Practice Final - 2019 Exam Answer Key

Introduction to Quantum Information Science

Your Name and EID: _____

Be sure to try all the problems! Don't spend too much time on any one of them. Also, even if you can't do the earlier parts of a problem, *be sure to look at the later parts,* since in some cases they're independent of the earlier parts. May you maintain your quantum coherence until the end of the exam!

[Optional] What is your favorite interpretation of quantum mechanics (Pick One)?

_____Copenhagen

_____Many-Worlds

_____Bohmian Mechanics

_____New Physics (Including Dynamical Collapse)

_____Other:

_____None

_____What does it even matter?

1. True or False? Write your answer on the provided line. [20 Points, 1 Point per Part]

__T__ **a)** Unitary matrices preserve the 2-norm of all complex vectors.

__F__ **b)** A pure state of n qubits is described by an n-dimensional complex unit vector.

__T__ **c)** The Bell inequality states that by using classical strategies, Alice and Bob can win the CHSH game with probability at most $\frac{3}{4}$.

__F__ **d)** Google's recent quantum supremacy experiment demonstrated the successful use of quantum error-correction.

__T__ **e)** Lattice-based cryptography is one proposal for secure post-quantum public-key cryptography.

__F__ **f)** The fastest known classical algorithms for factoring all take time c^n, for some $c > 1$, to factor an n-bit integer.

__T__ **g)** Grover's algorithm can find a marked item in a list of N items using $O(\sqrt{N})$ queries to the list, with high probability, even if the number of marked items is unknown at the start.

__T__ **h)** If Alice and Bob share a bipartite pure state, then their entanglement entropy is equal to the von Neumann entropy of Alice's local density matrix.

__T__ **i)** The eigenvalues of a unitary matrix are always complex numbers with absolute value 1.

__T__ **j)** The eigenvalues of a density matrix are always in $[0, 1]$.

__F__ **k)** For every density matrix, there is a unique probabilistic mixture of pure states that the density matrix represents.

__F__ **l)** If Alice and Bob share many entangled qubits, they can win the CHSH game with probability arbitrarily close to 1.

__T__ **m)** The only 2×2 matrices that are both unitary and stochastic are $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$.

__F__ **n)** In Simon's algorithm, once we have a state of the form $\frac{1}{\sqrt{2}}(|x\rangle + |x \oplus s\rangle)$, we can recover s with probability $\frac{1}{2}$ by measuring this state twice and taking the XOR of the measurements.

__T__ **o)** Fault-tolerant quantum computation requires a continual process of intermediate measurements and insertion of clean qubits.

__F__ **p)** As far as anyone knows, the use of qutrits rather than qubits as physical building blocks could lead to more problems being solvable in polynomial time by quantum computers.

__T__ **q)** While $\langle u|v\rangle$ and $\langle v|u\rangle$ might be different, they always have the same absolute value.

__T__ **r)** When applied to a list of size 4, with 1 marked item, Grover's algorithm succeeds after just a single iteration.

__F__ **s)** In QKD, if Eve knows only that some particular qubit is either $|+\rangle$ or $|-\rangle$, she cannot learn which without altering the qubit.

__F__ **t)** While there are many different proposals for physical realization of quantum computers, they all involve using the states of individual atomic nuclei or subatomic particles as qubits.

2. Short Answer [20 Points]

Consider the state:

$$|\psi\rangle = \frac{|00\rangle + |01\rangle + |10\rangle}{\sqrt{3}}$$

a) [5 points] Calculate the reduced density matrix of the second qubit of $|\psi\rangle$. *Solution.*

$$|\psi\rangle = \sqrt{\frac{2}{3}}\,|0\rangle\,|+\rangle + \frac{1}{\sqrt{3}}\,|10\rangle$$

Thus the reduced density matrix is

$$\frac{2}{3}\,|+\rangle\,\langle +| + \frac{1}{3}\,|0\rangle\,\langle 0| = \frac{2}{3}\begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix} + \frac{1}{3}\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} = \frac{1}{3}\begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$$

b) [5 points] Calculate $|\psi\rangle$'s entanglement entropy. You don't need to simplify your answer. *Solution. The characteristic polynomial is*

