Jonathan Nicolosi

Homework 4

1.

(d) can be inferred since NP-Complete is a complexity class which represents the set of all problems X in NP for which it is possible to reduce any other NP problem Y to X in polynomial time. This means we can solve Y quickly if we know how to solve X quickly.

(g) can be inferred because if Y is no easier than X, then X is in the same complexity class as Y.

2.

a. False.
SUBSET-SUM is NP-complete so it can be reduced to another NP-complete problem. However, we don't know that COMPOSITE is NP-complete, only that it is in NP. Hence, we cannot say for sure that SUBSET-SUM reduces to COMPOSITE.

b. True.
The given running time is polynomial in n and log t. Since SUBSET-SUM is NP-complete, this implies P = NP. Hence, every algorithm in NP, including COMPOSITE, would have a polynomial-time algorithm.

c. True.
If a P algorithm exists for an NP problem, it suggests that P algorithms exist for other NP problems as well.

d. False.
The class P is a subset of NP, and it is not empty. Proving P≠NP would only show that NP-complete problems cannot be solved in polynomial time.

3.

a. True.
An NP-complete problem can be reduced to another NP-complete problem.

b. False.
If 3-SAT $\leq_P$ 2-SAT you could solve 3-sat in polynomial time since 2-sat can be solved in polynomial time. But since 3-SAT is NP complete, this would imply P = NP.

c. True.
   A polynomial-time algorithm for one NP-complete problem yields polynomial time algorithms for all others. Hence, either all these problems are in P, or none are. P≠NP implies the latter.

4. A simple path is defined as a path where all vertices are visited exactly once. Hamiltonian paths abide by the same rule and they are NP-complete problems. K acts as a certificate and the certifier is designed to check if the path is correct. We know that LONG-PATH ∈ NP. To show that LONG-PATH is NP-Complete, you must show that Hamiltonian Path $\leq_P$ LONG-PATH.

   CheckHamiltonianPath(G)

   1. Construct G' from G by giving weight of 1 to all edges of G.
   2. For all pairs of vertices u, v ∈ V
   3. If (LONG-PATH(G', u, v, n-1 == 1) return 1
   4. Else return 0

   If G has a Hamiltonian Path, then CheckHamiltonianPath will return 1.

   Since G has a Hamiltonian Path, there exists vertices u, v and a path from u to v that visits each vertex exactly once. This path has length n-1 in G'.

5.
   a. If we have solution to this problem in polynomial time, we can assign k values from |V| down to 2 and check if it is colorable, and stop when k reaches some value that it is not colorable. Then we choose the minimum number from all those numbers make the graph colorable. That is the number we want in Graph-coloring problem. It is easy to know the time is still polynomial.

      If Graph-coloring problem is solvable in polynomial time, then we know the minimum number of colors needed, say l, then we can just compare the given number k in decision problem with l, if k < l, then we can answer NO, in decision problem; if k ≥ l, then we can answer YES in decision problem. It must be polynomial time solvable.

b. To prove that 4-COLOR is NP-Complete, we must show that it is a member of NP and that 3-COLOR, another NP-Complete problem, is polynomial-time reducible to it.

   First, we can show that 4-COLOR is a member of NP by providing a polynomial-time certifier for it. The certifier algorithm will compare the color of each node in the graph to the color assigned to each connected node and will reject the solution if the colors of two adjacent nodes match.

   Second, we show a reduction from 3-Color which is given as NP-Complete in the problem. Given a 3-COLOR problem for a graph G with n nodes, we can map it to a 4-COLOR problem by constructing a new graph G'. The graph G' has one additional node that is connected to every other node in the graph. This graph can be created with n+1 steps so the mapping is polynomial time. Since the new node is connected to every other node in the graph, it cannot share a color with any other node. So if G was three-colorable, G' can be four-colored by assigning the original nodes the same colors as in the three-color solution of G and assigning the new node the fourth color. For any G' that is four-colorable, the new node must have a color of its own since it is connected to every other node in the graph. Therefore, G can be three colored by using the same color assignments for all the remaining nodes. Therefore, G' is four-colorable if and only if G is three-colorable.