chang@umbc.edu

All Pairs Shortest Paths

- Floyd-Warshall (Dynamic Programming)

All pairs shortest paths

output: a shortest path tree for each source vertex

No negative weight edges

Run Dijkstra's Algorithm V times

array: $V \cdot \Theta(V^2) = \Theta(V^3)$

heap: $V \cdot \Theta(E \log V) = \Theta(E \cdot V \log V)$

Contains negative weight edges, but no cycles w/ neg. cost

Run Bellman-Ford V times $V \cdot \Theta(V E) = \Theta(V^2 E)$

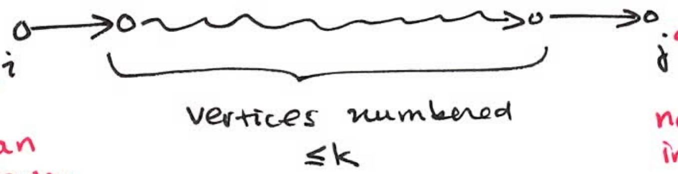
very, very slow

Floyd-Warshall $\Theta(V^3)$ a "fast" $\Theta(V^3)$ alg.

negative weight edges O.K.

Floyd-Warshall Algorithm as D.P.

$SP(i,j,k)$ = length of shortest path from i to j where intermediate vertices are numbered $\leq k$.



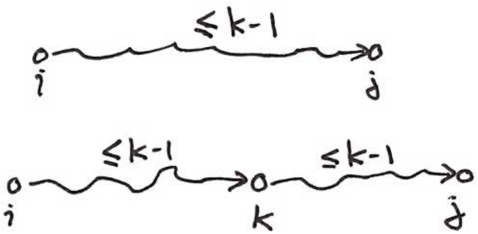
not an int. vertex

not an int. vertex

vertex numbering is totally arbitrary

$$SP(i,j,k) = \min\{SP(i,j,k-1),\ SP(i,k,k-1) + SP(k,j,k-1)\}$$

two choices: ① ②



use k ②
or
don't use k ①

no need to repeat k

Dynamic Programming Table:

3 dimensions, one each for i, j & k : $D^{(k)}[i,j]$

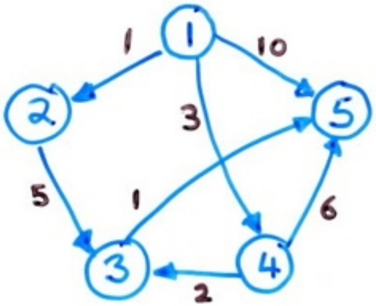
will reduce to 2D

Fill in starting with smaller values of k.

If we fill in the entire table, we have shortest paths for every pair of vertices.

"Recovering actual solution" not done in the usual way.

compute shortest path trees, not just lengths



$D^{(0)}$

	1	2	3	4	5
1	0	1	∞	3	10
2	∞	0	5	∞	∞
3	∞	∞	0	∞	1
4	∞	∞	2	0	6
5	∞	∞	∞	∞	0

$D^{(1)}$

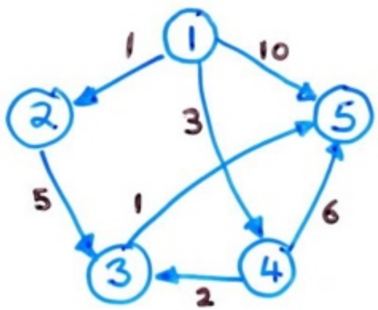
	1	2	3	4	5
1	0	1	∞	3	10
2	∞	0	5	∞	∞
3	∞	∞	0	∞	1
4	∞	∞	2	0	6
5	∞	∞	∞	∞	0

$P^{(0)}$

	1	2	3	4	5
1	nil	1	nil	1	1
2	nil	nil	2	nil	nil
3	nil	nil	nil	nil	3
4	nil	nil	4	nil	4
5	nil	nil	nil	nil	nil

$P^{(1)}$

	1	2	3	4	5
1	nil	1	nil	1	1
2	nil	nil	2	nil	nil
3	nil	nil	nil	nil	3
4	nil	nil	4	nil	4
5	nil	nil	nil	nil	nil



$D^{(1)}$

	1	2	3	4	5
1	0	1	∞	3	10
2	∞	0	5	∞	∞
3	∞	∞	0	∞	1
4	∞	∞	2	0	6
5	∞	∞	∞	∞	0

$D^{(2)}$

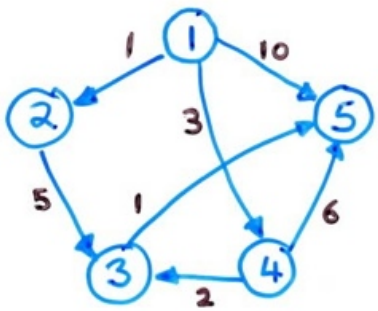
	1	2	3	4	5
1	0	1	6	3	10
2	∞	0	5	∞	∞
3	∞	∞	0	∞	1
4	∞	∞	2	0	6
5	∞	∞	∞	∞	0

$P^{(1)}$

	1	2	3	4	5
1	nil	1	nil	1	1
2	nil	nil	2	nil	nil
3	nil	nil	nil	nil	3
4	nil	nil	4	nil	4
5	nil	nil	nil	nil	nil

$P^{(2)}$

	1	2	3	4	5
1	nil	1	2	1	1
2	nil	nil	2	nil	nil
3	nil	nil	nil	nil	3
4	nil	nil	4	nil	4
5	nil	nil	nil	nil	nil



