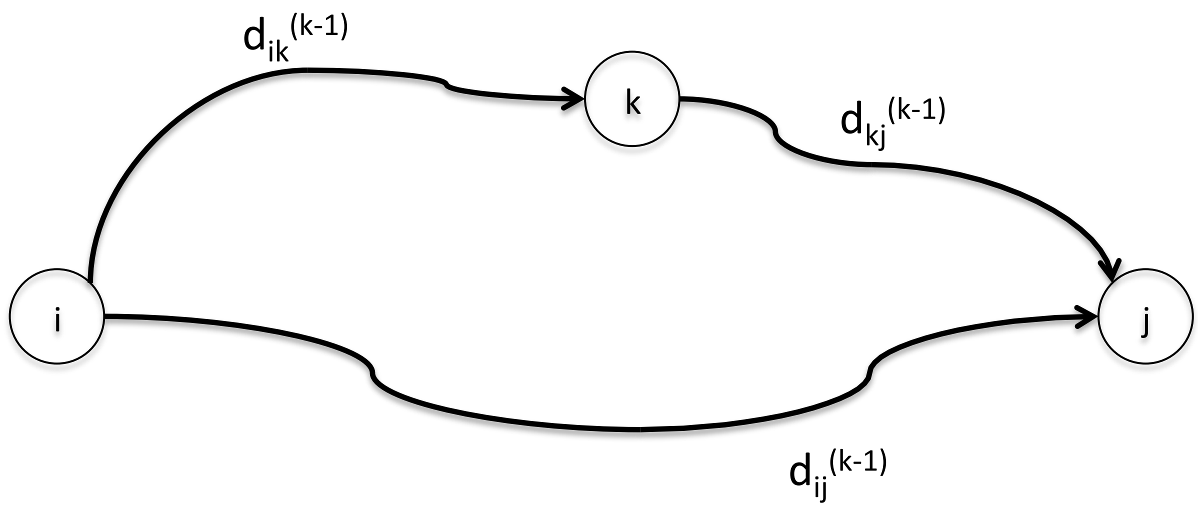
# Floyd-Warshall

CIS 315

# new restriction

- consider for each i and j, the length of the shortest path from i to j, with ….
- … all intermediate vertices in {1,2,…,k}
- relax this restriction, let k=0,1,2,…,n
- definition: $d_{ij}^{(k)}$ is the minimum length of a path from i to j with that restriction
- the n×n matrix $D^{(k)}$ has entries $d_{ij}^{(k)}$
- compute $D^{(0)}$, $D^{(1)}$, $D^{(2)}$, …, $D^{(n)}$

# does vertex k help?

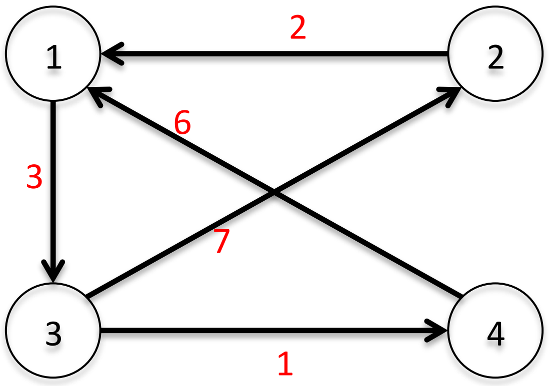



all intermediate vertices on a path numbered 1,2,…,k-1

update step: $d_{ij}^{(k)} = \mathrm{MIN}[d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}]$

# example

$$D^{(0)} = W = \begin{pmatrix} 0 & \infty & 3 & \infty \\ 2 & 0 & \infty & \infty \\ \infty & 7 & 0 & 1 \\ 6 & \infty & \infty & 0 \end{pmatrix}$$



