ESTRUTURAS DE DADOS

Grafos

Roteiro

- Aplicações da Estrutura
- Conceitos Básicos
- Formas de Representação
- Busca em Largura e Profundidade

Aplicações da Estrutura

As árvores vistas até agora são muito úteis para representar hierarquias em que um nó pai possui um ou mais filhos.

Quando removemos a restrição de que um filho pode ter apenas um pai, o poder de representação das árvores fica insuficiente.

Um grafo é capaz de representar esses casos.

Grafos podem nos ajudar a representar:

- Cidades conectadas por estradas.
- Pessoas conectadas por relações de amizade.
- Páginas web conectadas por links.
- Átomos conectados por ligações químicas.
- Filmes conectados por preferência dos usuários.
- Espécies conectadas filogeneticamente.
- Animais conectados por relações ecológicas.
- Estações conectadas por linhas de metrô.
- E assim por diante...

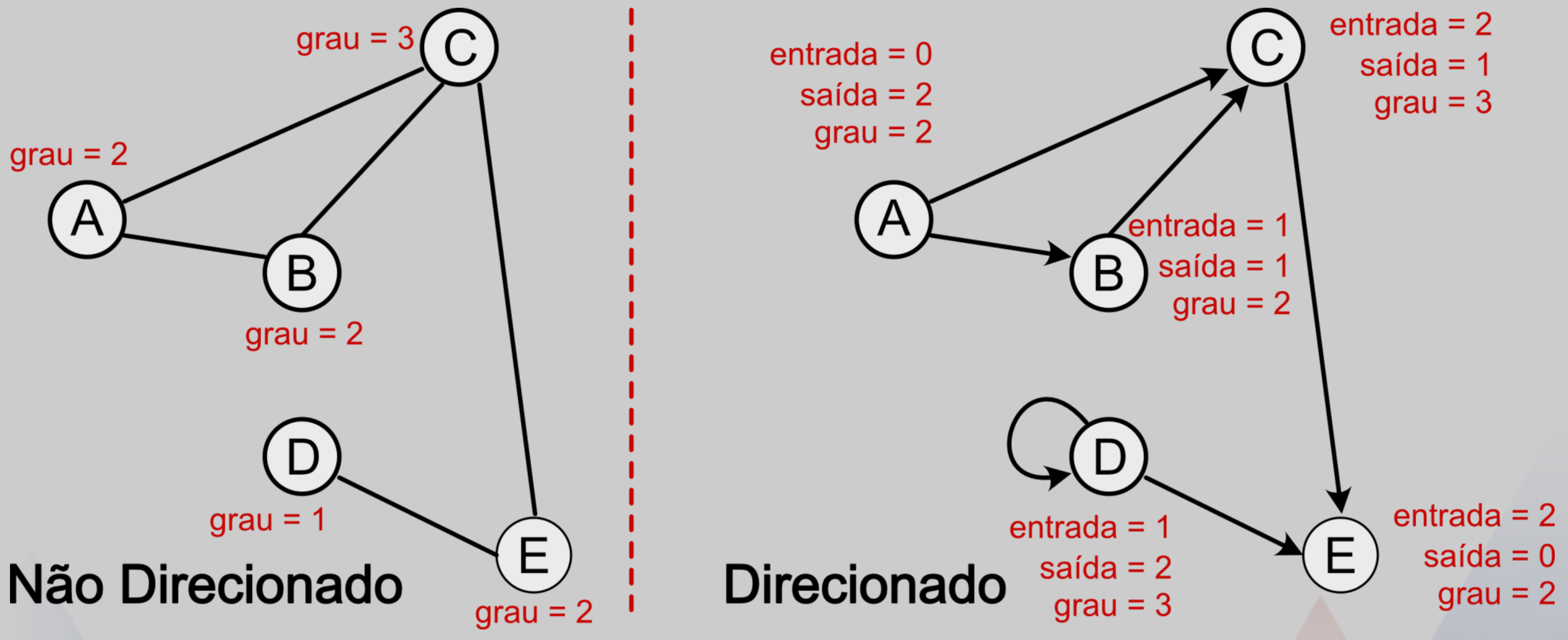
Conceitos Básicos

Um grafo $G = (V, E)$ é uma estrutura formada por um conjunto $V = (v_1, v_2, ..., v_n)$ de vértices e um conjunto $E = (e_1, e_2, ..., e_m)$ de arestas, onde cada aresta é um par de vértices.

