

# ESTRUTURA DE DADOS

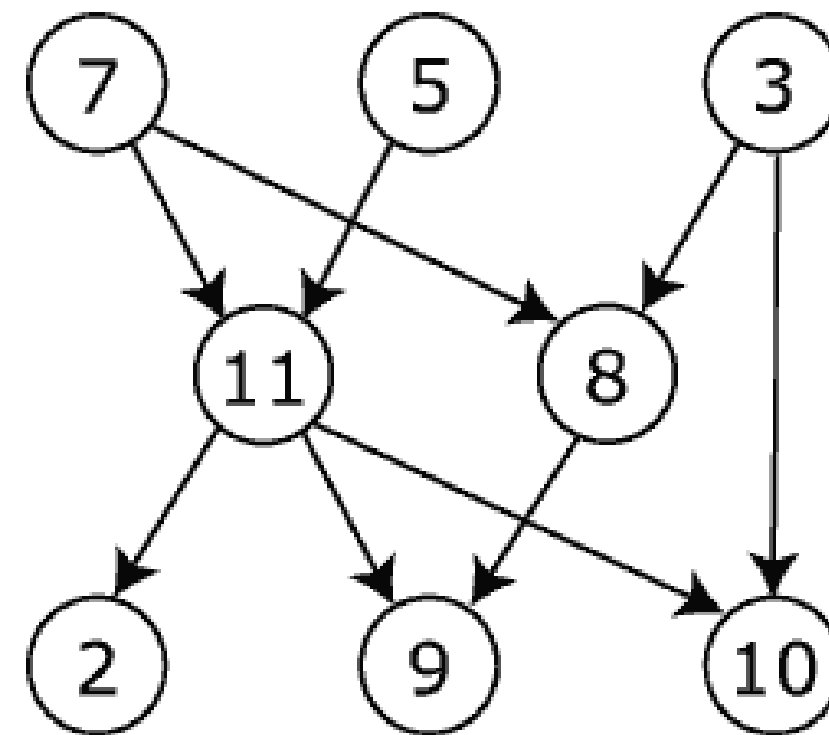
MÓDULO 07 | AULA 07



Grafos são estruturas de dados que consistem em um conjunto de vértices (ou nós) e um conjunto de arestas que conectam pares desses vértices. Eles são usados para modelar relações entre objetos.

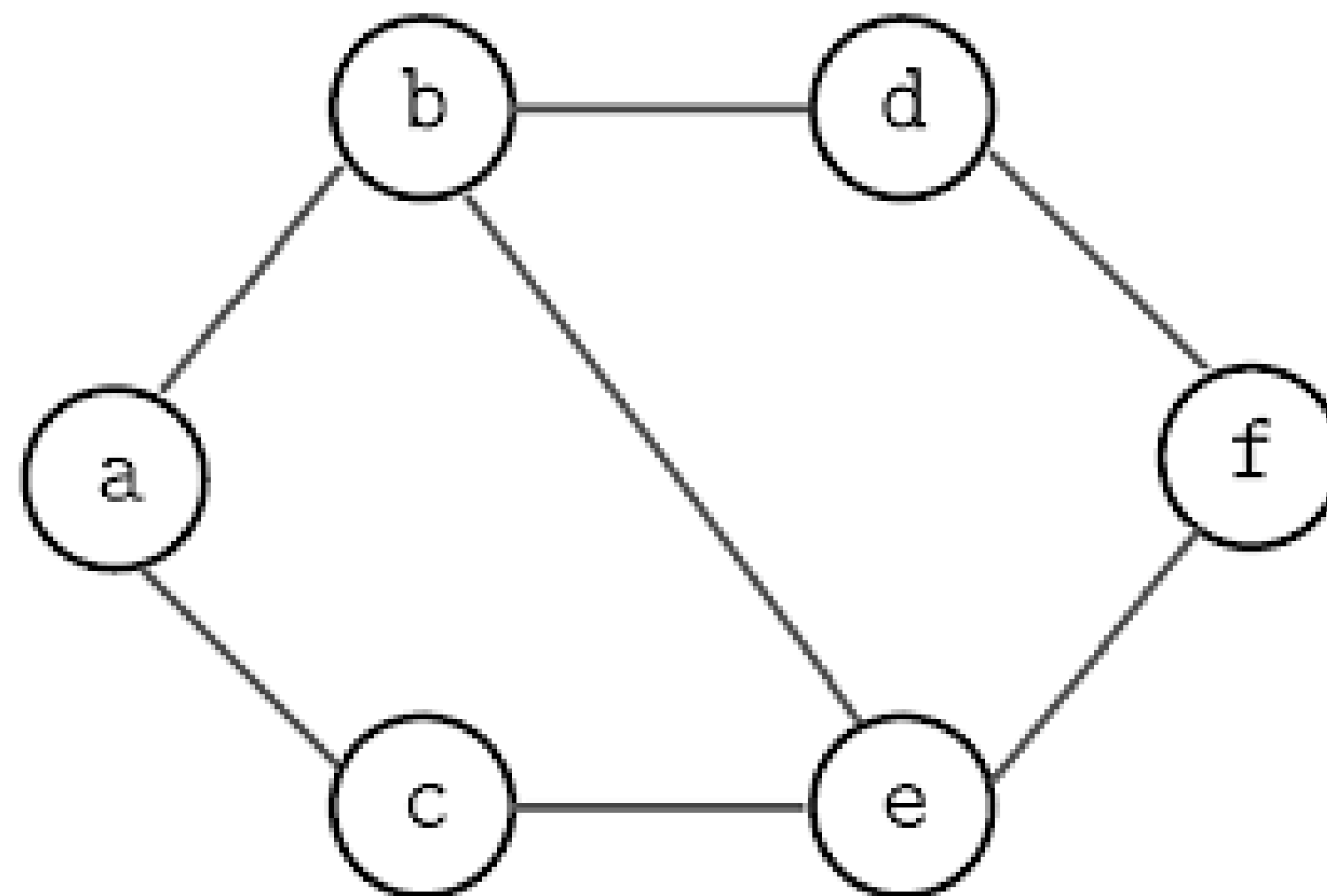
## TIPOS DE GRAFOS

**Grafos Dirigidos (Digrafos):** Grafos onde as arestas têm uma direção, indicando uma relação unidirecional entre dois vértices. Representado por setas. Úteis em cenários onde a relação entre os objetos não é simétrica, como sistemas de rotas de voo, onde uma rota pode existir de A para B, mas não necessariamente de B para A.



