

# Algoritmos e Estruturas de Dados II

## Aula 12 - Caminhos de Custo Mínimo - Floyd-Warshall

Prof. Luciano A. Digiampietri  
digiampietri@usp.br  
@digiampietri

# Árvore de Caminhos de Peso Mínimo

- A **Busca em Largura** fornece o caminho mais curto de um vértice de origem aos demais (Árvore de Caminhos de Menor Comprimento).

# Árvore de Caminhos de Peso Mínimo

- A **Busca em Largura** fornece o caminho mais curto de um vértice de origem aos demais (Árvore de Caminhos de Menor Comprimento).
  - O comprimento do caminho é dado pelo **número de arestas** que ele contém.

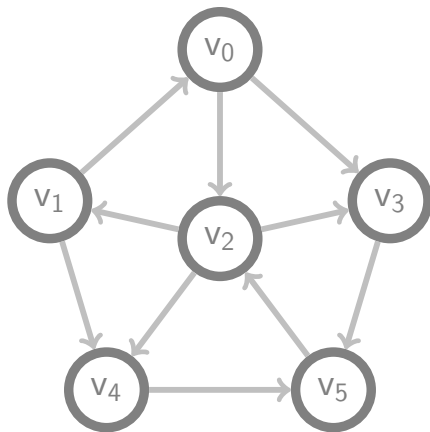
# Árvore de Caminhos de Peso Mínimo

- A **Busca em Largura** fornece o caminho mais curto de um vértice de origem aos demais (Árvore de Caminhos de Menor Comprimento).
  - O comprimento do caminho é dado pelo **número de arestas** que ele contém.
- Porém, se o grafo for ponderado, o peso pode representar, por exemplo, **a distância ou o tempo** para se percorrer a aresta.

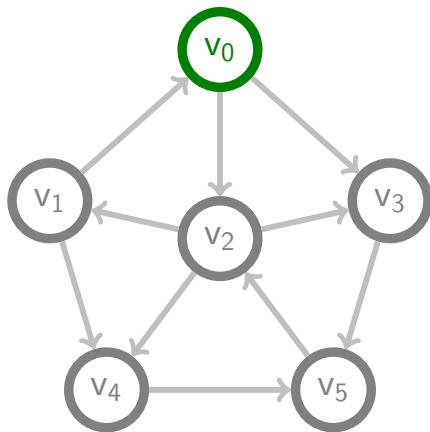
# Árvore de Caminhos de Peso Mínimo

- A **Busca em Largura** fornece o caminho mais curto de um vértice de origem aos demais (Árvore de Caminhos de Menor Comprimento).
  - O comprimento do caminho é dado pelo **número de arestas** que ele contém.
- Porém, se o grafo for ponderado, o peso pode representar, por exemplo, **a distância ou o tempo** para se percorrer a aresta.
  - O **custo de um caminho** será dado pela **soma dos pesos** das arestas e, em algumas situações, será melhor passar por mais arestas (com menor custo) do que por menos arestas (porém de maior custo).

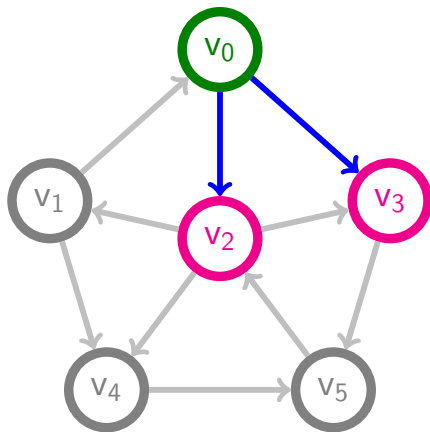
# Caminhos de Menor Comprimento (número de arestas) - Busca em Largura



# Caminhos de Menor Comprimento (número de arestas) - Busca em Largura

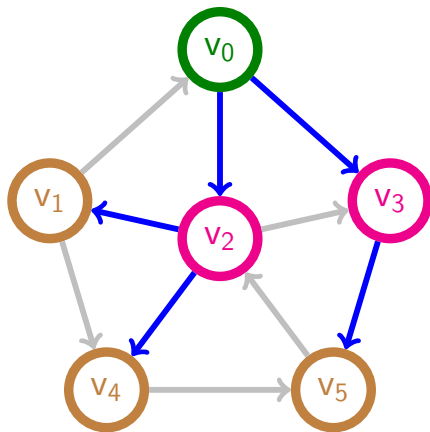


# Caminhos de Menor Comprimento (número de arestas) - Busca em Largura

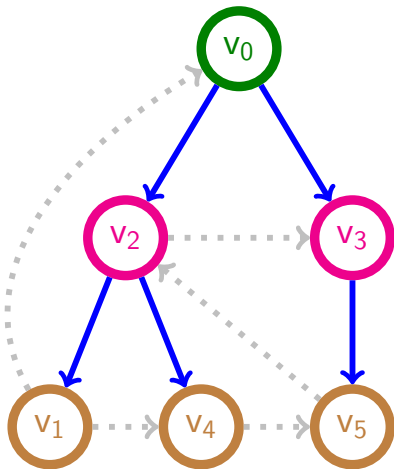




# Caminhos de Menor Comprimento (número de arestas) - Busca em Largura



# Árvore de Caminhos de Menor Comprimento



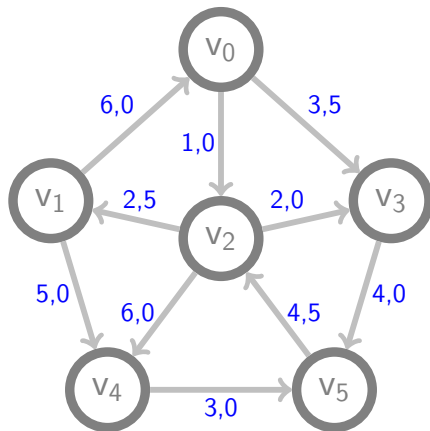
# Árvore de Caminhos de Peso Mínimo

- As Árvores de Caminhos de Peso Mínimo ou de Custo Mínimo (muitas vezes também chamadas de Árvores de Caminhos Mínimos<sup>1</sup>) representam os caminhos de menor peso (ou custo) a partir de um vértice de origem.

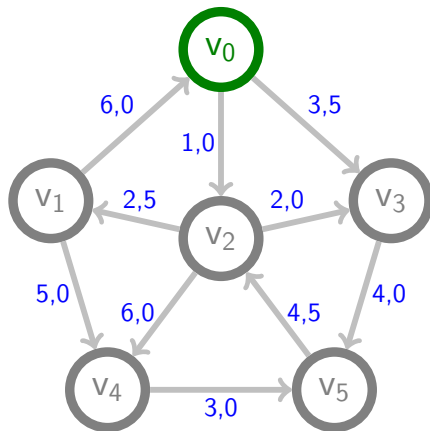
---

<sup>1</sup> Isto se confunde com caminhos de menor comprimento.

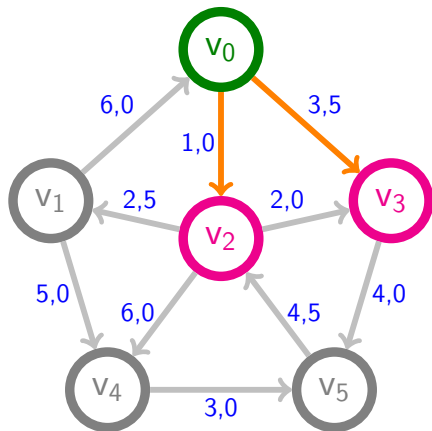
# Árvore de Caminhos de Peso Mínimo



# Árvore de Caminhos de Peso Mínimo

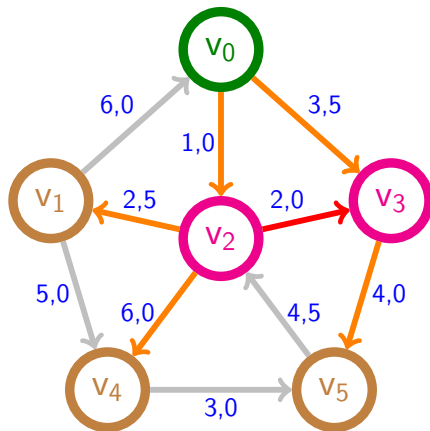


# Árvore de Caminhos de Peso Mínimo

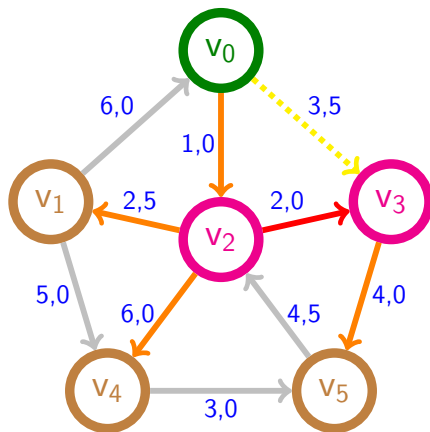


# Árvore de Caminhos de Peso Mínimo

Existe **um caminho de menor custo** ('atalho') de  $v_0$  para  $v_3$ , passando por  $v_2$ .



# Árvore de Caminhos de Peso Mínimo

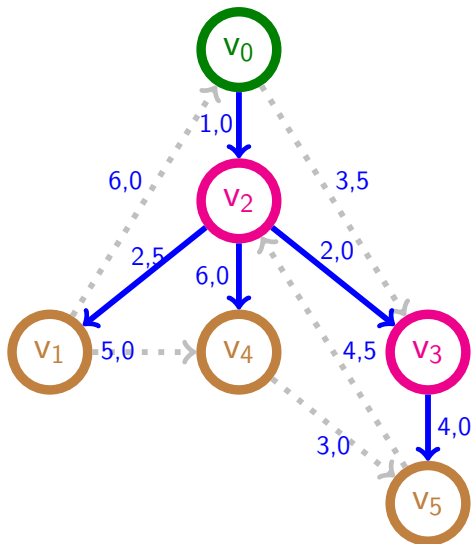


Existe **um caminho de menor custo** ('atalho') de  $v_0$  para  $v_3$ , passando por  $v_2$ .

