Experiment-7 Floyd’s Algorithm

Implement All-Pairs Shortest Paths Problem using Floyd‘s algorithm.

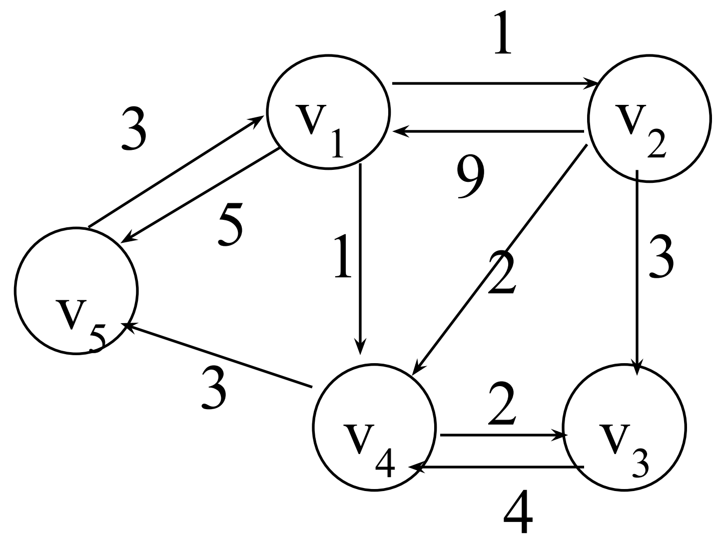
All pairs shortest path

- *The problem:* find the shortest path between every pair of vertices of a graph

- *A representation*: a weight matrix where
  W(i,j)=0 if i=j.
  W(i,j)=∞ if there is no edge between i and j.
  W(i,j)="weight of edge"

The weight matrix and the graph

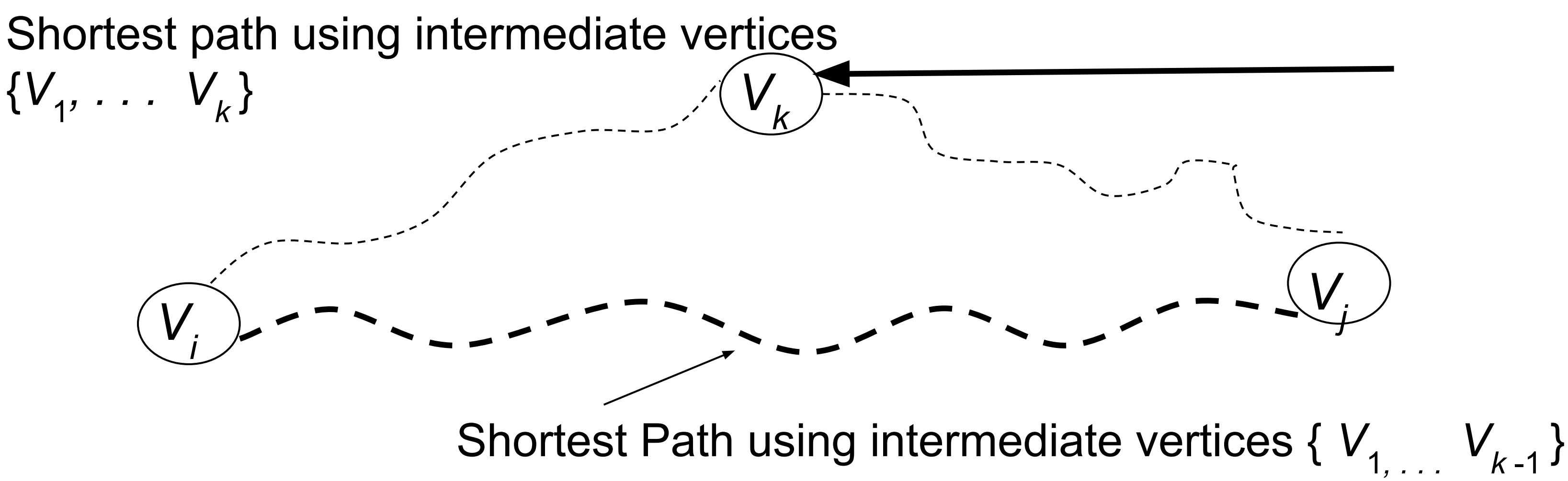
	1	2	3	4	5
1	0	1	∞	1	5
2	9	0	3	2	∞
3	∞	∞	0	4	∞
4	∞	∞	2	0	3
5	3	∞	∞	∞	0



The Recursive Definition

Case 1: A shortest path from v_i to v_j restricted to using only vertices from $\{v_1,v_2,\ldots,v_k\}$ as intermediate vertices does not use v_k. Then $D^{(k)}[i,j]= D^{(k-1)}[i,j]$.