## TIPOS DE GRAFOS

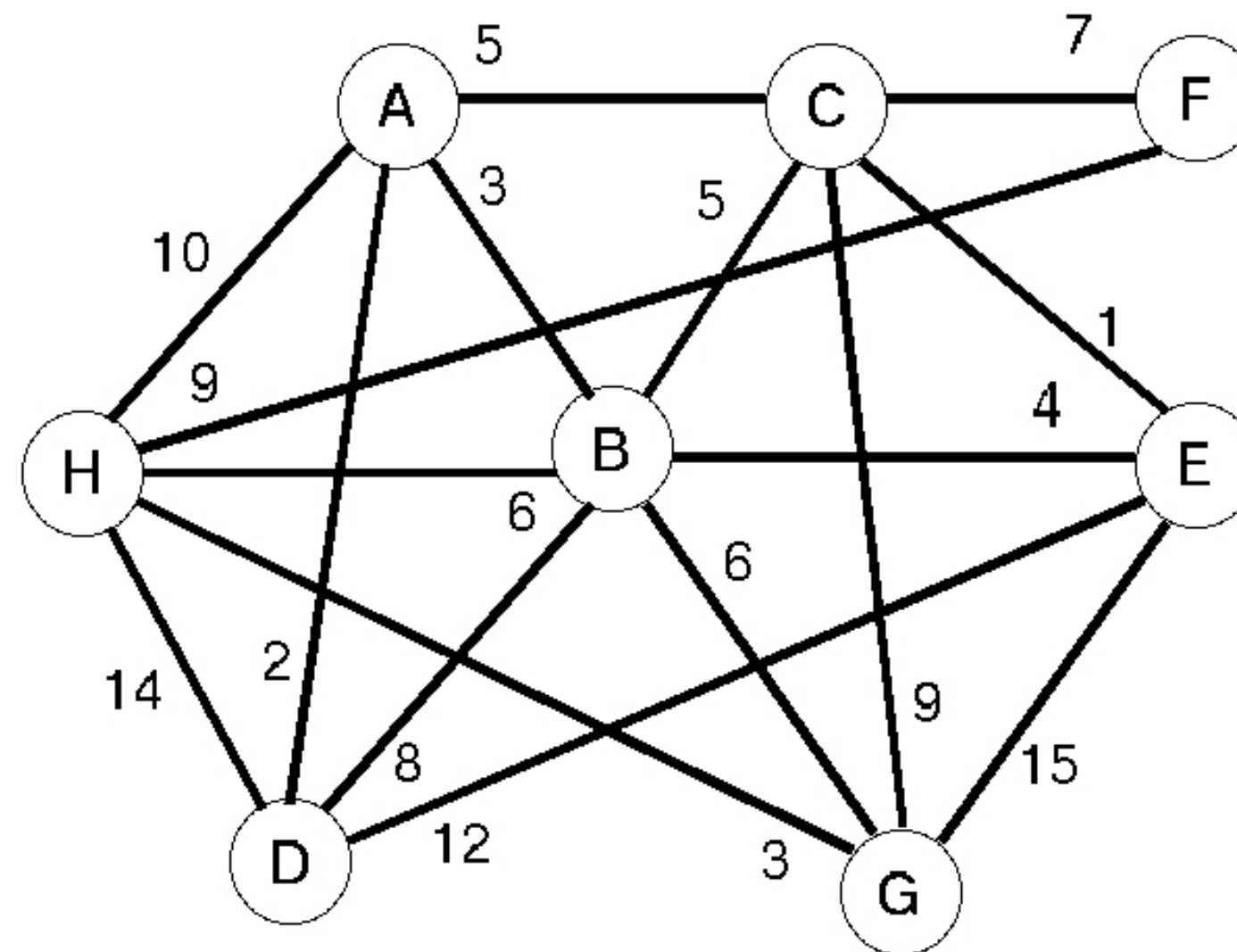
**Grafos Não Dirigidos:** Grafos onde as arestas não têm direção, indicando uma relação bidirecional entre dois vértices. Representado por linhas. Adequados para modelar relações simétricas, como redes sociais onde a amizade é mútua, ou redes de estradas onde o caminho é acessível em ambas as direções.





