Kelsey Neis - neis@umn.edu
HW 7 - Dec. 14
Section 001

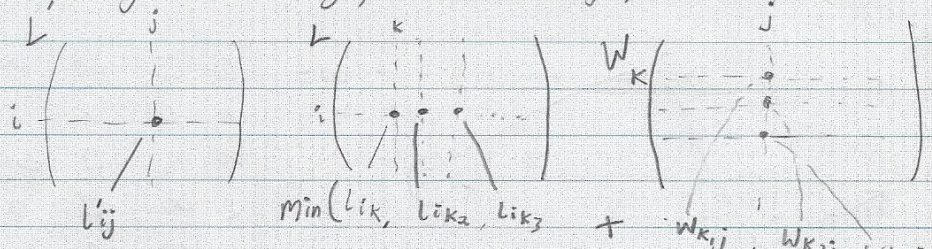
25.1

2) There are no edges between i and i, so the weight must be 0. Staying at a vertex should not add to the length of a path. Zero ensures this.

3) In the problem of shortest path, a weight of ∞ is an annihilator weight. It indicates there is no path between the vertices. ∞ is also the identity for min. And 0 is the identity for addition. In general matrix multiplication, 0 is the annihilator and 1 is the identity, so $L^{(0)}$ could be translated to a matrix with 1 on the diagonal and 0 off of it, which is the identity matrix I.

4) The important the computation happens on line 7,

$$l'_{ij} = \min(l'_{ij}, l_{ik} + w_{kj})$$



from 25.2, $l_{ij}^{(m)} = \min_{1 \le k \le n} \{ l_{ik}^{(m-1)} + w_{kj} \}$

min corresponds to +, so replace min w/ $\sum$:

$$= \sum_{k=1}^{n} a_{ik} \cdot b_{kj} = [A(BC)]_{ij}$$

+ corresponds to multiplication in this problem, because it represents the combined edges

$$= [(AB)C]_{ij}$$

$$\sum_{k}^{n} (AB)_{ik} C_{kj} = \sum_{k} \left(\sum_{l}^{n} A_{il} B_{lk} \right) C_{kj}$$

In case of min, the above shows that you can group the mins in any way. So, $\min_{1 \le k \le n} \{ l_{ik}^{(m-1)} + w_{kj} \} =$

$$= \min \{ \min_{1 \le k \le l} (l_{ik} + w_{kl}), \min_{l \le k \le j} (l_{ik} + w_{kj}) \}$$

5) Z^{n-1} (final product) $= (X_1, X_2, \ldots, X_n)$, where X_j = shortest path $s \to j$

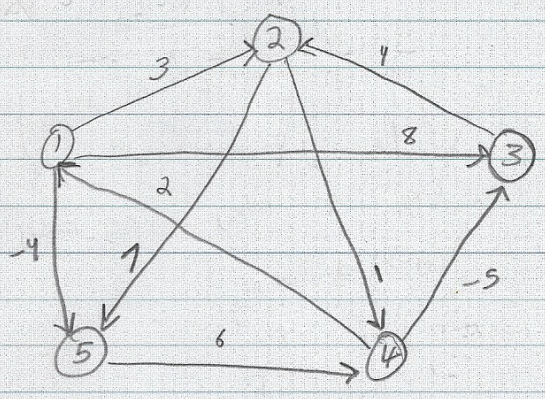
$W = i\begin{pmatrix} \infty & \cdots & j & \cdots \\ \vdots & & & \end{pmatrix}$ = weights of the edges in the graph

$Z^k = (X_1, X_2, \ldots X_n)$, where X_j = wt of min k-step path from s to j

to obtain Z^{n-1}, take $\min_{1 \le k \le n} \{ Z_k^{(m-1)} + w_{kj} \}$

$\Rightarrow \sum_{k=1}^{n} z_k^{(m-1)} \cdot w_{kj}$

(6)



$$W = \begin{pmatrix} 0 & 3 & \infty & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$L = \begin{pmatrix} 0 & 1 & -3 & 2 & (-4) \\ 3 & 0 & -4 & (1) & -1 \\ 7 & (4) & 0 & 5 & 3 \\ (2) & -1 & (-5) & 0 & -2 \\ 8 & 5 & 1 & (6) & 0 \end{pmatrix}$$

$$\pi_{ij} = \begin{cases} 0 \text{ (or NIL)}, & i = j \text{ or } L_{ij} = \infty \\ i, & l_{ij} = w_{ij} \\ k, & \text{otherwise} \end{cases}$$

