Group D1 Jiang Wenhan Li Rong Zhong Tiantian Zhou Ruidi

Ex 1

We just need to compare the max length of the edge in graph (we suppose the max length is $1_i^,$) Other edges' lengths are length 2_i.

If length $2_i <$ length $1_i^,$ then $E = E - 1_i^,$

else we just keep it in the graph.

Another situation: if $v_i \notin V$ or $v_i^, \notin V$, then $E = E + (v_i, v_i^,)$

Ex 2

I) assume there are 4 vertexes, the distance matrix is shown below:

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 999 & 999 \\ 1 & 999 & 0 & 999 \\ 1 & 999 & 999 & 0 \end{bmatrix}$$

Obviously, the tour will cost more if it is forbidden to visit a city (city 1) more than once.

II)

suppose we have to pass through city x for at least two times, then there must be two edges included in tour. We assume them as $\{x, a\}$ and $\{x, b\}$. According to the definition of Euclidean matrix, $d(x, a) + d(x, b) \geq d(a, b)$, so we can simply replace these two with edge $\{a, b\}$.

III)

The first step is to create a MST with Prim's algorithm.

Suppose the length of the chosen path is l, then the total length of the tour is $2l$ since the length of path returning is l as well.

In the case of optimal tour, it will turns to a MST if we deduce one edge from it. Thus, the optimal length $l_0 > l$. Therefore, the total tour length, which is $2l$, is less equal than $2l_0$, two times of the optimal length.

EX 3

I) It can be simplified using the methods of recursion. Firstly set the stop condition of the recursion as following:

1. when there is no edge coming out this vertex
2. when it reaches the destination

   Then we set the next recursion to the first possible edge out of vertex, and apply this again at the vertex linked by this edge.

II) Not safe. the worst case time complexity of Ford-Fulkerson algorithm is $O(m \times f)$ (m refers to the number of edges and f refers to the max flow). It is possible to cost a huge quantity of time to finish the algorithm in some extreme conditions (for example, five edges in total, four of them has capacity of 9999, but one of them has capacity of 1, it is possible to cost 9999 times to finish the algorithm). Thus, there is need to set limitation upon the running times.