$D^{(2)}$

	1	2	3	4	5
1	0	1	6	3	10
2	∞	0	5	∞	∞
3	∞	∞	0	∞	1
4	∞	∞	2	0	6
5	∞	∞	∞	∞	0

$D^{(3)}$

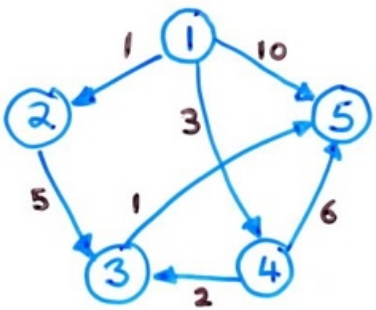
	1	2	3	4	5
1	0	1	6	3	7
2	∞	0	5	∞	6
3	∞	∞	0	∞	1
4	∞	∞	2	0	3
5	∞	∞	∞	∞	0

$P^{(2)}$

	1	2	3	4	5
1	nil	1	2	1	1
2	nil	nil	2	nil	nil
3	nil	nil	nil	nil	3
4	nil	nil	4	nil	4
5	nil	nil	nil	nil	nil

$P^{(3)}$

	1	2	3	4	5
1	nil	1	2	1	3
2	nil	nil	2	nil	3
3	nil	nil	nil	nil	3
4	nil	nil	4	nil	3
5	nil	nil	nil	nil	nil



$D^{(3)}$

	1	2	3	4	5
1	0	1	6	3	7
2	∞	0	5	∞	6
3	∞	∞	0	∞	1
4	∞	∞	2	0	3
5	∞	∞	∞	∞	0

$D^{(4)}$

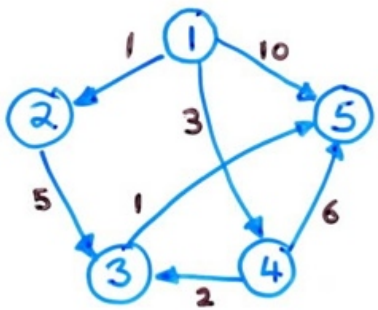
	1	2	3	4	5
1	0	1	5	3	6
2	∞	0	5	∞	6
3	∞	∞	0	∞	1
4	∞	∞	2	0	3
5	∞	∞	∞	∞	0

$P^{(3)}$

	1	2	3	4	5
1	nil	1	2	1	3
2	nil	nil	2	nil	3
3	nil	nil	nil	nil	3
4	nil	nil	4	nil	3
5	nil	nil	nil	nil	nil

$P^{(4)}$

	1	2	3	4	5
1	nil	1	4	1	3
2	nil	nil	2	nil	3
3	nil	nil	nil	nil	3
4	nil	nil	4	nil	3
5	nil	nil	nil	nil	nil



$D^{(4)}$

	1	2	3	4	5
1	0	1	5	3	6
2	∞	0	5	∞	6
3	∞	∞	0	∞	1
4	∞	∞	2	0	3
5	∞	∞	∞	∞	0

$D^{(5)}$

	1	2	3	4	5
1	0	1	5	3	6
2	∞	0	5	∞	6
3	∞	∞	0	∞	1
4	∞	∞	2	0	3
5	∞	∞	∞	∞	0

$P^{(4)}$

	1	2	3	4	5
1	nil	1	4	1	3
2	nil	nil	2	nil	3
3	nil	nil	nil	nil	3
4	nil	nil	4	nil	3
5	nil	nil	nil	nil	nil

$P^{(5)}$

	1	2	3	4	5
1	nil	1	4	1	3
2	nil	nil	2	nil	3
3	nil	nil	nil	nil	3
4	nil	nil	4	nil	3
5	nil	nil	nil	nil	nil

Floyd Warshall

/* Use Adjacency Matrix */

$D^{(0)}$ = adjacency matrix for graph G

for k=1 to n ← n = # of vertices

    for i=1 to n

        for j=1 to n

            if $D^{(k-1)}[i,j] > D^{(k-1)}[i,k] + D^{(k-1)}[k,j]$

                $D^{(k)}[i,j] = D^{(k-1)}[i,k] + D^{(k-1)}[k,j]$

                $P^{(k)}[i,j] = P^{(k-1)}[k,j]$

            else

                $D^{(k)}[i,j] = D^{(k-1)}[i,j]$

                $P^{(k)}[i,j] = P^{(k-1)}[i,j]$

If $j == k$, we check

$$D^{(k-1)}[i,j] > D^{(k-1)}[i,k] + D^{(k-1)}[k,j]$$

$$\begin{array}{c}\| \\ D^{(k-1)}[i,k]\end{array} \underset{\uparrow \text{ is never true}}{>} \begin{array}{c}\| \\ D^{(k-1)}[i,k]\end{array} + \cancelto{0}{D^{(k-1)}[k,k]}$$

So, $D^{(k-1)}[i,k] == D^{(k)}[i,k]$

Similarly, $D^{(k-1)}[k,j] == D^{(k)}[k,j]$

Furthermore, RHS of assignments are $D^{(k-1)}[i,k]$ & $D^{(k-1)}[k,j]$

$$D^{(k)}[i,j] = D^{(k-1)}[i,k] + D^{(k-1)}[k,j]$$

if we remove (k) & $(k-1)$, the result is the same!

Floyd-Warshall

D = adjacency matrix for graph G

```
for k=1 to n          = # of vertices
    for i=1 to n
        for j=1 to n
            if D[i,j] > D[i,k] + D[k,j]
                D[i,j] = D[i,k] + D[k,j]
                P[i,j] = P[k,j]
```

$\Theta(n^3)$

row i P[i, _] is the predecessor array for the shortest path tree w/ source i