**Assignment: 9**

1. Prove that Traveling Salesman Problem is NP-Complete.

Traveling Salesman Problem : *Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?*[1](https://en.wikipedia.org/wiki/Travelling_salesman_problem)

Part 1 : we have to prove that TSP fall under NP problem, so to prove that TSP fall under NP problem, we have to show that it can be verified in polynomial time. We can check if the path computed is covering all the nodes(this can be done in constant time), and secondly when we sum up the total cost of the edges and check if its minimum. Since finding the minimum is something which is subject to debate, as to check the minimum weight , we can’t come to conclusion unless we compute all the path to find the weight of different possible travellor’s path, and it can’t be computed in polynomial time. Hence solution cannot be determined in polynomial time hence first part fails.

Part 2 : To prove the problem fall in NP hard problem,

Now to prove that TSP is a NP hard problem ,we will take another NP complete problem and reduce it to TSP problem in polynomial time. So to prove TSP we will consider Hamilton Cycle which is know to be as NP complete problem. To make Hamiltonian circuit we consider each node as equivalent to city of TSP problem.

Given a graph G=(V,E), Hamiltonian path states does there exist a simply cycle in G that traverses every vertex exactly once,now to reduce HC to TSP we take G’=(V,E’), set all edge weights equal to 1, and let k = |V|=n, that is, k equals the number of nodes in G. Any edge not originally in G then receives a weight of 2. Then pass this modified graph into TSP, asking if there exists a Tour on G with cost at most k. If the answer to TSP is YES, then HC is YES. Likewise if TSP is NO, then HC is NO. So it means each edge should cost 1 , to form the Hamilton Cycle.which will result in giving TSP. So we have proven that G has a Hamiltonian cycle if and only if G’ has a tour of cost at most n. Thus TSP is NP-hard.

1. Prove that the following decision problem is NP-Complete LARGEST-COMMON-SUBGRAPH: Given two G1 and G2 and integer k, determine whether there is a graph G with > = k edges which is a subgraph of both G1 and G2 (Hint: reduce from CLIQUE)

Part 1 : we have to prove that Largest common subgraph fall under NP problem, so to prove that LCS fall under NP problem, we have to show that it can be verified in polynomial time. If we find a graph G we can verify if it is having all the vertices and edges are from both the graphs, G1 and G2 only. And non of the other node is missed which is common can be checked in polynomial time, hence its NP problem.

Part 2 : Now to prove if the Largest Common subgraph of given 2 graph is NP hard problem, we will take another NP complete problem and reduce it to Largest Common Subgraph. So to prove this we will consider Clique problem which is known to be as NP complete problem.

The reduction from the Clique Problem to finding maximum subgraph problem .A clique is a subset of nodes such that each pair of nodes is connected by an edge.

Given two subdivision graphs S(G1) = (V1 ∪ E1, E1, µ1, f0) and S(G2) = (V2 ∪ E2, E2, µ2, f0), the compatibility graph GC = (VC ∪ EC, EC, f0, νC )

Where VC = {(x1, xa) ∈ V1 × V2 | µ1(x1) = µ2(xa)}

EC = {(e1, ea) ∈ E1 × E2 | ν1(e1) = ν2(ea) and µ1(ends(e1)) = µ2(ends(e2))}

create a complete clique G'=(V,E') such that E' = {(u,v) | u != v, for each u,v in V).

The solution to the maximal clique problem is the same solution for the maximal subgraph problem for G and G'. Since clique problem is NP-Hard, so does this problem.

Thus, there is no known polynomial solution to this problem. Thus it is NP hard problem.

Since its NP as well NP hard, we can say it is NP complete problem.