CS 325 Spring 2018 – HW 7

Sheng Bian

Problem 1:

X reduces to Y implies X is no harder than Y.

a. No, it can't be inferred. X could be in NP.

b. No, it can't be inferred. Y could be in NP-hard.

c. No, it can't be inferred. X could be in P.

d. Yes, it can be inferred. Y can be both in NP and NP-complete.

e. No, it can't be inferred. X and Y could be both NP-complete. If Y is in NP-complete, X is no harder than Y so X could also be NP-complete.

f. No, it can't be inferred. Y could be harder than X.

g. Yes, it can be inferred. Since X is no harder than Y, X must be in P.

Problem 2:

a. False. SUBSET-SUM reduces to COMPOSITE is not correct since SUBSET-SUM is NP-complete and COMPOSITE is in NP. By definition, if X reduces to Y, then X is no harder than Y. NP-complete can't be reduced to NP problem because NP is easier than NP-complete.

b. True. $O(n^3)$ is a polynomial running time. Since SUBSET-SUM is NP-complete problem and it has polynomial running time algorithm, it implies all P = NP. Therefore, all problems in NP have polynomial running time algorithm, including COMPOSITE.

c. False. Even though COMPOSITE is in NP, it doesn't mean it's also in NP-complete. Therefore, polynomial algorithm for COMPOSITE doesn't imply P = NP.

d. False. P is a subset of NP and problems in P can be solved in polynomial time. P != NP only implies NP-complete problems can't be solved in polynomial time.

Problem 3:

a. True, 3-SAT can be reduced to TSP. 3-SAT and TSP are both NP-complete problems. By definition, X reduces to Y implies X is no harder than Y. Also, from lecture, 3-SAT can be first reduced to DIR-HAM-CYCLE, then reduced to HAM-CYCLE, finally reduced to TSP.

b. False. P != NP means there is no polynomial-time algorithm for 3-SAT. However, 3-SAT reduces to 2-SAT and 2-SAT is in P show that 3-SAT is also in P. Therefore, there is a contradiction.

c. True. If one NP-complete problem can be solved in polynomial time, all NP-complete problem can be solved in polynomial time. Therefore, either all NP-complete problems are in P or none of NP-complete problems are in P. P != NP implies none of problems are in P, then no NP-complete problem can be solved in polynomial time.

Problem 4:

1) Show that HAM-PATH $\in$ NP.

Given a graph G'(u, v) that has HAM-PATH. Let's start from the starting node u and traverse the path. Let's visit each vertex exactly once and reach the ending node v. It can definitely be completed in polynomial time. Therefore, HAM-PATH belongs to NP.

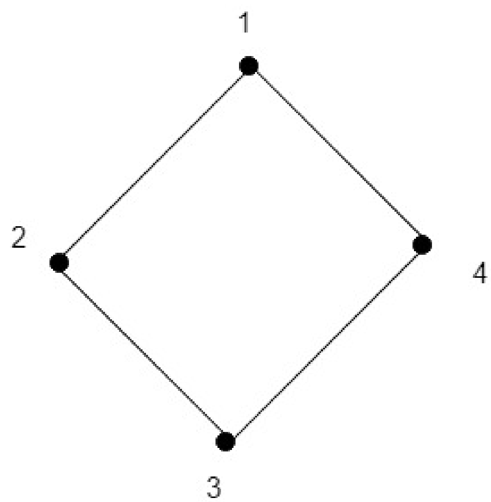
2) Show that R $\leq_P$ HAN-PATH for R $\in$ NP-Complete.

a. Select R=HAM-CYCLE because it has it has a similar structure to HAM-PATH. It known that HAM-CYCLE is NP-complete.

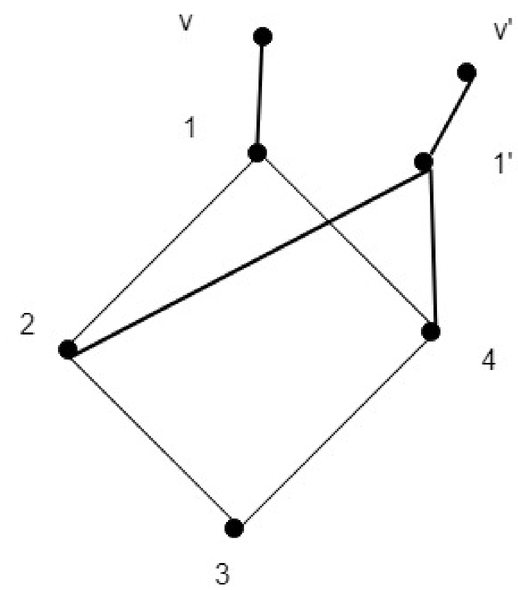
b. Show a polynomial algorithm to transform HAM-CYCLE into an instance of HAM-PATH.

Given a graph G we produce a new graph G' such that G has a HAM-CYCLE if and only if G' has a HAM-PATH. G' is created by choosing an arbitrary vertex u in G and adding a new vertex u'. u' is connected with the neighbour vertices of u. Then, add vertex v to connect with u and add vertex v' to connect with u'.

Below is an example of the transformation. The u is 1.



