CSI 604 – Spring 2016 – All-Pairs Shortest Path Algorithms

Note: $G(V, E)$ is a directed graph with n nodes, numbered 1, 2, …, n. $W = [w_{ij}]$ is an $n \times n$ matrix such that $w_{ii} = 0$ for each i, $1 \leq i \leq n$, and w_{ij} is the weight of the directed edge (i, j). The edge weights may be negative, but it is assumed that there are no negative weight cycles. The goal is to compute an $n \times n$ matrix $\Delta = [\delta_{ij}]$ such that δ_{ij} gives the length of a shortest path from i to j.

Note: In the following algorithm, D is the matrix that gives the length of shortest paths consisting of at most $r - 1$ edges, for some $r \geq 1$. The algorithm returns the matrix D' that gives the length of shortest paths consisting of at most r edges.

Extend-Shortest-Paths (D, W)

1. **for** $i = 1$ **to** n **do**

   for $j = 1$ **to** n **do**

   (a) $d'_{ij} = \infty$.

   (b) **for** $k = 1$ **to** n **do**

   $d'_{ij} = \min\left\{d'_{ij},\ d_{ik} + w_{kj}\right\}$.

2. Return $D' = \left[d'_{ij}\right]$.

Note: The following algorithm computes the product C of two $n \times n$ matrices A and B. Note the structural similarity between this algorithm and the Extend-Shortest-Paths algorithm above.

Matrix-Multiply $(A,\ B)$

1. **for** $i = 1$ **to** n **do**

   for $j = 1$ **to** n **do**

   (a) $c_{ij} = 0$.

   (b) **for** $k = 1$ **to** n **do**

   $c_{ij} = c_{ij} + a_{ik} * b_{kj}$.

2. Return $C = [c_{ij}]$.

Slow-All-Pairs-Shortest-Paths (W)

1. $D^{(1)} = W$.

2. **for** $r = 2$ **to** $n - 1$ **do**

   $D^{(r)} =$ Extend-Shortest-Paths$(D^{(r-1)},\ W)$.

3. Output $D^{(n-1)}$. /* This is the required Δ matrix. */

(over)

Faster-All-Pairs-Shortest-Paths (W)

1. $D^{(1)} = W;\ r = 1.$
2. **while** $n - 1 > r$ **do**
   (a) $D^{(2r)} =$ Extend-Shortest-Paths($D^{(r)},\ D^{(r)}$).
   (b) $r = 2 * r$.
3. Output $D^{(r)}$. /* This is the required Δ matrix. */

Floyd-Warshall (W)

1. $D^{(0)} = W$.
2. **for** $k = 1$ **to** n **do**

   for $i = 1$ **to** n **do**
   for $j = 1$ **to** n **do**
   $d_{ij}^{(k)} = \min\left\{d_{ij}^{(k-1)},\ d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right\}$.
3. Return $D^{(n)} = \left[d_{ij}^{(n)}\right]$.

Floyd-Warshall-Transitive-Closure (G)

1. **for** $i = 1$ **to** n **do**

   for $j = 1$ **to** n **do**
   if $(i = j)$ or $(i, j) \in E$
   then $t_{ij}^{(0)} = 1$
   else $t_{ij}^{(0)} = 0$
2. **for** $k = 1$ **to** n **do**

   for $i = 1$ **to** n **do**
   for $j = 1$ **to** n **do**
   $t_{ij}^{(k)} = t_{ij}^{(k-1)} \vee (t_{ik}^{(k-1)} \wedge t_{kj}^{(k-1)})$.
3. Return $T^{(n)} = \left[t_{ij}^{(n)}\right]$.
