

CSCE 440/640 Quantum Algorithms

Homework 2

Texas A&M U, Spring 2019
Lecturer: Fang Song

Jan. 28, 2019
Due: Feb. 13, 2019

Instructions. Only PDF format is accepted (type it or scan clearly). Your solutions will be graded on *correctness* and *clarity*. You should only submit work that you believe to be correct; if you cannot solve a problem completely, you will get significantly more partial credit if you clearly identify the gap(s) in your solution. For this problem set, a random subset of problems will be graded. Problems marked with “[G]” are required for graduate students. Undergraduate students will get bonus points for solving them.

You may collaborate with others on this problem set. However, you must *write up your own solutions* and *list your collaborators* for each problem.

1. (Quantum states and gates)

(a) (6 points) Let $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ and $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$.

- i) Suppose we have a qubit and we first apply X and then Z . Is it equivalent to first applying Z and then X ?
- ii) Suppose we have two qubits, and we apply X to both and then Z to both. Is it equivalent to applying Z to both and then applying X to both? Justify your answer.

(b) (6 points) (SWAP gate) A SWAP gate takes two inputs a and b and outputs b and a ; i.e., it swaps the values of two input registers. Show how to build a SWAP gate using only CNOT gates. (Hint: you’ll need 3 of them.)

(c) (5 points) [G] Show that every unitary one-qubit gate with real entries can be written as a rotation matrix, possibly preceded and followed by Z -gates. In other words, show that for every 2×2 real unitary U , there exist signs $s_1, s_2, s_3 \in \{1, -1\}$ and angle $\theta \in [0, 2\pi)$ such that

$$U = s_1 \begin{pmatrix} 1 & 0 \\ 0 & s_2 \end{pmatrix} \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & s_3 \end{pmatrix}.$$

2. (Product states versus entangled states) In each of the following, either express the 2-qubit state as a tensor product of 1-qubit states or prove that it cannot be expressed this way.

(a) (4 points) $\frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle - \frac{1}{2}|11\rangle$

(b) (4 points) $\frac{3}{4}|00\rangle + \frac{\sqrt{3}}{4}|01\rangle + \frac{\sqrt{3}}{4}|10\rangle + \frac{1}{4}|11\rangle$

3. (Distinguishing states by local measurements) Suppose Alice and Bob are physically separated from each other, and are each given one of the qubits of some 2-qubit state. They are required to distinguish between State I and State II with only local measurements. Namely they can each perform a local (one-qubit) unitary operation and then a measurement (in the computational basis) of their own qubit. After their measurements, they can send only classical bits to each other. (This is usually referred to as LOCC: local operation and classical communication.) In each case below, either give a perfect distinguishing procedure (that never errs) or explain why there is no perfect distinguishing procedure (i.e., that for any procedure the success probability must be less than 1).

- (a) (5 points) State I: $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$; State II: $\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$
- (b) (5 points) State I: $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$; State II: $\frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$
- (c) (5 points) [G] State I: $\frac{1}{\sqrt{2}}(|00\rangle + i|11\rangle)$; State II: $\frac{1}{\sqrt{2}}(|00\rangle - i|11\rangle)$

4. (Linear algebra)

- (a) (15 points) (Tensor product)
 - i) Show that $(A \otimes B) \cdot (C \otimes D) = (AC) \otimes (BD)$.
 - ii) Show that $(A \otimes B)^\dagger = A^\dagger \otimes B^\dagger$.
 - iii) If A and B are both invertible, show that so is $A \otimes B$.
 - iv) Show that $\text{tr}(A \otimes B) = \text{tr}(A) \cdot \text{tr}(B)$.
 - v) Show that if A and B are unitary matrices, then so is $A \otimes B$.
- (b) (6 points) Let $U = (v_1, \dots, v_n)$ be a unitary matrix and each $v_i \in \mathbb{C}^n$.
 - i) Show that $\{v_1, \dots, v_n\}$ form an orthonormal basis of \mathbb{C}^n .
 - ii) Show that the eigenvalues of any unitary U are of the form $e^{i\theta}$ for some $\theta \in [0, 2\pi)$.
- (c) (4 points) Show that for any $x \in \{0, 1\}^n$, $H^{\otimes n}|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \{0, 1\}^n} (-1)^{x \cdot y} |y\rangle$.
 $x \cdot y := \sum_{i=1}^n x_i y_i$ is the dot product over \mathbb{Z}_2^n .
- (d) (4 points) Let $x, y \in \{0, 1\}^n$ and let $s = x \oplus y$. Show that

$$H^{\otimes n} \frac{1}{\sqrt{2}}(|x\rangle + |y\rangle) = \frac{1}{\sqrt{2^{n-1}}} \sum_{z: z \cdot s = 0} (-1)^{x \cdot z} |z\rangle.$$

- (e) (8 points) For a vector $v = (v_0, \dots, v_{k-1}) \in \mathbb{C}^k$, let $\|v\| := \sqrt{\sum_{i=0}^{k-1} |v_i|^2}$, which is the usual Euclidean length of v . For any $k \times k$ matrix $M \in \mathbb{C}^{k \times k}$, define its *spectral norm* $\|M\|$ as $\|M\| = \max_{|\psi\rangle} \|M|\psi\rangle\|$, where the maximum is taken over quantum states (i.e., vectors $|\psi\rangle$ such that $\| |\psi\rangle \| = 1$). Define the distance between two $k \times k$ unitary matrices M_1 and M_2 as $\|M_1 - M_2\|$. Show that
 - i) $\|A - B\| \leq \|A - C\| + \|C - B\|$, for any three $k \times k$ matrices A , B , and C . (Namely, this distance measure satisfies the *triangle inequality*.)

ii) Show that, for any two $k \times k$ unitary matrices U_1 and U_2 , and any matrix A , $\|U_1AU_2\| = \|A\|$.

