Assignment 4

Görkem Güzeler

Spring 2021

1 Part A:

Input: Given set C of cities in Turkey and a set Sc of cities that the tourists are usually interested in visiting on their way to c ϵ C.

Output: It should compute a shortest itinerary Ic that includes all the cities in Sc in the path from İstanbul to each c ϵ C.

2 Part B:

In order to show the NP-completeness, we should prove both membership and hardness of the problem. Let our problem be L.

Membership: We should show if L is in NP. Problem definition already given above with input and outputs. We will use this information when building our algorithm. C and Sc are given as input, we think of them as vertices in our graph G. We should come up with Ic as output. For every input, our algorithm should return an Ic. At each vertex, there will be polynomial number of options to select a city from the Ic union C. To put through the idea in another way, there are k cities to visit and at each vertex we also have k cities to choose. Therefore, L is in NP.

Hardness: Let's use one of the NP-Complete problems to reduce into our problem. I choose Hamiltonian Cycle Problem to apply reduction. In a nutshell, we can define Hamiltonian Cycle Problem such that there is cycle in a graph that visits all of the vertices exactly once which is almost the same idea behind our problem. Therefore, we need to relate our problem with the Hamiltonian Cycle Problem. Let our problem be L and Hamiltonian Cycle Problem H.

There is a transitivity property of Hardness. We already know that H is NP-complete which means H is NP-hard and every problem x in NP can be reduced to H in polynomial time, thus x => H is already satisfied. If we show that H reduces into L (H => L), we will prove that L is NP-Hard (x => H):

We will convert our problem L to a decision problem by asking that "is there L ,that consists of given cities, with the cost of k at most?". Since our problem asks for the shortest itinerary, we will be keeping k as lower as we can. Given G(V,E), one of the vertices will be chosen as Istanbul while others will be split into two groups such as Sc and C. We will start from Istanbul, we will visit the cities in Sc while we are reaching to the cities in the C. Then, we check whether there exist an itinerary from Istanbul to c with visiting Sc union C exactly once with cost of at most k. If the answer is yes with the given cost k, we know that there is a cycle which visits each city exactly once and returns back to the source vertex. If we use it in our problem L, starts from Istanbul visits all the cities in Sc to reach to C and goes back to Istanbul. If the answer is no with the given cost, we know that there does not exist Hamilton Cycle which shows solution of L is not possible with cost k. We proved that H reduces into L (H => L). All in all, x => H and H => L leads to the result of x => L.

Since we proved L is in NP and L is NP-Hard, we proved it is NP-complete.