N, and assume $K \geq 1$ (i.e., that N is even). Prove that:

   - If $u \in \mathbb{Z}_N$ is odd, then the largest power of 2 dividing $\text{ord}(u)$ is 2^K.
   - If $u \in \mathbb{Z}_N$ is even, then the largest power of 2 dividing $\text{ord}(u)$ is *not* 2^K.

   (d) [**] Let N_1 and N_2 be even positive integers. Suppose we pick $u_1 \in \mathbb{Z}_{N_1}$ and $u_2 \in \mathbb{Z}_{N_2}$ independently and uniformly at random. Prove that with probability at least $1/2$, the largest power of 2 dividing $\text{ord}_{\mathbb{Z}_{N_1}}(u_1)$ is distinct from the largest power of 2 dividing $\text{ord}_{\mathbb{Z}_{N_2}}(u_2)$.

   (e) [**] Let N_1 and N_2 be positive integers. The group $\mathbb{Z}_{N_1} \times \mathbb{Z}_{N_2}$ is the set of all pairs (u_1, u_2) where $x \in \mathbb{Z}_{N_1}$, $y \in \mathbb{Z}_{N_2}$, with the operation $+$ defined by

   $$(u_1, u_2) + (v_1, v_2) = (u_1 + v_1 \pmod{N_1}, \quad u_2 + v_2 \pmod{N_2}),$$

   neutral element $(0, 0)$, and inverse of (u_1, u_2) being $(-u_1, -u_2)$.
   The *order* of $u = (u_1, u_2) \in \mathbb{Z}_{N_1} \times \mathbb{Z}_{N_2}$, denoted $\text{ord}_{\mathbb{Z}_{N_1} \times \mathbb{Z}_{N_2}}(u)$, is again defined to be the least positive integer r such that Equation (1) holds. Prove that

   $$\text{ord}_{\mathbb{Z}_{N_1} \times \mathbb{Z}_{N_2}}(u) = \text{LCM}\Big(\text{ord}_{Z_{N_1}}(u_1), \text{ord}_{Z_{N_2}}(u_2)\Big).$$

[1] This name invites comparisons with the "dihedral group", discussed in Lecture 17. You can also think of the cyclic group as all the permutations $\pi : \{1, 2, \ldots, N\} \to \{1, 2, \ldots, N\}$ that are self-isomorphisms of the *directed* cycle graph.

2. **[The last obligatory number theory problem?]** [**] Recall that in Shor's factoring algorithm the input was a large number $B = P \cdot Q$, assumed to be the product of two odd primes. We argued that it would be easy to factor B given a "nontrivial square root" R of 1 (mod B), meaning a number $R \in \mathbb{Z}_B^*$ with $R^2 \equiv 1$ and $R \not\equiv \pm 1$. The strategy for finding R was:

   - Pick $A \in \mathbb{Z}_B^*$ uniformly at random.
   - Use a quantum computer to determine the *order* L of A, meaning the least positive integer such that $A^L \equiv 1 \pmod{B}$.
   - Hope for two "lucky" things: (i) L is even; (ii) $A^{L/2} \neq -1 \pmod{B}$.

   If both lucky things happen, then we could take $R = A^{L/2} \pmod{B}$. (Note that $A^{L/2} \not\equiv 1$ because of the minimality of L.) As stated in class, one can show that the probability (over the choice of A) that both lucky things happen is at least $1/2$. You will show that in this problem.

