Quantum Computing, Final exam

Ronald de Wolf

March 28, 2013
13:00–16:00
SP G2.10

The exam is "open book", meaning you can bring any kind of paper you want but no electronic devices. You are allowed to answer in English or Dutch.

1. Let U be a 1-qubit unitary that we would like to implement in a controlled way, i.e., we want to implement a map $|c\rangle|b\rangle \mapsto |c\rangle U^c|b\rangle$ for all $c, b \in \{0,1\}$. Suppose there exist 1-qubit unitaries A, B, and C, such that $ABC = I$ and $AXBXC = U$ (remember that X is the NOT-gate). Give a circuit that acts on two qubits and implements a controlled-U gate, using CNOTs and (uncontrolled) A, B, and C gates.

2. (a) Suppose x is an n-bit string. What happens if we apply a Hadamard transform to each qubit of the n-qubit state $\frac{1}{\sqrt{2^n}} \sum_{y\in\{0,1\}^n} (-1)^{x\cdot y}|y\rangle$?

   (b) Give a quantum algorithm that uses T queries to n-bit string x, and that maps $|y\rangle \mapsto (-1)^{x\cdot y}|y\rangle$ for every $y \in \{0,1\}^n$ that contains at most T 1s (i.e., for every y of Hamming weight $\leq T$). You can argue on a high level, no need to write out circuits in detail.

   (c) Give a quantum algorithm that with high probability outputs x, using at most $n/2+2\sqrt{n}$ queries to x. *Hint: Approximate the state of part (a) using the subroutine of part (b), and see what happens if you apply Hadamards to the approximate state. Use the fact that* $\frac{1}{2^n}\sum_{w=0}^{n/2+2\sqrt{n}} \binom{n}{w}$ *is nearly 1.*

3. Consider the *sorting* problem: there are N numbers $a_1, \ldots, a_N$ and we want to sort these. We can only access the numbers by making *comparisons*. A comparison is similar to a black-box query: it takes 2 indices i, j as input and outputs whether $a_i < a_j$ or not. The output of a sorting algorithm should be the list of N indices, sorted in increasing order. It is known that for classical computers, $N\log(N) + O(N)$ comparisons are necessary and sufficient for sorting. Prove that a quantum algorithm needs at least $\Omega(N)$ comparisons for sorting.
   Hint: Show how you can use sorting to solve one of the problems we analyzed in the lecture (Chapter 8 and exercises), and then invoke the lower bound for that problem.

4. Consider the following communication complexity problem, called the "Hidden Matching Problem." Alice's input is some $x \in \{0,1\}^n$. Bob's input is a matching M, i.e., a partition of $\{1, \ldots, n\}$ into $n/2$ disjoint pairs (assume n is a power of 2). Their goal is that Bob outputs a pair $(i, j) \in M$ together with the parity $x_i \oplus x_j$ of the two bits indexed by that pair. It doesn't matter which pair $(i, j) \in M$ Bob outputs, as long as the additional bit of output equals the parity of the two indexed bits of x. Show that they can solve this problem with success

probability 1 using only a message of $\log n$ qubits from Alice to Bob (and no communication from Bob to Alice).

Hint: The matching M induces a projective measurement that Bob can do on the message he receives.

5. Shor's 9-qubit code allows to *correct* a bitflip and/or a phaseflip on one of its 9 qubits. Below we give a 4-qubit code which allows to *detect* a bitflip and/or a phaseflip. By this we mean that after the detection procedure we either have the original uncorrupted state back, or we know that an error occurred (though we do not know which one). The logical 0 and 1 are encoded as:

$$|\overline{0}\rangle = \tfrac{1}{2}(|00\rangle + |11\rangle) \otimes (|00\rangle + |11\rangle)$$
$$|\overline{1}\rangle = \tfrac{1}{2}(|00\rangle - |11\rangle) \otimes (|00\rangle - |11\rangle)$$

   (a) Show how we can detect a bitflip error.

   (b) Show how we can detect a phaseflip error.

   (c) Does that mean that we can now detect any 1-qubit error? Explain your answer.