A operação de atualizar a distância entre dois vértices, passando por outro(s) vértice(s) é chamada de **relaxamento**.



# Árvore de Caminhos de Peso Mínimo



# Caminhos de Peso Mínimo

Mas como encontramos os caminhos de custo (ou peso) mínimo?

- Existem diversas abordagens, aprenderemos três que usam o relaxamento.

# Caminhos de Peso Mínimo

Mas como encontramos os caminhos de custo (ou peso) mínimo?

- Existem diversas abordagens, aprenderemos três que usam o relaxamento.
  - **Algoritmo de Floyd-Warshall:** usado para encontrar os caminhos de peso mínimo entre todos os pares de vértices.

# Caminhos de Peso Mínimo

Mas como encontramos os caminhos de custo (ou peso) mínimo?

- Existem diversas abordagens, aprenderemos três que usam o relaxamento.
  - **Algoritmo de Floyd-Warshall:** usado para encontrar os caminhos de peso mínimo entre todos os pares de vértices.
  - **Bellman-Ford:** usado para encontrar os caminhos de peso mínimo entre um vértice de origem e os demais.

# Caminhos de Peso Mínimo

Mas como encontramos os caminhos de custo (ou peso) mínimo?

- Existem diversas abordagens, aprenderemos três que usam o relaxamento.
  - **Algoritmo de Floyd-Warshall:** usado para encontrar os caminhos de peso mínimo entre todos os pares de vértices.
  - **Bellman-Ford:** usado para encontrar os caminhos de peso mínimo entre um vértice de origem e os demais.
  - **Dijkstra:** usado para encontrar os caminhos de peso mínimo entre um vértice de origem e os demais.

# Caminhos de Peso Mínimo

Mas como encontramos os caminhos de custo (ou peso) mínimo?

- Existem diversas abordagens, aprenderemos três que usam o relaxamento.
  - **Algoritmo de Floyd-Warshall:** usado para encontrar os caminhos de peso mínimo entre todos os pares de vértices.
  - **Bellman-Ford:** usado para encontrar os caminhos de peso mínimo entre um vértice de origem e os demais.
  - **Dijkstra:** usado para encontrar os caminhos de peso mínimo entre um vértice de origem e os demais.
- Aprenderemos as especificidades de cada um destes algoritmos nesta e nas próximas aulas.

# Algoritmo de Floyd-Warshall

Publicado de forma independente em 1962<sup>2</sup>, este algoritmo tem as seguintes características:

---

<sup>2</sup>E em 1959 por Bernard Roy.

# Algoritmo de Floyd-Warshall

Publicado de forma independente em 1962<sup>2</sup>, este algoritmo tem as seguintes características:

- **Objetivo:** Encontrar os caminhos de custo mínimo entre todos os pares de vértices;

---

<sup>2</sup>E em 1959 por Bernard Roy.



# Algoritmo de Floyd-Warshall

Publicado de forma independente em 1962<sup>2</sup>, este algoritmo tem as seguintes características:

- **Objetivo:** Encontrar os caminhos de custo mínimo entre todos os pares de vértices;
- **Aplicação:** Grafos direcionados e ponderados (arestas podem ter pesos negativos, mas não pode haver um ciclo negativo);

---

<sup>2</sup>E em 1959 por Bernard Roy.

# Algoritmo de Floyd-Warshall

Publicado de forma independente em 1962<sup>2</sup>, este algoritmo tem as seguintes características:

- **Objetivo:** Encontrar os caminhos de custo mínimo entre todos os pares de vértices;
- **Aplicação:** Grafos direcionados e ponderados (arestas podem ter pesos negativos, mas não pode haver um ciclo negativo);
- **Complexidade de tempo:**  $O(V^3)$ .

---

<sup>2</sup>E em 1959 por Bernard Roy.

# Algoritmo de Floyd-Warshall

Publicado de forma independente em 1962<sup>2</sup>, este algoritmo tem as seguintes características:

- **Objetivo:** Encontrar os caminhos de custo mínimo entre todos os pares de vértices;
- **Aplicação:** Grafos direcionados e ponderados (arestas podem ter pesos negativos, mas não pode haver um ciclo negativo);
- **Complexidade de tempo:**  $O(V^3)$ .
- Abordagem utiliza Programação Dinâmica.

---

<sup>2</sup>E em 1959 por Bernard Roy.

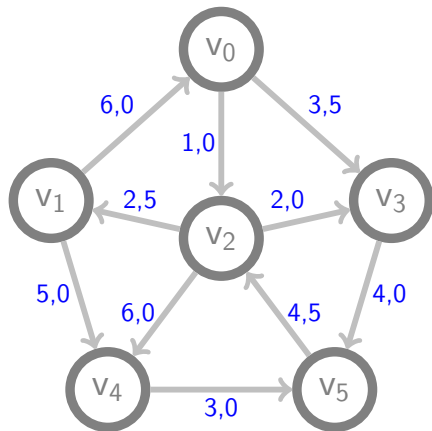
# Algoritmo de Floyd-Warshall

## Pseudocódigo:

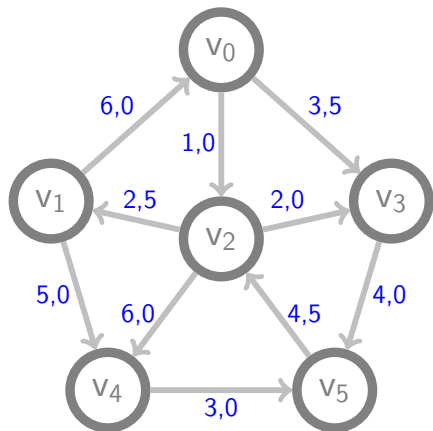
```
PARA k DE 1 A n
  PARA i DE 1 A n
    PARA j DE 1 A n
      SE  $\text{dist}[i,j] > \text{dist}[i,k] + \text{dist}[k,j]$  ENTÃO
         $\text{dist}[i,j] = \text{dist}[i,k] + \text{dist}[k,j]$ 
         $\text{pred}[i,j] = \text{pred}[k,j]$ 
```

Sendo  $n$  igual ao número de vértices, *dist* uma matriz iniciada com a matriz de pesos entre os pares de vértices (com 'infinito' para vértices não adjacentes) e *pred* a matriz de predecessores (anteriores).

# Algoritmo de Floyd-Warshall



# Algoritmo de Floyd-Warshall



## Matriz de Distâncias

|  | v0 | v1 | v2 | v3 | v4 | v5 |
|--|----|----|----|----|----|----|
|--|----|----|----|----|----|----|

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| v0 |  |  |  |  |  |  |
|----|--|--|--|--|--|--|

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| v1 |  |  |  |  |  |  |
|----|--|--|--|--|--|--|

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| v2 |  |  |  |  |  |  |
|----|--|--|--|--|--|--|

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| v3 |  |  |  |  |  |  |
|----|--|--|--|--|--|--|

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| v4 |  |  |  |  |  |  |
|----|--|--|--|--|--|--|

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| v5 |  |  |  |  |  |  |
|----|--|--|--|--|--|--|

## Matriz de Predecessores

|  | v0 | v1 | v2 | v3 | v4 | v5 |
|--|----|----|----|----|----|----|
|--|----|----|----|----|----|----|

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| v0 |  |  |  |  |  |  |
|----|--|--|--|--|--|--|

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| v1 |  |  |  |  |  |  |
|----|--|--|--|--|--|--|

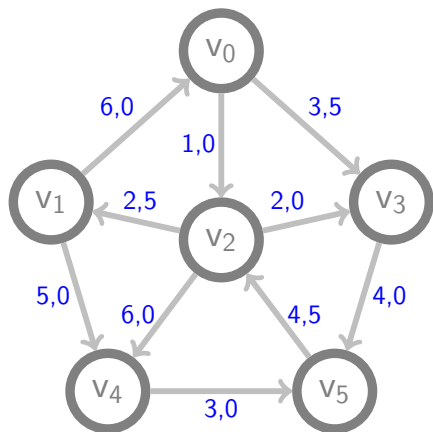
|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| v2 |  |  |  |  |  |  |
|----|--|--|--|--|--|--|

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| v3 |  |  |  |  |  |  |
|----|--|--|--|--|--|--|

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| v4 |  |  |  |  |  |  |
|----|--|--|--|--|--|--|

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| v5 |  |  |  |  |  |  |
|----|--|--|--|--|--|--|

# Algoritmo de Floyd-Warshall



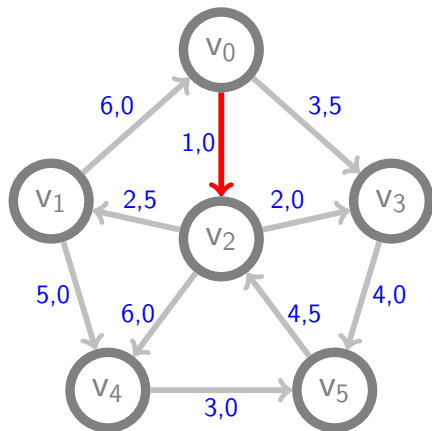
**Matriz de Distâncias**

|    | v0  | v1  | v2  | v3  | v4  | v5  |
|----|-----|-----|-----|-----|-----|-----|
| v0 | 0,0 | -   | 1,0 | 3,5 | -   | -   |
| v1 | 6,0 | 0,0 | -   | -   | 5,0 | -   |
| v2 | -   | 2,5 | 0,0 | 2,0 | 6,0 | -   |
| v3 | -   | -   | -   | 0,0 | -   | 4,0 |
| v4 | -   | -   | -   | -   | 0,0 | 3,0 |
| v5 | -   | -   | 4,5 | -   | -   | 0,0 |

**Matriz de Predecessores**

|    | v0 | v1 | v2 | v3 | v4 | v5 |
|----|----|----|----|----|----|----|
| v0 | 0  | -1 | 0  | 0  | -1 | -1 |
| v1 | 1  | 1  | -1 | -1 | 1  | -1 |
| v2 | -1 | 2  | 2  | 2  | 2  | -1 |
| v3 | -1 | -1 | -1 | 3  | -1 | 3  |
| v4 | -1 | -1 | -1 | -1 | 4  | 4  |
| v5 | -1 | -1 | 5  | -1 | -1 | 5  |

