CS 593/MA 595 - Intro to Quantum Computation
Theoretical Homework 8

Due Wednesday, November 19 at 11:59PM (upload to Brightspace)

1. The RSA cryptosystem works roughly as follows:

   Alice generates two (biggish) random prime numbers p, q and keeps them secret. She releases $N = pq$ to the public as her "public modulus." She computes $\phi(N) = (p-1)(q-1)$ and keeps it secret. She then picks a random number e such that $1 < e < \phi(N)$ is coprime to $(p-1)(q-1)$. She announces e to the public as the encryption key people should use to send her encrypted messages (details momentarily). Because she knows $\phi(N)$, she can compute d such that $de = 1 \mod \phi(N)$ efficiently (e.g., via the extended Euclidean algorithm); she will use d as her private decryption key.

   Bob can now send Alice a message M—so long as it is in the form of a number $2 \leq M < N$—by sending the encrypted message $M^e \mod N$ instead. Alice can then apply her decryption key to get M back: $(M^e)^d = M \mod N$.

   Suppose Eve has a quantum computer. Show that given access to Alice's public key data N and e, Eve can find the decryption key d in polynomial time.

2. Shor's algorithm for factoring the integer N uses a reduction to order-finding for multiplication mod N. Show, conversely, that if we know how to factor N efficiently, then we can find the order of any x mod N efficiently (for x coprime to N).

3. **[Do not submit–this is just for fun!]** Show that the following problem is NP: given a, r, N with $0 < a < N$ and $r > 0$, decide if r is the order of a mod N.