**T17T (cs170-hw) – CS 170 Project Write-up**

(Albert Weng, Ethan Chiou, Justin Bu, Spencer Axelrod)

General Algorithm/Approach

We actually used 3 different algorithms to tackle this project. Initially, we thought that given enough time (let’s say a few hours), we would be able to exactly solve all instances of size 50 with the optimal solution. After doing a little arithmetic, we quickly came to the conclusion that this option was not plausible given the designated amount of time. So we had to look for other routes.

First thing that came to mind was the greedy solution. We run this algorithm *N* times, where *N* is the number of nodes of the input. Each time we run this algorithm, we start at a node, and greedily traverse the shortest edge going out of our current node, if legal (that is, if we hadn’t traversed 3 nodes of the same color already). After running this algorithm starting at every node in the graph, we select the path that yields the lowest total path weight.

We then came across the ratio-2 approximation algorithm for regular TSP by taking advantage of the minimum spanning tree of the graph. We decided to apply the same algorithm given the new constraints of our traversal.

Lastly, we noticed that some submissions had fewer than 15 nodes in their graphs. For these smaller graphs, we decided to just enumerate all possible legal paths and select the smallest one, guaranteeing optimality.

Input Files

For our input files, we decided to start out with just the output of a random number generator that matched the constraints of the specifications. After discussing the algorithm a bit more, we came up with the idea that if our idea path followed a “diamond” structure, then a greedy solution would have a very small chance of finding the optimal solution of the problem. DRAW THE PICTURE SOMEWHERE HERE. Unfortunately, we ran out of time before we were able to incorporate this structure into all of our input files.

Code Organization/How To Run Code

Resources Used

*Algorithms* by U.V. Vazirani, C. H. Papadimitriou, and S. Dasgupta