

Lab07-Network Flow

CS214-Algorithm and Complexity, Xiaofeng Gao, Spring 2019.

* If there is any problem, please contact TA Mingran Peng.

* Name:Lynn Xiao Student ID: Email:

1. Remember the network problems in last lab?

Consider there is a network consists n computers. For some pairs of computers, a wire i exists in the pair, which means these two computers can communicate with delay t_i .

The distance of two computers are defined as their communication delay. Design an algorithm to compute the maximum distance in the network.

You need to provide the pseudo code and analyze the time complexity.

Solution.

Algorithm 1: Maximum Distance Algorithm

Input: Graph $G(V, E)$ (computer $\in V$ and wire $\in E$), the number of the computer n

Output: The maximum distance in the network

```
1 for  $i \leftarrow 1$  to  $n$  do
2   for  $j \leftarrow 1$  to  $n$  do
3     if wire  $w$  connects  $i, j$  then
4        $c_{ij} = t_w$ 
5     else  $c_{ij} = \infty$ 
6   for  $k \leftarrow 1$  to  $n$  do
7     for  $i \leftarrow 1$  to  $n$  do
8       for  $j \leftarrow 1$  to  $n$  do
9          $c_{ij} = \max\{c_{ik} + c_{kj}\}$ 
10  return max value of the matrix  $C$ 
```

This algorithm uses Floyd algorithm to compute the distance of each pair of computers. The time complexity of the Floyd algorithm is $O(n^3)$ and finding the maximum value is $O(n^2)$. So the time complexity of the algorithm is $O(n^3)$. \square

2. Suppose you are traveling through a country defined as a directed graph $G = (V, E)$. You start from vertex s and want to go to e . For each $i \in E$, there is a w_i regarding the cost for traveling via i . Some times $w_i < 0$ which means you can earn money by traveling. (Do not ask why.) Here you need to design an algorithm satisfying the following demands:
 - (a) Find the minimum cost from s to e . The problem guarantee that there is a path from s to e .
 - (b) Indicate whether there is a circle in G , that by traveling through this circle you can earn money continually.

You need to provide the pseudo code and analyze the time complexity.

Solution.

Algorithm 2: Minimum Traveling Cost Algorithm

Input: Graph $G(V, E)$, number of vertices s and number of edges e

Output: Whether there is a negative circle and the minimum cost from s to e

```
1 for node  $u \in v$  do
2    $M[0, u] = \infty$ 
3  $M[0, s] = 0$ 
4 for  $i \leftarrow 1$  to  $n$  do
5   for node  $u \in v$  do
6     if  $M[e]$  has been updated in previous iteration then
7       for edge  $(e, v) \in E$  do
8          $M[i, e] = \min\{M[i-1, e], M[i-1, v] + w(e, v)\}$ 
9 if  $M[n, u] = M[n-1, u]$  for all  $u \in v$  then
10   return  $M[n, e]$ 
11 else There is a negative circle in  $G$ .
```

□

3. (Bonus) Suppose you are a staff of SJTU who is in charge of arranging lessons. Suppose you have n time slots, n lessons and n professors. Clearly, you should assign exactly one time slot and one lesson to every professor. A lesson or a time slot should be assigned to exactly one professor.

For each professor, he will prefer some certain time slots among these n time slots, and prefer to taught some certain lessons among these n lessons.

A professor will be satisfied iff you arrange him both his preferred lesson and preferred time slot. Your goal is to satisfy as many professors as you can. Design an algorithm to output how many professors can you satisfy at most.

Notice that this problem is really hard, even draft idea is welcomed.

(Hint: Treat time slots, lessons, professors as nodes. Construct edges according to professors' preference. Use network flow algorithm to solve it.)

Solution.

We can construct a graph with n time slot nodes, n lesson nodes, n pre-professor nodes and n post-professor nodes. In the graph, directed edges of capacity 1 are used to connect them.

For the time slots and lessons that a professor prefers, we connect the time slot node to pre-professor node and post-professor node to lesson node. Also, pre-professor node is connected to the corresponding post-professor node. A source node connects to all time slot nodes. All lesson nodes connect to a target node.

Then to compute how many professors are satisfied, we only need to compute the max flow of the graph. So we can use Ford-Fulkerson algorithm with the time complexity $O(mn)$. □

Remark: You need to include your .pdf and .tex files in your uploaded .rar or .zip file.