   Recall from Problem 6, Homework 4 that, for prime P, the elements of $\mathbb{Z}_P^*$ are of the form $G, G^2, \ldots, G^{P-1}$ for some "generator" $G \in \mathbb{Z}_P^*$. If you think about it, this means that the "group $\mathbb{Z}_P^*$ with operation multiplication", is the same as the "cyclic group $\mathbb{Z}_{P-1}$ with operation addition", in the sense that if you write down the "operation tables" (multiplication table in the former case, addition table in the latter case) they're exactly the same — it's just that the "elements have been renamed". In mathspeak, the two groups are *isomorphic*. The "isomorphism" is that $G^u \in \mathbb{Z}_P^*$ maps to $u \in \mathbb{Z}_{P-1}$; then indeed, $G^u \cdot G^v = G^{u+v}$. A remark that will be important later: the element -1 of $\mathbb{Z}_P^*$ must be $G^{\frac{P-1}{2}}$ (because $(-1)^2 = 1$); this maps to the element $\frac{P-1}{2} \in \mathbb{Z}_{P-1}$, the unique element of order 2.

   Recall also the Chinese Remainder Theorem from Problem 7, Homework 5. It effectively shows that the group $\mathbb{Z}_B^*$ with operation multiplication is "equivalent" (isomorphic) to the group $\mathbb{Z}_P^* \times \mathbb{Z}_Q^*$ with operation componentwise multiplication. (The isomorphism maps $X \in \mathbb{Z}_B^*$ to the pair $(S, T) \in \mathbb{Z}_P^* \times \mathbb{Z}_Q^*$, where $S = X \pmod{P}$ and $T = X \pmod{Q}$. In particular, $1 \in \mathbb{Z}_B^*$ is mapped to $(1, 1) \in \mathbb{Z}_P^* \times \mathbb{Z}_Q^*$ and $-1 \in \mathbb{Z}_B^*$ is mapped to $(-1, -1) \in \mathbb{Z}_P^* \times \mathbb{Z}_Q^*$.) Combining all our "isomorphisms", we get that the group $\mathbb{Z}_B^*$ with operation multiplication is equivalent to the "product of cyclic groups" $\mathbb{Z}_{P-1} \times \mathbb{Z}_{Q-1}$ with operation componentwise addition. In particular, $-1 \in \mathbb{Z}_B^*$ is mapped to $(\frac{P-1}{2}, \frac{Q-1}{2})$.

   Upshot: To analyze our strategy for finding a nontrivial square-root R of 1 (mod B), we may equivalently analyze the following:

   - Pick $u_1 \in \mathbb{Z}_{P-1}$ and $u_2 \in \mathbb{Z}_{Q-1}$ independently and uniformly at random. Let $u = (u_1, u_2) \in \mathbb{Z}_{P-1} \times \mathbb{Z}_{Q-1}$.
   - Let $L = \mathrm{ord}_{\mathbb{Z}_{P-1} \times \mathbb{Z}_{Q-1}}(u)$.
   - Hope for two "lucky" things: (i) L is even; (ii) $u + u + \cdots + u$ ($L/2$ times) is not equal to $(\frac{P-1}{2}, \frac{Q-1}{2})$.

   Prove that if the largest power of 2 dividing $\mathrm{ord}_{\mathbb{Z}_{P-1}}(u_1)$ is distinct from the largest power of 2 dividing $\mathrm{ord}_{\mathbb{Z}_{Q-1}}(u_2)$, then *both* lucky things happen. Conclude that the probability of both lucky things happening is at least $1/2$. (Of course, you will want to use the last few parts of Problem 1 in this problem.)

3. [**Miller–Rabin primality test.**] Having completed the preceding problems, you are now in the *perfect* position to understand the Miller–Rabin algorithm for classically efficiently (though probabilistically) testing whether a given number is prime. If you have never done this before, take this once-in-a-lifetime opportunity to study one of the infinitely many expositions of this algorithm/proof findable on the Internet. My particular recommendation is Chapters 10.8.1, 10.8.2 (and Exercises 10.39, 10.40) from the book *The Nature of Computation* by Moore and Mertens.