# Algoritmo de Floyd-Warshall



**Matriz de Distâncias**

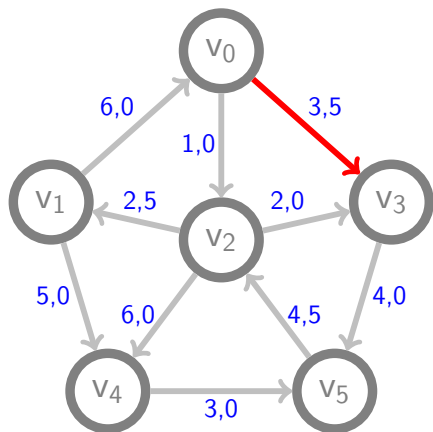
|    | v0  | v1  | v2  | v3  | v4  | v5  |
|----|-----|-----|-----|-----|-----|-----|
| v0 | 0,0 | -   | 1,0 | 3,5 | -   | -   |
| v1 | 6,0 | 0,0 | 7,0 | -   | 5,0 | -   |
| v2 | -   | 2,5 | 0,0 | 2,0 | 6,0 | -   |
| v3 | -   | -   | -   | 0,0 | -   | 4,0 |
| v4 | -   | -   | -   | -   | 0,0 | 3,0 |
| v5 | -   | -   | 4,5 | -   | -   | 0,0 |

**Matriz de Predecessores**

|    | v0 | v1 | v2 | v3 | v4 | v5 |
|----|----|----|----|----|----|----|
| v0 | 0  | -1 | 0  | 0  | -1 | -1 |
| v1 | 1  | 1  | 0  | -1 | 1  | -1 |
| v2 | -1 | 2  | 2  | 2  | 2  | -1 |
| v3 | -1 | -1 | -1 | 3  | -1 | 3  |
| v4 | -1 | -1 | -1 | -1 | 4  | 4  |
| v5 | -1 | -1 | 5  | -1 | -1 | 5  |



# Algoritmo de Floyd-Warshall



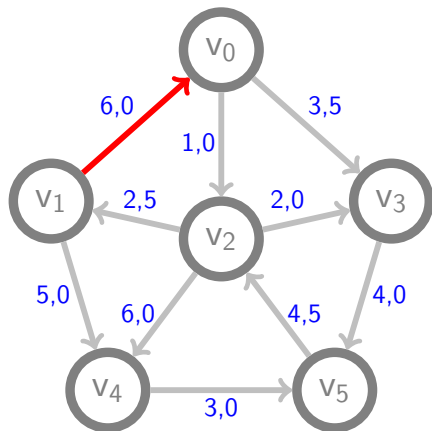
**Matriz de Distâncias**

|       | $v_0$ | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $v_5$ |
|-------|-------|-------|-------|-------|-------|-------|
| $v_0$ | 0,0   | -     | 1,0   | 3,5   | -     | -     |
| $v_1$ | 6,0   | 0,0   | 7,0   | 9,5   | 5,0   | -     |
| $v_2$ | -     | 2,5   | 0,0   | 2,0   | 6,0   | -     |
| $v_3$ | -     | -     | -     | 0,0   | -     | 4,0   |
| $v_4$ | -     | -     | -     | -     | 0,0   | 3,0   |
| $v_5$ | -     | -     | 4,5   | -     | -     | 0,0   |

**Matriz de Predecessores**

|       | $v_0$ | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $v_5$ |
|-------|-------|-------|-------|-------|-------|-------|
| $v_0$ | 0     | -1    | 0     | 0     | -1    | -1    |
| $v_1$ | 1     | 1     | 0     | 0     | 1     | -1    |
| $v_2$ | -1    | 2     | 2     | 2     | 2     | -1    |
| $v_3$ | -1    | -1    | -1    | 3     | -1    | 3     |
| $v_4$ | -1    | -1    | -1    | -1    | 4     | 4     |
| $v_5$ | -1    | -1    | 5     | -1    | -1    | 5     |

# Algoritmo de Floyd-Warshall



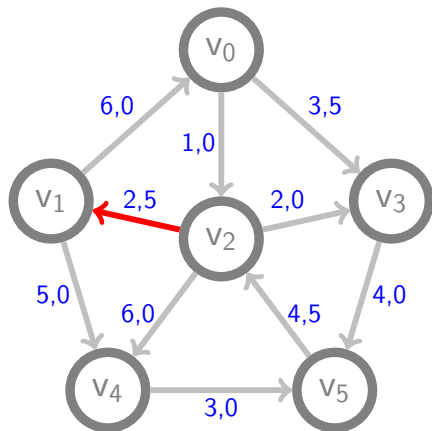
**Matriz de Distâncias**

|    | v0  | v1  | v2  | v3  | v4  | v5  |
|----|-----|-----|-----|-----|-----|-----|
| v0 | 0,0 | -   | 1,0 | 3,5 | -   | -   |
| v1 | 6,0 | 0,0 | 7,0 | 9,5 | 5,0 | -   |
| v2 | 8,5 | 2,5 | 0,0 | 2,0 | 6,0 | -   |
| v3 | -   | -   | -   | 0,0 | -   | 4,0 |
| v4 | -   | -   | -   | -   | 0,0 | 3,0 |
| v5 | -   | -   | 4,5 | -   | -   | 0,0 |

**Matriz de Predecessores**

|    | v0 | v1 | v2 | v3 | v4 | v5 |
|----|----|----|----|----|----|----|
| v0 | 0  | -1 | 0  | 0  | -1 | -1 |
| v1 | 1  | 1  | 0  | 0  | 1  | -1 |
| v2 | 1  | 2  | 2  | 2  | 2  | -1 |
| v3 | -1 | -1 | -1 | 3  | -1 | 3  |
| v4 | -1 | -1 | -1 | -1 | 4  | 4  |
| v5 | -1 | -1 | 5  | -1 | -1 | 5  |

# Algoritmo de Floyd-Warshall



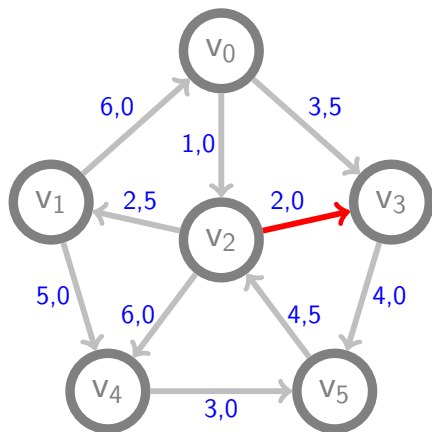
**Matriz de Distâncias**

|    | v0  | v1  | v2  | v3  | v4  | v5  |
|----|-----|-----|-----|-----|-----|-----|
| v0 | 0,0 | 3,5 | 1,0 | 3,5 | -   | -   |
| v1 | 6,0 | 0,0 | 7,0 | 9,5 | 5,0 | -   |
| v2 | 8,5 | 2,5 | 0,0 | 2,0 | 6,0 | -   |
| v3 | -   | -   | -   | 0,0 | -   | 4,0 |
| v4 | -   | -   | -   | -   | 0,0 | 3,0 |
| v5 | -   | -   | 4,5 | -   | -   | 0,0 |

**Matriz de Predecessores**

|    | v0 | v1 | v2 | v3 | v4 | v5 |
|----|----|----|----|----|----|----|
| v0 | 0  | 2  | 0  | 0  | -1 | -1 |
| v1 | 1  | 1  | 0  | 0  | 1  | -1 |
| v2 | 1  | 2  | 2  | 2  | 2  | -1 |
| v3 | -1 | -1 | -1 | 3  | -1 | 3  |
| v4 | -1 | -1 | -1 | -1 | 4  | 4  |
| v5 | -1 | -1 | 5  | -1 | -1 | 5  |

# Algoritmo de Floyd-Warshall



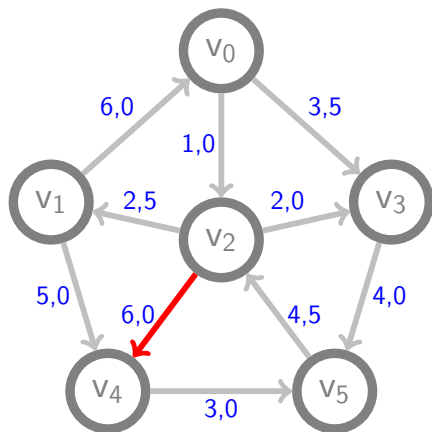
**Matriz de Distâncias**

|    | v0  | v1  | v2  | v3  | v4  | v5  |
|----|-----|-----|-----|-----|-----|-----|
| v0 | 0,0 | 3,5 | 1,0 | 3,0 | -   | -   |
| v1 | 6,0 | 0,0 | 7,0 | 9,5 | 5,0 | -   |
| v2 | 8,5 | 2,5 | 0,0 | 2,0 | 6,0 | -   |
| v3 | -   | -   | -   | 0,0 | -   | 4,0 |
| v4 | -   | -   | -   | -   | 0,0 | 3,0 |
| v5 | -   | -   | 4,5 | -   | -   | 0,0 |

**Matriz de Predecessores**

|    | v0 | v1 | v2 | v3 | v4 | v5 |
|----|----|----|----|----|----|----|
| v0 | 0  | 2  | 0  | 2  | -1 | -1 |
| v1 | 1  | 1  | 0  | 0  | 1  | -1 |
| v2 | 1  | 2  | 2  | 2  | 2  | -1 |
| v3 | -1 | -1 | -1 | 3  | -1 | 3  |
| v4 | -1 | -1 | -1 | -1 | 4  | 4  |
| v5 | -1 | -1 | 5  | -1 | -1 | 5  |