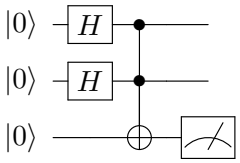
$$(\frac{2}{3} - \lambda)(\frac{1}{3} - \lambda) - \frac{2}{9} = \lambda^2 - \lambda - \frac{1}{9} \Rightarrow \lambda = \frac{1 \pm \sqrt{1 - \frac{4}{9}}}{2} = \frac{1}{2} \pm \frac{\sqrt{5}}{6}$$

From here we can use these eigenvalues to compute the entanglement entropy via

$$-\lambda_1 \log_2 \lambda_1 - \lambda_2 \log_2 \lambda_2$$

c) [5 points] Draw a quantum circuit, using Hadamard gates, Toffoli gates, and $\{|0\rangle, |1\rangle\}$ measurements, that prepares $|\psi\rangle$ from the all-0 initial state. Your circuit is allowed to use ancilla qubits, and is also allowed to prepare $|\psi\rangle$ only with $\frac{3}{4}$ success probability—for example, only if a measurement on ancilla qubit(s) yields some specific outcome.

Solution.



Before the measurement, we have the state

$$\frac{1}{2}(|000\rangle + |010\rangle + |100\rangle + |111\rangle)$$

Thus if we measure $|0\rangle$ on the third qubit then we'll get $|\psi\rangle$ on the first two qubits.

d) [5 points] What is the flaw in the following reasoning?

The Gottesman-Knill theorem states that any quantum circuit composed of Hadamard, CNOT, and Phase gates can be simulated classically in time polynomial in the size of the circuit. Simon's algorithm solves Simon's problem quantumly using only a polynomial number of Hadamard gates and $O(n)$ oracle queries. Therefore, Simon's problem can be solved classically in polynomial time using polynomially many oracle queries.

Solution. The quantum oracle might use non clifford gates, making it impossible to use the Gottesman-Knill theorem.

e) [3 points] Give a basis of eigenvectors for the 4×4 CNOT matrix, along with their associated eigenvalues.

Solution. We know $|00\rangle$ *and* $|01\rangle$ *are unaffected by CNOT. Similarly, we know that* $|+\rangle$ *and* $|-\rangle$ *are the eigenvalues of the NOT gate. Thus the eigenbasis is* $|00\rangle$, $|01\rangle$, $|1+\rangle$, *and* $|1-\rangle$ *with eigenvalues* 1, 1, 1, *and* -1 *respectively.*

3. Shor's Algorithm [10 Points]

Suppose we use Shor's algorithm to factor $N = 105$ into $3 \cdot 5 \cdot 7$. (Yes, N is now a product of 3 primes!) Suppose also that we make the choices $x = 2$ and $Q = 60000$.

a) [2 points] What is the order of the multiplicative group $\mathbb{Z}_N^\times$?
Solution. We still want items that are relatively prime to $N = 105$, so we compute

$$(3-1) \cdot (5-1) \cdot (7-1) = 48$$

b) [2 points] What is the period of the function $f(r) = x^r (\mathrm{mod} N)$?
Solution.

$$2^{12} = 1 \mod 105$$

c) [3 points] Suppose we factor $x^s - 1$ into $x^{s/2} - 1$ and $x^{s/2} + 1$, and then take the gcd of both factors with N itself. Which prime factors of N, if any, would be "peeled off" this way?

Solution.

$$2^6 \pm 1 = 63, 65$$

$$\gcd(105, 63) = \gcd(63, 42) = \gcd(42, 21) = \gcd(21, 0) = 21$$

$$\gcd(105, 65) = \gcd(65, 40) = \gcd(40, 15) = \gcd(15, 10) = 5$$

d) [3 points] After we apply the QFT to the $|r\rangle$ register and then measure that register, what are the possible results that we could observe?

Solution. Luckily $\frac{Q}{s} = 5000$ *is an integer, so we'll see integer multiples of* 5000.

