

Grafos – Definições Básicas e Representação

Estrutura de Dados Avançada — QXD0015



UNIVERSIDADE
FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Prof. Atílio Gomes Luiz
gomes.atilio@ufc.br

Universidade Federal do Ceará

1º semestre/2024

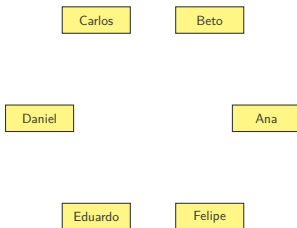


Redes Sociais

Como representar amizades em uma rede social?

Redes Sociais

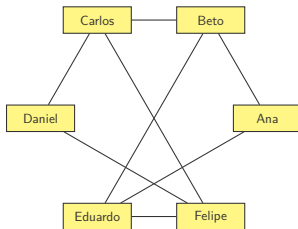
Como representar amizades em uma rede social?



Temos um conjunto de pessoas (Ana, Beto, Carlos, etc...)

Redes Sociais

Como representar amizades em uma rede social?



Temos um conjunto de pessoas (Ana, Beto, Carlos, etc...)

- Ligamos duas pessoas se elas se conhecem

Grafos

O **Grafo** como estrutura de dados é frequentemente utilizado para representar relações entre conjuntos de objetos

Grafos

O **Grafo** como estrutura de dados é frequentemente utilizado para representar relações entre conjuntos de objetos

- Chamamos esses objetos de **vértices**

Grafos

O **Grafo** como estrutura de dados é frequentemente utilizado para representar relações entre conjuntos de objetos

- Chamamos esses objetos de **vértices**
 - Ex: pessoas em uma rede social

Grafos

O **Grafo** como estrutura de dados é frequentemente utilizado para representar relações entre conjuntos de objetos

- Chamamos esses objetos de **vértices**
 - Ex: pessoas em uma rede social
- Chamamos as conexões entre os objetos de **arestas**

Grafos

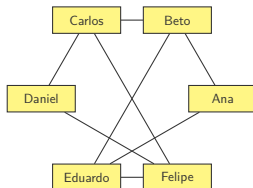
O **Grafo** como estrutura de dados é frequentemente utilizado para representar relações entre conjuntos de objetos

- Chamamos esses objetos de **vértices**
 - Ex: pessoas em uma rede social
- Chamamos as conexões entre os objetos de **arestas**
 - Ex: relação de amizade na rede social

Grafos

O **Grafo** como estrutura de dados é frequentemente utilizado para representar relações entre conjuntos de objetos

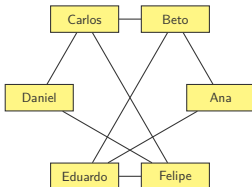
- Chamamos esses objetos de **vértices**
 - Ex: pessoas em uma rede social
- Chamamos as conexões entre os objetos de **arestas**
 - Ex: relação de amizade na rede social



Grafos

O **Grafo** como estrutura de dados é frequentemente utilizado para representar relações entre conjuntos de objetos

- Chamamos esses objetos de **vértices**
 - Ex: pessoas em uma rede social
- Chamamos as conexões entre os objetos de **arestas**
 - Ex: relação de amizade na rede social

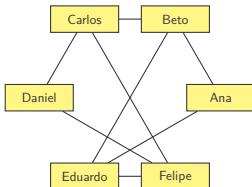


Representamos um grafo visualmente com os vértices representados por pontos e as arestas representadas por curvas ligando dois vértices.

Grafos

O **Grafo** como estrutura de dados é frequentemente utilizado para representar relações entre conjuntos de objetos

- Chamamos esses objetos de **vértices**
 - Ex: pessoas em uma rede social
- Chamamos as conexões entre os objetos de **arestas**
 - Ex: relação de amizade na rede social

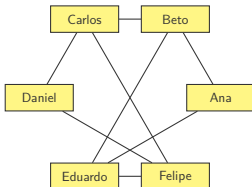


Representamos um grafo visualmente com os vértices representados por pontos e as arestas representadas por curvas ligando dois vértices.

Grafos

O **Grafo** como estrutura de dados é frequentemente utilizado para representar relações entre conjuntos de objetos

- Chamamos esses objetos de **vértices**
 - Ex: pessoas em uma rede social
- Chamamos as conexões entre os objetos de **arestas**
 - Ex: relação de amizade na rede social

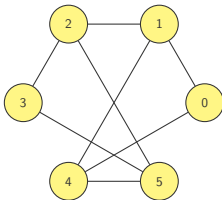


Representamos um grafo visualmente com os vértices representados por pontos e as arestas representadas por curvas ligando dois vértices.

