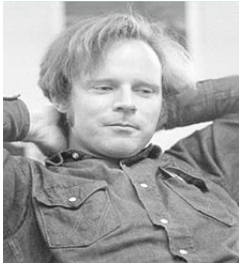
Grafos

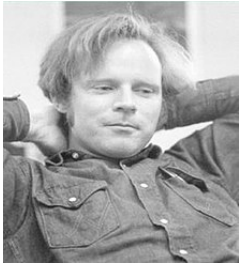
Algoritmo de Floyd-Warshall

Prof. Edson Alves

Faculdade UnB Gama

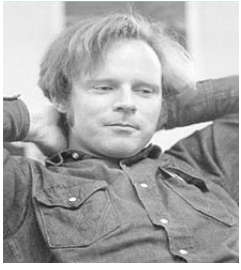


Robert W. Floyd
(1962)



Robert W. Floyd
(1962)

Stephen Warshall
(1962)



Robert W. Floyd
(1962)

Stephen Warshall
(1962)

Bernard Roy
(1959)

Características do algoritmo de Bellman-Ford

Características do algoritmo de Bellman-Ford

- ⋆ **Computa o caminho mínimo entre todos os pares de vértices de** $G(V, E)$

Características do algoritmo de Bellman-Ford

- ⋆ **Computa o caminho mínimo entre todos os pares de vértices de $G(V, E)$**
- ⋆ **É capaz de processar arestas negativas**

Características do algoritmo de Bellman-Ford

- ⋆ **Computa o caminho mínimo entre todos os pares de vértices de $G(V, E)$**
- ⋆ **É capaz de processar arestas negativas**
- ⋆ **Não processa, mas identifica ciclos negativos**

Características do algoritmo de Bellman-Ford

- ⋆ **Computa o caminho mínimo entre todos os pares de vértices de $G(V, E)$**
- ⋆ **É capaz de processar arestas negativas**
- ⋆ **Não processa, mas identifica ciclos negativos**
- ⋆ **As distâncias são reduzidas por meio do uso de vértices intermediários**

Características do algoritmo de Bellman-Ford

- ⋆ **Computa o caminho mínimo entre todos os pares de vértices de** $G(V, E)$
- ⋆ **É capaz de processar arestas negativas**
- ⋆ **Não processa, mas identifica ciclos negativos**
- ⋆ **As distâncias são reduzidas por meio do uso de vértices intermediários**
- ⋆ **Complexidade:** $O(V^3)$

Pseudocódigo

Pseudocódigo

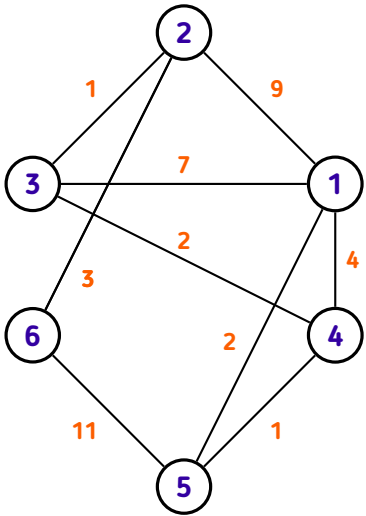
Entrada: uma grafo $G(V, E)$
Saída: uma matriz d tal que $d[u][v]$ é a distância mínima em G entre u e v

1. **Faça:**
   (a) $d[u][u] = 0$, **para todos** $u \in V$
   (b) $d[u][v] = w$, **se** $(u, v, w) \in E$
   (c) $d[u][v] = \infty$, **caso contrário**