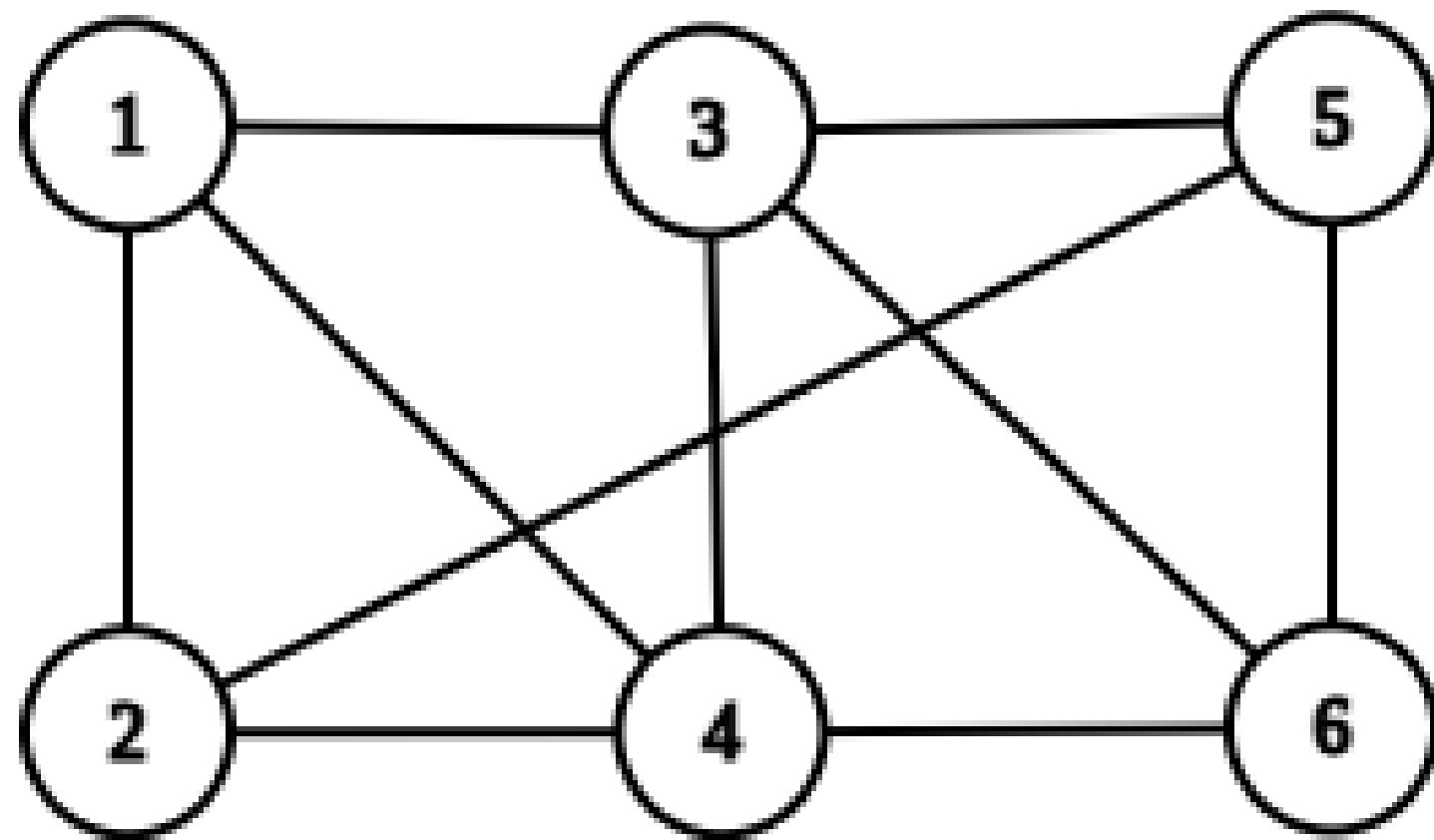
## TIPOS DE GRAFOS

**Grafos Ponderados:** Grafos cujas arestas possuem um peso ou custo associado. Esse peso pode representar distância, tempo, custo, ou qualquer outra métrica que quantifique a relação entre os vértices. Usados para modelar problemas de otimização, como encontrar o caminho mais curto ou mais barato, em mapas de rotas ou redes de telecomunicações.



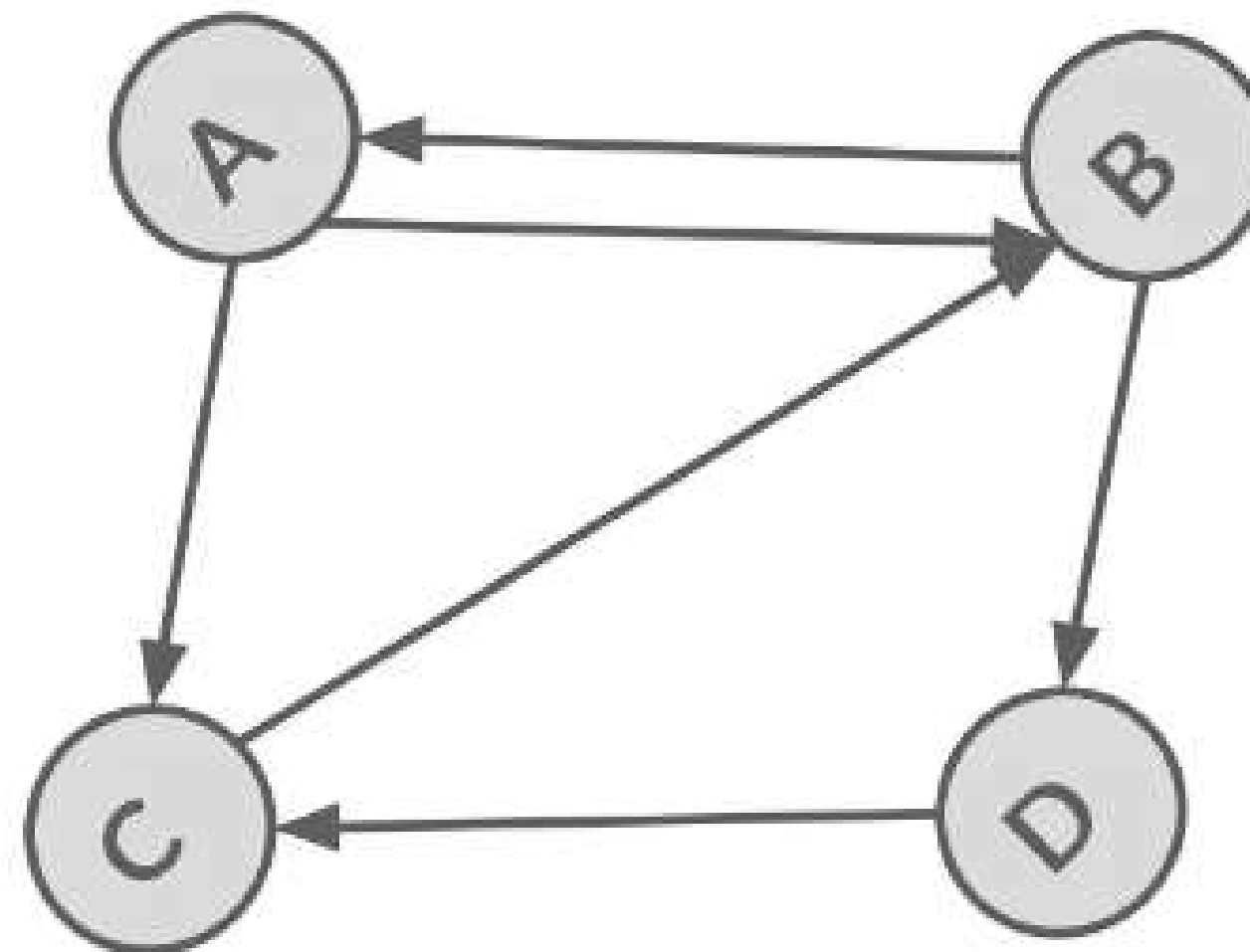
## TIPOS DE GRAFOS

**Grafos Não Ponderados:** Grafos onde as arestas não têm peso. A conexão entre dois vértices é considerada igual a qualquer outra conexão na rede. Utilizados quando apenas a conexão entre os vértices é importante, e não a magnitude dessa conexão, como em grafos de amizade simples ou redes de colaboração.