- **Um grafo é dito não direcionado se as relações representadas pelas arestas não têm sentido, ou seja, arestas podem ser seguidas em qualquer direção: $e_i = \{v_j, v_k\}$, onde $v_j, v_k \in E$.**
- **Um grafo é dito direcionado se as arestas são pares ordenados de vértices, saindo de um em direção ao outro: $e_i = (v_j, v_k)$, onde $v_j, v_k \in E$.**

Grau

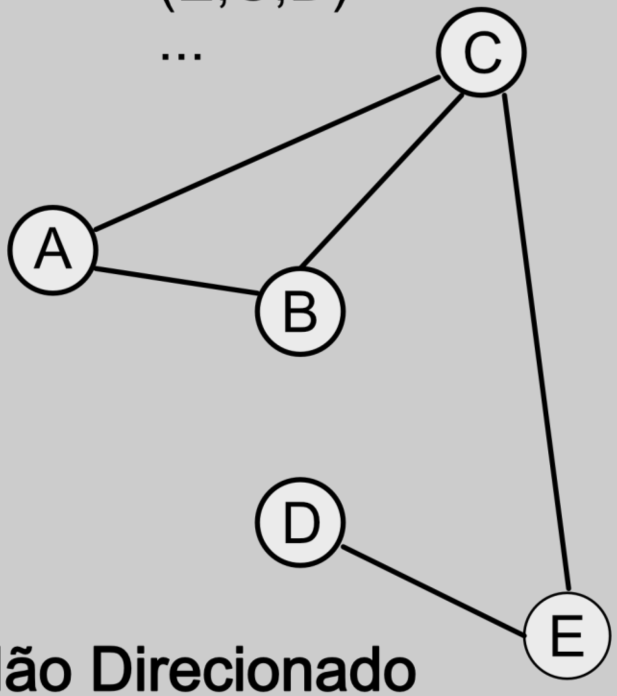
- Em **grafos não direcionados**, o grau de um vértice é o número de arestas que incidem nele.
  - Note que self-loops não são permitidos.
- Em **grafos direcionados**, o grau é o número de arestas que saem do vértice (grau de saída) mais o número de arestas que chegam (grau de entrada).



Caminho

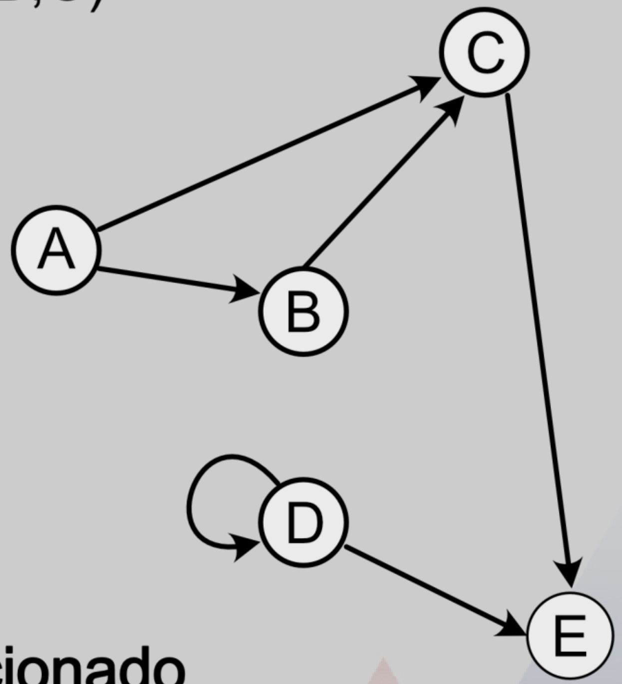
- **Um caminho de v_1 a v_n é uma sequência de vértices $C = (v_1, v_2, \ldots v_n)$, tal que $(v_i, v_{i+1}) \in E$ para todo i no intervalo de 1 a $n-1$.**

(A,C,E,D)
(C,A,B)
(E,C,D)
...



Não Direcionado

(A,C,E)
(D,D,E)
(A,B,C)
...



Direcionado

Formas de Representação

Tradicionalmente, um grafo é representado por uma matriz de adjacências ou por uma coleção de listas de adjacência.

Na definição das representações, vamos assumir que grafos não ponderados (sem pesos nas arestas), depois ilustraremos as representações com grafos ponderados.