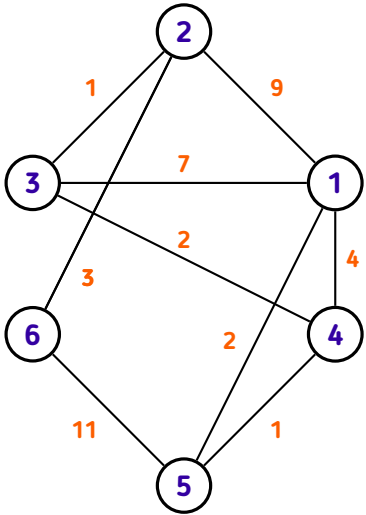
2. **Para cada vértice** k **e todos os pares** $(u, v) \in V^2$**, faça**

$$d[u][v] = \min(d[u][v], d[u][k] + d[k][v])$$

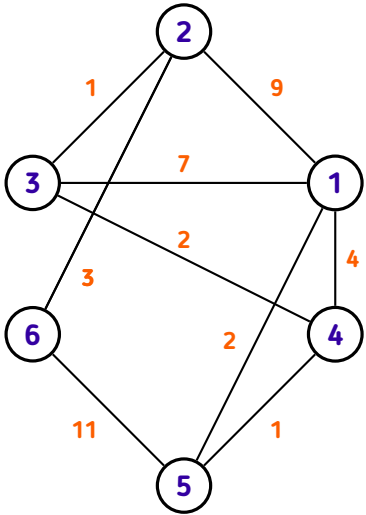
3. **Retorne** d



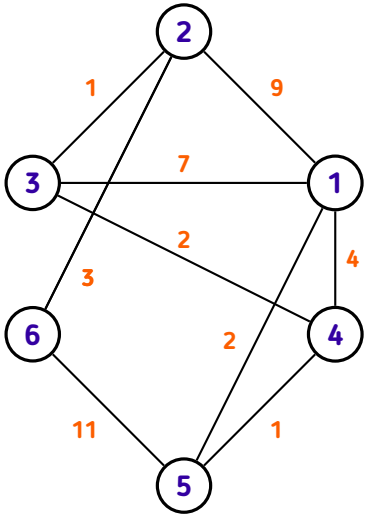
	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						



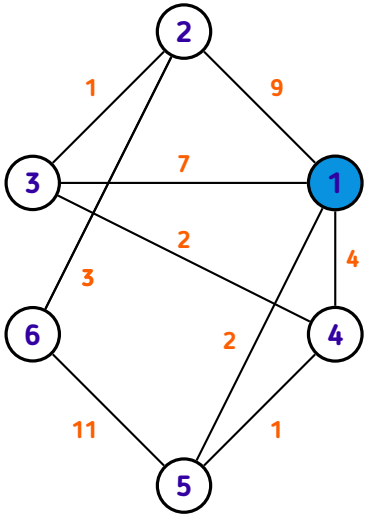
	1	2	3	4	5	6
1	0					
2		0				
3			0			
4				0		
5					0	
6						0



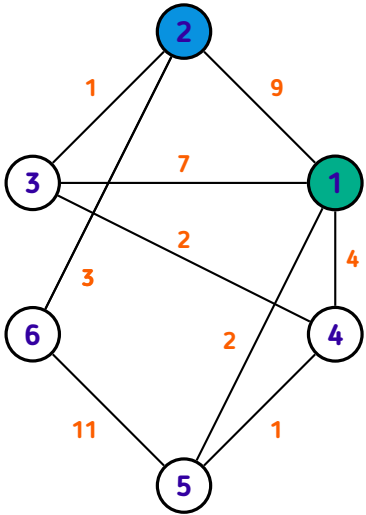
	1	2	3	4	5	6
1	0	9	7	4	2	
2	9	0	1			3
3	7	1	0	2		
4	4		2	0	1	
5	2			1	0	11
6		3			11	0



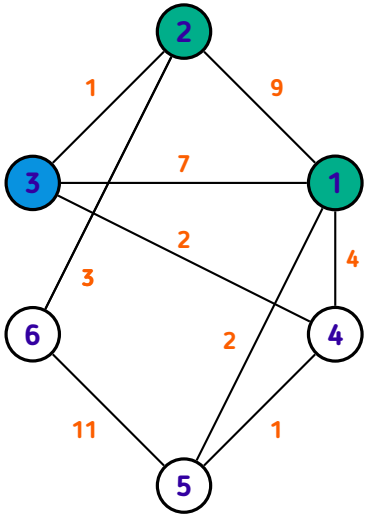
	1	2	3	4	5	6
1	0	9	7	4	2	∞
2	9	0	1	∞	∞	3
3	7	1	0	2	∞	∞
4	4	∞	2	0	1	∞
5	2	∞	∞	1	0	11
6	∞	3	∞	∞	11	0