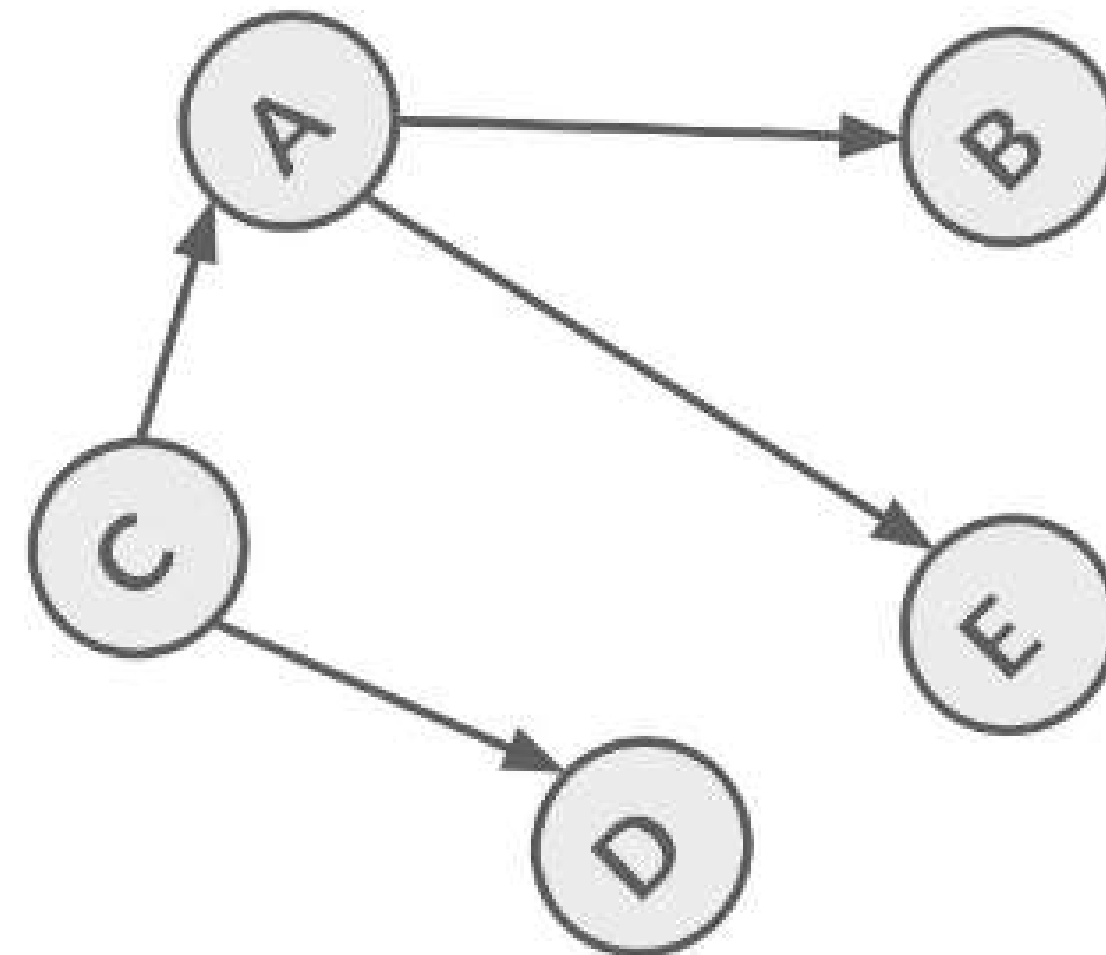
## TIPOS DE GRAFOS

**Grafos Cíclicos:** Grafos que contêm pelo menos um ciclo, ou seja, um caminho fechado onde o ponto inicial e final são o mesmo vértice. Comuns em modelagem de redes que permitem loops ou circuitos, como ciclos de negócios, circuitos elétricos, ou até mesmo modelagem de dependências em sistemas.



## TIPOS DE GRAFOS

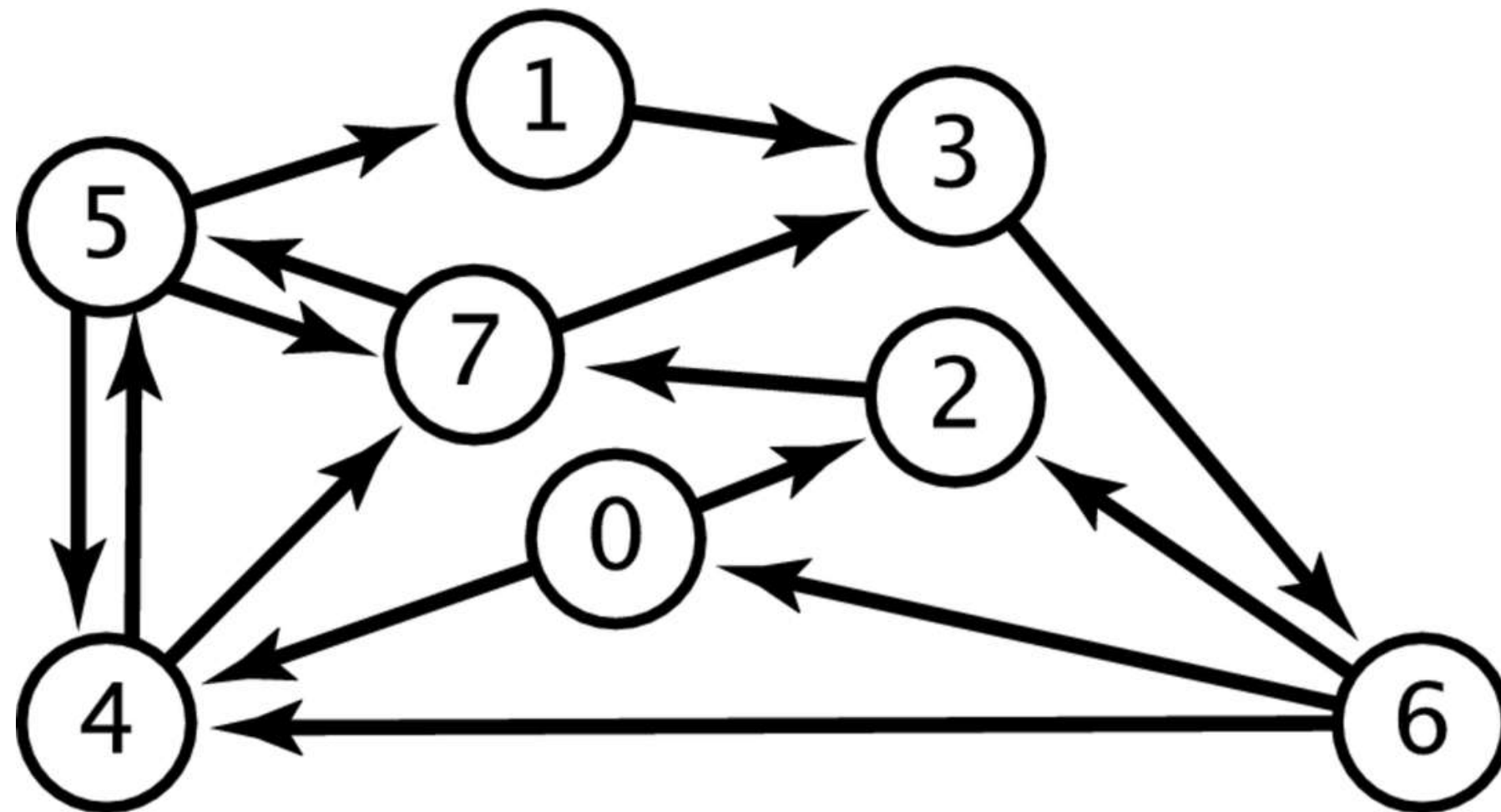
**Grafos Acíclicos Não Dirigidos:** Grafos sem ciclos, onde não é possível começar de um vértice e retornar a ele seguindo um caminho contínuo. Usados em situações onde ciclos não fazem sentido ou são proibidos, como em árvores genealógicas.