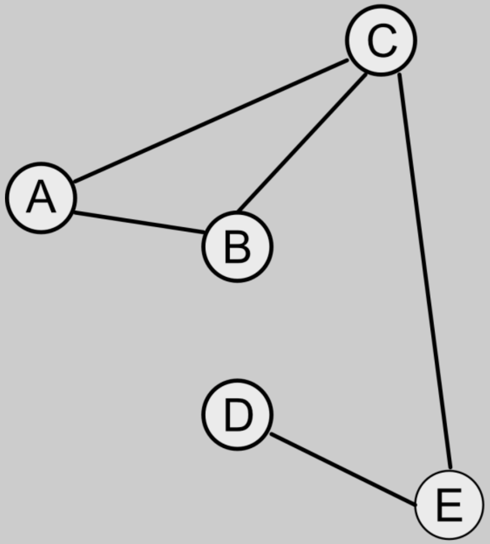
Seja $G = (V, E)$ um grafo com n vértices, uma matriz de adjacências A é uma matriz $n \times n$ tal que:

- $A[i, j] = 1$, se houver uma aresta indo do vértice i para o vértice j.
- $A[i,j] = 0$, caso contrário.

Uma lista de adjacências de um grafo com n vértices consiste de um arranjo de n listas encadeadas, uma para cada vértice no grafo.

- Para cada vértice u, a lista de u contém todos os vizinhos de u.

