University of Warsaw, PhD Open, Exam *Quantum Computing*

Ronald de Wolf (CWI Amsterdam, rdewolf@cwi.nl)

5 questions. Handed out: January 9, 2010

1. Suppose we have a qubit which is promised to be either in the state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ or in the state $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. Give a quantum circuit that determines the state of this qubit, using a unitary gate and a measurement in the computational basis.

2. Consider 1-qubit gates $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ and $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, and 2-qubit CNOT (controlled-X).

   (a) Show that $HXH = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$. This is known as the *phase-flip* operation.

   (b) Give the 4×4 unitary matrix for the 2-qubit operation $(I \otimes H)\text{CNOT}(I \otimes H)$.

   (c) Draw a 2-qubit circuit which transforms the initial state $|00\rangle$ into the state $\frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$.

3. Given a function $f : \{0,1\}^n \to \{0,1\}$ with the promise that there exists a string $s \in \{0,1\}^n$ such that $f(x) = x \cdot s$ for all $x \in \{0,1\}^n$. Here $x \cdot s = \sum_{i=1}^n x_i s_i \mod 2$ denotes inner product of bit strings modulo 2. We would like to learn what s is.

   (a) Give a quantum algorithm that makes only 1 query to f, and that computes s with success probability 1 (as usual, a query is the unitary transformation $O_f : |x\rangle \to (-1)^{f(x)}|x\rangle$).
   Hint: Use the Deutsch-Jozsa algorithm.

   (b) Argue that any classical algorithm to compute s needs to evalute f at least n times.

4. Suppose we have a database with $N = 2^n$ binary slots, containing t ones (solutions) and $N - t$ zeroes.

   (a) Show that we can use Grover's algorithm to find the positions of *all* t ones, using an expected number of $O(t\sqrt{N})$ queries to the database. You can argue on a high level, no need to draw actual quantum circuits.

   (b) Show that this can be improved to an expected number of $O(\sqrt{tN})$ queries.
   Hint: Recall that if there are i solutions, Grover's algorithm finds a solution using an expected number of $O(\sqrt{N/i})$ queries.

5. Consider the *sorting* problem: there are n distinct 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 $i, j \in \{1, \ldots, n\}$ 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 well-known that for classical computers, $\log_2(n!) = n \log_2(n) + O(n)$ comparisons are necessary and sufficient for sorting. Prove that a quantum sorting algorithm needs at least $\Omega(n)$ comparisons.
   Hint: View this in a two-party communication setting: Alice knows the numbers $a_1, \ldots, a_n$, and hence she knows which one among the $n!$ permutations is the one that puts the numbers in order. Bob would run an optimal quantum sorting algorithm (say with T comparisons), using $O(\log n)$ qubits of communication (to and from Alice) to implement each comparison. In the end Bob learns the permutation. Then apply Holevo's theorem to lower bound the total amount of quantum communciation that must have been sent. Then conclude a lower bound on T. Again, you can argue on a high level.