4. **[Order-finding reduces to factoring.]** Having completed the proof that factoring a number B classically efficiently (though probabilistically) reduces to "order-finding" in $\mathbb{Z}_B^*$, show the opposite reduction, thereby showing that factoring and order-finding are of "equivalent" complexity. More precisely, suppose you had a subroutine that could factor any number $B = P \cdot Q$ (for simplicity, just consider the product of two primes). Using it, give an efficient (deterministic) algorithm for finding the order of $A \in \mathbb{Z}_B^*$, given A and B.

5. **[The semiclassical Quantum Fourier Transform — or, how to do it with 1-qubit gates.]** The content of this problem is due to CMU's own (emeritus) Prof. Bob Griffiths and Chi-Sheng Niu, from 1996. It is of considerable practical importance for the implementation of any quantum algorithm that computes the quantum Fourier transform and then immediately measures the result (as Shor's algorithm does).

   (a) [**] Recall the "controlled phase gates" used in Lecture 14: if the control qubit is $|1\rangle$ then the unitary V_j is applied to the target qubit, where V_j leaves $|0\rangle$ alone and puts a phase of $\overline{\omega}_N^{2^{n-1-j}}$ onto $|1\rangle$. Show that these 2-qubit controlled phase gates are "symmetric", in the sense that they have the same operation if you reverse which qubit is the control and which qubit is the target.

   (b) [**] As a consequence of the previous problem, instead of drawing the circuit for the quantum Fourier transform like this —

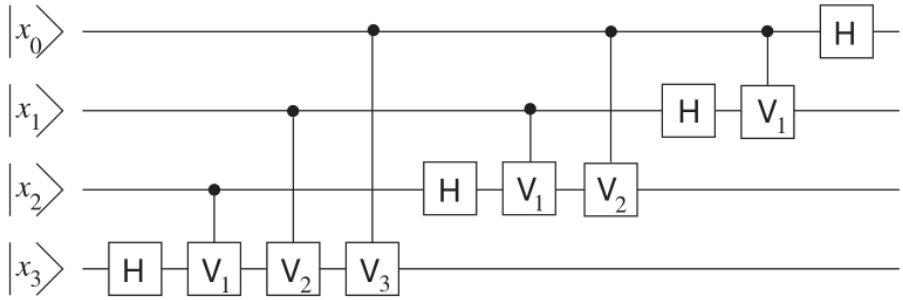

Figure 1: Original form of the QFT; the top output wire is the *most* significant bit of S.

   — you can equivalently draw it like this —

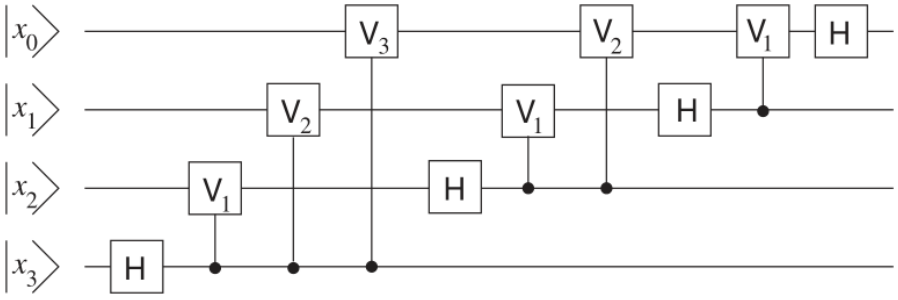

Figure 2: Equivalent form of the QFT.

   Incidentally, you can much more beautifully draw the circuit like this:

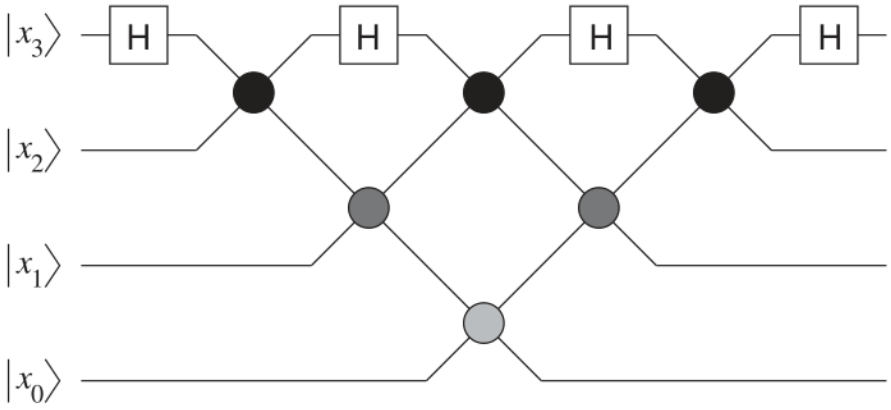

Figure 3: Beautiful form of the QFT circuit diagram. Since the controlled phase gates are symmetric, they can be depicted symmetrically, as big dots connecting two wires. The shading of these dots indicates how much phase is applied; the darker the dot, the larger the phase (i.e., the smaller the j in V_j, or the more "important" the phase). By the way, the very pale dots are the ones that you simply omit if you're doing the approximate version of the QFT.

Returning to Figure 2, notice that in this form, all the qubits' lifetimes end by being control bits. Suppose that (as in Shor's algorithm), we plan on immediately measuring all the qubits output by this circuit. Reminiscent of the Principle of Deferred Measurement, show that we'll obtain equivalent results if we measure each qubit after its Hadamard gate, and then use the outcome to *classically* control whether or not to apply the 1-qubit V_j gates, as in this diagram:[2]

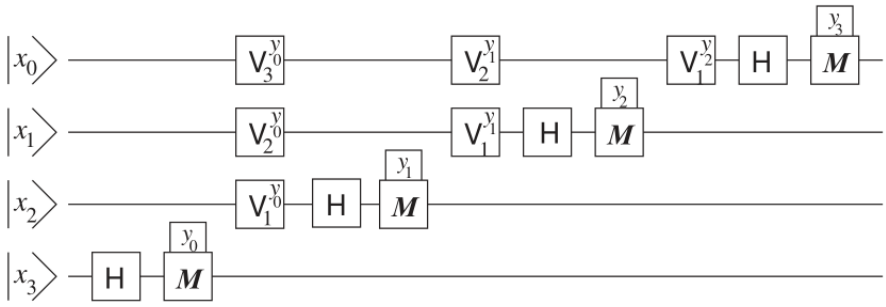

Figure 4: Equivalent form to Figure 2, assuming we are planning to measure at the end. $\boldsymbol{M}$ stands for measurement, and y_0, y_1, y_2, y_3 are what we usually called s_0, s_1, s_2, s_3.

[2] I borrowed all three diagrams from Mermin's book *Quantum Computer Science: An Introduction*, though he borrowed the beautiful third diagram from Griffiths and Niu.

6. **[Experimentally realizing Shor's algorithm.]** Both of the papers mentioned in this problem consider doing Shor's algorithm with the semiclassical quantum Fourier transform described in the previous problem.

   (a) [**] Read the paper Pretending to factor large numbers on a quantum computer by Smolin, Smith, and Vargo. Write a paragraph summarizing their main critique of prior experiments.

   (b) What was your favorite joke or easter egg in the paper?

   (c) [**] Read the paper Realization of a scalable Shor algorithm by Monz et al. Do you feel it adequately addresses the criticisms in the Smolin–Smith–Vargo paper? Why or why not?

7. **[Cosets.]** Let G be a finite group with operation $\circ$. Let H be a subgroup of G (potentially generated by any number of elements). Prove the following facts about cosets of H, used in Lecture 17.

   (a) For all $x \in G$, the coset xH has the same number of elements as the subgroup H.

   (b) Any two cosets xH and yH are either identical, or disjoint.

   (c) The set of all cosets of H forms a partition of G (meaning every element of G is in exactly one distinct coset).

