CS 538: Problem Set 1

Section: MW 2-3:15 pm

Total: 100pts Due: 02/27/2017

Instructions:
1. I leave plenty of space on each page for your computation. If you need more sheet, please attach your work right behind the corresponding problem. If your answer is incorrect but you show the computation process, then partial credits will be given. Please staple your solution and use the space wisely.

2. This assignment contains two parts. You are allowed to work on the homework in a group (you can stick to your notes-taking group or form a different one). No late assignment is accepted. Identical solutions (same wording, paragraph, code), turned in by different groups (persons), will be considered cheating.

3. Full credit will be given only to the correct solution which is described clearly. Convoluted and obtuse descriptions might receive low marks, even when they are correct. Also, aim for concise solutions, as it will save you time spent on write-ups, and also help you conceptualize the key idea of the problem.

First Name:

Last Name:

Group ID:

Score: **/ 100 + 15**

Problem 1 Eulerian Cycle: 10pts

Please show that for a connected multi-graph $G = (V, E)$, it has an Eulerian cycle if and only if every vertex has an even degree.
(A) G has an Eulerian cycle $\rightarrow$ all vertices have an even degree

(B) All vertices in G have an even degree $\rightarrow$ there is an Eulerian cycle in G

Problem 2 Eulerian Cycle: 10pts

Show a *polynomial time* algorithm α that acts on a connected graph G , in which every vertex has even degree, outputs an Eulerian cycle.

Problem 3 Vertex Cover: 10pts

Suppose we are given an algorithm listed below. Does this algorithm guarantee a 2OPT? If yes, explain why; if not, please give an counter example.

Algorithm 1 Vertex Cover Made Easy

Require: Graph $G = (V, E)$
Ensure: Output the vertex cover of 2OPT
  Start of algorithm
  Vertex cover $C := \emptyset$
  while $\exists(u, v) \in E$ such that $u \notin C$ and $v \notin C$ **do**
    $C := C \cup \{u\}$
  Return C
  End of algorithm

Problem 4 Steiner Tree: 20pts

Please briefly describe a 2OPT algorithm for a steiner tree problem (X, d) where $X = R \cup S$ (R is the set of required vertices and S is the set of optional vertices) and d is the distance function that DOES NOT have to satisfy triangle inequality.

Problem 5 Programming: Travelling Salesman Problem G_TSP_R ← M_TSP_R (Eulerian(MST + PM)): 50pts (5*2+8*5)

Please use the language of your choice to do that following:
Suppose we have n cities to visit, the distance between two arbitrary cities are symmetric, i.e. $d(x,y) = d(y,x)$ and it does not have to satisfy the triangle inequality.

(a-1) Design an arbitrary non-negative symmetric distance function d and represent this function in an n by n matrix with all zero terms in the lower diagnoal and the diagonal line.

(a-2) Simplify d into distance function d' that such that it satisfies the triangle inequality and verify it (Dijkstra's algorithm might help).

(b) You will be given a matrix (only the uppr right diagonal, see below) that encodes the distance (satisfies the triangle inequality) between cities (let say name them city $1, \cdots, 5$). For instance, $d''(1,5) = d''(5,1) = 4$.

$$d'' = \begin{pmatrix} 0 & 7 & 12 & 7 & 4 \\ & 0 & 8 & 9 & 4 \\ & & 0 & 14 & 10 \\ & & & 0 & 5 \\ & & & & 0 \end{pmatrix}$$

(b-1) Generate a Minimal Spanning Tree T for these 5 cities such that $cost_{d''}(T) = 21$

(b-2) Generate a 2*OPT M_TSP_R cycle from the result (b-1) by use of DFS on T with OPT(M_TSP_R) ≥ 42; then simplify it such that cost can be lowered to 37

(b-3) Implement perfect matching such that $M = Perfect\_Matching(T)$. From the output of your perfect matching to confirm that OPT(M_TSP_R) $\geq 2 * cost_{d''}(M) = 34$

(b-4) By using the algorithm you mentioned earlier for finding Eulerian cycle in a graph (T + M) , show that you can find an Eulerian cycle with cost 38. (and yes, $38 \leq 1.5 * 34$)

(b-5) Optimize (b-4) by use of the optimization approach used in (b-2) to show you can further reduce the cost of 38 to 36.

(c-1) Bonus (5pts) : In the M_TSP_R algorithm, let $O :=$ the set of vertices that have odd degrees in T. Why finding the perfect maching (O, M) and adding it back to the T would guarantee an Eulerian cycle in graph $(T + M)$?

(c-2) Bonus (10pts): Briefly describe how you would extend this implmentation to any arbitrary nonegative symmetric distance funciton d (e.g. in (a - 1)) such that you can can easily find a 1.5 OPT of **G_TSP_R** ?