## TIPOS DE GRAFOS

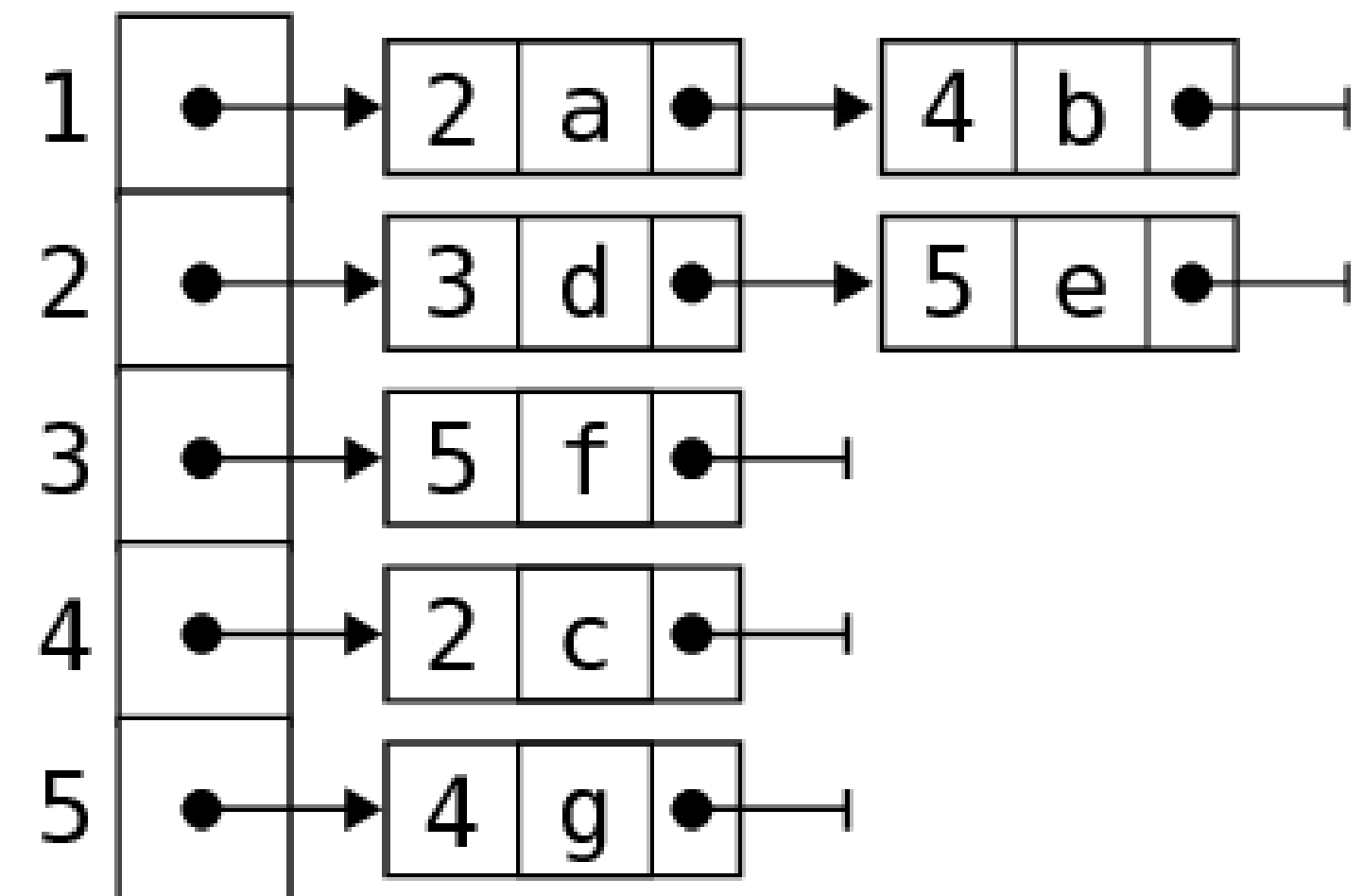
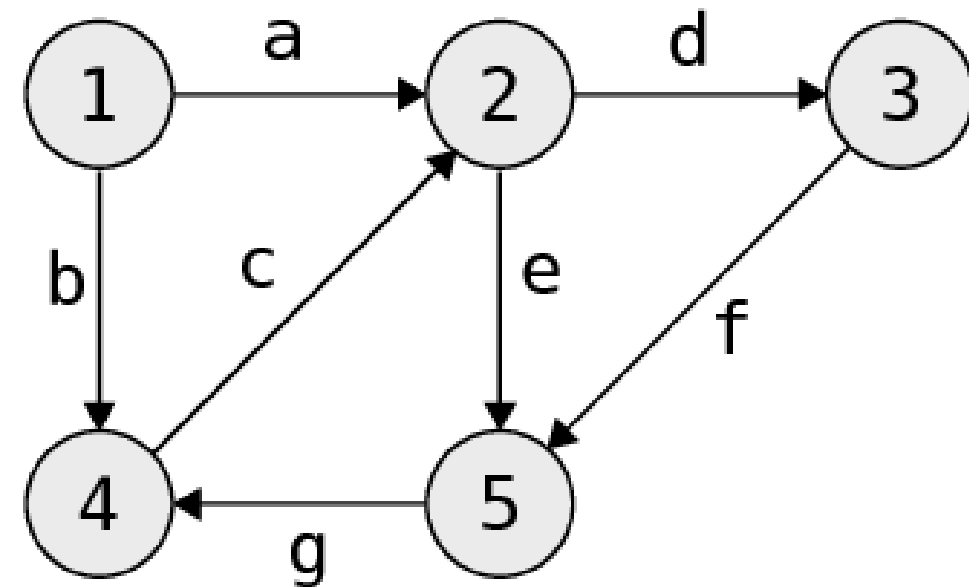
**Grafos Acíclicos Dirigidos (DAG - Directed Acyclic Graphs):** Uma categoria especial de grafos dirigidos sem ciclos. Extremamente úteis em programação, para modelar processos ou sistemas que têm uma ordem clara de precedência, como planilhas, compilação de programas, e em muitos algoritmos de otimização e processamento de dados.