Case 2: A shortest path from v_i to v_j restricted to using only vertices from $\{v_1,v_2,\ldots,v_k\}$ as intermediate vertices does use v_k. Then $D^{(k)}[i,j]= D^{(k-1)}[i,k]+ D^{(k-1)}[k,j]$.

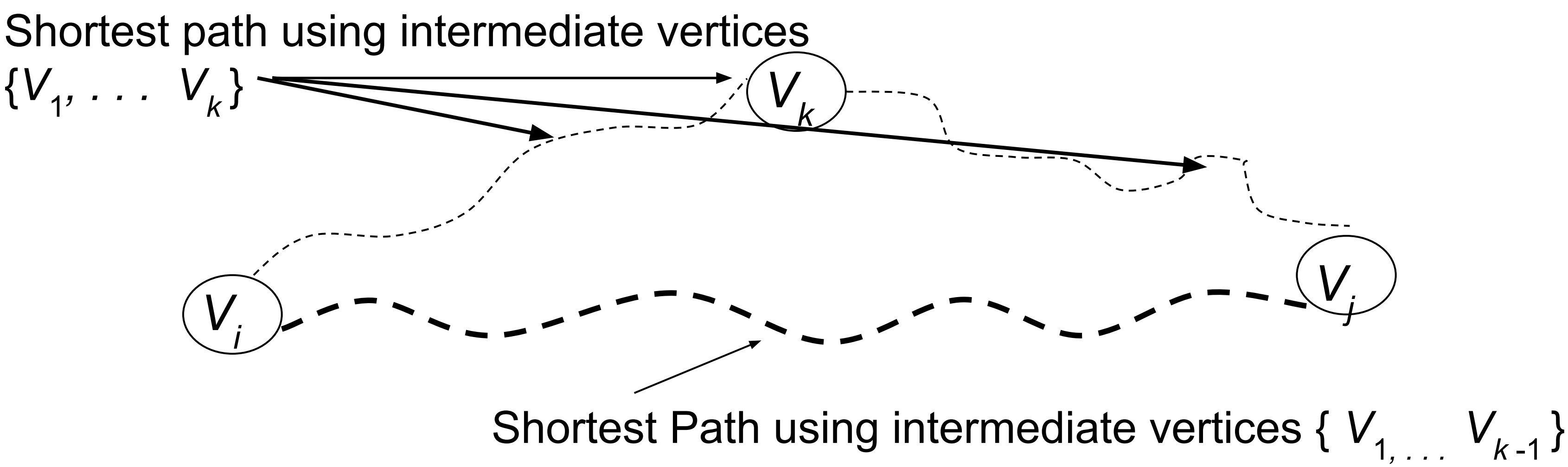


The recursive definition

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

  We conclude:

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



Floyd's Algorithm Using n+1 *D* matrices

Floyd//Computes shortest distance between all pairs of //nodes, and saves P to enable finding shortest paths

1. $D^0 \leftarrow W$ // initialize *D* array to *W* [ ]

2. $P \leftarrow 0$ // initialize P array to [0]