1. Identify edges where $w_{ij} = l_{ij}$: $\pi_{ij} = i$
These will be the π values for all other vertices. In the same iteration, set π_{ij} to NIL if either $i = j$ or $l_{ij} = \infty$

$$\begin{pmatrix} \text{NIL} & & & & (1) \\ & \text{NIL} & & (2) & \\ & (3) & \text{NIL} & & \\ (4) & & (4) & \text{NIL} & \\ & & & (5) & \text{NIL} \end{pmatrix}$$

2. For the remaining empty entries,
- if only one distinct non-NIL value in the column, set π_{ij} to that value

- if more than one distinct value exists in the column, compare the l values for the current row and select the min of the l value plus the weight of the candidate predecessor

j	1	2	3	4	5
	NIL	3	4	5	①
	4	NIL	4	②	1
	4	③	NIL	2	1
	④	3	④	NIL	1
	4	3	4	⑤	NIL

Only one distinct value for a direct shortest path to vertex ①, so all other indirect paths must go through it

In column 3, there are two possible predecessors for getting to vertex ④

Example: The example in 25.1 has one column where the choice for predecessor is not trivial. To get to vertex ④, you can either go through ② or ⑤. To solve this, all we need to do is to take the min of l_{i2} plus wt of last edge. For example, $\pi_{14} = \min(l_{12} + w_{24}, l_{15} + w_{54}) = 2$. The index of the winning j is π_{14}, so ⑤, and $\pi_{34} = \min(l_{32} + w_{24}, l_{35} + w_{54}) = 5$, where $j = 2$

More formally: $\pi_{ij} = \begin{cases} NIL, & i = j \text{ or } l_{ij} = \infty \\ i, & l_{ij} = w_{ij} \\ \min(l_{ik} + w_{kj}), & \text{otherwise, where } k \text{ is the set of distinct predecessors of } j \end{cases}$

Runtime: To get the direct shortest paths (circled above), requires n^2 time to iterate through the L matrix. For determining the rest of the predecessors, you're looking at up to $n-2$ entries in the matrix (exclude $i=j$ and self). That adds the third dimension making it $\Theta(n^3)$

7) In the k for loop for Extend-Shortest-Path, the winning k index will be the predecessor of l_{ij}.

```
Extend-Shortest-Path(L, Π, W)
1  n = L.rows; let L' = (l'ij) nxn matrix; let Π' new nxn matrix
2  for i = 1 to n
3      for j = 1 to n
4          l'ij = ∞ ; π'ij = πij
5          for k = 1 to n
```

```
6       l'ij = min(l'ij, lik + wkj)
7       if l'ij = lik + wkj then
8           πij = πkj
9   return L', Π'
```

```
Slow-all-pairs-shortest-paths(W)
1.  n = W.rows ; L(1) = W ; Π(1) new nxn matrix
2   for p = 1 to n
3       for q = 1 to n
4           if L(1)pq = ∞ then
5               Π(1)pq = NIL
6           else
7               Πpq = p
8   for m = 2 to n-1
9       let L(m) be a new nxn matrix
10      let Π(m) new nxn matrix
11      L(m), Π(m) = Extend-Shortest-Paths(L(m-1), Π(m-1), W)
12  return L(n-1), Π(n-1)
```

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

```
1   n = W.rows
2   L = W
3   L' = W
4   m = 1
5   while m < n-1
6       L = L'
7       L' = Extended-Shortest-Paths(L, L)
8       m = 2m
9   return L'
```

Since previous matrices are not used, we can overwrite them, only keeping the last one to get the next matrix.

25.2

4) At the beginning of each iteration of the k for loop, the D matrix is in the state it needs to be to compute the next step. As line 6 relaxes the paths the two inner loops are always moving forward in the matrix, so modifications to it do not affect further computations. In this way, the D matrix from the previous iteration of k can be recycled.

To use the example from 25.4:

$D =$ (already computed)

$$\begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & 5 & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \rightarrow \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & \infty \\ 2 & 5 & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

(computed)

$K=2$, $i=3$, $j=4$

(not yet computed) (not computed)

$$d_{34} = \min(d_{34}, d_{32} + d_{24}) = \underline{5}$$

(d_{34}: "∞", d_{32}: "4", d_{24}: "1")