4. QSampling [18 Points]

In the Graph Isomorphism problem, we're given as input two n-vertex undirected graphs G and H. The problem is to determine whether they're isomorphic—in other words, whether there's any permutation of the vertex labels that makes G and H equal.

a) [5 points] Given as input an n-vertex graph G, describe how to sample, in classical poly(n) time, from a probability distribution D_G over graphs such that:

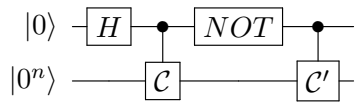
- Whenever the graphs G and H are isomorphic, $D_G = D_H$.
- Whenever G and H are non-isomorphic, D_G and D_H have disjoint supports (i.e., no graph appears with nonzero probability in both of them).

Solution. We can simply uniformly sample over permutations of G. If G and H are isomorphic, then their distributions D_G and D_H must be the same, since they are just permutations of one another. If they are not isomorphic, then by definition no permutation of G will turn into a graph that is a permutation of H meaning that they have disjoint supports.

b) [4 points] Given a probability distribution $D = (p_x)$ over n-bit strings x, define the "QSampling state" of D to be

$$|\psi_D\rangle := \sum_{x \in \{0,1\}^n} \sqrt{p_x}\,|x\rangle$$

Given two probability distributions D and D', suppose that the quantum circuit $\mathcal{C}$ maps $|0^n\rangle$ to $|\psi_D\rangle$, while the circuit $\mathcal{C}'$ maps $|0^n\rangle$ to $|\psi_{D'}\rangle$. Then what is the output state of the circuit shown below, which acts on $n+1$ qubits?



Solution.

$$\frac{1}{\sqrt{2}}(|0\rangle \sum_{x \in \{0,1\}^n} \sqrt{p_x}\,|x\rangle + |1\rangle \sum_{x \in \{0,1\}^n} \sqrt{p'_x}\,|x\rangle)$$

c) [5 points] Now suppose we measure the first qubit of that output state in the $\{|+\rangle, |-\rangle\}$ basis. What is the probability of the outcome $|+\rangle$ if $D = D'$? What about if D and D' have disjoint supports?

Solution. If $D = D'$, then we actually have an unentangled state, and the first qubit is simply $|+\rangle$, so we will always get it.

If D and D' have disjoint probabilities, then the two states in superposition are orthogonal. Thus, we have the maximally mixed state on the first qubit, so the probability to measure $|+\rangle$ is $\frac{1}{2}$.

d) [4 points] Suppose your distributions D_G from part (a) could be efficiently QSampled. Using your previous work, explain how Graph Isomorphism could then be solved in BQP (quantum polynomial time). *Solution. We can simply use the previous circuit and see if we ever measure $|-\rangle$ in the first qubit. Since we know that either the distributions are equivalent if the graphs are isomorphic, or disjoint if not, then we will only measure $|-\rangle$ if the graphs are not isomorphic. With exponentially low probability over n tries, the probability of failure can easily be made very small.*

e) [5 points, extra credit] So then why *doesn't* this approach immediately imply a fast quantum algorithm for Graph Isomorphism? Explain what could go wrong in passing from fast algorithms to sample D_G and D_H, to fast algorithms to QSample them.

Solution. It isn't obviously easy to QSample distributions that can even be classically sampled. One might be tempted to use the state

$$\frac{1}{n!} \sum_{\sigma \in S_{|V|}} |\sigma(G)\rangle |\sigma\rangle$$

where σ is a graph permutation, but this second register actually produces garbage.

5. Inner Product [10 Points]

Suppose Alice and Bob hold n-bit strings $x = (x_1, ..., x_n)$ and $y = (y_1, ..., y_n)$ respectively. One thing they might want to learn is the mod-2 inner product of their strings,

$$x_1 y_1 + \ldots + x_n y_n (\text{mod} 2).$$

a) [4 points] Suppose Alice and Bob had a quantum communication protocol in which they are allowed to exchange up to T qubits and to perform arbitrary local unitary transformations to their qubits (possibly including ancilla qubits), that ended with Bob knowing the above inner product, with success probability 1. Explain how, by exchanging the same number of qubits T, Bob could also prepare an n-qubit state of the form

$$\frac{1}{\sqrt{2^n}} \sum_{z \in \{0,1\}^n} (-1)^{x \cdot z} |z\rangle ,$$

where x is an n-bit string held by Alice.

