組別: 姓名: 學號: 分數: 批改者:

一人寫一份，可討論。上台講解且正確者，全組自行加一分。

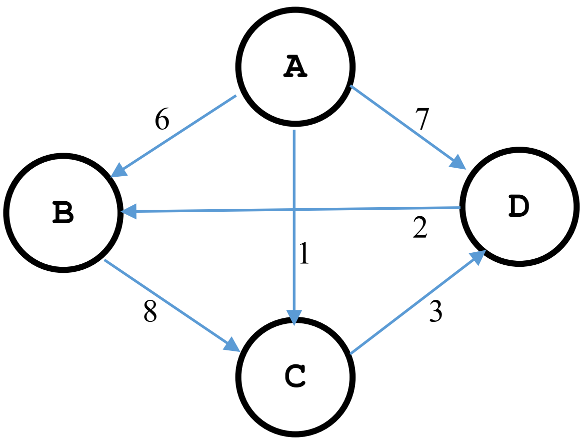
Algorithms

Classwork # Video 8.1-8.4

Each problem is 3 points.

3 = correct. 2 = one mistake. 1 = many mistakes. 0 = no answer.

1. **APSP Algorithm [8.1]**
   Run APSP algorithm we taught in 8.1 on the following graph. Please show all L metrics in your steps. What is the cost from A to D? What is the cost from D to C?



Solutions:

$$L^{(1)} = \begin{bmatrix} 0 & 6 & 1 & 7 \\ \infty & 0 & 8 & \infty \\ \infty & \infty & 0 & 3 \\ \infty & 2 & \infty & 0 \end{bmatrix} \qquad L^{(2)} = \begin{bmatrix} 0 & 6 & 1 & 4 \\ \infty & 0 & 8 & 11 \\ \infty & 5 & 0 & 3 \\ \infty & 2 & 10 & 0 \end{bmatrix}$$

$$L^{(3)} = \begin{bmatrix} 0 & 6 & 1 & 4 \\ \infty & 0 & 8 & 11 \\ \infty & 5 & 0 & 3 \\ \infty & 2 & 10 & 0 \end{bmatrix}$$

From A to D: 4

From D to C: 10

2. Floyd-Warshall Algorithm [8.2]

Suppose that we modify the definition of the predecessor matrix Π from (1) to (2), does Floyd-Warshall Algorithm can still work correctly? Please answer with reasons.

$$\pi_{ij}^{(k)} = \begin{cases} \pi_{ij}^{(k-1)} & if\ \ d_{ij}^{(k-1)} \le d_{ik}^{(k-1)} + d_{kj}^{(k-1)} \\ \pi_{kj}^{(k-1)} & if\ \ d_{ij}^{(k-1)} > d_{ik}^{(k-1)} + d_{kj}^{(k-1)} \end{cases} \quad i \neq j \quad \textbf{(1)}$$

$$\pi_{ij}^{(k)} = \begin{cases} \pi_{ij}^{(k-1)} & if\ \ d_{ij}^{(k-1)} < d_{ik}^{(k-1)} + d_{kj}^{(k-1)} \\ \pi_{kj}^{(k-1)} & if\ \ d_{ij}^{(k-1)} \ge d_{ik}^{(k-1)} + d_{kj}^{(k-1)} \end{cases} \quad i \neq j \quad \textbf{(2)}$$

Solutions:

We can still generate a correct π matrix. These two methods are only different when there exist more than two shortest path. When updating the π matrix, (1) will choose to save the original path, and (2) will choose to save a new path.

3. Transitive Closure of a Dynamic Graph [8.3]

Suppose that we wish to maintain the transitive closure of a directed graph $G = (V, E)$ as we insert edges into E. That is, after each edge has been inserted, we want to update the transitive closure of the edges inserted so far. Assume that we represent the transitive closure as a Boolean matrix. Show how to update the transitive closure $G^* = (V, E^*)$ of a graph $G = (V, E)$ in $O(V^2)$ time when a new edge is added to G.

Solutions:

1. If we add an edge E' between vertices (x_1, x_2).
2. Then we have to consider each pair of vertices (u, v) in G. Because each pair of vertices may be connective through E'.
3. If there are no path between (u, v) originally, we have to check whether there is a path between (u, x_1) and (x_2, v). If both (u, x_1) and (x_2, v) are connected in G, (u, v) is connected in G^*.
4. If (u, v) is connected in G, it remains connected in G^*.
5. To consider every pair of vertices once time, the runtime is $O(V^2)$.

4. Transitive Closure [8.3]

Give an $O(VE)$-time algorithm for computing the transitive closure of a directed graph $G = (V, E)$. Assume $V, E \geq 1$.

Solutions:

By using BFS from vertex u, we can determine whether other vertices are reachable from u. If we only initialize all G once in $O(V)$, the BFS remain $O(V)$. In this way, we can get the transitive closure in $V \times O(E) = O(VE)$ time by starting from each vertex in G.

5. APSP Algorithm [8.1]

Modify **SLOW-ALL-PAIRS-SHORTEST-PATHS** so that it can determine whether the graph contains a negative-weight cycle in the same time complexity.

Solutions:

Because the length of negative-weight cycle is at most n, and the length of SHORTEST-PATHS should not change after L^{n-1}. We can compute for one step more than the original, which compute all the way up to L^n. There are negative weight cycles if and only if any elements change or diagonal has negative number. Because we only do one time more than original method, the time complexity remain in the same.