Grafos

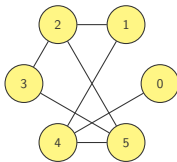
O **Grafo** como estrutura de dados é frequentemente utilizado para representar relações entre conjuntos de objetos

- Chamamos esses objetos de **vértices**
 - Ex: pessoas em uma rede social
- Chamamos as conexões entre os objetos de **arestas**
 - Ex: relação de amizade na rede social

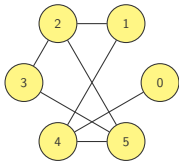


Representamos um grafo visualmente com os vértices representados por pontos e as arestas representadas por curvas ligando dois vértices.

Grafo não direcionado – Definição

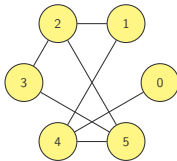


Grafo não direcionado – Definição



Um **grafo não direcionado** G é um par ordenado (V, E) tal que:

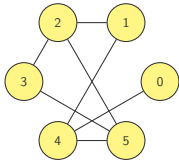
Grafo não direcionado – Definição



Um **grafo não direcionado** G é um par ordenado (V, E) tal que:

- V é um conjunto finito e não vazio cujos elementos são chamados **vértices** do grafo.

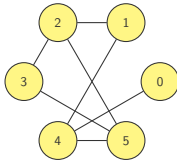
Grafo não direcionado – Definição



Um **grafo não direcionado** G é um par ordenado (V, E) tal que:

- V é um conjunto finito e não vazio cujos elementos são chamados **vértices** do grafo.
 - **Exemplo:** $V = \{0, 1, 2, 3, 4, 5\}$

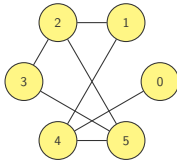
Grafo não direcionado – Definição



Um **grafo não direcionado** G é um par ordenado (V, E) tal que:

- V é um conjunto finito e não vazio cujos elementos são chamados **vértices** do grafo.
 - **Exemplo:** $V = \{0, 1, 2, 3, 4, 5\}$
- E é um conjunto de pares de elementos **não ordenados** de V . Cada elemento em E é chamado de **aresta**.

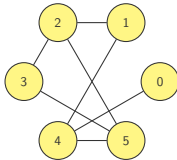
Grafo não direcionado – Definição



Um **grafo não direcionado** G é um par ordenado (V, E) tal que:

- V é um conjunto finito e não vazio cujos elementos são chamados **vértices** do grafo.
 - **Exemplo:** $V = \{0, 1, 2, 3, 4, 5\}$
- E é um conjunto de pares de elementos **não ordenados** de V . Cada elemento em E é chamado de **aresta**.
 - **Exemplo:** $E = \{\{0, 4\}, \{5, 3\}, \{1, 2\}, \{2, 5\}, \{4, 5\}, \{3, 2\}, \{1, 4\}\}$

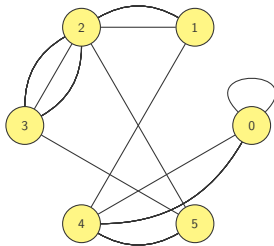
Grafo não direcionado – Definição



Um **grafo não direcionado** G é um par ordenado (V, E) tal que:

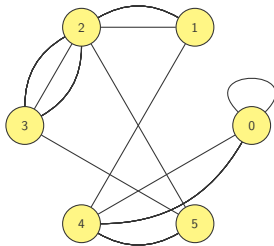
- V é um conjunto finito e não vazio cujos elementos são chamados **vértices** do grafo.
 - **Exemplo:** $V = \{0, 1, 2, 3, 4, 5\}$
- E é um conjunto de pares de elementos **não ordenados** de V . Cada elemento em E é chamado de **aresta**.
 - **Exemplo:** $E = \{\{0, 4\}, \{5, 3\}, \{1, 2\}, \{2, 5\}, \{4, 5\}, \{3, 2\}, \{1, 4\}\}$
- Chamamos grafos não direcionados simplesmente de grafos.

Grafo Simples – Definição



- Para alguns problemas, faz sentido permitir que dois vértices sejam ligados por várias arestas (**arestas múltiplas**) e também permite-se arestas que possuam o mesmo vértice como extremos (**laços**).