Solution. Let's say Alice and Bob's joint state is $|x\rangle |0^k\rangle |y\rangle |0^l\rangle |0\rangle$ where the second and fourth registers are Alice and Bob's ancilla, and $|c\rangle$ is where we want to write the answer, and is held by Bob. Then we can view the protocol as a unitary U that maps

$$U |x\rangle |0^k\rangle |y\rangle |0^l\rangle |c\rangle = |x\rangle |a\rangle |y\rangle |b\rangle |c \oplus x \cdot y \bmod 2\rangle$$

Basically, we have something similar to an xor oracle. If Bob puts $|-\rangle$ into the answer register, then like the xor-to-phase oracle protocol we can get the state

$$(-1)^{x \cdot y \bmod 2} |x\rangle |a\rangle |y\rangle |b\rangle |-\rangle$$

If Bob also puts his first register into the state $\frac{1}{\sqrt{2^n}} \sum_{z \in \{0,1\}^n} |z\rangle$, then by linearity we'll end up with $\frac{1}{\sqrt{2^n}} \sum_{z \in \{0,1\}^n} (-1)^{x \cdot z} |z\rangle$ as desired.

b) [6 points] Assume Alice and Bob have no preshared entanglement. Recall Holevo's Theorem, which implies that in order to communicate n bits to Bob reliably, Alice must send Bob at least n qubits. Using Holevo's Theorem together with part (a), prove that Alice and Bob must exchange at least n qubits, even if they only want to learn the inner product mod 2 of their input strings x and y.

Solution. Note that

$$H^{\otimes n}\frac{1}{\sqrt{2^n}}\sum_{z\in\{0,1\}^n}(-1)^{x\cdot z}\left|z\right\rangle = \left|x\right\rangle.$$

This means that we've learned at least n-bits of information in this protocol. By Holevo's theorem, that means we need at least n-qubits.

c) [6 points, extra credit] Now suppose we're no longer working mod 2, and Alice and Bob want to know whether their inner product

$$x_1y_1 + \ldots + x_ny_n$$

is zero or nonzero as an integer. (In other words, whether there's an i such that $x_i = y_i = 1$.) Describe a protocol by which Alice and Bob can accomplish this, with high probability, by exchanging only $O(\sqrt{n}\log n)$ qubits in total. The qubits can be spread across as many rounds of communication as necessary, and can be sent in either direction.

Solution. Let's see if we can run grover's algorithm to search for a marked item where $x_i \cdot y_i = 1$. To do so, we'll need to implement the oracle

$$O\left|i\right\rangle = (-1)^{x_i\cdot y_i}\left|i\right\rangle$$

WLOG, we'll have Alice have the final state when the oracle is implemented, such that she starts with the state $\left|i\right\rangle\left|0\right\rangle\left|0\right\rangle$. Let Alice prepare the state $\left|i\right\rangle\left|x_i\right\rangle\left|0\right\rangle$, which is free, since we are allowed arbitrary local operations and Alice fully knows what x is (this is equivalent to applying the xor oracle for $f(i) = x_i$). This state requires $\log n + 2$ bits so we'll have Alice send them to Bob.

Now let Bob start with the state $\left|i\right\rangle\left|x_i\right\rangle\left|0\right\rangle\left|-\right\rangle$ using the qubits Alice sent over, plus an ancilla. We'll start by having Bob turn his state into $\left|i\right\rangle\left|x_i\right\rangle\left|y_i\right\rangle\left|-\right\rangle$ using the same method Alice used. Let

U be the 2-bit AND $|i\rangle|x_i\rangle|y_i\rangle|0\rangle \to |i\rangle|x_i\rangle|y_i\rangle|x_i \cdot y_i\rangle$ which can be done unitarily using classical reversible logic. Using the xor-to-phase oracle protocol again, we can turn our state into

$$(-1)^{x_i \cdot y_i}|i\rangle|x_i\rangle|y_i\rangle|-\rangle$$

Now we need to uncompute to isolate $(-1)^{x_i \cdot y_i}|i\rangle$ from the ancilla. Let Bob uncompute the $|y_i\rangle$, then send the first $\log n + 2$ qubits over to Alice (notice that the $|-\rangle$ remained unentangled) who will then uncompute the $|x_i\rangle$ so that we arrive at

$$(-1)^{x_i \cdot y_i}|i\rangle|0\rangle|0\rangle$$

and have implemented the oracle necessary to perform Grover search.