8. **[Graph Isomorphism via the Hidden Subgroup Problem.]** Suppose G is an undirected graph with vertex set $\{1, 2, \ldots, n\}$ and edge set E. Let S_n denote the group of all permutations $\pi : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, n\}$, with the usual operation of composition $\circ$. Then $\mathrm{Aut}(G)$ is defined to be the subset of all permutations $\pi \in S_n$ that are *automorphisms* (self-isomorphisms) of G. Informally, this means that if you apply π to the vertex-names of G, you get the same graph G back. A bit more formally, it means that

$$E = \Big\{\{\pi(u), \pi(v)\} : \{u, v\} \in E\Big\}.$$

   (a) [**] Show that if π and $\sigma \in \mathrm{Aut}(G)$ then $\pi \circ \sigma$ and π^{-1} are also in $\mathrm{Aut}(G)$. Conclude that $\mathrm{Aut}(G)$ is a subgroup of S_n. (I suppose you should do this according to our definition that H is a subgroup of G if H is obtained from some "generators" $h_1, \ldots, h_k \in G$ by starting with the neutral element e and applying $\circ$ with $h_1, \ldots, h_k, h_1^{-1}, \ldots, h_k^{-1}$ "as much as possible". Note that there is no harm in taking "more generators than you need".)

   (b) [**] Let G be a graph with vertex set $\{1, 2, \ldots, n\}$, and let COLORS be the set of all $n \times n$ adjacency matrices of undirected graphs on vertex set $\{1, 2, \ldots, n\}$. Define $f_G : S_n \to \mathrm{COLORS}$ by letting $f_G(\pi)$ be the adjacency matrix of the graph obtained by permuting G's vertices according to π. Prove that f_G is "$\mathrm{Aut}(G)$-periodic", as defined in Lecture 17: for each coset of $\mathrm{Aut}(G)$, f_G gives the same "color" to all the elements of the coset; and, f_G gives different colors to different cosets.

   (c) Argue that there is an "efficient" (poly(n)-step) classical algorithm that, given n-vertex graph G (in adjacency matrix format) outputs (the description of) a classical circuit C computing the function f_G from the previous part — and hence there is also an efficient classical algorithm computing (the description of) a quantum circuit Q_F that "implements" f_G.

   (d) [**] Let G_1 and G_2 be graphs, each with vertex set $\{1, 2, \ldots, n\}$. We assume that G_1 and G_2 are connected graphs. Recall that G_1 and G_2 are said to be *isomorphic* if there exists a permutation $\pi \in S_n$ such that when π is applied to the vertices of G_1, the result is G_2. As a remark, the "Graph Isomorphism problem" — i.e., the computational task of determining whether two given G_1, G_2 are isomorphic or not — is not known to be solvable efficiently by any classical algorithm.[3]

   Let $G = G_1 \sqcup G_2$ be the $2n$-vertex graph formed by taking a disjoint copy of G_1 and G_2, and consider $\mathrm{Aut}(G) \subseteq S_{2n}$. Show that G_1 and G_2 are isomorphic if and only if there exists $\sigma \in \mathrm{Aut}(G)$ that "maps between halves", meaning $\sigma(i) = j$ for some $i \leq n$ and $j > n$.

   (e) [**] Suppose an algorithm obtains some generators $h_1, \ldots, h_k$ for a subgroup H of S_{2n}. Show that it can efficiently determine if there exists $\sigma \in H$ that "maps between halves".

   (f) [**] Suppose there were an efficient quantum algorithm for solving the Hidden Subgroup Problem for the group S_{2n}. Show that there would be an efficient quantum algorithm for solving the Graph Isomorphism problem.

[3] Recently Babai showed the problem could be solved in $n^{\mathrm{polylog}\, n}$ steps, but he is also on record as believing that it cannot be solved in fewer than $n^{\log n}$ steps.

9. **[Visualization of quantum physics.]** Watch this video, which I thought was nice.