Grafo Simples – Definição



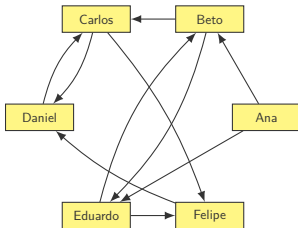
- Para alguns problemas, faz sentido permitir que dois vértices sejam ligados por várias arestas (**arestas múltiplas**) e também permite-se arestas que possuam o mesmo vértice como extremos (**laços**).
- Grafos que não possuem laços nem arestas múltiplas são chamados **grafos simples**.

Seguindo e sendo seguido

Em alguns problemas em grafos, as relações a serem modeladas não são simétricas. Por exemplo, como representar seguidores em redes sociais?

Seguindo e sendo seguido

Em alguns problemas em grafos, as relações a serem modeladas não são simétricas. Por exemplo, como representar seguidores em redes sociais?



- A **Ana** segue o **Beto** e o **Eduardo**
- Ninguém segue a **Ana**
- O **Daniel** é seguido pelo **Carlos** e pelo **Felipe**
- O **Eduardo** segue o **Beto** que o segue de volta

Grafo direcionado – Definição

Um **grafo direcionado** ou **digrafo** G é um par ordenado (V, E) tal que:

Grafo direcionado – Definição

Um **grafo direcionado** ou **digrafo** G é um par ordenado (V, E) tal que:

- V é um conjunto **vértices** finito e não vazio.

Grafo direcionado – Definição

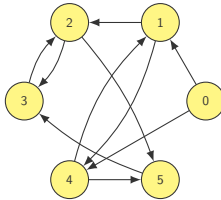
Um **grafo direcionado** ou **digrafo** G é um par ordenado (V, E) tal que:

- V é um conjunto **vértices** finito e não vazio.
- E é um conjunto de **pares ordenados** de vértices. Os elementos de E são chamados de **arestas**. Cada aresta $(u, v) \in E$ é tal que u indica **origem** e v indica **destino**. A aresta **sai** de u e **entra** em v .

Grafo direcionado – Definição

Um **grafo direcionado** ou **digrafo** G é um par ordenado (V, E) tal que:

- V é um conjunto **vértices** finito e não vazio.
- E é um conjunto de **pares ordenados** de vértices. Os elementos de E são chamados de **arestas**. Cada aresta $(u, v) \in E$ é tal que u indica **origem** e v indica **destino**. A aresta **sai** de u e **entra** em v .

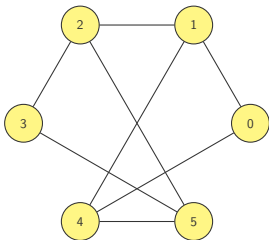


Desenhamos um grafo direcionado com os vértices representados por pontos e as arestas representadas por curvas com uma seta na ponta ligando os seus extremos.

Grafos direcionados e não direcionados

Podemos ver um **grafo não direcionado** como um **grafo direcionado**.

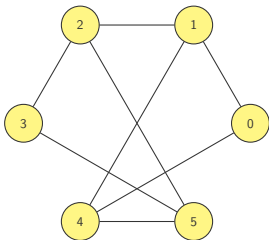
Basta considerar cada aresta como dois arcos.



Grafos direcionados e não direcionados

Podemos ver um **grafo não direcionado** como um **grafo direcionado**.

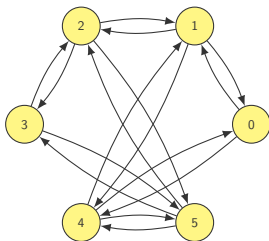
Basta considerar cada aresta como dois arcos.



Grafos direcionados e não direcionados

Podemos ver um **grafo não direcionado** como um **grafo direcionado**.

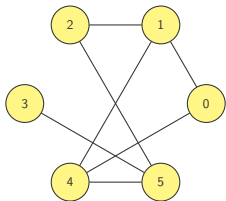
Basta considerar cada aresta como dois arcos.



Uma série de definições

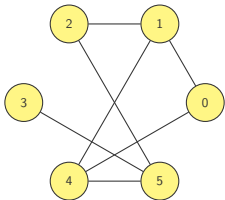


Adjacência e Incidência



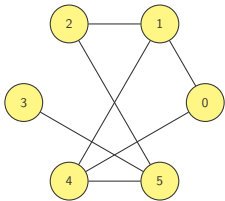
- Dado $G = (V, E)$ **não direcionado** e $\{u, v\} \in E$, dizemos que $\{u, v\}$ **incide** nos vértices u e v e vice-versa.