Grafo Não Direcionado



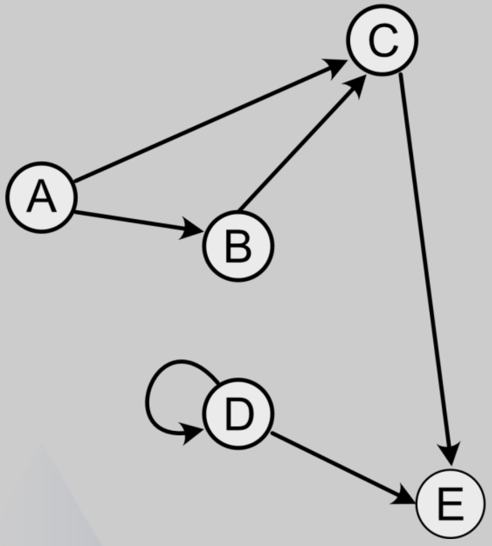
Matriz de Adjacências

	A	B	C	D	E
A	0	1	1	0	0
B	1	0	1	0	0
C	1	1	0	0	1
D	0	0	0	0	1
E	0	0	1	1	0

Listas de Adjacências

A → B → C →
B → C → A →
C → E → A → B →
D → E →
E → D → C →

Grafo Direcionado



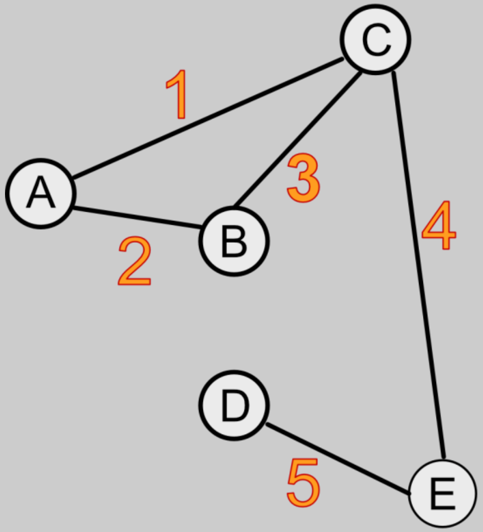
Matriz de Adjacências

	A	B	C	D	E
A	0	1	1	0	0
B	0	0	1	0	0
C	0	0	0	0	1
D	0	0	0	1	1
E	0	0	0	0	0

Listas de Adjacências

A → B → C →
B → C →
C → E →
D → E → D →
E →

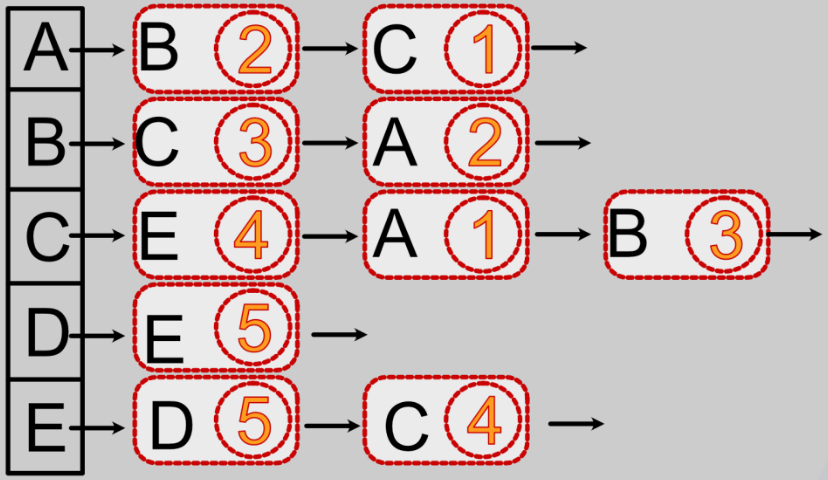
Grafo Não Direcionado



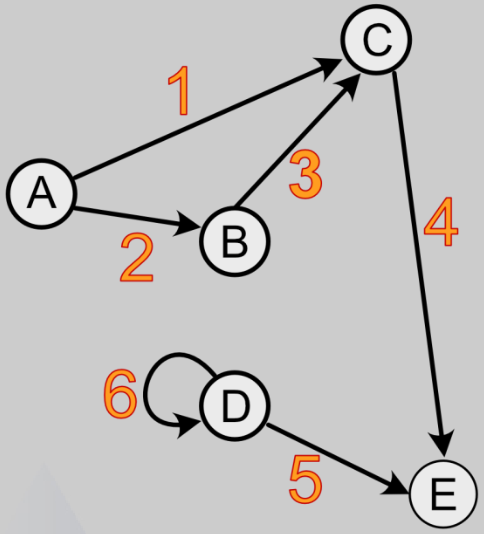
Matriz de Adjacências

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

Listas de Adjacências



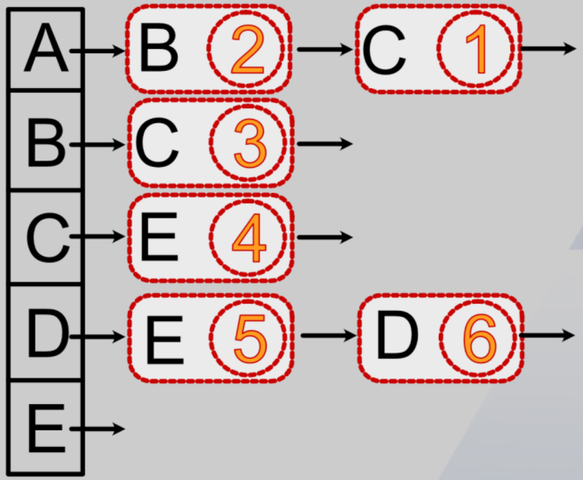
Grafo Direcionado



Matriz de Adjacências

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

Listas de Adjacências



Algumas vantagens e desvantagens:

1. Com matrizes de adjacências, alocaremos espaço para a matriz inteira no momento da declaração da matriz, antes de sabermos o número de vértices e o número de arestas.
2. Se o grafo for denso (muitas arestas em relação ao número de vértices), a lista de adjacências ocupa espaço demais.
3. As matrizes de adjacências ocupam o mesmo espaço em grafos esparsos e densos.
4. Buscas são melhores com listas de adjacências, pois já temos os adjacentes de um nó.
5. Testar se existe uma aresta entre dois vértices, dados os índices, é melhor com matrizes de adjacência.
6. Encontrar os predecessores de um nó é melhor com matrizes de adjacência, pois basta olhar a coluna do nó na matriz.
   - Precisaríamos varrer todas as listas se usássemos listas de adjacência.

A forma de representação que usaremos depende de vários fatores

1. As Matrizes de adjacência ocupam o mesmo espaço em grafos esparsos e densos.
2. Buscas são melhores com listas de adjacências, pois já temos os adjacentes de um nó.
3. Testar se existe uma aresta entre dois vértices dados é melhor com matrizes de adjacência.
4. Encontrar os predecessores de um nó é melhor com matrizes de adjacência, pois basta olhar a coluna do nó. Precisaríamos varrer todas as listas se usássemos listas de adjacência.

ESTRUTURAS DE DADOS

Grafos