CMSC491 Introduction to Quantum Computation, VCU
Fall 2015, Assignment 6

Due: Tuesday, October 27 at start of class

Total marks: 20 marks + 10% bonus for typing your solutions in LaTeX.

1 Exercises

1. (8 marks) This question will practice working through Simon's algorithm.

   (a) (2 marks) Consider function $f : \{0,1\}^2 \mapsto \{0,1\}^2$ such that $f(00) = 10$, $f(01) = 11$, $f(10) = 10$ and $f(11) = 11$. This function satisfies the promise required for Simon's problem, i.e. $f(x) = f(y)$ iff $x = y \oplus s$. What is the value of s for f?

   (b) (6 marks) Suppose $f : \{0,1\}^n \mapsto \{0,1\}^n$ is an n-bit function which slightly violates the promise required of Simon's algorithm in that f is "almost" one-to-one in the following sense: For each distinct input $x \in \{0,1\}^n$, the output $f(x)$ is unique, *except* for inputs 0^n and 1^n, which are the only pair of inputs satisfying $f(0^n) = f(1^n)$. Thus, we are very "close" to the $s = 0^n$ case, and we expect the analysis to go "similarly".

   Recall that right before the measurement in Simon's algorithm, our quantum state looks like

   $$\sum_{y \in \{0,1\}^n} |y\rangle \left(\frac{1}{2^n} \sum_{x \in \{0,1\}^n} (-1)^{x \cdot y} |f(x)\rangle \right).$$

   Pick an arbitrary $\hat{y} \in \{0,1\}^n$. Show that when the first register is now measured in the standard basis, the probability of outcome $\hat{y}$ is given by

   $$\frac{1}{2^n} \pm \frac{1}{2^{2n-1}},$$

   where the $+$ occurs if the parity of y is even, and the $-$ occurs if the parity of y is odd. Here, the parity of y is defined as $\bigoplus_{i=1}^{n} y_i$. (Hint: The purpose of this question is to make sure you follow the analysis of Simon's algorithm. Thus, begin by following the analysis for the $s = 0^n$ case, and then add in ideas from the $s \neq 0^n$ case analysis to deal with the fact that f is not exactly one-to-one.)

2. (5 marks)

   (a) (1 mark) How many bits are required to write the integer 10291823 in binary?

   (b) (4 marks) This question tests an important subtlety in the definition of "polynomial-time". One of the most famous open problems in classical complexity theory is whether the problem of factoring a given integer N into its prime factors is solvable in polynomial time. Given positive integer N as input, why is the runtime of the following naive approach to the factoring problem not polynomial-time?

   1: Set $m := 2$.
   2: Set $S := \emptyset$ for S a multi-set.
   3: **while** $m \leq \lceil \sqrt{N} \rceil$ **do**

```
    if m divides N then
        Set S ∪ {m}.
        Set N = N/m.
    else
        Set m = m + 1.
    end if
end while
Return the set S of divisors found.
```

(Hint: Think about how we define "polynomial-time" — is it in terms of $|N|$, or the number of bits needed to encode N?)

3. (7 marks) Consider the NAND gate, which consists of an AND gate followed by a NOT gate. NAND is a special gate in that it is *universal*, meaning any Boolean function simulated using just NAND gates. **Using the construction from class**, give a quantum (i.e. reversible) implementation of a classical NAND gate. Your answer should be a drawing of a circuit specifying all inputs and outputs.