# Algoritmo de Floyd-Warshall



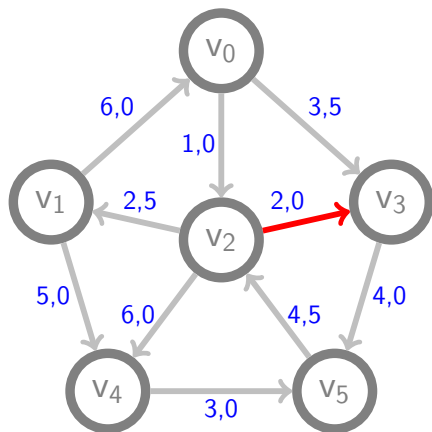
**Matriz de Distâncias**

|    | v0  | v1  | v2  | v3  | v4  | v5  |
|----|-----|-----|-----|-----|-----|-----|
| v0 | 0,0 | 3,5 | 1,0 | 3,0 | 7,0 | -   |
| v1 | 6,0 | 0,0 | 7,0 | 9,5 | 5,0 | -   |
| v2 | 8,5 | 2,5 | 0,0 | 2,0 | 6,0 | -   |
| v3 | -   | -   | -   | 0,0 | -   | 4,0 |
| v4 | -   | -   | -   | -   | 0,0 | 3,0 |
| v5 | -   | -   | 4,5 | -   | -   | 0,0 |

**Matriz de Predecessores**

|    | v0 | v1 | v2 | v3 | v4 | v5 |
|----|----|----|----|----|----|----|
| v0 | 0  | 2  | 0  | 2  | 2  | -1 |
| v1 | 1  | 1  | 0  | 0  | 1  | -1 |
| v2 | 1  | 2  | 2  | 2  | 2  | -1 |
| v3 | -1 | -1 | -1 | 3  | -1 | 3  |
| v4 | -1 | -1 | -1 | -1 | 4  | 4  |
| v5 | -1 | -1 | 5  | -1 | -1 | 5  |

# Algoritmo de Floyd-Warshall



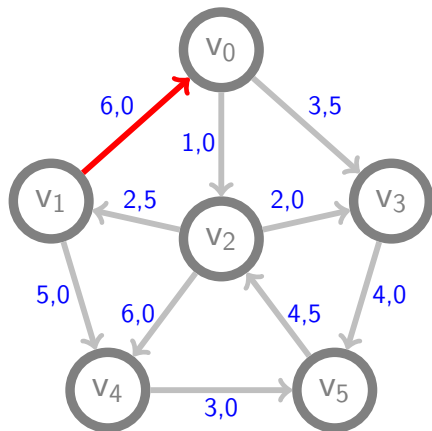
**Matriz de Distâncias**

|    | v0  | v1  | v2  | v3  | v4  | v5  |
|----|-----|-----|-----|-----|-----|-----|
| v0 | 0,0 | 3,5 | 1,0 | 3,0 | 7,0 | -   |
| v1 | 6,0 | 0,0 | 7,0 | 9,0 | 5,0 | -   |
| v2 | 8,5 | 2,5 | 0,0 | 2,0 | 6,0 | -   |
| v3 | -   | -   | -   | 0,0 | -   | 4,0 |
| v4 | -   | -   | -   | -   | 0,0 | 3,0 |
| v5 | -   | -   | 4,5 | -   | -   | 0,0 |

**Matriz de Predecessores**

|    | v0 | v1 | v2 | v3 | v4 | v5 |
|----|----|----|----|----|----|----|
| v0 | 0  | 2  | 0  | 2  | 2  | -1 |
| v1 | 1  | 1  | 0  | 2  | 1  | -1 |
| v2 | 1  | 2  | 2  | 2  | 2  | -1 |
| v3 | -1 | -1 | -1 | 3  | -1 | 3  |
| v4 | -1 | -1 | -1 | -1 | 4  | 4  |
| v5 | -1 | -1 | 5  | -1 | -1 | 5  |

# Algoritmo de Floyd-Warshall



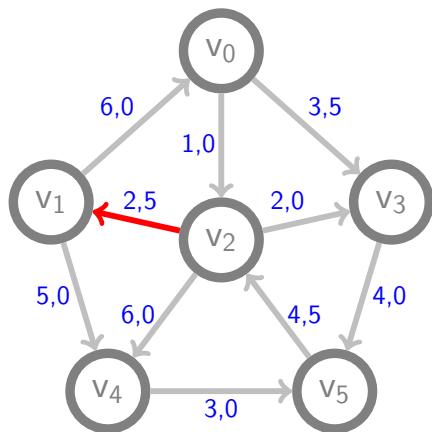
**Matriz de Distâncias**

|    | v0   | v1  | v2  | v3  | v4  | v5  |
|----|------|-----|-----|-----|-----|-----|
| v0 | 0,0  | 3,5 | 1,0 | 3,0 | 7,0 | -   |
| v1 | 6,0  | 0,0 | 7,0 | 9,0 | 5,0 | -   |
| v2 | 8,5  | 2,5 | 0,0 | 2,0 | 6,0 | -   |
| v3 | -    | -   | -   | 0,0 | -   | 4,0 |
| v4 | -    | -   | -   | -   | 0,0 | 3,0 |
| v5 | 13,0 | -   | 4,5 | -   | -   | 0,0 |

**Matriz de Predecessores**

|    | v0 | v1 | v2 | v3 | v4 | v5 |
|----|----|----|----|----|----|----|
| v0 | 0  | 2  | 0  | 2  | 2  | -1 |
| v1 | 1  | 1  | 0  | 2  | 1  | -1 |
| v2 | 1  | 2  | 2  | 2  | 2  | -1 |
| v3 | -1 | -1 | -1 | 3  | -1 | 3  |
| v4 | -1 | -1 | -1 | -1 | 4  | 4  |
| v5 | 1  | -1 | 5  | -1 | -1 | 5  |

# Algoritmo de Floyd-Warshall



**Matriz de Distâncias**

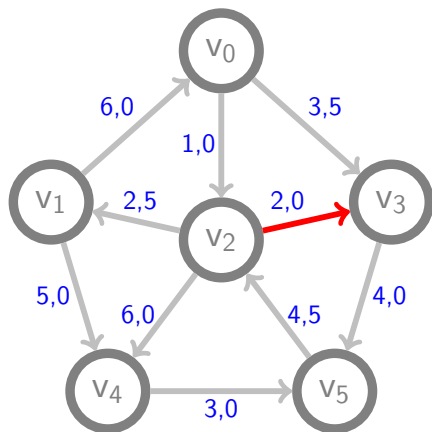
|       | $v_0$ | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $v_5$ |
|-------|-------|-------|-------|-------|-------|-------|
| $v_0$ | 0,0   | 3,5   | 1,0   | 3,0   | 7,0   | -     |
| $v_1$ | 6,0   | 0,0   | 7,0   | 9,0   | 5,0   | -     |
| $v_2$ | 8,5   | 2,5   | 0,0   | 2,0   | 6,0   | -     |
| $v_3$ | -     | -     | -     | 0,0   | -     | 4,0   |
| $v_4$ | -     | -     | -     | -     | 0,0   | 3,0   |
| $v_5$ | 13,0  | 7,0   | 4,5   | -     | -     | 0,0   |

**Matriz de Predecessores**

|       | $v_0$ | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $v_5$ |
|-------|-------|-------|-------|-------|-------|-------|
| $v_0$ | 0     | 2     | 0     | 2     | 2     | -1    |
| $v_1$ | 1     | 1     | 0     | 2     | 1     | -1    |
| $v_2$ | 1     | 2     | 2     | 2     | 2     | -1    |
| $v_3$ | -1    | -1    | -1    | 3     | -1    | 3     |
| $v_4$ | -1    | -1    | -1    | -1    | 4     | 4     |
| $v_5$ | 1     | 2     | 5     | -1    | -1    | 5     |



# Algoritmo de Floyd-Warshall



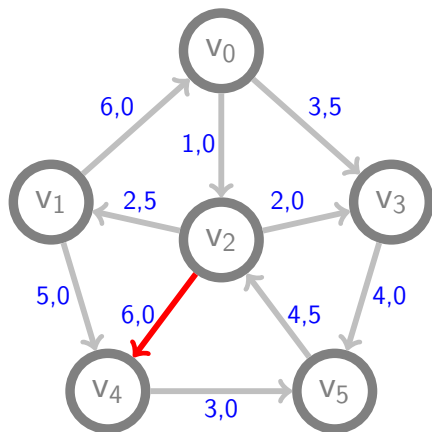
**Matriz de Distâncias**

|       | $v_0$ | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $v_5$ |
|-------|-------|-------|-------|-------|-------|-------|
| $v_0$ | 0,0   | 3,5   | 1,0   | 3,0   | 7,0   | -     |
| $v_1$ | 6,0   | 0,0   | 7,0   | 9,0   | 5,0   | -     |
| $v_2$ | 8,5   | 2,5   | 0,0   | 2,0   | 6,0   | -     |
| $v_3$ | -     | -     | -     | 0,0   | -     | 4,0   |
| $v_4$ | -     | -     | -     | -     | 0,0   | 3,0   |
| $v_5$ | 13,0  | 7,0   | 4,5   | 6,5   | -     | 0,0   |

**Matriz de Predecessores**

|       | $v_0$ | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $v_5$ |
|-------|-------|-------|-------|-------|-------|-------|
| $v_0$ | 0     | 2     | 0     | 2     | 2     | -1    |
| $v_1$ | 1     | 1     | 0     | 2     | 1     | -1    |
| $v_2$ | 1     | 2     | 2     | 2     | 2     | -1    |
| $v_3$ | -1    | -1    | -1    | 3     | -1    | 3     |
| $v_4$ | -1    | -1    | -1    | -1    | 4     | 4     |
| $v_5$ | 1     | 2     | 5     | 2     | -1    | 5     |