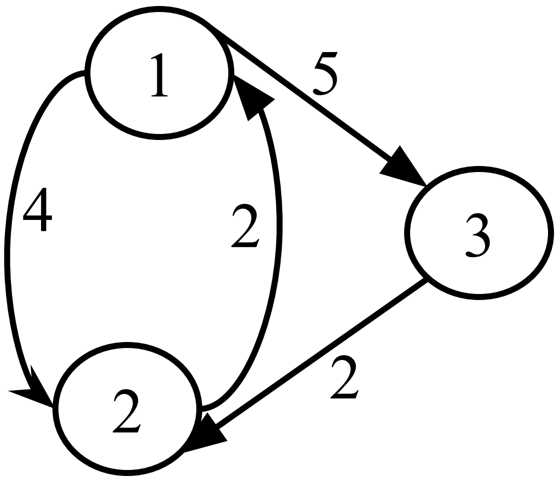
3. **for** $k \leftarrow 1$ **to** n

4. do for $i \leftarrow 1$ **to** n

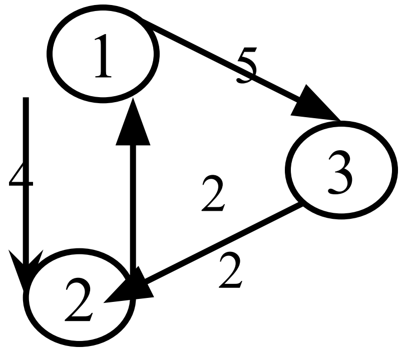
5. do for $j \leftarrow 1$ **to** n

6. $D^k[i, j] = \min(D^{k-1}[i,j]$ **or** $(D^{k-1}[i, k] + D^{k-1}[k, j]))$

Example



$$W = D^0 = \begin{array}{c|c|c|c|} & 1 & 2 & 3 \\ \hline 1 & 0 & 4 & 5 \\ \hline 2 & 2 & 0 & \infty \\ \hline 3 & \infty & 2 & 0 \\ \hline \end{array}$$



$$
D^0 = \begin{array}{c|ccc|}
 & 1 & 2 & 3 \\
\hline
1 & 0 & 4 & 5 \\
2 & 2 & 0 & \infty \\
3 & \infty & 2 & 0 \\
\hline
\end{array}
$$

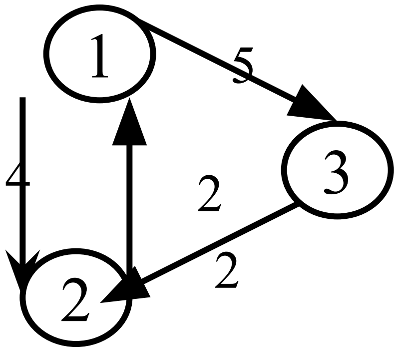
k = 1
Vertex 1 can be intermediate node

$$
D^1 = \begin{array}{c|ccc|}
 & 1 & 2 & 3 \\
\hline
1 & 0 & 4 & 5 \\
2 & 2 & 0 & 7 \\
3 & \infty & 2 & 0 \\
\hline
\end{array}
$$

$D^1[2,3] = \min(D^0[2,3], D^0[2,1]+D^0[1,3])$

$= \min (\propto, 7)$

$= 7$



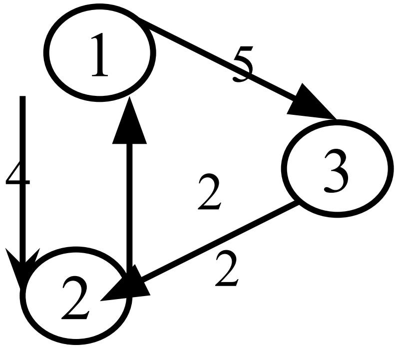
$$D^1 = \begin{array}{c|c|c|c|} & 1 & 2 & 3 \\ \hline 1 & 0 & 4 & 5 \\ \hline 2 & 2 & 0 & 7 \\ \hline 3 & \infty & 2 & 0 \\ \hline \end{array}$$

k = 2
Vertices 1, 2
can be
intermediate

$$D^2 = \begin{array}{c|c|c|c|} & 1 & 2 & 3 \\ \hline 1 & 0 & 4 & 5 \\ \hline 2 & 2 & 0 & 7 \\ \hline 3 & 4 & 2 & 0 \\ \hline \end{array}$$

$D^2[1,3] = \min(D^1[1,3], D^1[1,2]+D^1[2,3])$
$= \min (5, 4+ \propto)$
$= 5$

$D^2[3,1] = \min(D^1[3,1], D^1[3,2]+D^1[2,1])$
$= \min (\propto, 2+2)$
$= 4$



$$D^2 = \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 0 & 4 & 5 \\ 2 & 2 & 0 & 7 \\ 3 & 4 & 2 & 0 \end{array}$$

