December 10, 2015

CS 577 - Homework 11
Sejal Chauhan, Vinothkumar Siddharth, Mihir Shete

1 Graded written problem

Part a: Two Travelling Turkey Problem (TTTP) as mentioned in the written problem has the following contraints:

- Each city is visited at least once
- Number of distinct routes has to be as small as possible
- Each city can be visited any number of times
- Path starts and ends at the same city (DC)
- Maximum of the total effort of the two turkeys is minimized

Formally we can describe the desicion version of the problem as :

Given a doubly-edged complete graph G with different edge weights, find a spanning tree which has at most cost k for the maximum of the two costs among all possible spanning trees. For the graph G, find if there exists a spanning tree whose cost is atmost k for either of the summation of the edges else report that no solution exists. It can be checked in polynomial time, if such a spanning tree exists. Hence the above formulization fits in the framework of NP-complete desicion problem under polynomial time reductions, as described in the lecture notes.

Part b: Decision problem is NP-complete

To prove that TTTP decision problem is NP-complete, we will show that Partition Set(PS) that is a known NP-complete problem, can be reduced to TTTP problem. Given a set S that must be partitioned such the the sum of the two subsets should be equal. We need to pre-process the subset problem in such a manner that it cn be given to TTTP as an input. For the PS problem, our input is an array of length n. We create a root node, and for every element in S we add two vertices with edge weight as (S[i], 0) and (0, S[i]). The edge weight between these two vertices is (0,0). Construct this graph with all the elements in S and make it a completed connected graph with all the other vertices is (,). This is a polynomial time task.

This graph is now given as an input to the TTTP problem. Each edge is doubly weighted and TTTP has to find whether such a solution for this completed graph exists or not wherein k is equal to half of the total sum of the set S. TTTP will find a spanning tree with the given vertices that will ensure every element of the set is considered. Else it will report that no such solution exists. Let us consider that the edges that were considered to construct the spanning tree is (S[0],0), (0,S[1]), (0, S[2]).. etc. We will now carry out ordered sum for all these edges and obtain (S1,S2), both should be equal to k. This addition is a polynomial time task. Now, the elements in S1 and S2 is the required subsets solution for the PS NP problem.

This is how we can successfully reduce PS to TTTP.