Name(s)______________________________________ ____

Algorithms, Assignment 07: Complexity Classification

1. For the Satisfiability Problem (SAT), you are given a Boolean expression and then asked if there is an assignment of true and false values to the variables that lead to the expression evaluating to true.

   Example Boolean expressions:

$w \land \overline{w}$	: cannot be satisfied
$(x \lor y) \land (\bar{x} \lor \bar{y})$	: satisfied by $x = \bar{y}$
$a \land \overline{((b \lor \bar{c}) \land d)}$	: satisfied by $a = T, d = F$

   3-CNF-SAT (sometimes called 3SAT and CNF means conjunctive normal form) is a particular type of Boolean expression that looks like this:

   $$(x_1 \lor \overline{x_2} \lor \overline{x_4}) \land (x_2 \lor x_3 \lor x_5) \land (x_1 \lor x_4 \lor \overline{x_6})$$

   In 3-CNF-SAT we have some number of clauses (enclosed in parentheses), each clause has three literals OR'd together, and the clauses are AND'd together.

   The 3-CNF-SAT problem can be stated as: given an expression in the 3-CNF form, is there an assignment of literals that makes the expression evaluate to true.

   3-CNF-SAT is NP-Complete.

   a. Describe an algorithm for verifying a solution (an assignment of the literals) of a 3-CNF expression and state the running time of the verification algorithm.

   b. Does 3-CNF-SAT belong to the class P?

   c. Does 3-CNF-SAT belong to the class NP?

   d. Is 3-CNF-SAT NP-Hard?

Name(s)____________________________________ ____

2. A graph clique is a completely connected subcomponent of a graph. That is, every vertex in a clique has an edge connecting it to every other vertex in the clique.

   A simple question to ask is: given a graph G, what is the size of its largest clique?

   a. Write a decision version of the above question.

   b. The decision version of the clique problem is in **NP**. Given an algorithm for verifying a solution to the problem.

   c. The decision version of the clique problem is **NP-Hard**. Describe the process of proving that clique is in fact NP-Hard by stating a reduction involving the clique and 3-CNF-SAT problems. You do not need to show the transformation process, just state the reduction process at a high-level:
      i. The direction of the reduction (what is being reduced to what).
      ii. What the reduction implies.

Name(s)______________________________________ ____

d. (Bonus) Describe the transformation needed to complete part (c). You should spend some time thinking about this first, but then I'd recommend stepping through this animation one slide at a time—after each slide you should attempt to complete the process yourself.

Name(s)_____________________________________ ____

3. The graph partitioning problem **is NP-Complete**. Graph partitioning…

    a. … can be solved with a polynomial time algorithm.

        True   False   Probably False (but maybe True)

    b. … is in NP

        True   False   Probably False (but maybe True)

    c. … is NP-Hard.

        True   False   Probably False (but maybe True)

    d. … can be reduced to the clique problem.

        True   False   Probably False (but maybe True)

    e. … can be reduced to a breadth-first search.

        True   False   Probably False (but maybe True)

    f. … can have its decision variant verified in polynomial time.

        True   False   Probably False (but maybe True)

    g. … can have the 3-CNF-SAT problem reduced to it.

        True   False   Probably False (but maybe True)

    h. … can have the shortest path problem reduced to it.

        True   False   Probably False (but maybe True)

Name(s)_____________________________________ ____

4. Consider these two problems:

   PRIME = $\{\langle x \mid x \geq 2\rangle : x$ is a prime number$\}$

   NOT-PRIME = $\{\langle x \mid x \geq 2\rangle : x$ is **not** a prime number$\}$

   The notation above is a common for denoting problems when discussing computational complexity. You can read them as:

   "PROBLEM NAME" = {⟨"Problem Input"⟩ : "Decision being made"}

   Here is one more example:

   MST-K = $\{\langle G, k\rangle : G$ has a minimum spanning tree with a total cost $\leq k\}$

   "The MST-K decision problem states that for a given graph G and an input K we must output whether or not (yes or no) G has a minimum spanning tree (MST) with a total cost less than or equal to k."

   In a 2004 paper, Agrawal, Kayal, and Saxena proved that PRIME $\in P$. This denotes that the PRIME problem belongs to the set P (it can be solved in polynomial time).

   Answer true or false and justify answers to the following statements:

   a. PRIME $\in P$

   b. NOT-PRIME $\in P$

   c. PRIME $\leq_p$ CLIQUE (the PRIME problem can be reduced to the CLIQUE problem)

   d. CLIQUE $\leq_p$ PRIME (the CLIQUE problem can be reduced to the PRIME problem)