5. (Errors in randomized algorithms) Suppose you want to write a computer program C to compute a Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$, mapping n bits to 1 bit. If C is a deterministic algorithm, then “ C successfully computes f ” has a clear meaning that that $C(x) = f(x)$ for all inputs $x \in \{0,1\}^n$. But what if C is a probabilistic algorithm?

(a) (8 points) The best thing is if C is a *zero-error* algorithm with failure probability p . Namely

- on every input x , the output of $C(x)$ is either $f(x)$ or \perp (denoting failure).
- on every input x we have $\Pr[C(x) = \perp] \leq p$ (NB. the probability is only over the internal randomness of C , not the random choice of x).

i) If you have a zero-error algorithm C for f with failure probability 90%, show how to convert it to a zero-error algorithm C' with failure probability at most 2^{-500} . The “slowdown” should only be a factor of a few thousand.

ii) Alternatively, show how to convert C to an algorithm C'' for f which: (i) always outputs the correct answer, meaning $C''(x) = f(x)$ for all x ; (ii) has expected running time only a few powers of 2 worse than that of C . (Hint: look up the mean of a geometric random variable.)

(b) (5 points) The second best thing is if C is a one-sided error algorithm for f , with failure probability p . There are two kinds of such algorithms, “no-false-positives” and “no-false-negatives”. For simplicity, let’s just consider “no false-negatives” (the other case is symmetric);

- on every input x , the output $C(x)$ is either 0 or 1;
- on every input x such that $f(x) = 1$, the output $C(x)$ is also 1;
- on every input x such that $f(x) = 0$, we have $\Pr[C(x) = 1] \leq p$.

Show how to convert a no-false-negatives algorithm C for f with failure probability 90% to another no-false-negatives algorithm C' for f with failure probability at most 2^{-500} . The “slowdown” should only be a factor of a few thousand.

(c) (5 points) The third possibility (which is rare in practice) is if C is a two-sided error algorithm for f , with failure probability p . Namely,

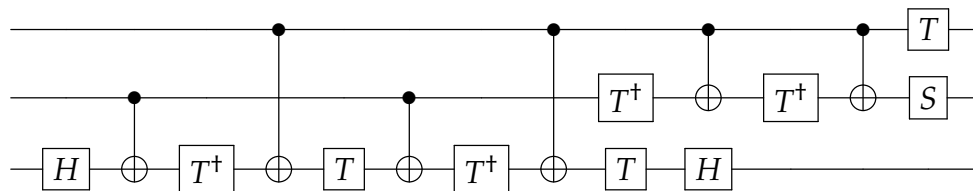
- on every input x , the output $C(x)$ is either 0 or 1.
- on every input x , we have $\Pr[C(x) \neq f(x)] \leq p$.

If you have a two-sided error algorithm C for f with failure probability 40%, show how to convert it to a two-sided error algorithm C' for f with failure probability at most 2^{-500} . The “slowdown” should only be a factor of a few dozen thousand. (Hint: look up the Chernoff bound.)

6. (Simple search algorithms) In the context of this question, we are interested in exact solutions (with failure probability zero).

- (a) (6 points) (1-out-of-4 search) Consider a black-box function $f : \{0,1\}^2 \rightarrow \{0,1\}$ with the property that there is a unique $x \in \{0,1\}^2$ such that $f(x) = 1$ and the goal is to determine x . How many classical queries are necessary to solve this problem? Design a quantum algorithm that finds x using 1 quantum query.
- (b) (6 points) [G] (2-out-of-4 search) Given a black-box for a function $f : \{0,1\}^2 \rightarrow \{0,1\}$ with exactly two $x \in \{0,1\}^2$ such that $f(x) = 1$ and the goal is to determine both x 's. Prove that 3 classical queries are necessary to solve this problem and that 2 quantum queries are sufficient to solve this problem.
7. (Simulating classical circuits) Let $f : \{0,1\}^2 \rightarrow \{0,1\}^2$ be a function such that $f(ab) = 0$ if $a = b = 1$ and $f(ab) = 1$ otherwise.
- (a) (3 points) Design a circuit using your favorite gate set (e.g., AND, OR, NOT) to compute f .
- (b) (3 points) Turn your circuit into a reversible circuit using Toffoli gate $T : a, b, c \mapsto a, b, a \wedge b \oplus c$ and other reversible gates. You may need to introduce ancilla bits
- (c) (4 points) Turn your reversible circuit into a unitary quantum circuit that implements the unitary $U_f : |x\rangle|y\rangle \mapsto |x\rangle|f(x) \oplus y\rangle$.
8. (Playing with quantum circuits)

- (a) (Exercise) Play around both the graphic composer and QASM editor [IBM Q experience](#) (or some other tools, e.g., [Quirky](#) and [Quantum playground](#)). Test the teleportation protocol.
- (b) (10 points) Determine the behavior of the following quantum circuit by implementing it (in graphic interface or programming it): You'll want to precede



this circuit by all 8 possible ways of doing or not doing NOT gates on the relevant 3 qubits, so as to see what this circuit does to each of the basic states $|000\rangle, |001\rangle, \dots, |111\rangle$.

9. (Watch in your leisure time. No grades.)
- (a) [Many Worlds Interpretation](#).
- (b) [Fast Fourier Transform](#).
- (c) The enormity of the [number \$2^{256}\$](#) .