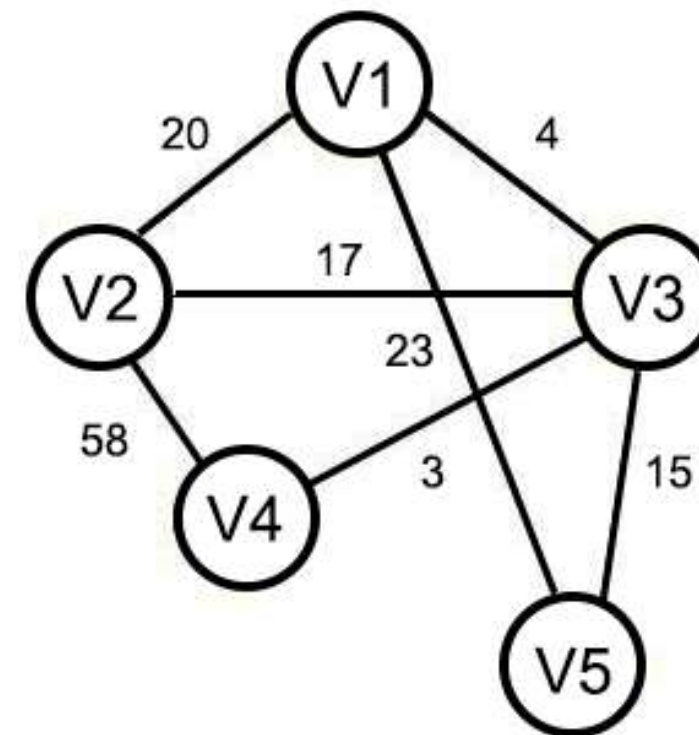
## REPRESENTAÇÃO DE GRAFOS

**Lista de Adjacências:** Discussão sobre outra forma de representar grafos, mais eficiente em termos de espaço para grafos esparsos, onde cada vértice mantém uma lista dos vértices aos quais está conectado.



## REPRESENTAÇÃO DE GRAFOS

**Matriz de Adjacências:** Explicação de como os grafos podem ser representados por matrizes, onde as linhas e colunas representam os vértices e as entradas indicam a presença de uma aresta.

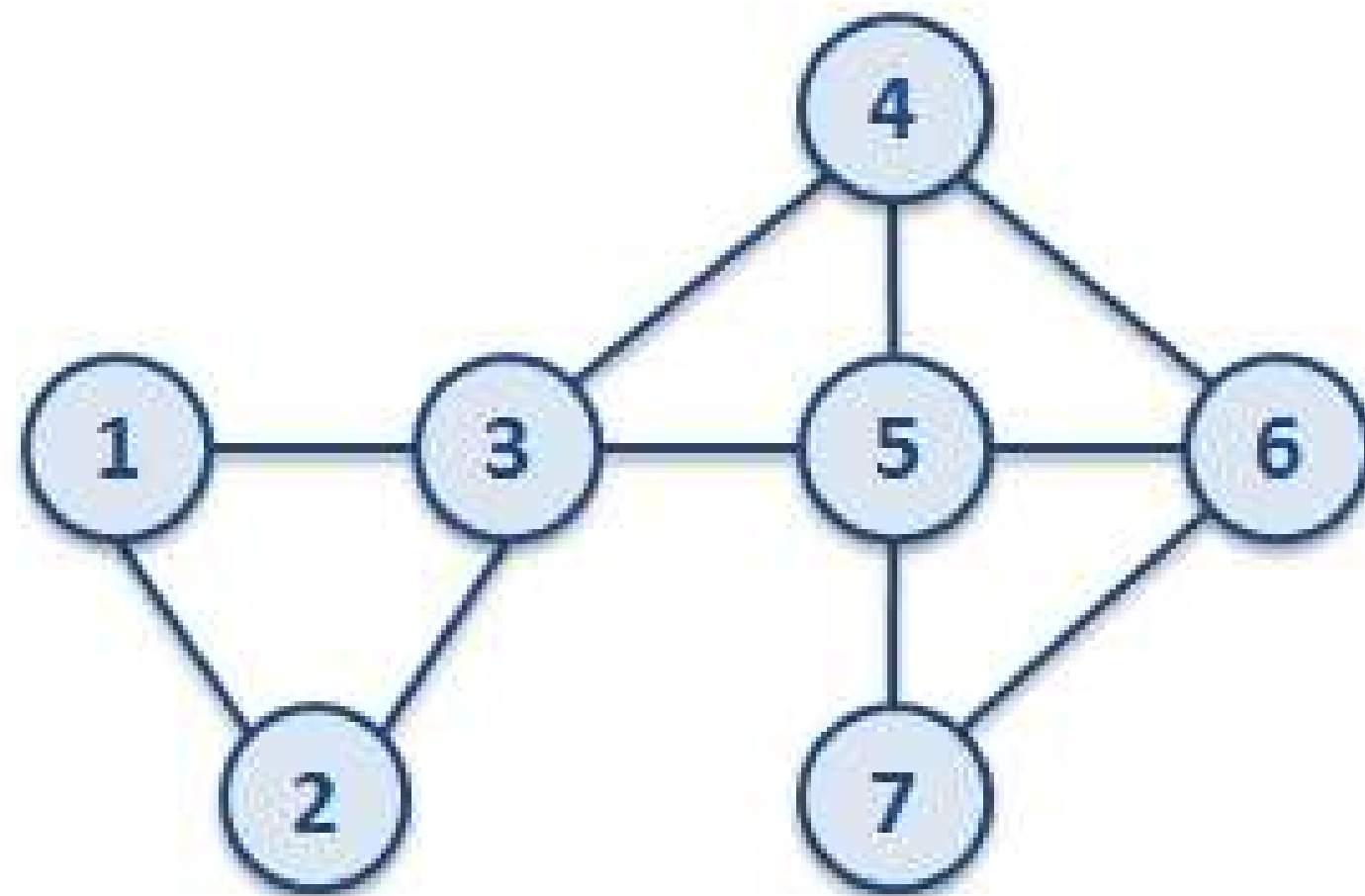


\*zero é um valor escolhido em código para considerar não ter nenhuma ligação entre os dois grafos, porém se seu grafos tiver zero como um valor valido deve se escolher outro valor.