G



G'

c. Prove you are able to “solve” HAM-CYCLE by using HAM-PATH. Therefore HAM-PATH is as hard as HAM-CYCLE.

Show that the graph G has a HAM-CYCLE if and only if graph G’ has a HAM-PATH.

i) Now if G has a HAM-CYCLE 1, 2, 3, 4, 1, then G’ has a HAM-PATH v, 1, 2, 3, 4, 1’, v’.

ii) If G’ has a HAM-PATH v, 1, 2, 3, 4, 1’, v’, we can remove new added vertices v and v’. Then the HAM-PATH becomes 1, 2, 3, 4, 1’. Then remove 1’, the graph becomes G and there is a HAM-CYCLE 1, 2, 3, 4, 1.

Since both 1) and 2) are true, then HAM-PATH is in NP-Complete

Problem 5:

1) Show that LONG-PATH $\in$ NP.

Given a graph G(u, v) that has LONG-PATH. Let’s start from the starting node u and traverse the path. Let’s visit each vertex exactly once and reach the ending node v. Since it has a simple path in G from u to v of length at least k, it can obviously be completed in polynomial time. Therefore, LONG-PATH belongs to NP.

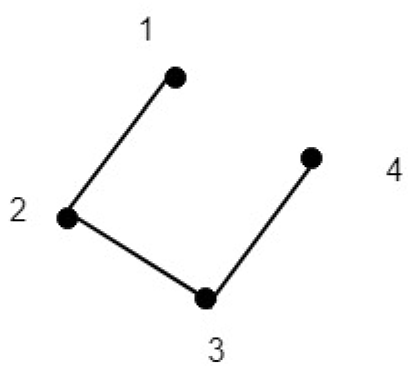
2) Show that R $\leq_P$ LONG-PATH for R $\in$ NP-Complete.

a. Select R=HAM-PATH because it has it has a similar structure to LONG-PATH. It known that HAM-PATH is NP-complete.

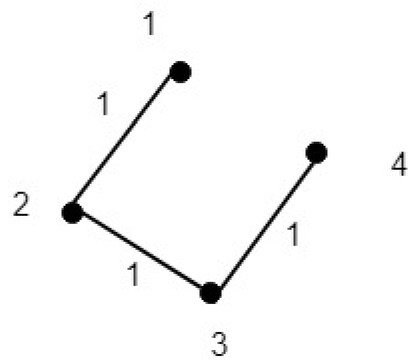
b. Show a polynomial algorithm to transform HAM-PATH into an instance of LONG-PATH.

Given a graph G we produce a new graph G' such that G has a HAM-PATH if and only if G' has a LONG-PATH. G' is created by simply add weight of 1 to each edge. Then find a path from u to v that has crossed at k number of edges.

Below is an example of the transformation.



G



G'

c. Prove you are able to "solve" HAM-PATH by using LONG-PATH. Therefore LONG-PATH is as hard as HAM-PATH.

Show that the graph G has a HAM-PATH if and only if graph G' has a LONG-PATH.

i) Now if G has a HAM-PATH 1, 2, 3, 4, then G' has a LONG-PATH 1, 2, 3, 4, and from 1 to 4 of length at least 3.

ii) if G' has a LONG-PATH 1, 2, 3, 4, and from 1 to 4 of length at least 3, after remove weight of each edge, G has HAM PATH 1, 2, 3, 4.

Since both 1) and 2) are true, then LONG-PATH is in NP-Complete

Extra Credit:

1) Show that TPP-D $\in$ NP.

Given a graph G(u, v) that has TDD-D. Let's start from the starting node u and traverse the path. Let's visit each vertex at most once and reach the ending node v. Since TDD requires start and finish at the same market place and visit each market place at most once, it can obviously be completed in polynomial time. Therefore, TDD-D belongs to NP.

2) Show that R $\leq_P$ TDD-D for R $\in$ NP-Complete.

a. Select R= Travelling Salesman Problem because it has it has a similar structure to TDD-D. It known that TSP is NP-complete.

b. Show a polynomial algorithm to transform TSP into an instance of TDD-D.

Given a graph G we produce a new graph G' such that G has a TSP if and only if G' has a TDD-D. Since G requires the shortest route to visit each and return to the original city given that a list of cities and the distances between each pair of cities. TPP-D is generalization of TSP. We can do the transformation by adding each good is available at one marketplace and each marketplace only sells one good. The shopping list of goods include all the goods at all the marketplaces. The total cost is goods price plus travel costs.

c. Prove you are able to "solve" TSP by using TDD-D. Therefore TDD-D is as hard as TSP.

The condition is adding each good is available at one marketplace and each marketplace only sells one good. The shopping list of goods include all the goods at all marketplace.

i) If TSP can be solved, then we find the shortest possible route that visits each city and returns to the origin city. We can just traverse the same path to solve the TDD-D.

ii) If TDD-D can be solved, then we have already traverse the shortest path through all the market places. The reason is good prices are fixed since each good is only available at one marketplace. Then we must choose the shortest path to get lowest travel cost. Therefore, TSP can be solve by traversing the same path.

Since both 1) and 2) are true, then TPP-D is in NP-Complete