Shannon Jeffers
CS 325 HW 5
HW7

1. A. Can not infer. Y is at least as hard as X, meaning it could be harder than X so X cold just be P which is in the set of NP

   B. Can not infer for the same reason as A. Y could be harder than X.

   C. Can not infer. P and NP are also in NP, so X could be either of those.

   D. Can infer . X is no harder than Y

   E. Cannot infer. Y is at LEAST as hard as X meaning it could be the same hardness.

   F. Cannot infer, Y is at least as hard as X meaning it could be harder.

   G. Can infer because X is at least as hard as Y.

   Ultimately we can infer both D and G because If you have a way to effectively solve Y you also have a way to effectively solve X because X is no harder than Y.

2. A. No it does not follow. We know that SUBSET-SUM is NP complete which means it can be reduced to any other NP complete (or harder) problem. We only know that COMPOSITE is NP (which is a large umbrella) so we cannot say with 100% certainty that SUBSET-SUM can be reduced to COMPOSITE.

   B. Yes, this we can infer. It is said if we can solve 1 NP-Complete algorithm in polynomial time, we can solve all of them in polynomial time. If SUBSET-SUM can be solved in polynomial time we are basically saying that P=NP, which would indicate anything in NP could be solved in polynomial time.

   C. No this does not follow. In order for P=NP COMPOSITE would have to be confirmed as NP-Complete. Since we have not confirmed COMPOSITE to be NP complete, having a polynomial time does not mean that P=NP. Since P is a subset of NP it is possible to have NP problems that are also P, but it does not mean all of them are.

   D. No this definitely does not follow. P is a subset of NP which means that some problems can be solved in polynomial time that are NP problems but not all problems. NP-Complete problems cannot be solved in polynomial time.

3. A. This one is true. They are both NP-complete problems so one can be reduced to the other. There may be a couple steps along the way but it can be done. Going based off of the lectures, 3-SAT can be reduced to DIR-HAM-CYCLE to HAM-CYCLE which can then be reduced to TSP. The chart in the lecture slides displays this information.

   B. False. There is an existing polynomial time algorithm for 2-SAT, but 3-SAT is NP-Complete. If P != NP then there is no polynomial time algorithm that exists for NP-Complete problems such as

3-SAT. If 3-SAT was reducable to 2-SAT which we already know has a polynomial algorithm, then 3-SAT would also have a polynomial algorithm because the algorithm that 3-SAT is reducing to must be equal or harder than the 3-SAT. This is a contradiction that there is not a polynomial algorithm for 3-SAT.

C. True. If a single NP-Complete problem could be solved in Polynomial time then ALL NP-Complete problems could be solved in Polynomial time. If all NP-Complete problems could be solved in polynomial time we would have, by definition, P=NP. So it follows that since we have P!=NP by definition there are no NP-Complete problems that can be solved in polynomial time.

4. The first thing we need to do in order to prove that HAM-PATH is NP-Complete is to prove that it can be verified in polynomial time. We need input in the form of {G, u, v} as well as a certificate that is simply a sequence of vertices ranging from 1 to n. In order to verify that the solution in the certificate given is a HAM-PATH we simply need to traverse each vertex in the certificate sequence and ensure that each vertex is visited exactly once. This traversal can certainly be done in polynomial time, which would put HAM-PATH into the NP category. Now to prove the complete part of NP-Complete we must find a problem that can be reduced into a HAM-PATH.

   From the NP-Completeness slide in the lecture we can see that a HAM-CYCLE can be reduced into a HAM-PATH. A HAM-CYCLE is simply a HAM-PATH that begin and end at the same vertex and we know that HAM-CYCLE is NP-complete so it is a good candidate. To convert from HAM-CYCLE we will create a graph, G', which is a copy of the graph used in the HAM-CYCLE, G, and add a couple vertices. The vertices that need to be added are u' and v'. These vertices will only be connected to the corresponding versions of themselves in the copy of the graph i.e. u' connects with only u and v' connects with only v. G' will have a HAM-PATH if (and only if) G has a HAM-CYCLE with edge = {u, v}. The HAM-PATH algorithm will need to be ran on each G' for all edges in G and if there is no HAM-PATH present then there is no HAM-CYLCE in the original graph.

5. The first step in determining if this problem is NP-Complete is to make sure that you can verify a given certificate for the problem in polynomial time. The information we need to do this verification would be the certificate or solution itself and a set of edges for G. We would need to verify that the edges are in the graph as well as sum up the edges to see if they match the solution k. This can be done in O(E) time (meaning it is based on the number of edges) and therefor it is polynomial. This proves that the LONG PATH problem is NP

   The next step would be to find a problem that can be reduced into the LONG PATH problem to prove that the problem is NP-Complete. From the problem above we know that the HAM-PATH problem is NP-Complete and from the chart in the lecture we know that the HAM-PATH problem can be reduced into the LONG PATH problem. The HAM-PATH problem we are looking to find the longest path without repeating an edge. The difference between this and the LONG PATH problem is that the LONG PATH problem is looking to find a path of at least length k. If we assume a connected graph, we can set all the edge weights in the graph to be equal to 1. Since each edge will only contribute one, we need k to be set equal to the number of all the vertices-1. Using the LONG PATH problem we can now see if there is a PATH of at least length k. If there is this would indicate that the solution is also a HAM-PATH and therefor shows that HAM-PATH can be reduced to LONG PATH and solved by LONG PATH. This proving that LONG PATH is NP-Complete.