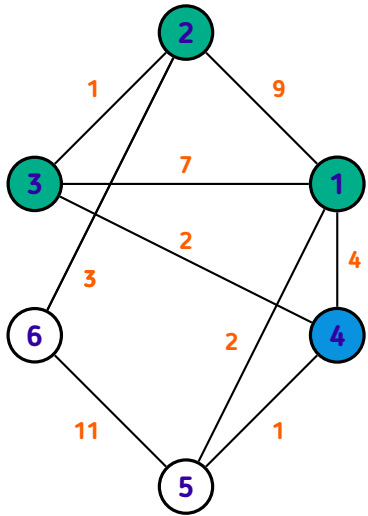
	1	2	3	4	5	6
1	0	9	7	4	2	∞
2	9	0	1	**13**	**11**	3
3	7	1	0	2	**9**	∞
4	4	**13**	2	0	1	∞
5	2	**11**	**9**	1	0	11
6	∞	3	∞	∞	11	0



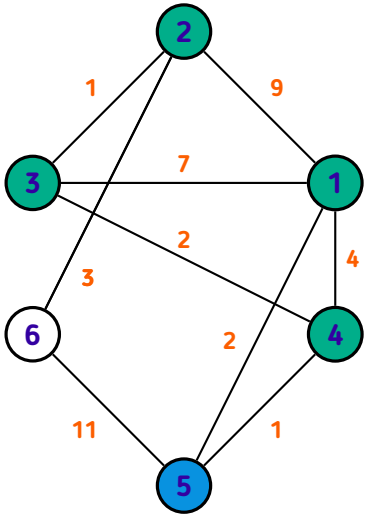
	1	2	3	4	5	6
1	0	9	7	4	2	**12**
2	9	0	1	13	11	3
3	7	1	0	2	9	**4**
4	4	13	2	0	1	**16**
5	2	11	9	1	0	11
6	**12**	3	**4**	**16**	11	0



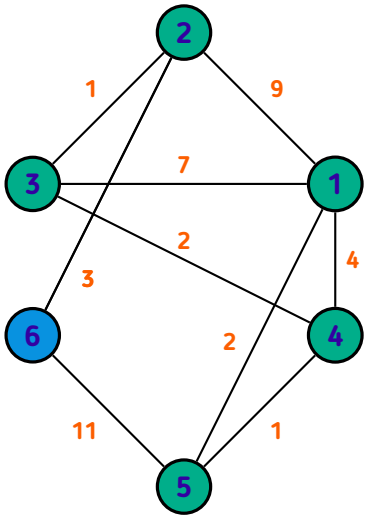
	1	2	3	4	5	6
1	0	**8**	7	4	2	**11**
2	**8**	0	1	**3**	**10**	3
3	7	1	0	2	9	4
4	4	**3**	2	0	1	**6**
5	2	**10**	9	1	0	11
6	**11**	3	4	**6**	11	0



	1	2	3	4	5	6
1	0	**7**	**6**	4	2	**10**
2	**7**	0	1	3	**4**	3
3	**6**	1	0	2	**3**	4
4	4	3	2	0	1	6
5	2	**4**	**3**	1	0	**7**
6	**10**	3	4	6	**7**	0



	1	2	3	4	5	6
1	0	**6**	**5**	**3**	2	**9**
2	**6**	0	1	3	4	3
3	**5**	1	0	2	3	4
4	**3**	3	2	0	1	6
5	2	4	3	1	0	7
6	**9**	3	4	6	7	0



	1	2	3	4	5	6
1	0	6	5	3	2	9
2	6	0	1	3	4	3
3	5	1	0	2	3	4
4	3	3	2	0	1	6
5	2	4	3	1	0	7
6	9	3	4	6	7	0

```
vector<vector<int>> floyd_warshall(int N)
{
    vector<vector<int>> dist(N + 1, vector<int>(N + 1, oo));

    for (int u = 1; u <= N; ++u)
        dist[u][u] = 0;

    for (int u = 1; u <= N; ++u)
        for (auto [v, w] : adj[u])
            dist[u][v] = w;

    for (int k = 1; k <= N; ++k)
        for (int u = 1; u <= N; ++u)
            for (int v = 1; v <= N; ++v)
                dist[u][v] = min(dist[u][v], dist[u][k] + dist[k][v]);

    return dist
}
```

