Benjamin Tate
5/30/2017
CS 325 -- Section 400
Homework #5

1.
   a. We cannot infer whether or not X is NP-complete, because X reduces to Y in polynomial time, not necessarily the other way around, so we don't have enough information.
   b. False. Since we know from the reduction that Y is at least as hard as X, we can infer that Y is NP-hard, but we would need to know that Y is also in NP to infer that it is NP-complete
   c. Not necessarily. X doesn't necessarily have to be as hard as Y, and only knowing X is in NP doesn't indicate that it is also NP-complete, because for all we know, X might also be in P.
   d. True. Since Y must be at least as hard as X and X is NP-complete, Y must be NP-complete as well.
   e. False. As shown above, there are circumstances that satisfy the given information in which both X and Y are NP-complete.
   f. Not necessarily. Since Y is *at least* as hard as X, Y could be impossible to solve in polynomial time even if X is in P. The fact that X reduces to Y in polynomial time does not affect this.
   g. True. Since X reduces to Y in polynomial time, and Y can be solved in polynomial time, X can be solved in polynomial time by reducing to Y and solving Y.
2.
   a. This statement does not necessarily follow from the information we are given. While we know that since SUBSET-SUM is NP-complete, it can be reduced to another NP-complete problem, we do not know that COMPOSITE is NP-complete (only that it is in NP). Therefore, COMPOSITE may also be in P, making it simpler than an NP-complete problem. This contradicts the implication that SUBSET-SUM is no harder than COMPOSITE, which is implied by the statement SUBSET-SUM $\leq_p$ COMPOSITE.
   b. True. If an NP-complete problem has a polynomial time solution, that means that any problem that is in NP can also be solved in polynomial time, since NP problems cannot be any harder than NP-complete problems. Therefore, since there is a polynomial time solution for SUBSET-SUM (an NP-complete problem), there must be one for COMPOSITE, which is in NP.
   c. Not necessarily. While we know that COMPOSITE is in NP, even if it has a polynomial time solution, SUBSET-SUM can still be harder and have an NP solution. However, since SUBSET-SUM is NP-complete, P ≠ NP if SUBSET-SUM has a polynomial time solution.
   d. This is also not necessarily true. As in the example in problem 2c, COMPOSITE may have a polynomial solution even when P ≠ NP, and since COMPOSITE is in NP, the statement does not follow from the given facts.
3.
   a. True. Since we know that 3-SAT and TSP are both NP-complete problems, we know that one can be reduced to the other.
   b. False. We know that 2-SAT has a polynomial solution, so 3-SAT ≤ 2-SAT would imply that 3-SAT also has a polynomial time solution. However, since 3-SAT is known to be NP-complete, this would imply that P=NP, and it is stated that P ≠ NP, so this isn't possible.
   c. True. When P ≠ NP, NP-complete problems cannot be in P, meaning that no problem that can be solved in polynomial time can be NP-complete.
4. Since there is a simple reduction from HAMILTONIAN-PATH to LONG-PATH (replace k with n, and since we have an ordered set of vertices, we can just check if there is a path that includes at least k vertices), we can say for certain that LONG-PATH is NP-hard, because it is at least as hard as HAMILTONIAN-PATH, a well-known NP-complete algorithm. Furthermore, verifying a solution for LONG-PATH requires only checking if the solution makes a path from u to v with a distance of at least k, which would take polynomial time. Since LONG-PATH is verifiable in polynomial time, we now know that it is in NP. Since LONG-PATH is in NP and is also NP-hard, it follows that it is NP-complete.
5.
   a. **Decision problem:**
      K-COLOR: Given an undirected graph G and an integer k, can the vertices of G be colored using no more than k different colors, such that each vertex is colored differently from all adjacent vertices?

The graph-coloring problem is easily reducible to K-COLOR in polynomial time, because if we have a solution to the graph-coloring problem for a given graph G, we need only check if the solution is ≤ k for to answer K-COLOR for the same graph. Since GRAPH-COLOR $\leq_p$ K-COLOR, we know that K-COLOR is at least as hard as GRAPH-COLOR. Therefore, K-COLOR cannot be solved in polynomial time if GRAPH-COLOR can't, and since GRAPH-COLOR can be reduced in polynomial time to K-COLOR, we know that if GRAPH-COLOR can be solved in polynomial time, then so can K-COLOR.

b. We can easily determine that 4-COLOR can be verified in polynomial time, because verifying a solution would only take checking each vertex against its adjacent vertices, so we know already that 4-COLOR is in NP. Next, we can show that 3-COLOR is reducible to 4-COLOR. To do this, we consider a function that creates a graph G', such that G' is 4-colorable iff the original graph G is 3-colorable. As a simple example, consider G' to be identical to G, except with one node added, which is adjacent to all other nodes. Then to 4-color G', we can use the same colors as G for the nodes in common, and color the new node with the 4th color. Since this can be done in n+1 steps, it is a polynomial reduction. Now that we have proven that 3-COLOR $\leq_p$ 4-COLOR, we know that 4-COLOR is at least as hard as 3-COLOR, making it NP-hard. Finally since 4-COLOR is in NP and is also NP-hard, we know that 4-COLOR is NP-complete.