# Algoritmo de Floyd-Warshall



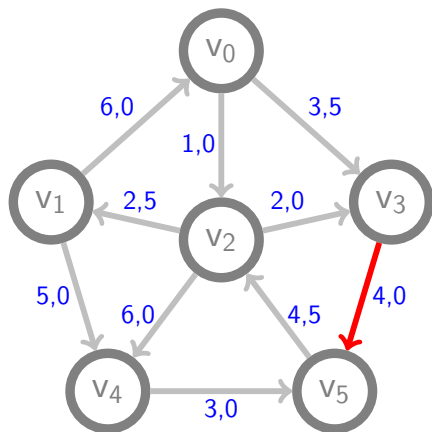
**Matriz de Distâncias**

|    | v0   | v1  | v2  | v3  | v4   | v5  |
|----|------|-----|-----|-----|------|-----|
| v0 | 0,0  | 3,5 | 1,0 | 3,0 | 7,0  | -   |
| v1 | 6,0  | 0,0 | 7,0 | 9,0 | 5,0  | -   |
| v2 | 8,5  | 2,5 | 0,0 | 2,0 | 6,0  | -   |
| v3 | -    | -   | -   | 0,0 | -    | 4,0 |
| v4 | -    | -   | -   | -   | 0,0  | 3,0 |
| v5 | 13,0 | 7,0 | 4,5 | 6,5 | 10,5 | 0,0 |

**Matriz de Predecessores**

|    | v0 | v1 | v2 | v3 | v4 | v5 |
|----|----|----|----|----|----|----|
| v0 | 0  | 2  | 0  | 2  | 2  | -1 |
| v1 | 1  | 1  | 0  | 2  | 1  | -1 |
| v2 | 1  | 2  | 2  | 2  | 2  | -1 |
| v3 | -1 | -1 | -1 | 3  | -1 | 3  |
| v4 | -1 | -1 | -1 | -1 | 4  | 4  |
| v5 | 1  | 2  | 5  | 2  | 2  | 5  |

# Algoritmo de Floyd-Warshall



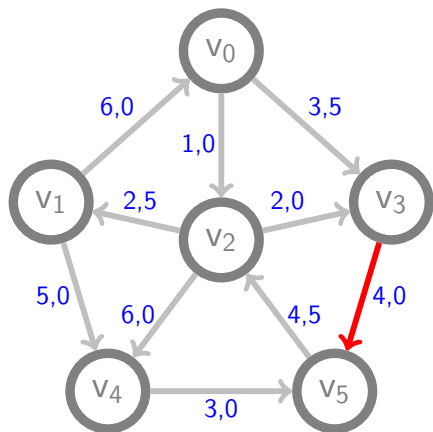
**Matriz de Distâncias**

|    | v0   | v1  | v2  | v3  | v4   | v5  |
|----|------|-----|-----|-----|------|-----|
| v0 | 0,0  | 3,5 | 1,0 | 3,0 | 7,0  | 7,0 |
| v1 | 6,0  | 0,0 | 7,0 | 9,0 | 5,0  | -   |
| v2 | 8,5  | 2,5 | 0,0 | 2,0 | 6,0  | -   |
| v3 | -    | -   | -   | 0,0 | -    | 4,0 |
| v4 | -    | -   | -   | -   | 0,0  | 3,0 |
| v5 | 13,0 | 7,0 | 4,5 | 6,5 | 10,5 | 0,0 |

**Matriz de Predecessores**

|    | v0 | v1 | v2 | v3 | v4 | v5 |
|----|----|----|----|----|----|----|
| v0 | 0  | 2  | 0  | 2  | 2  | 3  |
| v1 | 1  | 1  | 0  | 2  | 1  | -1 |
| v2 | 1  | 2  | 2  | 2  | 2  | -1 |
| v3 | -1 | -1 | -1 | 3  | -1 | 3  |
| v4 | -1 | -1 | -1 | -1 | 4  | 4  |
| v5 | 1  | 2  | 5  | 2  | 2  | 5  |

# Algoritmo de Floyd-Warshall



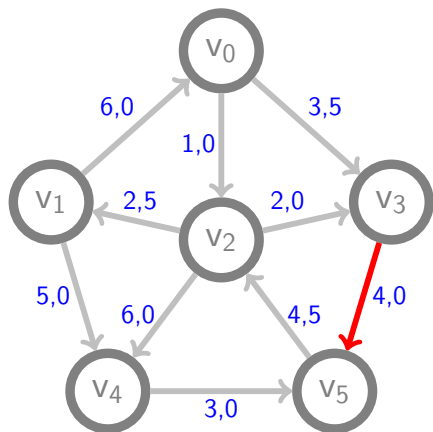
**Matriz de Distâncias**

|    | v0   | v1  | v2  | v3  | v4   | v5   |
|----|------|-----|-----|-----|------|------|
| v0 | 0,0  | 3,5 | 1,0 | 3,0 | 7,0  | 7,0  |
| v1 | 6,0  | 0,0 | 7,0 | 9,0 | 5,0  | 13,0 |
| v2 | 8,5  | 2,5 | 0,0 | 2,0 | 6,0  | -    |
| v3 | -    | -   | -   | 0,0 | -    | 4,0  |
| v4 | -    | -   | -   | -   | 0,0  | 3,0  |
| v5 | 13,0 | 7,0 | 4,5 | 6,5 | 10,5 | 0,0  |

**Matriz de Predecessores**

|    | v0 | v1 | v2 | v3 | v4 | v5 |
|----|----|----|----|----|----|----|
| v0 | 0  | 2  | 0  | 2  | 2  | 3  |
| v1 | 1  | 1  | 0  | 2  | 1  | 3  |
| v2 | 1  | 2  | 2  | 2  | 2  | -1 |
| v3 | -1 | -1 | -1 | 3  | -1 | 3  |
| v4 | -1 | -1 | -1 | -1 | 4  | 4  |
| v5 | 1  | 2  | 5  | 2  | 2  | 5  |

# Algoritmo de Floyd-Warshall



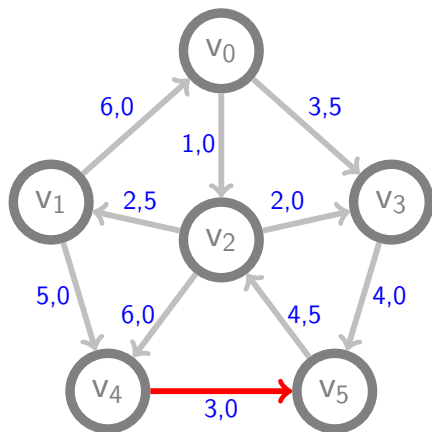
**Matriz de Distâncias**

|    | v0   | v1  | v2  | v3  | v4   | v5   |
|----|------|-----|-----|-----|------|------|
| v0 | 0,0  | 3,5 | 1,0 | 3,0 | 7,0  | 7,0  |
| v1 | 6,0  | 0,0 | 7,0 | 9,0 | 5,0  | 13,0 |
| v2 | 8,5  | 2,5 | 0,0 | 2,0 | 6,0  | 6,0  |
| v3 | -    | -   | -   | 0,0 | -    | 4,0  |
| v4 | -    | -   | -   | -   | 0,0  | 3,0  |
| v5 | 13,0 | 7,0 | 4,5 | 6,5 | 10,5 | 0,0  |

**Matriz de Predecessores**

|    | v0 | v1 | v2 | v3 | v4 | v5 |
|----|----|----|----|----|----|----|
| v0 | 0  | 2  | 0  | 2  | 2  | 3  |
| v1 | 1  | 1  | 0  | 2  | 1  | 3  |
| v2 | 1  | 2  | 2  | 2  | 2  | 3  |
| v3 | -1 | -1 | -1 | 3  | -1 | 3  |
| v4 | -1 | -1 | -1 | -1 | 4  | 4  |
| v5 | 1  | 2  | 5  | 2  | 2  | 5  |

# Algoritmo de Floyd-Warshall



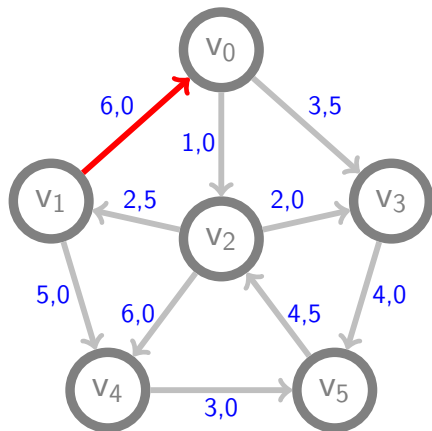
**Matriz de Distâncias**

|    | v0   | v1  | v2  | v3  | v4   | v5  |
|----|------|-----|-----|-----|------|-----|
| v0 | 0,0  | 3,5 | 1,0 | 3,0 | 7,0  | 7,0 |
| v1 | 6,0  | 0,0 | 7,0 | 9,0 | 5,0  | 8,0 |
| v2 | 8,5  | 2,5 | 0,0 | 2,0 | 6,0  | 6,0 |
| v3 | -    | -   | -   | 0,0 | -    | 4,0 |
| v4 | -    | -   | -   | -   | 0,0  | 3,0 |
| v5 | 13,0 | 7,0 | 4,5 | 6,5 | 10,5 | 0,0 |

**Matriz de Predecessores**

|    | v0 | v1 | v2 | v3 | v4 | v5 |
|----|----|----|----|----|----|----|
| v0 | 0  | 2  | 0  | 2  | 2  | 3  |
| v1 | 1  | 1  | 0  | 2  | 1  | 4  |
| v2 | 1  | 2  | 2  | 2  | 2  | 3  |
| v3 | -1 | -1 | -1 | 3  | -1 | 3  |
| v4 | -1 | -1 | -1 | -1 | 4  | 4  |
| v5 | 1  | 2  | 5  | 2  | 2  | 5  |

# Algoritmo de Floyd-Warshall



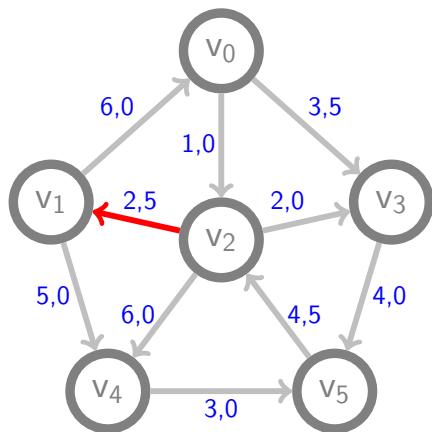
**Matriz de Distâncias**

|    | v0   | v1  | v2  | v3  | v4   | v5  |
|----|------|-----|-----|-----|------|-----|
| v0 | 0,0  | 3,5 | 1,0 | 3,0 | 7,0  | 7,0 |
| v1 | 6,0  | 0,0 | 7,0 | 9,0 | 5,0  | 8,0 |
| v2 | 8,5  | 2,5 | 0,0 | 2,0 | 6,0  | 6,0 |
| v3 | 17,0 | -   | -   | 0,0 | -    | 4,0 |
| v4 | -    | -   | -   | -   | 0,0  | 3,0 |
| v5 | 13,0 | 7,0 | 4,5 | 6,5 | 10,5 | 0,0 |