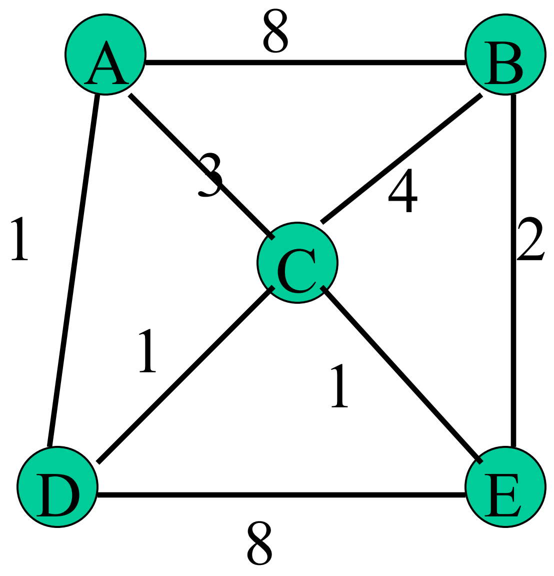
k = 3
Vertices 1, 2, 3
can be
intermediate

$$D^3 = \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 0 & 4 & 5 \\ 2 & 2 & 0 & 7 \\ 3 & 4 & 2 & 0 \end{array}$$

$D^3[1,2] = \min(D^2[1,2],\ D^2[1,3]+D^2[3,2])$
$= \min(4,\ 5+2)$
$= 4$

Floyd's Algorithm

- Back to the shortest-path problem.

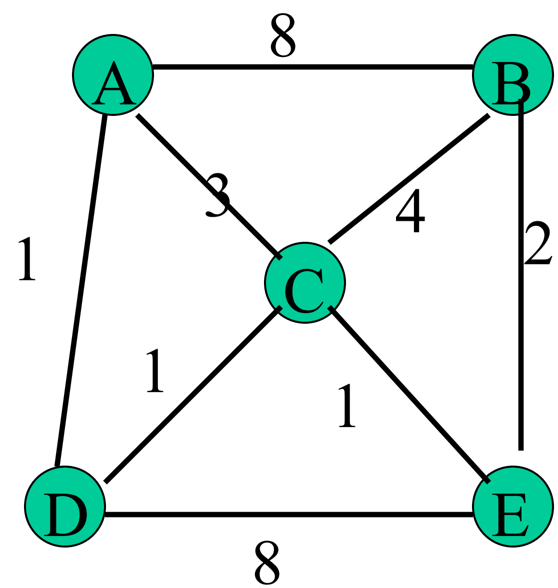


	A	B	C	D	E
A	0	8	3	1	∞
B	8	0	4	∞	2
C	3	4	0	1	1
D	1	∞	1	0	8
E	∞	2	1	8	0

Floyd’s Algorithm

	A	B	C	D	E
A	0	8	3	1	∞
B	8	0	4	∞	2
C	3	4	0	1	1
D	1	∞	1	0	8
E	∞	2	1	8	0

- First step, where can we go without using an intermediate vertex
- M^k, where k the set of intermediate vertices



