UNIVERSITY OF CALIFORNIA, DAVIS
Department of Electrical and Computer Engineering

EEC289Q: Modeling and Optimization for Computer Engineers

HW Assignment #3

via a Canvas assignment (please note that late submission will not be possible)

1. A path in a graph is a sequence of nodes each of which is connected to the next node in the sequence. A cycle is a path that does not intersect itself and ends at the starting node of the sequence. Stated differently, a cycle is a sequence of nodes v_1, v_2, v_3, … , v_n, v_1 in which, all nodes in the sequence are distinct (i.e., except for v_1, no other node appears in the sequence more than once), and there is an edge between any pair of adjacent nodes in the sequence. Cost of a cycle is defined as sum of the cost of the edges that constitute the cycle (i.e., the edges between pairs of adjacent nodes in the cycle).

Given graph G (V, E), the Traveling Salesman Problem (TSP) refers to finding the cycle that visits all nodes in V, and has the smallest total cost. For this homework problem, the objective is to solve TSP in the practical sense on two example 1000-node graphs posted on canvas given 1 min computation time budget on a typical personal computer. The term "practical sense" implies that we are interested in reducing the cost of solution to the extent possible, rather than guaranteeing optimality over all possible cycles in the graph.

Both example graphs are complete graphs, i.e., there exists an edge between any pair of nodes, though they are different in their edge costs:

A) The nodes represent points on 2-d plane. Edge costs are the Euclidean distance between the points.
B) Edge costs are sampled form a uniform probability distribution over the range 0 to 100.

Your solution should be submitted as two files (one pdf file and one text file) including the following information:

PDF file:

1. Describe your algorithm for both graphs A and B: Please use visuals, flowcharts, pseudocode and/or text as appropriate to clearly describe the final algorithm (out of several that you may have tried) that you used.
2. Separately for graphs A and B: how many cycles does your algorithm evaluate and what is the cost of the best cycle that it could find? Note that you should limit your code to run NO more than 1 min. Report the number of visited cycles in scientific format with one digit before the exponent (e.g., 4e12), and round the cycle cost up to two digits after the decimal point.
3. A link to the github repo of your code. Please make sure it is a public repo, and everyone has read/view access.

Text File:

Save your solution, i.e., the sequence of node indices that make up the best cycle that your algorithm found, in a text file named "solution_SID.txt", where SID in the file name should be replaced with your 9-digit student ID. The cycle should be outputted as a sequence of node indices separated by commas. For example, a cycle may read like:

232, 149, 980, 12, 30, 54, …. 678, 232 (30 points)

2. Suppose we are given an array A[1 .. m][1 .. n] of non-negative real numbers. We want to round A to an integer matrix, by replacing each entry x in A with either floor(x) or ceiling(x), without changing the sum of entries in any row or column of A.
   For example:

$$\begin{bmatrix} 1.2 & 3.4 & 2.4 \\ 3.9 & 4.0 & 2.1 \\ 7.9 & 1.6 & 0.5 \end{bmatrix} \longmapsto \begin{bmatrix} 1 & 4 & 2 \\ 4 & 4 & 2 \\ 8 & 1 & 1 \end{bmatrix}$$

   Describe and analyze an efficient algorithm that either rounds A in this fashion, or reports correctly that no such rounding is possible. (10 points).