CS 6846: Quantum Computing 31st Oct, 2024

Homework 3

Instructor: Shweta Agrawal *Due: 8th Nov, 5 PM*

Name	Roll No.

Instructions:

1. Please type up your solutions using latex.
2. Please email TA and instructor.
3. You may collaborate with other students. Please mention the names of your collaborators or any other source that you use for the solution.
4. Please type up your solutions *individually* without any help.

Problem 1: Rethinking Shor's Algorithm (5 points).

In class, we saw that Shor's algorithm has the following broad steps: i) create a uniform superposition of inputs, ii) evaluate the function f_a on this superposition, iii) measure the registers containing the function value to get $f_a(x) = y$, iv) apply QFT on data registers to obtain a superposition of points that are separated by M/r (ignore the rounding issue for now), v) measure to obtain a multiple of M/r, vi) repeat to obtain many such multiples and recover r via GCD, since M is known.
Step (iii) gives a superposition of points that are separated by period r. In more detail, for some x_0, we obtained a state:

$$\frac{1}{\sqrt{M/r}} \sum_{j=0}^{M/r-1} |x_0 + jr, y\rangle$$

Suppose we measure the data registers at this stage, i.e. before applying the QFT. The rationale is that we already have a superposition of points that are separated by r, so why can't we obtain many values of $x_0 + jr$, subtract pairs to remove the x_0 to obtain many multiples of r and then take GCD to recover r? Does this work? If so, why? If not, why?

Problem 2: Period Finding

Consider the function $f(a) = 7^a \mod 10$.

a) (1 point) What is the period r of f?

b) (4 points) Show how Shor's algorithm finds the period of f, using a Fourier transform over $Q = 128$ elements. Write down all intermediate superpositions of the algorithm for this case (don't just copy the general expressions from the notes, but instantiate them with actual numbers as much as possible, including the value of the period found in (a)). You may assume you're lucky, meaning the first run of the algorithm already gives a measurement outcome $b = \frac{cQ}{r}$ with c coprime to r.

Problem 3: Breaking Another PKE (5 points)

Let $P : \{0,1\}^n \to \{0,1\}^n$ be a permutation; that is, a function without any collisions. Let $Q(x) = P(x \oplus k_0) \oplus k_1$ for some secret keys k_0, k_1. It is known that if you can only make classical queries to these two functions, then you cannot recover k_0, k_1. This fact is used in the design of encryption schemes: P is a public permutation that everyone knows, and you turn it into a private permutation Q as above. Then Q can be used to encrypt messages (decryption will require the ability to compute the inverse of P, but we will ignore it for this problem).
Show that quantum queries to both P and Q allow for the recovery of k_0, k_1.
Hint: try defining a function f based on P and Q such that f is an instance of Simon's problem.

Problem 4: One Time Pad (3 +3 points)

(a) Prove that a classical one time pad satisfies information theoretic security.

(b) Prove that a quantum one-time pad results in a maximally mixed state.

Problem 5: Fourier formulas (3+3 points).

(a) Prove that if $s \neq 000\ldots0$

$$\implies \hat{f}(s) = \frac{1}{2}\left(\mathbb{E}_{\mathbf{x}\sim\{0,1\}^n}[f(\mathbf{x}) \mid \chi_s(\mathbf{x}) = +1] - \mathbb{E}_{\mathbf{x}\sim\{0,1\}^n}[f(\mathbf{x}) \mid \chi_s(\mathbf{x}) = -1]\right),$$

where the $\mid$ notation denotes "conditional expectation" and $\mathbb{E}_{\mathbf{x}\sim\{0,1\}^n}[\cdot]$ denotes "the expected value, when $\mathbf{x}$ is chosen uniformly at random from $\{0,1\}^n$".

(b) Let $f : \{0,1\}^n \to \mathbb{C}$. Now for $y \in \{0,1\}^n$, define the function $f^{+y} : \{0,1\}^n \to \mathbb{C}$ by $f^{+y}(x) = f(x+y)$. (Here the addition is in $\mathbb{F}_2^n$; i.e., coordinate-wise mod 2.) Compute $\widehat{f^{+y}}(s)$ in terms of $\hat{f}(s)$. How does performing Fourier sampling of f^{+y} compare to performing Fourier sampling on f?

Problem 6: Addition by Fourier transforms

Consider the task of constructing a quantum circuit to compute $|x\rangle \rightarrow |x+s \mod 2^n\rangle$, where s is a fixed constant and $0 \leq x < 2^n$.

(a) (5 points) Show that one efficient way to do this, for values of s such as 1, is to first perform a quantum Fourier transform, then to apply single qubit phase shifts, then an inverse Fourier transform.

(b) (3 points) How many operations are required and what values of s can be added easily this way?