Identificação de um caminho mínimo

Identificação de um caminho mínimo

- **O algoritmo de Dijkstra computa as distâncias mínimas, mas não os caminhos mínimos**

Identificação de um caminho mínimo

⋆ **O algoritmo de Dijkstra computa as distâncias mínimas, mas não os caminhos mínimos**

⋆ **Para determinar um caminho mínimo, é preciso definir uma matriz auxiliar** pred**, onde** $\text{pred}[u][v] =$ **antecessor de** v **no caminho mínimo de** u **a** v

Identificação de um caminho mínimo

⋆ **O algoritmo de Dijkstra computa as distâncias mínimas, mas não os caminhos mínimos**

⋆ **Para determinar um caminho mínimo, é preciso definir uma matriz auxiliar** pred**, onde** $\text{pred}[u][v]$ = **antecessor de** v **no caminho mínimo de** u **a** v

⋆ **No início do algoritmo,**

(a) $\text{pred}[u][u] = u, \forall u \in V$

(b) $\text{pred}[u][v] = u$**, se** $(u, v) \in E$

(c) $\text{pred}[u][v] = \text{undef}$**, caso contrário**

Identificação de um caminho mínimo

Identificação de um caminho mínimo

- **Se** (u, v) **atualizar** $d[v]$**, faça** $\text{pred}[v] = u$

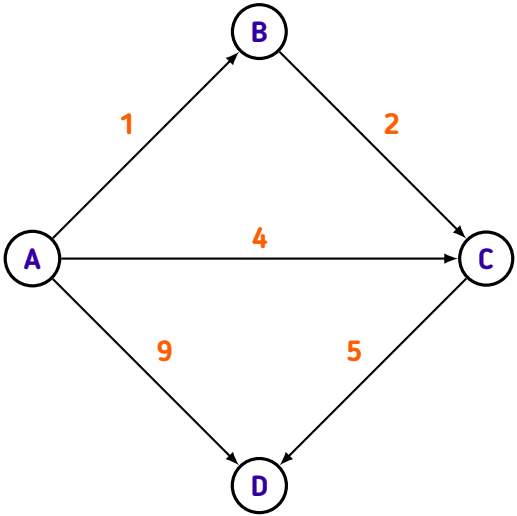
Identificação de um caminho mínimo

⋆ **Se** (u, v) **atualizar** $d[v]$**, faça** $\mathsf{pred}[v] = u$

⋆ **A sequência**

$$p = \{(u, \mathsf{pred}^{k-1}[u][v]), \ldots, (\mathsf{pred}[\mathsf{pred}[u][v]], \mathsf{pred}[u][v]), (\mathsf{pred}[u][v], v)\}$$

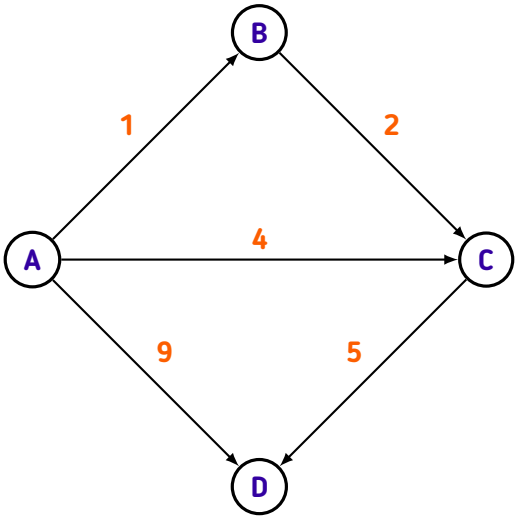
é um caminho mínimo de u **a** v **composto por** k **arestas e tamanho** $d[u][v]$



