Lab07-Network Flow

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

∗ If there is any problem, please contact TA Mingran Peng.
∗ Name:Tianyao Shi Student ID:517021910623 Email: sthowling@sjtu.edu.cn

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. Just as stated in last lab, we can establish a graph $G = (V, E)$ where V is the set of computer as vertices, E is the set of wires as edges with weight $w(e_i) = t_i$. The background of this problem ensures that $\forall i, t_i \geq 0$, so there is not negative cycle.

   As demonstrated in wechat group, the actual definition of the distance between two computers is there *minimum communication delay*. Therefore the problem takes shape that is first to find the shortest path between all pairs, then pick up the longest shortest path between all pairs. Since there is not negative cycle, we can simply run Flyod-Warshall on this graph to get a matrix, then pick up the value of the largest element(s) in the matrix. Algorithm 1 shows the process.

   Algorithm 1: Maximum distance in the network

   input : An $n \times n$ adjacent matrix A of graph G, where $a_{ij} = w(e_{ij})$.
   output: Maximum length of shortest paths between all pairs.

   ```
   max← 0;
   for k ← 1 to n do
       for j ← 1 to n do
           for i ← 1 to n do
               if a_ij > a_ik + a_kj then
                   a_ij ← a_ik + a_kj;
   for j ← 1 to n do
       for i ← 1 to n do
           if a_ij >max then
               max← a_ij;
   return max;
   ```

   The time complexity of Flyod-Warshall to compute all-pair shortest path is $\Theta(n^3)$, and traversing the matrix to find the largest element is of complexity $\Theta(n^2)$, so in all the time complexity of this algorithm is $\Theta(n^3)$, where $n = |V|$. □

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.

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.)

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