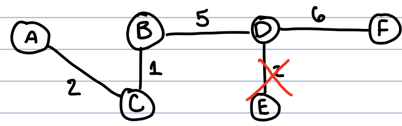
$\{V, W\}$

Visited	B	C	D	E	F	
{A}	A, 4	A, 2	A, ∞	A, ∞	A, ∞	
{A, C}	C, 3	x	C, 10	C, 12	C, ∞	
{A, C, B}	x	x	B, 8	C, 12	B, ∞	
{A, C, B, D}	x	x	x	D, 10	D, 14	
{A, C, B, D, E}	x	x	x	x	D, 14	

Solution:



$$Cost(A, V_i) = \min(\{A, V_i\}, \{A, \{V_k\}, V_i\})$$

$\{V, W\}$

Visited	B	C	D	E	F	
{A}	A, 4	A, 2	A, ∞	A, ∞	A, ∞	

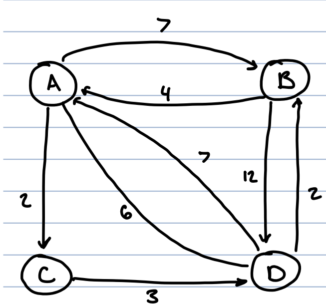
A	B	C	D	E	F
0	1	2	3	4	5
∅	(0, 4)	(0, 2)	(0, ∞)	(0, ∞)	(0, ∞)

Visited

0	1	2	3	4	5
1	0	1	0	0	0

P	A	B	C	D	E	F
A	0					
B		0				
C			0			
D				0		
E					0	
F						0

P	A	B	C	D
A	∞	0	0	0
B	0	∞	0	0
C	0	0	∞	0
D	0	0	0	∞



D''	A	B	C	D
A	∞	7	2	6
B	4	∞	∞	12
C	∞	∞	∞	3
D	7	1	∞	∞

P	A	B	C	D
A	∞	D	0	C
B	0	∞	A	C
C	D	D	∞	0
D	B	0	B	∞

Floyd-Marshall algorithm

use D^{-1}

Ⓐ

D^0	A	B	C	D
A	∞	7	2	6
B	4	∞	6	10
C	∞	∞	∞	3
D	7	1	9	∞

Cost(B,C) = min{ B→C, B→A→C } = 6
∞ , 4+2=6

Cost(B,D) = min{ B→D, B→A→D } = 10
12 , 4+6=10

Cost(C,B) = min{ C→B, C→A→B } = x
∞ , ∞

Cost(C,D) = min{ C→D, C→A→D }
3, ∞

Cost(D,B) = min{ D→B, D→A→B } =
1 ,

use D^0

Ⓑ

D^1	A	B	C	D
A	∞	7	2	6
B	4	∞	6	10
C	∞	∞	∞	3
D	5	1	7	∞

min{ A→C , A→B→C }
2 , 7...

min{ A→D, A→B→D }
6 , 7...

min{ C→A, C→B→A }
∞ ∞

min{ C→D, C→B→D }
3 , ∞..

min{ D→A, D→B→A } = 5
7, 1+4=5

min{ D→C, D→B→C } = 7
9 , 1+6=7

Ⓒ use D^1

D^2	A	B	C	D
A	∞	7	2	5
B	4	∞	6	9
C	∞	∞	∞	3
D	5	1	7	∞

min{ A→B, A→C→B }
7, 2+∞

min{ A→D, A→C→D } = 5
6 , 2+3=5

min{ B→A, B→C→A }
4 , 6...

min{ B→D, B→C→D } = 9
10 , 6+3=9

min{ A→B, A→C→B }

Ⓓ → j

i ↓

	3 D	0 A	1 B	2 C	3 D
0	A	∞	6	2	5
1	B	4	∞	6	9
2	C	8	4	∞	3
3	D	5	1	7	∞

* General formula : $M_k[i][j] = \min\{ M_{k-1}[i][j], M_{k-1}[i][k] + M_{k-1}[k][j] \}$

$M_3[0][1] = \min\{ M_2[0][1], M_2[0][3] + M_2[3][1] \} = 6$
7 , 5 + 1 = 6

A, C
2 , 2

B, A   4, ∞
B, C   6, 9
C, A   ∞, 3+5=8
C, B   ∞, 3+1=4

P	A	B	C	D
A	∞	D	D	C
B	O	∞	A	C
C	D	D	∞	O
D	B	O	B	∞

$Min(A,D) = \underbrace{A \rightarrow D}_{(1)} \rightarrow B$

$A \rightarrow C \rightarrow \underbrace{D \rightarrow B}_{(2)}$

0

$2+3+1=6$

$Min(D,A) = \underbrace{D \rightarrow B}_{(1)} \rightarrow A$

0

$= D \rightarrow \underbrace{B \rightarrow A}_{0}$

$= D \rightarrow B \rightarrow A$

Psudo Code:

```
function MinPath(D , P):
    for (k=0; k<n; k++)
        Create matrix M $ initialize it to -1
        for (i=0; i<n; i++)
            for(j=0; j<n; j++)
                if i=k $ j=k Then M[i][j] = D[i][j]
                else
                    M[i][j] = min( D[i][j], D[i][k] + D[k][j]
                    if M[i][j] < D[i][j] then P[i][j] = K
    D=M
  return P;
```

Bellman-Ford