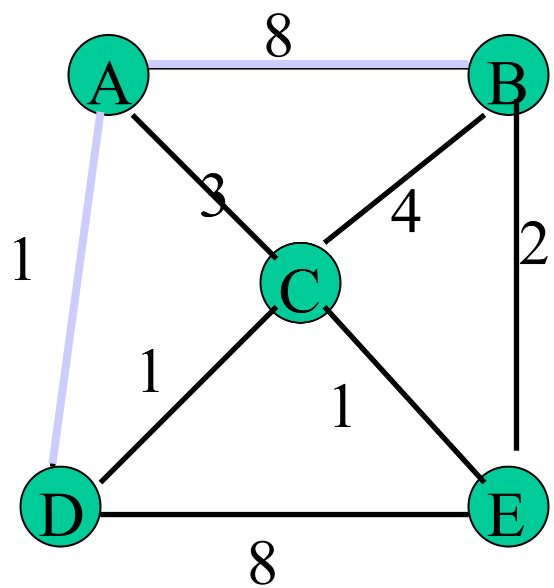
$M^{\{\}}$

	A	B	C	D	E
A	0	8	3	1	∞
B	8	0	4	∞	2
C	3	4	0	1	1
D	1	∞	1	0	8
E	∞	2	1	8	0

	A	B	C	D	E
A	0	8	3	1	∞
B	8	0	4	∞	2
C	3	4	0	1	1
D	1	∞	1	0	8
E	∞	2	1	8	0

Floyd's Algorithm

- Second step, where can we go if we use A as an intermediate vertices



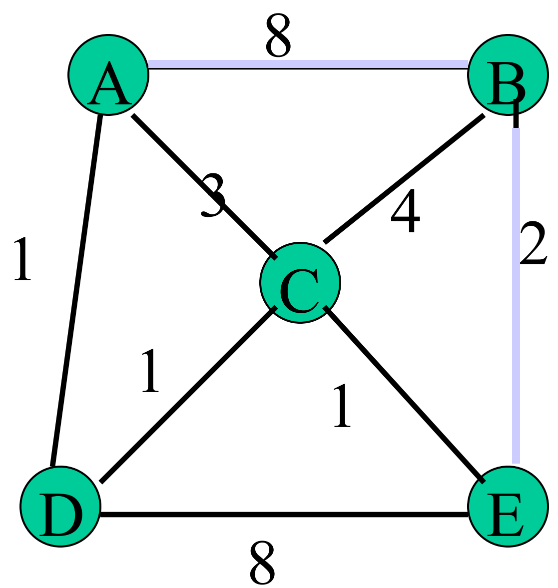
$M^{\{A\}}$

	A	B	C	D	E
A	0	8	3	1	∞
B	8	0	4	9	2
C	3	4	0	1	1
D	1	9	1	0	8
E	∞	2	1	8	0

Floyd's Algorithm

	A	B	C	D	E
A	0	8	3	1	∞
B	8	0	4	∞	2
C	3	4	0	1	1
D	1	∞	1	0	8
E	∞	2	1	8	0

- Third step, where can we go if we use A and B as intermediate vertices



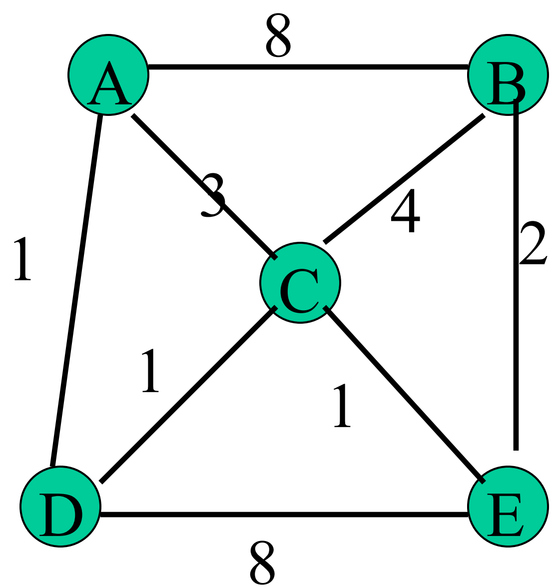
$M^{\{A,B\}}$

	A	B	C	D	E
A	0	8	3	1	10
B	8	0	4	9	2
C	3	4	0	1	1
D	1	9	1	0	8
E	10	2	1	8	0

Floyd's Algorithm

	A	B	C	D	E
A	0	8	3	1	∞
B	8	0	4	∞	2
C	3	4	0	1	1
D	1	∞	1	0	8
E	∞	2	1	8	0

- Fourth step, where can we go if we use A, B, and C as intermediate vertices