Each round takes $O(\log n)$ steps, and Grover requires $O(\sqrt{n})$ steps, giving us a communication complexity of $O(\sqrt{n}\log n)$.

6. k-SUM [10 Points]

In the famous k-SUM problem, we're given a list of integers $x_1, \ldots, x_n$, and are asked whether there are k distinct indices, $i_1 < \ldots < i_k$, such that $x_{i_1} + \ldots + x_{i_k} = 0$.

For this problem, you can ignore factors of $\log n$ in the running time (indeed, that is what the $\widetilde{O}$ notation means).

a) [5 points] Assume k is even AND that we are allowed multi-sets (aka repeated elements are allowed). Describe a classical algorithm that solves the k-SUM problem in $\widetilde{O}(n^{k/2})$ time, beating the trivial upper bound of $\widetilde{O}(n^k)$.

Solution. We present two solutions, starting with the simpler one.

Compute all $O(n^{k/2})$ sums and sort them in $\widetilde{O}(n^{k/2})$ time into the list T. We can then iterate through the list T. Let's say that we have entry s. We can then binary search for an entry $-s$ in time $O(k \log n)$.

The more complicated solutions involves a meet-in-the-middle approach. We'll once again compute all $O(n^{k/2})$ sums of subsets and sort them in $\widetilde{O}(n^{k/2})$ time into the list T. Then using the 2SUM approach of meet-in-the-middle (this is a standard coding interview question), we'll start with a pointer at the beginning of T (called the left pointer) and at the end (the right pointer). Let s_L and s_R be the sum for the left and right pointer respectively. We then proceed as follows:

1. *If $s_L + s_R = 0$ then we are done.*
2. *Else if $s_L + s_R < 0$ then move the left pointer one to the right.*

3. Else if $s_L + s_R > 0$ then move the right pointer one to the left.

4. Repeat.

The reason this algorithm works, is because if $s_L + s_R < 0$ then any of the values left of s_L are would also sum to < 0 when added with s_R. The same logic works for s_R and the values right of it if $s_L + s_R > 0$. We will argue that if there exists an s'_L and s'_R such that $s'_L + s'_R = 0$ then we will always check this value.

Since we move iteratively by 1 entry on the list, it is impossible for us to skip over considering s'_L or s'_R. WLOG, let us say that the left pointer has just arrived at s'_L before the right pointer arrives at s'_R (we only move one pointer at a time, so one of these has to happen first). Then by the nature of the algorithm we will continue moving the right pointer until we reach s'_R or a value equivalent to it.

This meet in the middle steps takes $\widetilde{O}(n^{k/2} \cdot k)$ since the pointers collectively pass over each subset of size $k/2$ at most once. This gives us our total runtime of $\widetilde{O}(n^{k/2})$ if we assume k to be a constant.

b) [5 points] Assume k is divisible by 3 and that we are again allowed multi-sets. Describe a quantum algorithm that solves the k-SUM problem in $\widetilde{O}(n^{k/3})$ time.

Solution. This is a standard trick in quantum query complexity. We iterate through all subsets of size $k/3$ and store the sum in a sorted list T in $\widetilde{O}(k \cdot n^{k/3})$ time. We want to Grover search for a subset of size $2k/3$ that adds up to 0 when combined with an entry of T. Since we can create a Grover oracle in $O(k \log n)$ time using binary search over T, each of the $\sqrt{n^{2k/3}}$ iterations of Grover will have a cost of $O(k \log n)$. This takes $O(\sqrt{n^{2k/3}} \cdot k \log n) = \widetilde{O}(n^{k/3} \cdot k^2)$ time for the Grover subroutine, giving the total runtime of $\widetilde{O}(n^{k/3})$ if we assume k to be a constant.

c) [5 points, extra credit] Suppose we wanted to prove that the algorithm from (b) was the fastest possible quantum algorithm for k-SUM. Could that be shown via a lower bound on k-SUM's quantum query complexity? Why or why not?

Solution. Query complexity can prove a bound of at most $\Omega(n)$. For $k > 3$ we cannot meet the upper-bound set by our algorithm so this can't be used to show that our algorithm is tight.