$M =$

	A	B	C	D
A				
B				
C				
D				

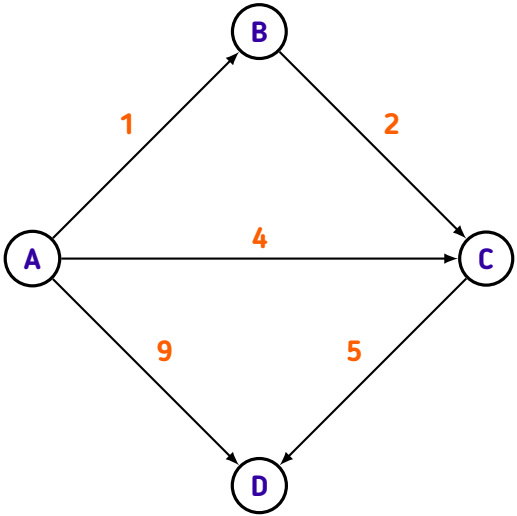
$$m_{ij} = \text{dist}[i][j]_{\text{pred}[i][j]}$$



$M =$

	A	B	C	D	
	0_A				A
		0_B			B
			0_C		C
				0_D	D

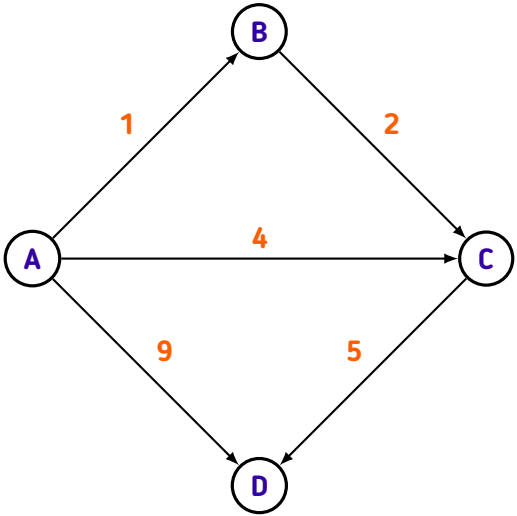
$$m_{ij} = \text{dist}[i][j]_{\text{pred}[i][j]}$$



$M =$

	A	B	C	D	
	0_A	1_A	4_A	9_A	A
		0_B	2_B		B
			0_C	5_C	C
				0_D	D

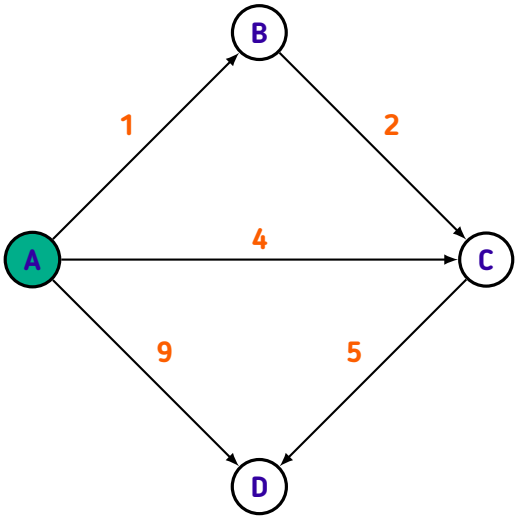
$$m_{ij} = \text{dist}[i][j]_{\text{pred}[i][j]}$$



$M =$

	A	B	C	D	
	0_A	1_A	4_A	9_A	A
	∞_-	0_B	2_B	∞_-	B
	∞_-	∞_-	0_C	5_C	C
	∞_-	∞_-	∞_-	0_D	D

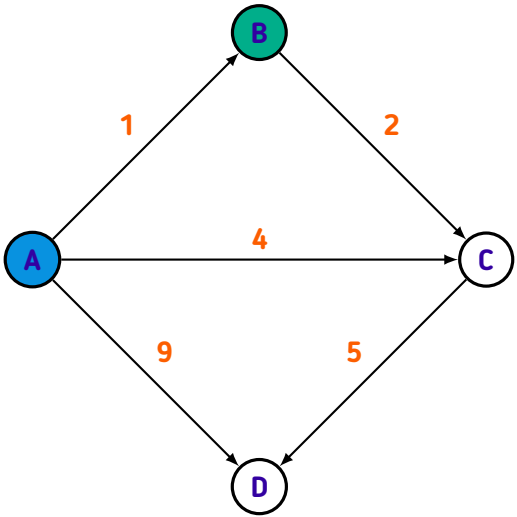
$$m_{ij} = \text{dist}[i][j]_{\text{pred}[i][j]}$$