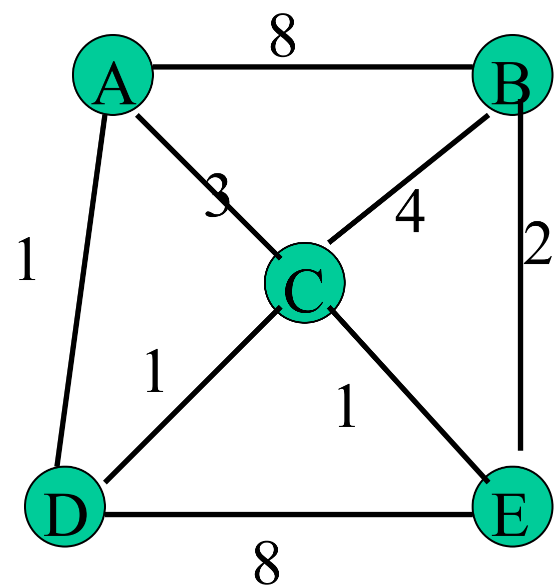
$$M^{\{A,B,C\}}$$

	A	B	C	D	E
A	0	8	3	1	10
B	8	0	4	9	2
C	3	4	0	1	1
D	1	9	1	0	8
E	10	2	1	8	0

Floyd's Algorithm

	A	B	C	D	E
A	0	8	3	1	∞
B	8	0	4	∞	2
C	3	4	0	1	1
D	1	∞	1	0	8
E	∞	2	1	8	0

- Fourth step, where can we go if we use A, B, and C as intermediate vertices



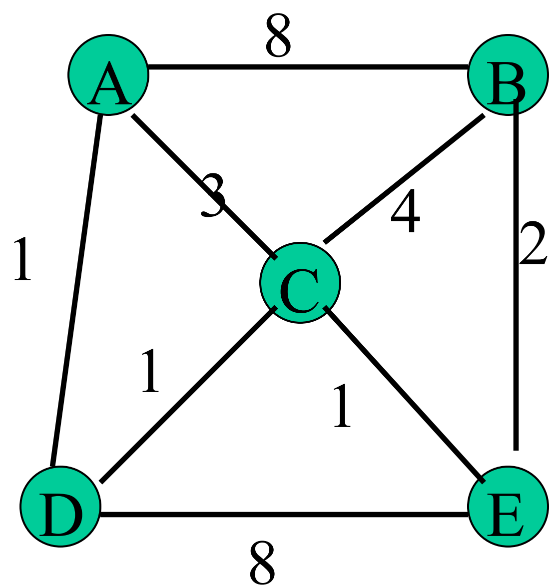
$M^{\{A,B,C\}}$

	A	B	C	D	E
A	0	7	3	1	4
B	7	0	4	5	2
C	3	4	0	1	1
D	1	5	1	0	2
E	4	2	1	2	0

Floyd's Algorithm

	A	B	C	D	E
A	0	8	3	1	∞
B	8	0	4	∞	2
C	3	4	0	1	1
D	1	∞	1	0	8
E	∞	2	1	8	0

- Fifth step, where can we go if we use A, B, C, and D as intermediate vertices



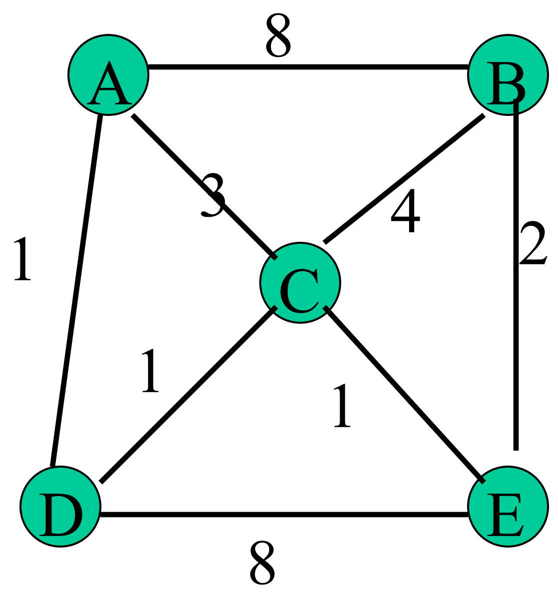
$M^{\{A,B,C,D\}}$

	A	B	C	D	E
A	0	7	2	1	4
B	7	0	4	5	2
C	2	4	0	1	1
D	1	5	1	0	2
E	4	2	1	2	0

Floyd's Algorithm

- Fifth step, where can we go if we use A, B, C, and D as intermediate vertices
- **OK, here is the answer.**