	V1	V2	V3	V4	V5
V1	0*	20	4	0*	23
V2	20	0*	17	58	0*
V3	4	17	3	3	15
V4	0*	58	3	0*	0*
V5	23	0*	15	0*	0*

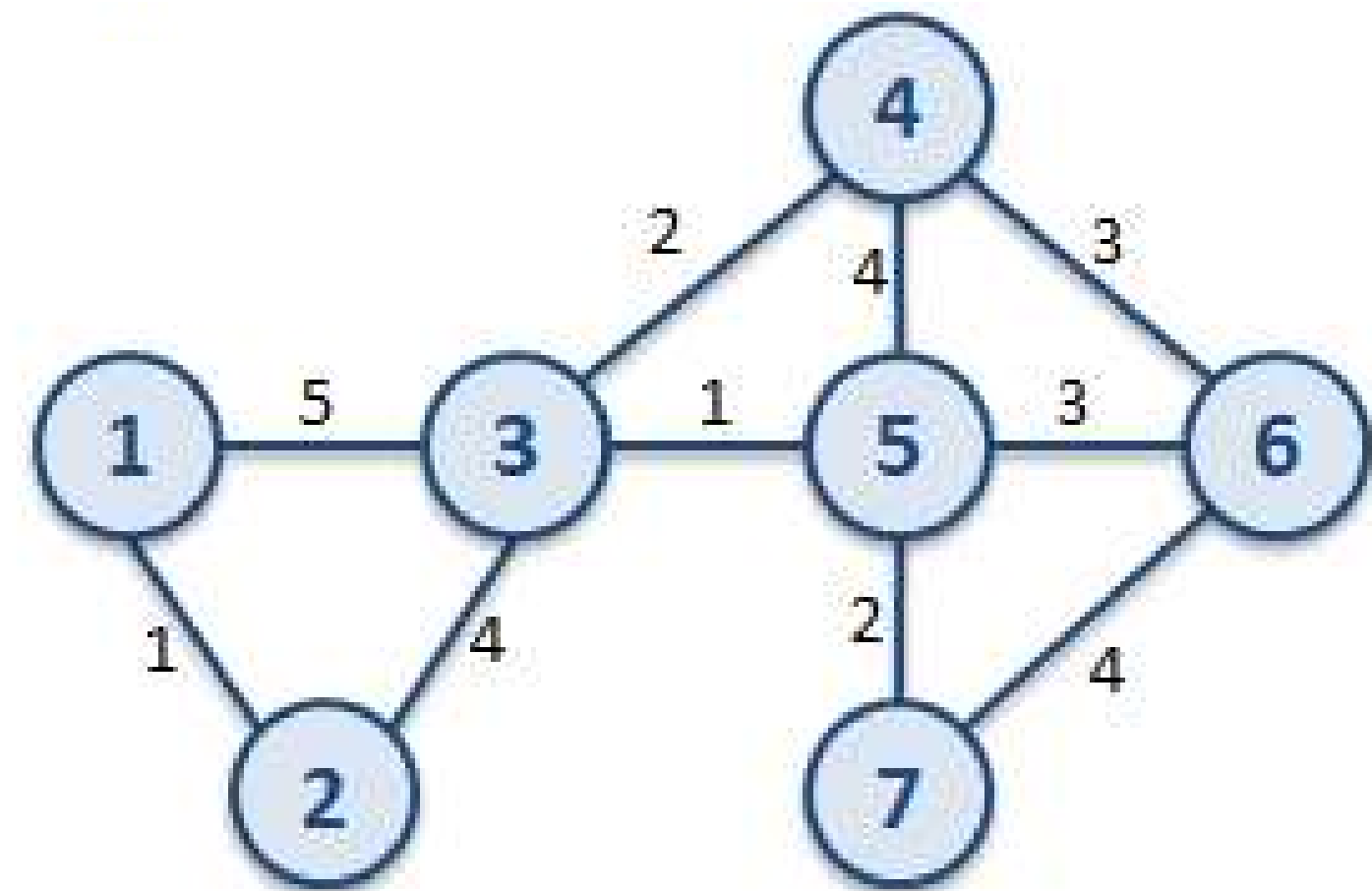
Imagem: Paulo Martins

# MATRIZ DE ADJACÊNCIA



$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

NÃO PONDERADOS



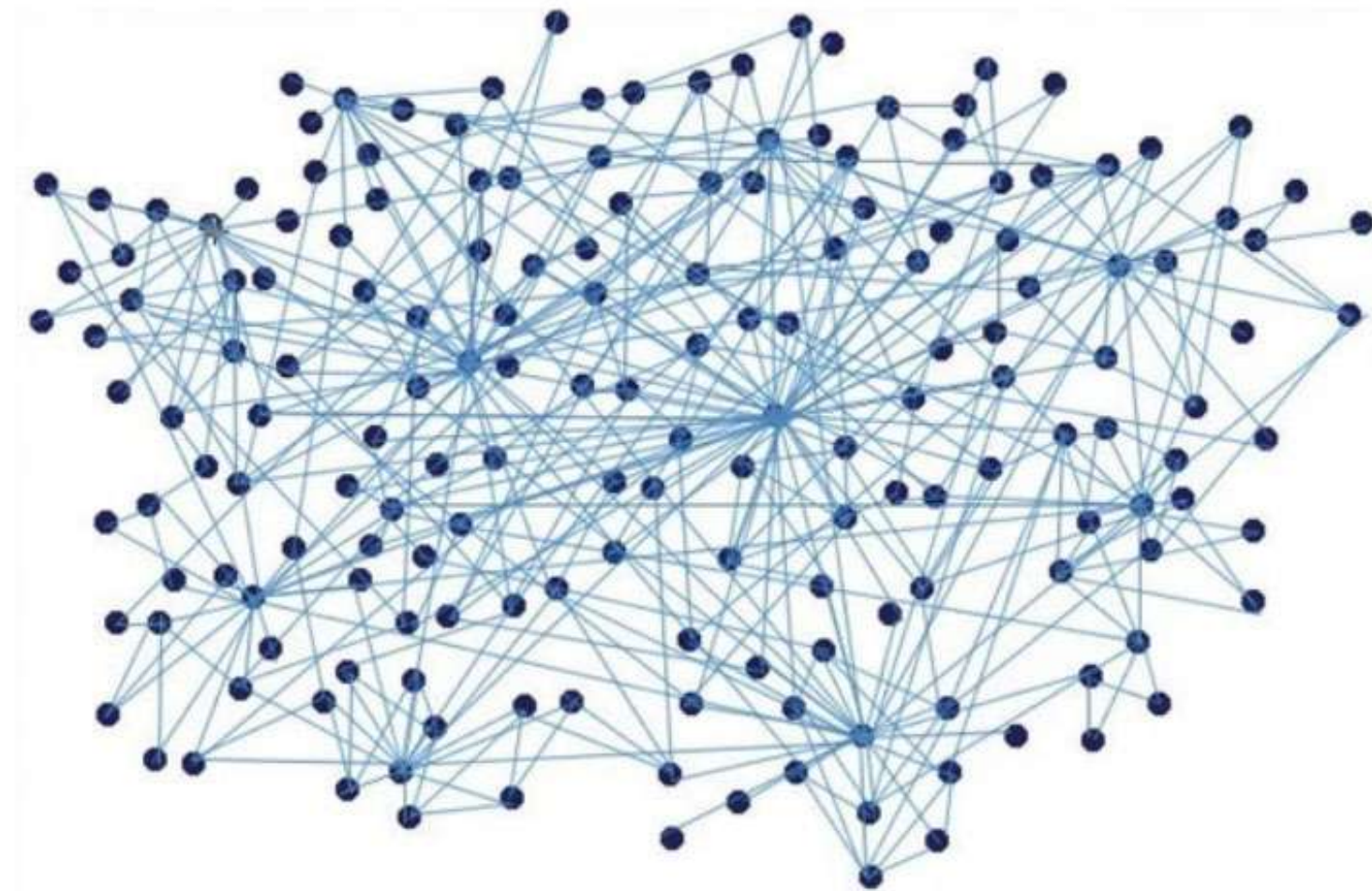
$$A = \begin{bmatrix} 0 & 1 & 5 & 0 & 0 & 0 & 0 \\ 1 & 0 & 4 & 0 & 0 & 0 & 0 \\ 5 & 4 & 0 & 2 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 & 4 & 3 & 0 \\ 0 & 0 & 1 & 4 & 0 & 3 & 2 \\ 0 & 0 & 0 & 3 & 3 & 0 & 4 \\ 0 & 0 & 0 & 0 & 2 & 4 & 0 \end{bmatrix}$$