# vertex 1 helps

$$D^{(1)} = \begin{pmatrix} 0 & \infty & 3 & \infty \\ 2 & 0 & 5 & \infty \\ \infty & 7 & 0 & 1 \\ 6 & \infty & 9 & 0 \end{pmatrix}$$

updated

# the main algorithm

```
input: n×n weight matrix W

D(0) = W

for k = 1 to n
   for i = 1 to n
      for j = 1 to n
          d_ij(k) = MIN[d_ij(k-1), d_ik(k-1) + d_kj(k-1) ]
```

$$d_{ij}^{(k)} = \text{MIN}[d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}]$$

remember: $d_{ij}^{(k)}$ is our shorthand for $D^{(k)}[i,j]$

both time and space are $O(n^3)$,
space can be reduced to $O(n^2)$

# remaining steps on example

$$D^{(1)} = \begin{pmatrix} 0 & \infty & 3 & \infty \\ 2 & 0 & 5 & \infty \\ \infty & 7 & 0 & 1 \\ 6 & \infty & 9 & 0 \end{pmatrix}$$

$$D^{(2)} = \begin{pmatrix} 0 & \infty & 3 & \infty \\ 2 & 0 & 5 & \infty \\ 9 & 7 & 0 & 1 \\ 6 & \infty & 9 & 0 \end{pmatrix}$$

$$D^{(2)} = \begin{pmatrix} 0 & \infty & 3 & \infty \\ 2 & 0 & 5 & \infty \\ 9 & 7 & 0 & 1 \\ 6 & \infty & 9 & 0 \end{pmatrix}$$

$$D^{(3)} = \begin{pmatrix} 0 & 10 & 3 & 4 \\ 2 & 0 & 5 & 6 \\ 9 & 7 & 0 & 1 \\ 6 & 16 & 9 & 0 \end{pmatrix}$$

$$D^{(3)} = \begin{pmatrix} 0 & 10 & 3 & 4 \\ 2 & 0 & 5 & 6 \\ 9 & 7 & 0 & 1 \\ 6 & 16 & 9 & 0 \end{pmatrix}$$

$$D^{(4)} = \begin{pmatrix} 0 & 10 & 3 & 4 \\ 2 & 0 & 5 & 6 \\ 7 & 7 & 0 & 1 \\ 6 & 16 & 9 & 0 \end{pmatrix}$$

# how to reconstruct the short path [IGNORE]

- text constructs a series of matrices $\prod^{(k)}$
- entries $\prod^{(k)}[i,j]=\pi_{ij}^{(k)}$
- defined to be the predecessor to node j on the shortest path from i to j with intermediate vertices in {1,2,…,k}
- it's either $\pi_{ij}^{(k-1)}$ or $\pi_{kj}^{(k-1)}$, depending on which element is the min
- space will be $\theta(n^3)$

# transitive closure "the same"

- subproblem: $t_{ij}^{(k)}$ is true iff there is a path from i to j with intermediate vertices in {1,2,…,k}
- base case: $t_{ij}^{(0)}$ is true iff [ i=j OR (i,j) $\in$ E ]
- recurrence: $t_{ij}^{(k)} = t_{ij}^{(k-1)} \vee [t_{ik}^{(k-1)} \wedge t_{kj}^{(k-1)}]$
- take the shortest path algorithm and modify initialization and inside the loop

# Floyd-Warshall (trans closure)

```
input: n×n adjacency matrix A

for i = 1 to n
   for j = 1 to n
     if i=j OR A[i,j]=1
        then t_ij^(0) = true
        else t_ij^(0) = false


for k = 1 to n
   for i = 1 to n
      for j = 1 to n
          t_ij^(k) = t_ij^(k-1)∨[t_ik^(k-1)∧ t_kj^(k-1)]
```

time and space same as with shortest path version

# dynamic programming philosophy

1) define a subproblem
2) define a recurrence showing how to derive the solution to the subproblem from smaller subproblems
3) fill out a table, in a bottom-up manner, storing optimum values for the subproblems
4) optionally, use values saved in the previous step to construct solution instance (usually top-down)