	A	B	C	D	E
A	0	8	3	1	∞
B	8	0	4	∞	2
C	3	4	0	1	1
D	1	∞	1	0	8
E	∞	2	1	8	0



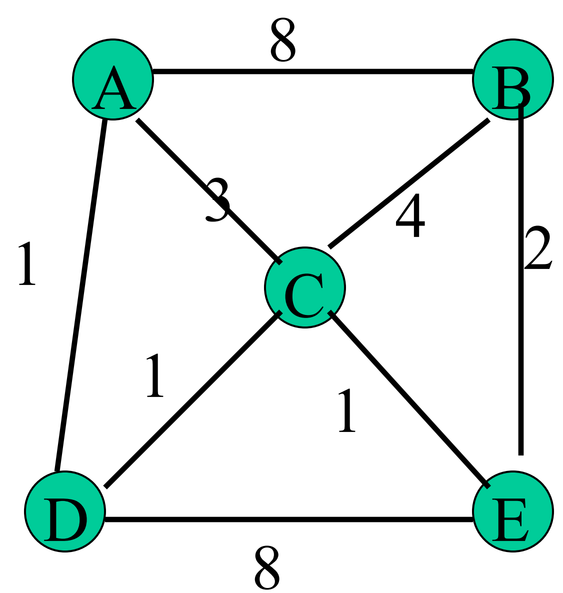
$M^{\{A,B,C,D\}}$

	A	B	C	D	E
A	0	6	2	1	3
B	6	0	4	5	2
C	2	4	0	1	1
D	1	5	1	0	2
E	3	2	1	2	0

Floyd's Algorithm

	A	B	C	D	E
A	0	8	3	1	∞
B	8	0	4	∞	2
C	3	4	0	1	1
D	1	∞	1	0	8
E	∞	2	1	8	0

- Final step, where can we go if we use all the vertices as intermediates.



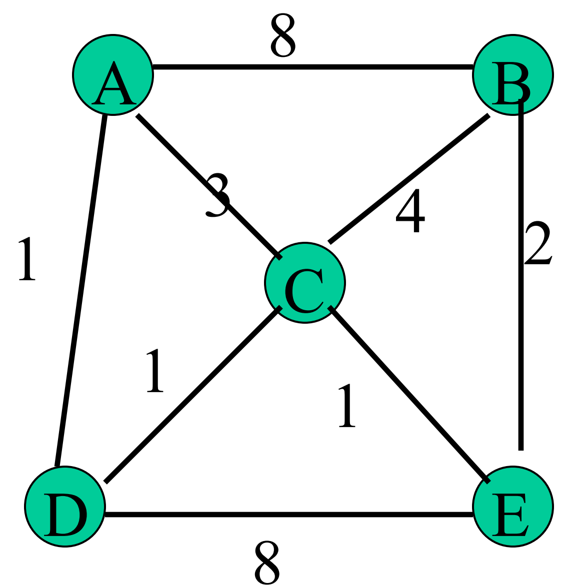
$M^{\{A,B,C,D,E\}}$

	A	B	C	D	E
A	0	6	2	1	3
B	6	0	4	5	2
C	2	4	0	1	1
D	1	5	1	0	2
E	3	2	1	2	0

Floyd's Algorithm

	A	B	C	D	E
A	0	8	3	1	∞
B	8	0	4	∞	2
C	3	4	0	1	1
D	1	∞	1	0	8
E	∞	2	1	8	0

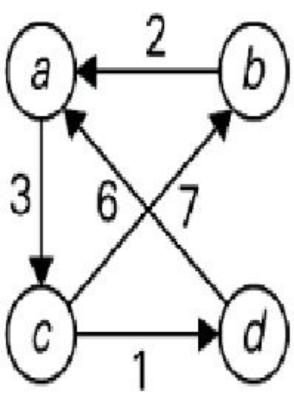
- Final step, where can we go if we use all the vertices as intermediates.



$$M^{\{A,B,C,D,E\}}$$

	A	B	C	D	E
A	0	5	2	1	3
B	5	0	3	5	2
C	2	3	0	1	1
D	1	5	1	0	2
E	3	2	1	2	0

Floyd's Algorithm: All pairs shortest paths



(a)

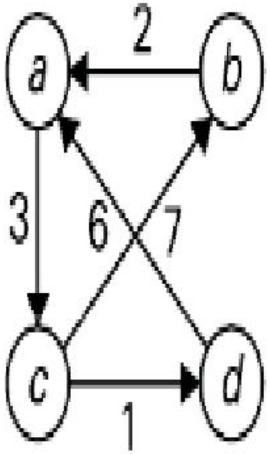
$$W = \begin{array}{cc} & \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{bmatrix} 0 & \infty & 3 & \infty \\ 2 & 0 & \infty & \infty \\ \infty & 7 & 0 & 1 \\ 6 & \infty & \infty & 0 \end{bmatrix} \end{array}$$

