Homework 7 8.370/18.435/2.111 Due 11/9

1. Let's look at a piece of the factoring algorithm a little more closely. As an example, let's factor 21. Suppose we use the unitary transformation which acts as:

$$U\,|y \bmod (21)\rangle = |2y \bmod (21)\rangle$$

on binary representations of the numbers between 0 and 20 (for this problem, it's irrelevant what it does outside this range).

For the factoring algorithm, recall that we start with the state $|1 \bmod (21)\rangle$, and use the phase estimation algorithm on the transformation U. If we have an eigenvector with eigenvalue $e^{2\pi i a/b}$, assume that phase estimation combined with continued fractions returns a/b in reduced fraction form with probability 1.

   (a) The state $|1 \pmod{21}\rangle$ can be represented as a superposition of eigenvectors of U. What are these eigenvectors? How many of them are there? (You don't have to write them all down; but make sure the grader can tell that you know what they are.)

   (b) Suppose the algorithm returns a/b, and if b is even, we compute $2^{b/2}$ and try to use it to factor. What is the probability we get the right factors this way? (If b is odd, count this as a failure.)

2. In the last problem, we saw that even if the phase estimation and continued fractions pieces of the algorithm work perfectly, and we choose a g in $[1, N-1]$ that is relatively prime to N, use the unitary

$$U : |y \pmod{N}\rangle \to |gy \pmod{N}\rangle$$

and get the correct period r, the factoring algorithm still has some probability of failing.

   (a) How bad is this problem for large N?

   Note: You may need the following theorem from number theory: Euler's totient function $\phi(n)$ is the number of positive integers less than or equal to n that are relatively prime to n. For all n, $\phi(n) \geq \frac{Cn}{\ln \ln n}$ for some constant C.

   (b) Can you think of anything to do to make the factoring algorithm somewhat more efficient for large N? (Assume that quantum computation is expensive and classical computation is relatively cheap.)

3. (20 points)

For a prime p and two numbers a, b, with $1 < a, b < p$, consider the quantum state

$$\frac{1}{\sqrt{p}} \sum_{j=0}^{p-1} |j \pmod p\rangle \, |aj + b \pmod p\rangle$$

(a) Show that if you are given one copy of this quantum state, then with a quantum computer, you can find a with high probability (i.e., with probability going to 1 as p goes to ∞).

(b) Show that you can also find b with high probability (i.e., with probability going to 1 as p goes to ∞).

(c) Show that no quantum algorithm can identify a with probability 1.

Hint: there are two techniques that we've seen in class that you might use to attack parts (a) and (b). One is the quantum Fourier transform and the other is phase estimation. One of these techniques works substantially better than the other.