**Matriz de Predecessores**

|    | v0 | v1 | v2 | v3 | v4 | v5 |
|----|----|----|----|----|----|----|
| v0 | 0  | 2  | 0  | 2  | 2  | 3  |
| v1 | 1  | 1  | 0  | 2  | 1  | 4  |
| v2 | 1  | 2  | 2  | 2  | 2  | 3  |
| v3 | 1  | -1 | -1 | 3  | -1 | 3  |
| v4 | -1 | -1 | -1 | -1 | 4  | 4  |
| v5 | 1  | 2  | 5  | 2  | 2  | 5  |

# Algoritmo de Floyd-Warshall



**Matriz de Distâncias**

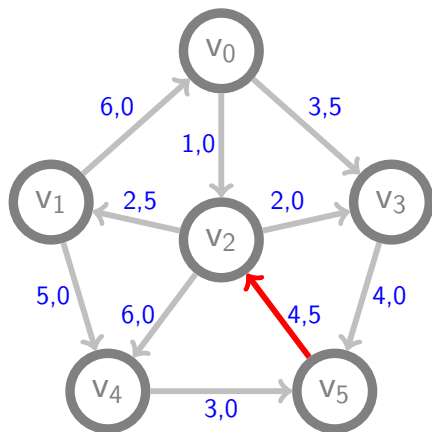
|    | v0   | v1   | v2  | v3  | v4   | v5  |
|----|------|------|-----|-----|------|-----|
| v0 | 0,0  | 3,5  | 1,0 | 3,0 | 7,0  | 7,0 |
| v1 | 6,0  | 0,0  | 7,0 | 9,0 | 5,0  | 8,0 |
| v2 | 8,5  | 2,5  | 0,0 | 2,0 | 6,0  | 6,0 |
| v3 | 17,0 | 11,0 | -   | 0,0 | -    | 4,0 |
| v4 | -    | -    | -   | -   | 0,0  | 3,0 |
| v5 | 13,0 | 7,0  | 4,5 | 6,5 | 10,5 | 0,0 |

**Matriz de Predecessores**

|    | v0 | v1 | v2 | v3 | v4 | v5 |
|----|----|----|----|----|----|----|
| v0 | 0  | 2  | 0  | 2  | 2  | 3  |
| v1 | 1  | 1  | 0  | 2  | 1  | 4  |
| v2 | 1  | 2  | 2  | 2  | 2  | 3  |
| v3 | 1  | 2  | -1 | 3  | -1 | 3  |
| v4 | -1 | -1 | -1 | -1 | 4  | 4  |
| v5 | 1  | 2  | 5  | 2  | 2  | 5  |



# Algoritmo de Floyd-Warshall



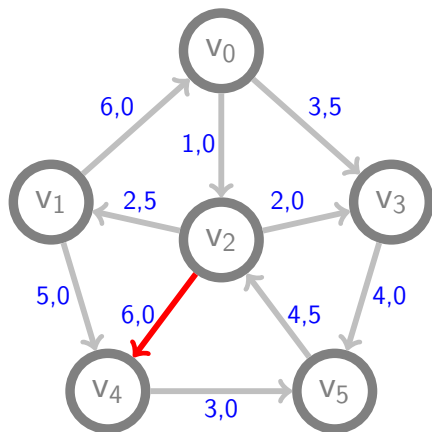
**Matriz de Distâncias**

|    | v0   | v1   | v2  | v3  | v4   | v5  |
|----|------|------|-----|-----|------|-----|
| v0 | 0,0  | 3,5  | 1,0 | 3,0 | 7,0  | 7,0 |
| v1 | 6,0  | 0,0  | 7,0 | 9,0 | 5,0  | 8,0 |
| v2 | 8,5  | 2,5  | 0,0 | 2,0 | 6,0  | 6,0 |
| v3 | 17,0 | 11,0 | 8,5 | 0,0 | -    | 4,0 |
| v4 | -    | -    | -   | -   | 0,0  | 3,0 |
| v5 | 13,0 | 7,0  | 4,5 | 6,5 | 10,5 | 0,0 |

**Matriz de Predecessores**

|    | v0 | v1 | v2 | v3 | v4 | v5 |
|----|----|----|----|----|----|----|
| v0 | 0  | 2  | 0  | 2  | 2  | 3  |
| v1 | 1  | 1  | 0  | 2  | 1  | 4  |
| v2 | 1  | 2  | 2  | 2  | 2  | 3  |
| v3 | 1  | 2  | 5  | 3  | -1 | 3  |
| v4 | -1 | -1 | -1 | -1 | 4  | 4  |
| v5 | 1  | 2  | 5  | 2  | 2  | 5  |

# Algoritmo de Floyd-Warshall



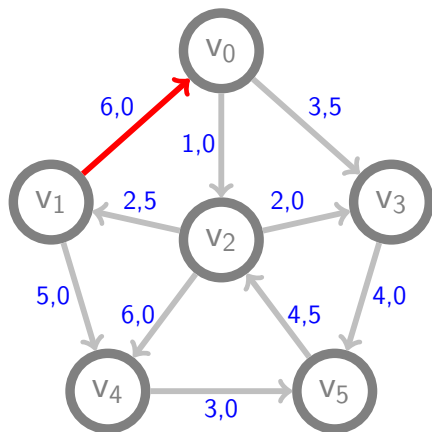
**Matriz de Distâncias**

|    | v0   | v1   | v2  | v3  | v4   | v5  |
|----|------|------|-----|-----|------|-----|
| v0 | 0,0  | 3,5  | 1,0 | 3,0 | 7,0  | 7,0 |
| v1 | 6,0  | 0,0  | 7,0 | 9,0 | 5,0  | 8,0 |
| v2 | 8,5  | 2,5  | 0,0 | 2,0 | 6,0  | 6,0 |
| v3 | 17,0 | 11,0 | 8,5 | 0,0 | 14,5 | 4,0 |
| v4 | -    | -    | -   | -   | 0,0  | 3,0 |
| v5 | 13,0 | 7,0  | 4,5 | 6,5 | 10,5 | 0,0 |

**Matriz de Predecessores**

|    | v0 | v1 | v2 | v3 | v4 | v5 |
|----|----|----|----|----|----|----|
| v0 | 0  | 2  | 0  | 2  | 2  | 3  |
| v1 | 1  | 1  | 0  | 2  | 1  | 4  |
| v2 | 1  | 2  | 2  | 2  | 2  | 3  |
| v3 | 1  | 2  | 5  | 3  | 2  | 3  |
| v4 | -1 | -1 | -1 | -1 | 4  | 4  |
| v5 | 1  | 2  | 5  | 2  | 2  | 5  |

# Algoritmo de Floyd-Warshall



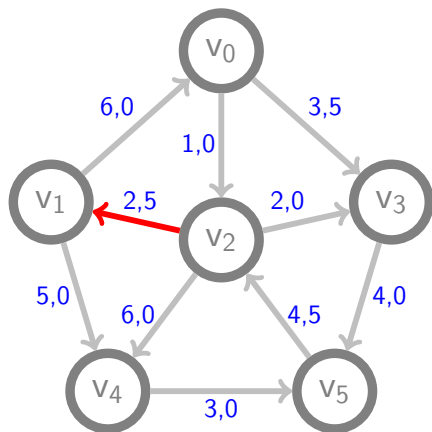
**Matriz de Distâncias**

|    | v0   | v1   | v2  | v3  | v4   | v5  |
|----|------|------|-----|-----|------|-----|
| v0 | 0,0  | 3,5  | 1,0 | 3,0 | 7,0  | 7,0 |
| v1 | 6,0  | 0,0  | 7,0 | 9,0 | 5,0  | 8,0 |
| v2 | 8,5  | 2,5  | 0,0 | 2,0 | 6,0  | 6,0 |
| v3 | 17,0 | 11,0 | 8,5 | 0,0 | 14,5 | 4,0 |
| v4 | 16,0 | -    | -   | -   | 0,0  | 3,0 |
| v5 | 13,0 | 7,0  | 4,5 | 6,5 | 10,5 | 0,0 |

**Matriz de Predecessores**

|    | v0 | v1 | v2 | v3 | v4 | v5 |
|----|----|----|----|----|----|----|
| v0 | 0  | 2  | 0  | 2  | 2  | 3  |
| v1 | 1  | 1  | 0  | 2  | 1  | 4  |
| v2 | 1  | 2  | 2  | 2  | 2  | 3  |
| v3 | 1  | 2  | 5  | 3  | 2  | 3  |
| v4 | 1  | -1 | -1 | -1 | 4  | 4  |
| v5 | 1  | 2  | 5  | 2  | 2  | 5  |

# Algoritmo de Floyd-Warshall



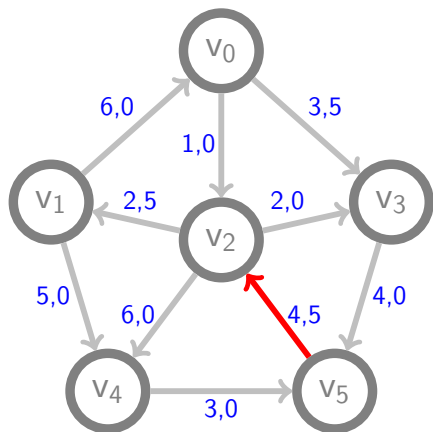
**Matriz de Distâncias**

|    | v0   | v1   | v2  | v3  | v4   | v5  |
|----|------|------|-----|-----|------|-----|
| v0 | 0,0  | 3,5  | 1,0 | 3,0 | 7,0  | 7,0 |
| v1 | 6,0  | 0,0  | 7,0 | 9,0 | 5,0  | 8,0 |
| v2 | 8,5  | 2,5  | 0,0 | 2,0 | 6,0  | 6,0 |
| v3 | 17,0 | 11,0 | 8,5 | 0,0 | 14,5 | 4,0 |
| v4 | 16,0 | 10,0 | -   | -   | 0,0  | 3,0 |
| v5 | 13,0 | 7,0  | 4,5 | 6,5 | 10,5 | 0,0 |