Adjacência e Incidência



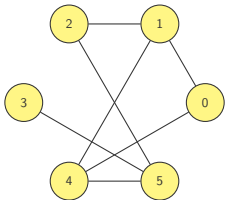
- Dado $G = (V, E)$ **não direcionado** e $\{u, v\} \in E$, dizemos que $\{u, v\}$ **incide** nos vértices u e v e vice-versa.
- Dizemos também que u e v são **adjacentes**.

Adjacência e Incidência



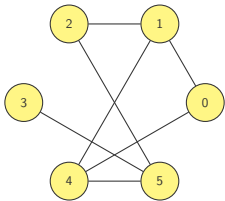
- Dado $G = (V, E)$ **não direcionado** e $\{u, v\} \in E$, dizemos que $\{u, v\}$ **incide** nos vértices u e v e vice-versa.
- Dizemos também que u e v são **adjacentes**.
- Dizemos que u e v são os **extremos** da aresta (u, v) .

Adjacência e Incidência



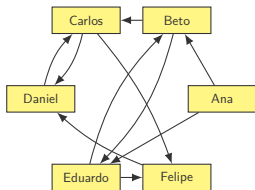
- Dado $G = (V, E)$ **não direcionado** e $\{u, v\} \in E$, dizemos que $\{u, v\}$ **incide** nos vértices u e v e vice-versa.
- Dizemos também que u e v são **adjacentes**.
- Dizemos que u e v são os **extremos** da aresta (u, v) .
- Vértices adjacentes a um vértice u formam a **vizinhança** de u e são chamados **vizinhos** de u .

Adjacência e Incidência



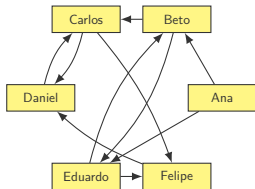
- Dado $G = (V, E)$ **não direcionado** e $\{u, v\} \in E$, dizemos que $\{u, v\}$ **incide** nos vértices u e v e vice-versa.
- Dizemos também que u e v são **adjacentes**.
- Dizemos que u e v são os **extremos** da aresta (u, v) .
- Vértices adjacentes a um vértice u formam a **vizinhança** de u e são chamados **vizinhos** de u .
 - **Exemplo:** os vértices 0, 1 e 5 são vizinhos de 4.

Adjacência e Incidência



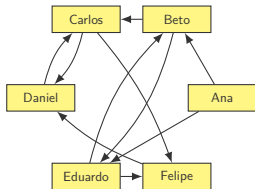
- Dado $G = (V, E)$ **direcionado** e $(u, v) \in E$, dizemos que (u, v) **sai** do vértice u e **entra** ou **incide** no vértice v .

Adjacência e Incidência



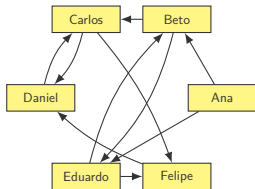
- Dado $G = (V, E)$ **direcionado** e $(u, v) \in E$, dizemos que (u, v) **sai** do vértice u e **entra** ou **incide** no vértice v .
- Dizemos também que v é **adjacente** a u .

Adjacência e Incidência



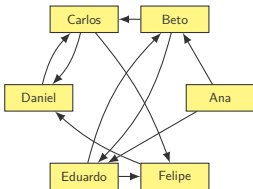
- Dado $G = (V, E)$ **direcionado** e $(u, v) \in E$, dizemos que (u, v) **sai** do vértice u e **entra** ou **incide** no vértice v .
- Dizemos também que v é **adjacente** a u .
- Dizemos que u e v são os **extremos** da aresta (u, v) .

Adjacência e Incidência



- Dado $G = (V, E)$ **direcionado** e $(u, v) \in E$, dizemos que (u, v) **sai** do vértice u e **entra** ou **incide** no vértice v .
- Dizemos também que v é **adjacente** a u .
- Dizemos que u e v são os **extremos** da aresta (u, v) .
- Vértices adjacentes a um vértice u formam a **vizinhança** de u e são chamados **vizinhos** de u .

