Let $d^{(k)}[i,j]$ be the weight of a shortest path from $i$ to $j$ whose intermediate vertices are drawn from the set $\{1, 2, 3, \ldots, k\}$

If $k=0$, then $d^{(0)}[i,j]=w(i,j)$

If $k>0$, then $d^{(k)}[i,j]$ is the minimum of $d^{(k-1)}[i,j]$ and $d^{(k-1)}[i,k]+d^{(k-1)}[k,j]$

i.e. $d^{(k)}[i,j]=\min(d^{(k-1)}[i,j],d^{(k-1)}[i,k]+d^{(k-1)}[k,j])$

The all-pairs shortest paths are the entries $d^{(n)}[i,j]$ in other words, $\delta(i,j)=d^{(n)}[i,j]$

Floyd-Warshall algorithm (W) Let $D^{(k)}=\{d^{(k)}[i,j]\}$

$D^{(0)}=W$ (i.e. for $i=1$ to $n$ for $j=1$ to $n$ $d^{(0)}[i,j]=w(i,j)$ )

for $k=1$ to $n$

for $i=1$ to $n$

for $j=1$ to $n$

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

return $D^{(n)}$

Running time: $O(V^3)$
since \( p \) is a simple path (no repeated vertices), \( p_1 \) and \( p_2 \) do not contain \( k \)

\( p_1 \) must be the shortest \( i \)-to-\( k \) path that uses only \( \{1, 2, 3, \ldots, k-1\} \)

and \( p_2 \) must be the shortest \( k \)-to-\( j \) path that uses only \( \{1, 2, 3, \ldots, k-1\} \)

since both \( p_1 \) and \( p_2 \) are known, we can compute \( p \) easily

To determine \( p \), take the shortest of

1) the shortest \( i \)-to-\( j \) path that uses only \( \{1, 2, 3, \ldots, k-1\} \)

2) the shortest \( i \)-to-\( k \) path \( p_1 \) that uses only \( \{1, 2, 3, \ldots, k-1\} \)

plus the shortest \( k \)-to-\( j \) path \( p_2 \) that uses only \( \{1, 2, 3, \ldots, k-1\} \)
Suppose we know the shortest path for every pair of vertices $s$ and $t$ such that all intermediate vertices on this path are numbered $1, 2, \ldots, k-1$

There are only two possibilities for $p$:

1) $k$ is not an intermediate vertex of $p$

then all intermediate vertices of $p$ come from $\{1, 2, 3, \ldots, k-1\}$
then $p$ is the same as the shortest $i$-to-$j$ path that uses only $\{1, 2, 3, \ldots, k-1\}$ as intermediate vertices, which is known

2) $k$ is an intermediate vertex of $p$

then we can break $p$ into $i$-to-$k$ path $p_1$ and $k$-to-$j$ path $p_2$
**TOPIC 20: All Pairs Shortest Paths (APSP)**

using Dijkstra: $O(V^3)$ or $O(VE \lg V)$ good only for non-negative weights

using Bellman-Ford: $O(V^2E)$

New algorithm: we assume no negative weight cycles

Size of output for APSP is $O(V^2)$; thus use adjacency-matrix representation

**Definition:** intermediate vertex of a simple path $p = \langle v_1, v_2, ..., v_s \rangle$ is any vertex of $p$ other than $v_1$ or $v_s$.

Let vertices be numbered $\{1, 2, 3, ..., n\}$. Consider a subset $\{1, 2, 3, ..., k\}$

For any pair of vertices $i$ and $j$, consider all paths from $i$ to $j$ whose intermediate vertices are drawn from $\{1, 2, 3, ..., k\}$ (i.e. cannot be numbered $> k$)

Let $p$ be a minimum-weight path from among these paths

Ex: $i=2$, $j=6$

for $k=4$, there is only 1 candidate path for $p$

path $2, 3, 1, 4, 6$ cost=$8+7+2+4=21$

for $k=5$, there are 2 candidate paths for $p$

path $2, 3, 1, 4, 6$ cost=$21$

path $2, 3, 5, 6$ cost=$8+1+2=11$ this is $p$