$M =$

	A	B	C	D	
	0_A	1_A	4_A	9_A	A
	∞-	0_B	2_B	∞-	B
	∞-	∞-	0_C	5_C	C
	∞-	∞-	∞-	0_D	D

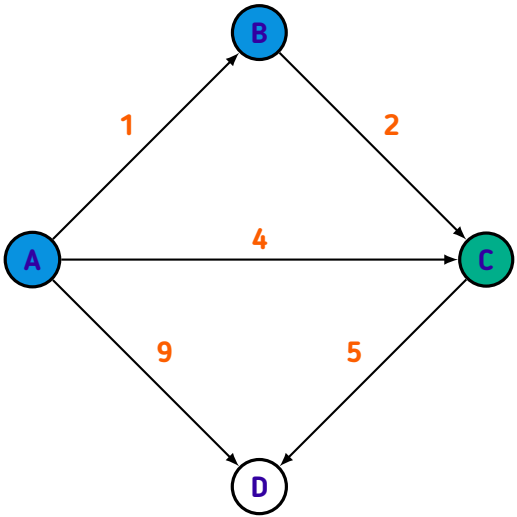
$$m_{ij} = \text{dist}[i][j]_{\text{pred}[i][j]}$$



$M =$

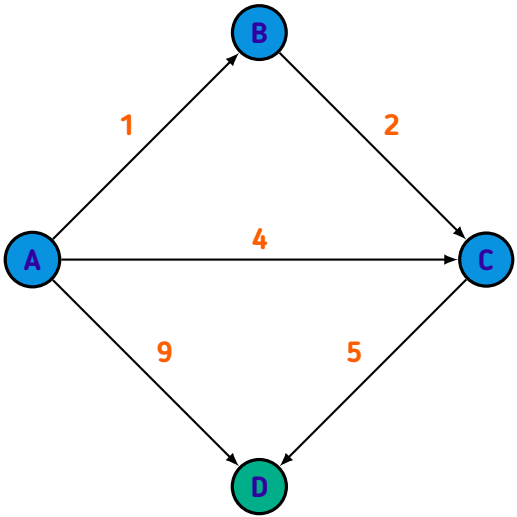
	A	B	C	D	
	0_A	1_A	3_B	9_A	A
	∞-	0_B	2_B	∞-	B
	∞-	∞-	0_C	5_C	C
	∞-	∞-	∞-	0_D	D

$$m_{ij} = \text{dist}[i][j]_{\text{pred}[i][j]}$$



$$M = \begin{array}{c|cccc} & \mathbf{A} & \mathbf{B} & \mathbf{C} & \mathbf{D} \\ \hline \mathbf{A} & 0_{\mathbf{A}} & 1_{\mathbf{A}} & 3_{\mathbf{B}} & \mathbf{8_C} \\ \mathbf{B} & \infty_{\text{-}} & 0_{\mathbf{B}} & 2_{\mathbf{B}} & \mathbf{7_C} \\ \mathbf{C} & \infty_{\text{-}} & \infty_{\text{-}} & 0_{\mathbf{C}} & 5_{\mathbf{C}} \\ \mathbf{D} & \infty_{\text{-}} & \infty_{\text{-}} & \infty_{\text{-}} & 0_{\mathbf{D}} \end{array}$$

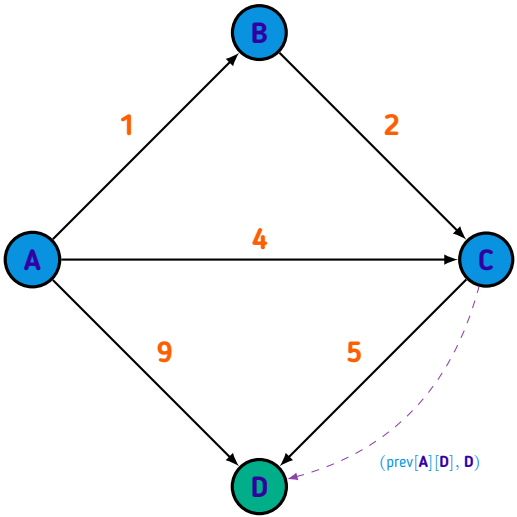
$$m_{ij} = \text{dist}[i][j]_{\text{pred}[i][j]}$$



