Assignment 7 – Due Thursday, December 4

1. Textbook, page 412, Problem 10.17

2. Textbook, page 412, Problem 10.19 [*Hint:* You are allowed to use results that we proved in class, and with that capability this entire proof can be written in a single line. Don't make it harder than it really is!]

3. Following the definition of the class BPP, the book states (page 369) "We defined this class with an error probability of $\frac{1}{3}$, but any constant error probability would yield an equivalent definition as long as it is strictly between 0 and $\frac{1}{2}$. What if, instead of the constant probability being strictly less than $\frac{1}{2}$ we changed the inequality to be a strict inequality? In other words, consider saying that probabilistic polynomial time Turing machine M recognizes language A if

   1. $w \in A$ implies Pr[M accepts w] $> \frac{1}{2}$

   2. $w \notin A$ implies Pr[M rejects w] $> \frac{1}{2}$

   Is the set of all such languages the same as BPP? Justify your answer.

4. The proof of Lemma 10.30 in the book (page 390) uses a simulation of an interactive proof to prove that $IP \subseteq PSPACE$. The purpose of the proof was to show that this simulation used only polynomial space, and the time wasn't considered. For this problem, analyze the time required by the simulation algorithm.

5. The definition of an interactive proof system gives the verifier the ability to use randomization, and this is in fact critical to the definition. For this problem, prove that if a language A has an interactive proof system in which the verifier is deterministic, then $A \in NP$ (so if randomization could be removed from *all* interactive proof systems, then $IP = NP$). [*Hint:* If the verifier is deterministic, then the prover knows exactly what message the verifier will send at each step, so does not even have to wait for the verifier's messages — the prover can simply compute all of its response messages at the very beginning and supply that to the verifier.]

6. (a) An AND gate is a common construction in conventional circuits, but can't be constructed directly in a quantum circuit. What basic principle would be violated in a gate that implemented an AND operation in the same way a boolean gate does (2

inputs and 1 output)? Despite this problem, we can do something similar: Design a gate that *includes* the AND functionality. The gate must have at least 2 inputs and at least 1 output, and the result of the AND must be easily extractable from the output. Give both a description and a unitary matrix that defines the operation (pages 28-30 of "An Introduction to Quantum Computing" are particularly important to this question).

(b) In the handout that discusses open problems in quantum complexity, the statement was made that "Since any quantum computer running in polynomial time can be fairly easily simulated in PSPACE, as was pointed out in [BV93], we are unlikely to be able to prove anytime soon that BQP is larger than P." (Note that BQP is the quantum version of the class BPP that we studied. Why is this "unlikely"?

(c) Are polynomial time mapping reductions compatible with the notion of quantum computing? In other words, if one NP-complete problem is solvable in polynomial time on a quantum computer, does that mean that *all* NP-complete problems are solvable in polynomial time on a quantum computer?