**Matriz de Predecessores**

|    | v0 | v1 | v2 | v3 | v4 | v5 |
|----|----|----|----|----|----|----|
| v0 | 0  | 2  | 0  | 2  | 2  | 3  |
| v1 | 1  | 1  | 0  | 2  | 1  | 4  |
| v2 | 1  | 2  | 2  | 2  | 2  | 3  |
| v3 | 1  | 2  | 5  | 3  | 2  | 3  |
| v4 | 1  | 2  | -1 | -1 | 4  | 4  |
| v5 | 1  | 2  | 5  | 2  | 2  | 5  |

# Algoritmo de Floyd-Warshall



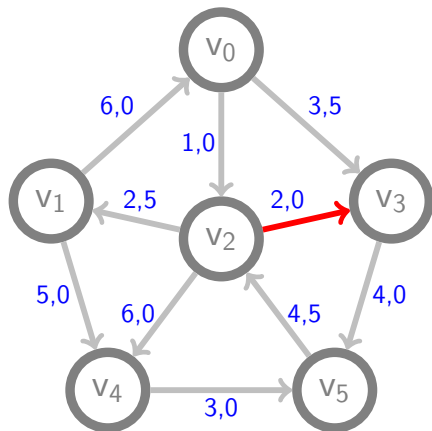
**Matriz de Distâncias**

|    | v0   | v1   | v2  | v3  | v4   | v5  |
|----|------|------|-----|-----|------|-----|
| v0 | 0,0  | 3,5  | 1,0 | 3,0 | 7,0  | 7,0 |
| v1 | 6,0  | 0,0  | 7,0 | 9,0 | 5,0  | 8,0 |
| v2 | 8,5  | 2,5  | 0,0 | 2,0 | 6,0  | 6,0 |
| v3 | 17,0 | 11,0 | 8,5 | 0,0 | 14,5 | 4,0 |
| v4 | 16,0 | 10,0 | 7,5 | -   | 0,0  | 3,0 |
| v5 | 13,0 | 7,0  | 4,5 | 6,5 | 10,5 | 0,0 |

**Matriz de Predecessores**

|    | v0 | v1 | v2 | v3 | v4 | v5 |
|----|----|----|----|----|----|----|
| v0 | 0  | 2  | 0  | 2  | 2  | 3  |
| v1 | 1  | 1  | 0  | 2  | 1  | 4  |
| v2 | 1  | 2  | 2  | 2  | 2  | 3  |
| v3 | 1  | 2  | 5  | 3  | 2  | 3  |
| v4 | 1  | 2  | 5  | -1 | 4  | 4  |
| v5 | 1  | 2  | 5  | 2  | 2  | 5  |

# Algoritmo de Floyd-Warshall



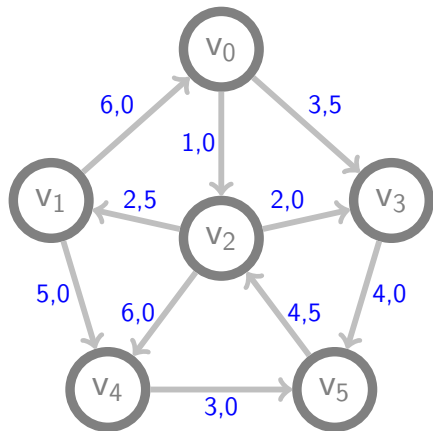
**Matriz de Distâncias**

|    | v0   | v1   | v2  | v3  | v4   | v5  |
|----|------|------|-----|-----|------|-----|
| v0 | 0,0  | 3,5  | 1,0 | 3,0 | 7,0  | 7,0 |
| v1 | 6,0  | 0,0  | 7,0 | 9,0 | 5,0  | 8,0 |
| v2 | 8,5  | 2,5  | 0,0 | 2,0 | 6,0  | 6,0 |
| v3 | 17,0 | 11,0 | 8,5 | 0,0 | 14,5 | 4,0 |
| v4 | 16,0 | 10,0 | 7,5 | 9,5 | 0,0  | 3,0 |
| v5 | 13,0 | 7,0  | 4,5 | 6,5 | 10,5 | 0,0 |

**Matriz de Predecessores**

|    | v0 | v1 | v2 | v3 | v4 | v5 |
|----|----|----|----|----|----|----|
| v0 | 0  | 2  | 0  | 2  | 2  | 3  |
| v1 | 1  | 1  | 0  | 2  | 1  | 4  |
| v2 | 1  | 2  | 2  | 2  | 2  | 3  |
| v3 | 1  | 2  | 5  | 3  | 2  | 3  |
| v4 | 1  | 2  | 5  | 2  | 4  | 4  |
| v5 | 1  | 2  | 5  | 2  | 2  | 5  |

# Algoritmo de Floyd-Warshall



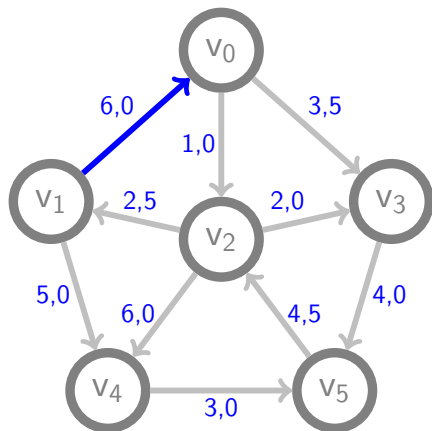
## Matriz de Distâncias

|    | v0   | v1   | v2  | v3  | v4   | v5  |
|----|------|------|-----|-----|------|-----|
| v0 | 0,0  | 3,5  | 1,0 | 3,0 | 7,0  | 7,0 |
| v1 | 6,0  | 0,0  | 7,0 | 9,0 | 5,0  | 8,0 |
| v2 | 8,5  | 2,5  | 0,0 | 2,0 | 6,0  | 6,0 |
| v3 | 17,0 | 11,0 | 8,5 | 0,0 | 14,5 | 4,0 |
| v4 | 16,0 | 10,0 | 7,5 | 9,5 | 0,0  | 3,0 |
| v5 | 13,0 | 7,0  | 4,5 | 6,5 | 10,5 | 0,0 |

## Matriz de Predecessores

|    | v0 | v1 | v2 | v3 | v4 | v5 |
|----|----|----|----|----|----|----|
| v0 | 0  | 2  | 0  | 2  | 2  | 3  |
| v1 | 1  | 1  | 0  | 2  | 1  | 4  |
| v2 | 1  | 2  | 2  | 2  | 2  | 3  |
| v3 | 1  | 2  | 5  | 3  | 2  | 3  |
| v4 | 1  | 2  | 5  | 2  | 4  | 4  |
| v5 | 1  | 2  | 5  | 2  | 2  | 5  |

# Algoritmo de Floyd-Warshall



**Matriz de Distâncias**

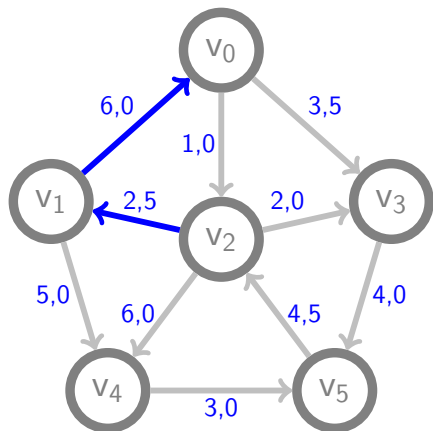
|    | v0   | v1   | v2  | v3  | v4   | v5  |
|----|------|------|-----|-----|------|-----|
| v0 | 0,0  | 3,5  | 1,0 | 3,0 | 7,0  | 7,0 |
| v1 | 6,0  | 0,0  | 7,0 | 9,0 | 5,0  | 8,0 |
| v2 | 8,5  | 2,5  | 0,0 | 2,0 | 6,0  | 6,0 |
| v3 | 17,0 | 11,0 | 8,5 | 0,0 | 14,5 | 4,0 |
| v4 | 16,0 | 10,0 | 7,5 | 9,5 | 0,0  | 3,0 |
| v5 | 13,0 | 7,0  | 4,5 | 6,5 | 10,5 | 0,0 |

**Matriz de Predecessores**

|    | v0 | v1 | v2 | v3 | v4 | v5 |
|----|----|----|----|----|----|----|
| v0 | 0  | 2  | 0  | 2  | 2  | 3  |
| v1 | 1  | 1  | 0  | 2  | 1  | 4  |
| v2 | 1  | 2  | 2  | 2  | 2  | 3  |
| v3 | 1  | 2  | 5  | 3  | 2  | 3  |
| v4 | 1  | 2  | 5  | 2  | 4  | 4  |
| v5 | 1  | 2  | 5  | 2  | 2  | 5  |



# Algoritmo de Floyd-Warshall



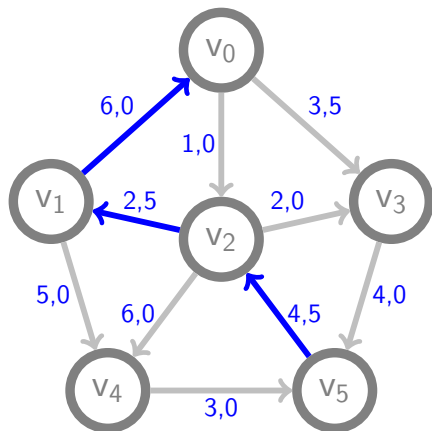
**Matriz de Distâncias**

|    | v0   | v1   | v2  | v3  | v4   | v5  |
|----|------|------|-----|-----|------|-----|
| v0 | 0,0  | 3,5  | 1,0 | 3,0 | 7,0  | 7,0 |
| v1 | 6,0  | 0,0  | 7,0 | 9,0 | 5,0  | 8,0 |
| v2 | 8,5  | 2,5  | 0,0 | 2,0 | 6,0  | 6,0 |
| v3 | 17,0 | 11,0 | 8,5 | 0,0 | 14,5 | 4,0 |
| v4 | 16,0 | 10,0 | 7,5 | 9,5 | 0,0  | 3,0 |
| v5 | 13,0 | 7,0  | 4,5 | 6,5 | 10,5 | 0,0 |