(b)

$$D = \begin{array}{cc} & \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{bmatrix} 0 & 10 & 3 & 4 \\ 2 & 0 & 5 & 6 \\ 7 & 7 & 0 & 1 \\ 6 & 16 & 9 & 0 \end{bmatrix} \end{array}$$

(c)

FIGURE 8.5 (a) Digraph. (b) Its weight matrix. (c) Its distance matrix.



$$D^{(0)} = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & \infty & 3 & \infty \\ 2 & 0 & \infty & \infty \\ \infty & 7 & 0 & 1 \\ 6 & \infty & \infty & 0 \end{bmatrix} \end{array}$$

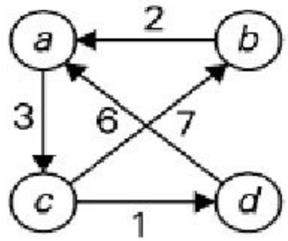
Lengths of the shortest paths with no intermediate vertices ($D^{(0)}$ is simply the weight matrix).

$$D^{(1)} = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & \infty & 3 & \infty \\ 2 & 0 & \mathbf{5} & \infty \\ \infty & 7 & 0 & 1 \\ 6 & \infty & \mathbf{9} & 0 \end{bmatrix} \end{array}$$

Lengths of the shortest paths with intermediate vertices numbered not higher than 1, i.e. just *a* (note two new shortest paths from *b* to *c* and from *d* to *c*).

$$D^{(2)} = \begin{array}{c c} & \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{bmatrix} 0 & \infty & 3 & \infty \\ 2 & 0 & 5 & \infty \\ \mathbf{9} & 7 & 0 & 1 \\ 6 & \infty & 9 & 0 \end{bmatrix} \end{array}$$

Lengths of the shortest paths
with intermediate vertices numbered
not higher than 2, i.e. a and *b*
(note a new shortest path from *c* to *a*).



$$D^{(0)} = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & \infty & 3 & \infty \\ 2 & 0 & \infty & \infty \\ \infty & 7 & 0 & 1 \\ 6 & \infty & \infty & 0 \end{bmatrix} \end{array}$$

Lengths of the shortest paths with no intermediate vertices ($D^{(0)}$ is simply the weight matrix).

$$D^{(1)} = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & \infty & 3 & \infty \\ 2 & 0 & \mathbf{5} & \infty \\ \infty & 7 & 0 & 1 \\ 6 & \infty & \mathbf{9} & 0 \end{bmatrix} \end{array}$$

Lengths of the shortest paths with intermediate vertices numbered not higher than 1, i.e. just *a* (note two new shortest paths from *b* to *c* and from *d* to *c*).

$$D^{(2)} = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & \infty & 3 & \infty \\ 2 & 0 & 5 & \infty \\ \mathbf{9} & 7 & 0 & 1 \\ 6 & \infty & 9 & 0 \end{bmatrix} \end{array}$$

Lengths of the shortest paths with intermediate vertices numbered not higher than 2, i.e. a and *b* (note a new shortest path from *c* to *a*).

$$D^{(3)} = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & \mathbf{10} & 3 & \mathbf{4} \\ 2 & 0 & 5 & \mathbf{6} \\ 9 & 7 & 0 & 1 \\ 6 & \mathbf{16} & 9 & 0 \end{bmatrix} \end{array}$$

Lengths of the shortest paths with intermediate vertices numbered not higher than 3, i.e. a, *b,* and *c* (note four new shortest paths from *a* to *b,* from a to *d,* from *b* to *d,* and from *d* to *b*).

$$D^{(4)} = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & 10 & 3 & 4 \\ 2 & 0 & 5 & 6 \\ \mathbf{7} & 7 & 0 & 1 \\ 6 & 16 & 9 & 0 \end{bmatrix} \end{array}$$

Lengths of the shortest paths with intermediate vertices numbered not higher than 4, i.e. a, *b, c,* and *d* (note a new shortest path from *c* to *a*).