6.045J/18.400J: Automata, Computability and Complexity Prof. Ron Rivest

Handout 15a: Quiz 3, II

Write your full name on each page.

Name: ______________________________

Problem 1: Trick Questions: (3 points each)

1. True or false: If $\mathcal{P} = \mathcal{NP}$ then the problem of determining whether or not a given integer is prime is solvable in polynomial time. Explain briefly.

2. True or false: If L is polytime reducible to a finite language, then L is in P. Explain briefly.

3. True or false: There is no language in $\mathcal{NP}$ that is recognizable in less than linear time. (That is, requiring less than n steps for inputs of length n.) Explain briefly.

4. If the complement of the reverse of a language L is recognizable in polynomial time, and if L is in $\mathcal{NP}$, then the set of palindromes in L is recognizable in polynomial time. (A palindrome is equal to its reversal.) Explain briefly.

5. It is $\mathcal{NP}$-complete to determine if a given input formula has two or more satisfying assignments. Explain briefly.

Name:

Problem 2: **(15 points)** Show the following language to be $\mathcal{NP}$-complete:

$$SETPACK = \left\{ \langle S, C, k \rangle \ : \ \begin{array}{l} S \text{ is a finite set, } C \text{ is a collections of subset of } S, \\ \text{and } k \text{ is an integer such that } C \text{ contains } k \text{ mutually} \\ \text{disjoint sets} \end{array} \right\}$$

Hint: Consider *X3C*.

Name:

Problem 3: **(15 points)** The *SQUARESUM* problem is as follows: you have before you a set of integers a_1, a_2, $\ldots a_n$, an integer K and an integer J. You must determine if you can put the integers a_1 through a_n into disjoint sets A_1 through A_K so that

$$\sum_{i=1}^{K} \left(\sum_{a \in A_i} a \right)^2 \leq J$$

Show that *SQUARESUM* is $\mathcal{NP}$-complete. (Hint: consider the case where $K = 2$.)

Name: ____________________

Problem 4: **(15 points)** Show that if $\mathcal{P} = \mathcal{NP}$, there exists a polynomial time algorithm that takes in a boolean formula ϕ and returns either

- An assignment that satisfies ϕ, if one exists, or
- A special output $\perp$ if one does not.

Note that your algorithm must return the assignment itself!

Name: ______________________________

Problem 5: **(15 points)** We say that two languages, L_1 and L_2 are *polynomial time isomorphic* if there exists a mapping $f : \Sigma^* \rightarrow \Sigma^*$ such that:

1. f is a bijection,
2. f is a poly-time mapping from L_1 to L_2, and
3. f^{-1} is a poly-time mapping from L_2 to L_1.

It is conjectured that every $\mathcal{NP}$-complete language is polynomial time isomorphic to every other $\mathcal{NP}$-complete language. Prove that if this were the case, then $\mathcal{P} \neq \mathcal{NP}$.