PONDERADOS

## PERCURSO EM GRAFOS

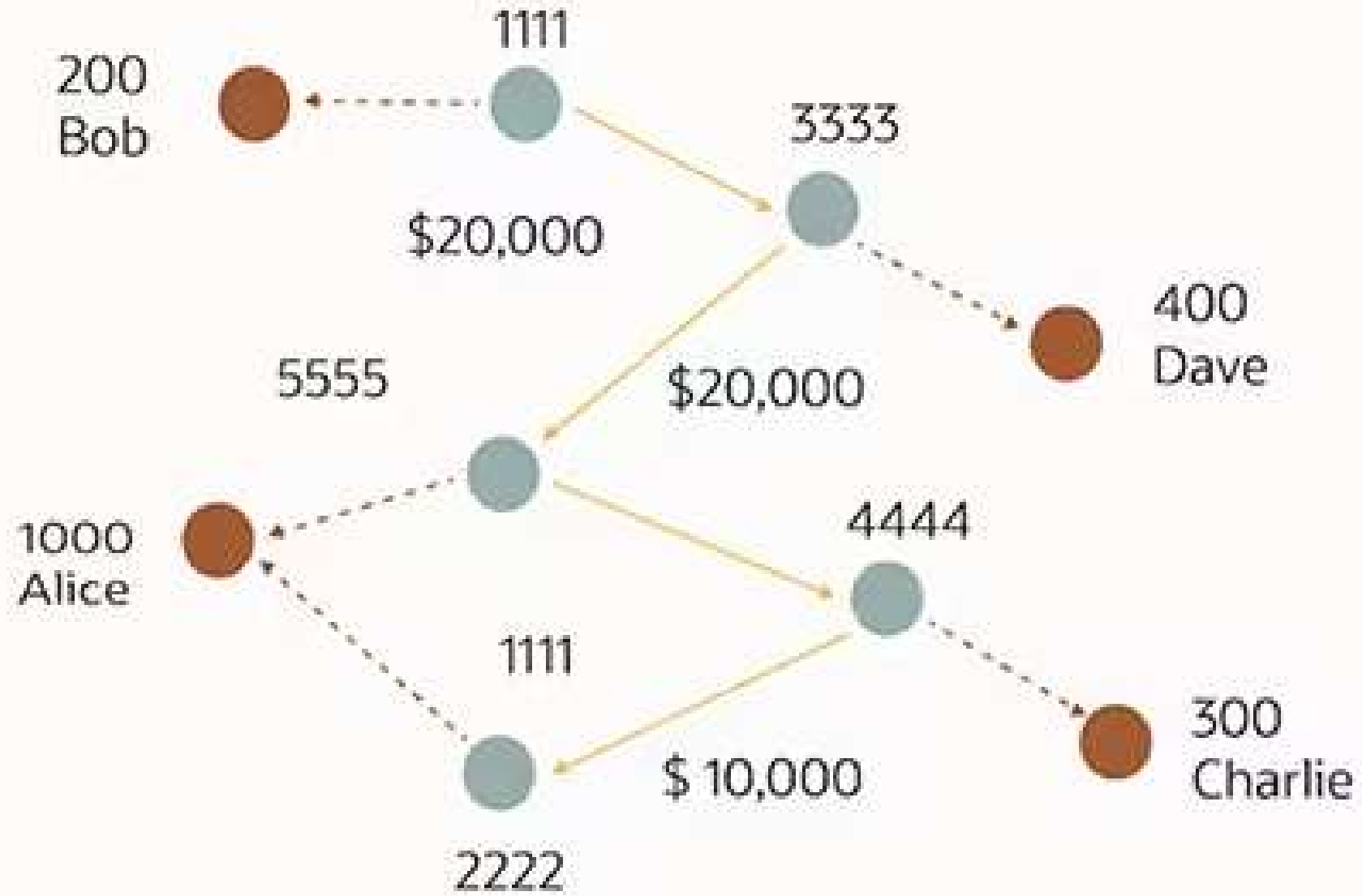
**Busca em Largura (BFS):** Descrição do algoritmo de busca em largura, que explora todos os vértices em um determinado nível antes de mover-se para os vértices no próximo nível.

**Busca em Profundidade (DFS):** Explicação do algoritmo de busca em profundidade, que explora tão profundamente quanto possível ao longo de cada ramo antes de retroceder.





## Caso de uso do banco de dados de grafos: lavagem de dinheiro



```
PATH p := () -[with SAME_INFO=true]-> ()
SELECT c1, a1, en, a2, c2
WHERE
  (c1:CUSTOMER) -> (a1:ACCOUNT) -> (en:ENTITY) -> (a2:ACCOUNT) <- (c2:CUSTOMER),
  (c1) -/:p*/-> (c2)
```