**Matriz de Predecessores**

|    | v0 | v1 | v2 | v3 | v4 | v5 |
|----|----|----|----|----|----|----|
| v0 | 0  | 2  | 0  | 2  | 2  | 3  |
| v1 | 1  | 1  | 0  | 2  | 1  | 4  |
| v2 | 1  | 2  | 2  | 2  | 2  | 3  |
| v3 | 1  | 2  | 5  | 3  | 2  | 3  |
| v4 | 1  | 2  | 5  | 2  | 4  | 4  |
| v5 | 1  | 2  | 5  | 2  | 2  | 5  |

# Algoritmo de Floyd-Warshall



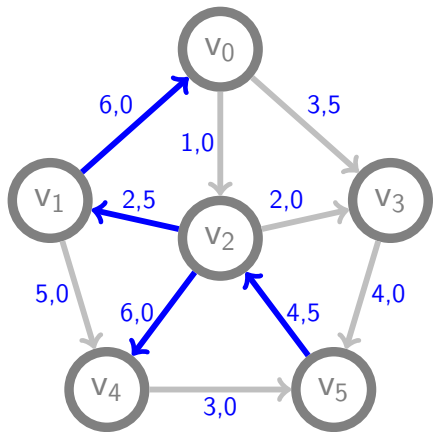
**Matriz de Distâncias**

|    | v0   | v1   | v2  | v3  | v4   | v5  |
|----|------|------|-----|-----|------|-----|
| v0 | 0,0  | 3,5  | 1,0 | 3,0 | 7,0  | 7,0 |
| v1 | 6,0  | 0,0  | 7,0 | 9,0 | 5,0  | 8,0 |
| v2 | 8,5  | 2,5  | 0,0 | 2,0 | 6,0  | 6,0 |
| v3 | 17,0 | 11,0 | 8,5 | 0,0 | 14,5 | 4,0 |
| v4 | 16,0 | 10,0 | 7,5 | 9,5 | 0,0  | 3,0 |
| v5 | 13,0 | 7,0  | 4,5 | 6,5 | 10,5 | 0,0 |

**Matriz de Predecessores**

|    | v0 | v1 | v2 | v3 | v4 | v5 |
|----|----|----|----|----|----|----|
| v0 | 0  | 2  | 0  | 2  | 2  | 3  |
| v1 | 1  | 1  | 0  | 2  | 1  | 4  |
| v2 | 1  | 2  | 2  | 2  | 2  | 3  |
| v3 | 1  | 2  | 5  | 3  | 2  | 3  |
| v4 | 1  | 2  | 5  | 2  | 4  | 4  |
| v5 | 1  | 2  | 5  | 2  | 2  | 5  |

# Algoritmo de Floyd-Warshall



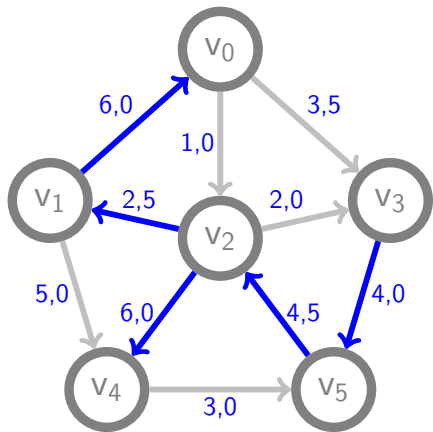
## Matriz de Distâncias

|    | v0   | v1   | v2  | v3  | v4   | v5  |
|----|------|------|-----|-----|------|-----|
| v0 | 0,0  | 3,5  | 1,0 | 3,0 | 7,0  | 7,0 |
| v1 | 6,0  | 0,0  | 7,0 | 9,0 | 5,0  | 8,0 |
| v2 | 8,5  | 2,5  | 0,0 | 2,0 | 6,0  | 6,0 |
| v3 | 17,0 | 11,0 | 8,5 | 0,0 | 14,5 | 4,0 |
| v4 | 16,0 | 10,0 | 7,5 | 9,5 | 0,0  | 3,0 |
| v5 | 13,0 | 7,0  | 4,5 | 6,5 | 10,5 | 0,0 |

## Matriz de Predecessores

|    | v0 | v1 | v2 | v3 | v4 | v5 |
|----|----|----|----|----|----|----|
| v0 | 0  | 2  | 0  | 2  | 2  | 3  |
| v1 | 1  | 1  | 0  | 2  | 1  | 4  |
| v2 | 1  | 2  | 2  | 2  | 2  | 3  |
| v3 | 1  | 2  | 5  | 3  | 2  | 3  |
| v4 | 1  | 2  | 5  | 2  | 4  | 4  |
| v5 | 1  | 2  | 5  | 2  | 2  | 5  |

# Algoritmo de Floyd-Warshall



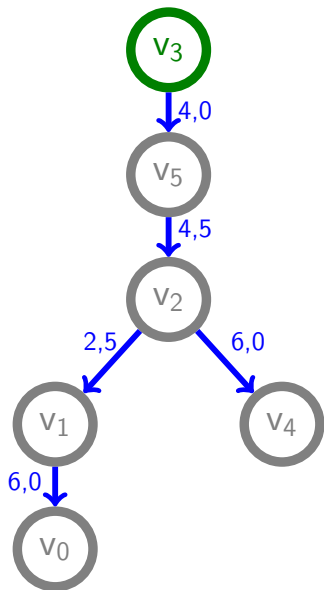
## Matriz de Distâncias

|    | v0   | v1   | v2  | v3  | v4   | v5  |
|----|------|------|-----|-----|------|-----|
| v0 | 0,0  | 3,5  | 1,0 | 3,0 | 7,0  | 7,0 |
| v1 | 6,0  | 0,0  | 7,0 | 9,0 | 5,0  | 8,0 |
| v2 | 8,5  | 2,5  | 0,0 | 2,0 | 6,0  | 6,0 |
| v3 | 17,0 | 11,0 | 8,5 | 0,0 | 14,5 | 4,0 |
| v4 | 16,0 | 10,0 | 7,5 | 9,5 | 0,0  | 3,0 |
| v5 | 13,0 | 7,0  | 4,5 | 6,5 | 10,5 | 0,0 |

## Matriz de Predecessores

|    | v0 | v1 | v2 | v3 | v4 | v5 |
|----|----|----|----|----|----|----|
| v0 | 0  | 2  | 0  | 2  | 2  | 3  |
| v1 | 1  | 1  | 0  | 2  | 1  | 4  |
| v2 | 1  | 2  | 2  | 2  | 2  | 3  |
| v3 | 1  | 2  | 5  | 3  | 2  | 3  |
| v4 | 1  | 2  | 5  | 2  | 4  | 4  |
| v5 | 1  | 2  | 5  | 2  | 2  | 5  |

# Algoritmo de Floyd-Warshall



**Matriz de Distâncias**

|    | v0   | v1   | v2  | v3  | v4   | v5  |
|----|------|------|-----|-----|------|-----|
| v0 | 0,0  | 3,5  | 1,0 | 3,0 | 7,0  | 7,0 |
| v1 | 6,0  | 0,0  | 7,0 | 9,0 | 5,0  | 8,0 |
| v2 | 8,5  | 2,5  | 0,0 | 2,0 | 6,0  | 6,0 |
| v3 | 17,0 | 11,0 | 8,5 | 0,0 | 14,5 | 4,0 |
| v4 | 16,0 | 10,0 | 7,5 | 9,5 | 0,0  | 3,0 |
| v5 | 13,0 | 7,0  | 4,5 | 6,5 | 10,5 | 0,0 |

**Matriz de Predecessores**

|    | v0 | v1 | v2 | v3 | v4 | v5 |
|----|----|----|----|----|----|----|
| v0 | 0  | 2  | 0  | 2  | 2  | 3  |
| v1 | 1  | 1  | 0  | 2  | 1  | 4  |
| v2 | 1  | 2  | 2  | 2  | 2  | 3  |
| v3 | 1  | 2  | 5  | 3  | 2  | 3  |
| v4 | 1  | 2  | 5  | 2  | 4  | 4  |
| v5 | 1  | 2  | 5  | 2  | 2  | 5  |

# Algoritmo de Floyd-Warshall - inicialização

```
void FloydWarshall(Grafo* g, Peso** dist, int** pred){
    int i, j, k, n;
    n = g->numVertices;
    for(i=0; i<n; i++){
        for(j=0; j<n; j++){
            if(g->matriz[i][j] != ARESTA_INVALIDA){
                dist[i][j] = g->matriz[i][j];
                pred[i][j] = i;
            }else{
                dist[i][j] = INFINITO;
                pred[i][j] = -1;
            }
        }
        dist[i][i] = 0;
        pred[i][i] = i;
    } ...
}
```

# Algoritmo de Floyd-Warshall

```
for(k=0;k<n;k++)  
    for(i=0;i<n;i++)  
        for(j=0;j<n;j++)  
            if(dist[i][j] > dist[i][k] + dist[k][j]){  
                dist[i][j] = dist[i][k] + dist[k][j];  
                pred[i][j] = pred[k][j];  
            }  
}
```

# Algoritmo de Floyd-Warshall

```
for(k=0;k<n;k++)  
    for(i=0;i<n;i++)  
        for(j=0;j<n;j++)  
            if(dist[i][j] > dist[i][k] + dist[k][j]){  
                dist[i][j] = dist[i][k] + dist[k][j];  
                pred[i][j] = pred[k][j];  
            }  
}
```

$O(V^3)$



# Algoritmos e Estruturas de Dados II

## Aula 12 - Caminhos de Custo Mínimo - Floyd-Warshall

Prof. Luciano A. Digiampietri  
digiampietri@usp.br  
@digiampietri