$M =$

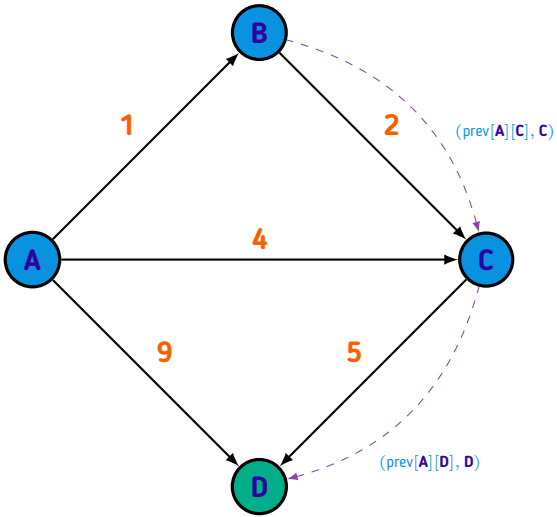
	A	B	C	D	
	0_A	1_A	3_B	8_C	A
	∞_-	0_B	2_B	7_C	B
	∞_-	∞_-	0_C	5_C	C
	∞_-	∞_-	∞_-	0_D	D

$$m_{ij} = \text{dist}[i][j]_{\text{pred}[i][j]}$$



$$M = \begin{array}{c|cccc} & \mathbf{A} & \mathbf{B} & \mathbf{C} & \mathbf{D} \\ \hline \mathbf{A} & 0_{\mathbf{A}} & 1_{\mathbf{A}} & 3_{\mathbf{B}} & 8_{\mathbf{C}} \\ \mathbf{B} & \infty_{\text{-}} & 0_{\mathbf{B}} & 2_{\mathbf{B}} & 7_{\mathbf{C}} \\ \mathbf{C} & \infty_{\text{-}} & \infty_{\text{-}} & 0_{\mathbf{C}} & 5_{\mathbf{C}} \\ \mathbf{D} & \infty_{\text{-}} & \infty_{\text{-}} & \infty_{\text{-}} & 0_{\mathbf{D}} \end{array}$$

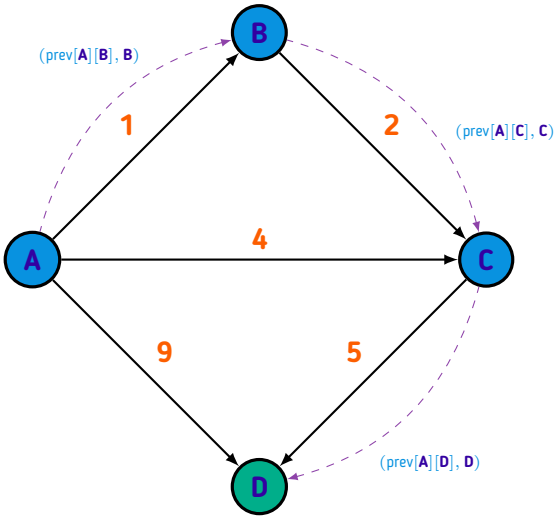
$$m_{ij} = \text{dist}[i][j]_{\text{pred}[i][j]}$$



$M =$

A	B	C	D	
0_A	1_A	3_B	8_C	A
∞_-	0_B	2_B	7_C	B
∞_-	∞_-	0_C	5_C	C
∞_-	∞_-	∞_-	0_D	D

$$m_{ij} = \text{dist}[i][j]_{\text{pred}[i][j]}$$



$M =$

A	B	C	D	
0_A	1_A	3_B	8_C	A
∞-	0_B	2_B	7_C	B
∞-	∞-	0_C	5_C	C
∞-	∞-	∞-	0_D	D

$$m_{ij} = \text{dist}[i][j]_{\text{pred}[i][j]}$$