Adjacência e Incidência



- Dado $G = (V, E)$ **direcionado** e $(u, v) \in E$, dizemos que (u, v) **sai** do vértice u e **entra** ou **incide** no vértice v .
- Dizemos também que v é **adjacente** a u .
- Dizemos que u e v são os **extremos** da aresta (u, v) .
- Vértices adjacentes a um vértice u formam a **vizinhança** de u e são chamados **vizinhos** de u .
 - **Exemplo:** Felipe é vizinho de Eduardo e Carlos, mas Ana não.

Tamanho do grafo

Considere um grafo $G = (V, E)$

- Denotamos por $|V|$ a cardinalidade do conjunto de vértices

Tamanho do grafo

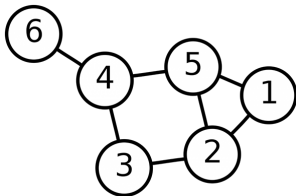
Considere um grafo $G = (V, E)$

- Denotamos por $|V|$ a cardinalidade do conjunto de vértices
- e por $|E|$ a cardinalidade do conjunto de arestas

Tamanho do grafo

Considere um grafo $G = (V, E)$

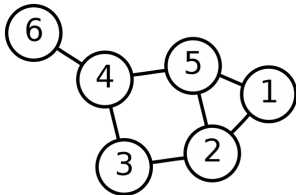
- Denotamos por $|V|$ a cardinalidade do conjunto de vértices
- e por $|E|$ a cardinalidade do conjunto de arestas
- no exemplo abaixo, temos $|V| = 6$ e $|E| = 7$



Tamanho do grafo

Considere um grafo $G = (V, E)$

- Denotamos por $|V|$ a cardinalidade do conjunto de vértices
- e por $|E|$ a cardinalidade do conjunto de arestas
- no exemplo abaixo, temos $|V| = 6$ e $|E| = 7$

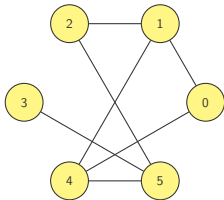


- O **tamanho** do grafo G é dado por $|V| + |E|$

Grau de um vértice

Quem tem mais amigos no Facebook? E seguidores no Instagram?

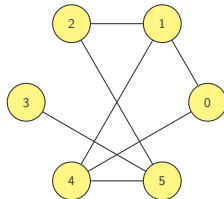
Dado um grafo **simples** e **não direcionado** $G = (V, E)$, o **grau** de um vértice $v \in V$, denotado por $d(v)$, é o número de arestas incidentes em u .



Grau de um vértice

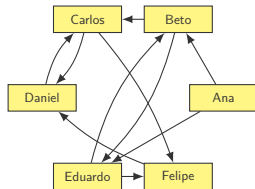
Quem tem mais amigos no Facebook? E seguidores no Instagram?

Dado um grafo **simples** e **não direcionado** $G = (V, E)$, o **grau** de um vértice $v \in V$, denotado por $d(v)$, é o número de arestas incidentes em v .



Dado um grafo **direcionado** $G = (V, E)$, o **grau de saída** $d_{out}(v)$ de um vértice $v \in V$, é o número de arestas que saem de v . O **grau de entrada** $d_{in}(v)$ de v , é o número de arestas que entram em v .

O **grau** $d(v)$ de v é a soma $d_{out}(v) + d_{in}(v)$.



Definição - Graus mínimo e máximo

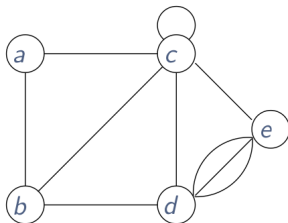
- Um vértice de grau 0 é dito **vértice isolado**.
- Um vértice de grau 1 é dito **vértice terminal**.
- Um vértice de grau $|V| - 1$ é dito **vértice universal**.

Definição - Graus mínimo e máximo

- Um vértice de grau 0 é dito **vértice isolado**.
- Um vértice de grau 1 é dito **vértice terminal**.
- Um vértice de grau $|V| - 1$ é dito **vértice universal**.
- O **grau mínimo** de G é o menor grau dentre todos os graus de vértices de G e é denotado por $\delta(G)$.
- O **grau máximo** de G é o maior grau dentre todos os graus de vértices de G e é denotado por $\Delta(G)$.



Contando graus dos vértices



$$d(a) = 2$$

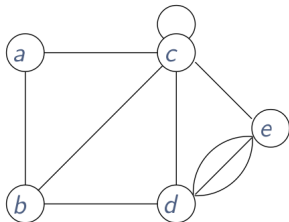
$$d(b) = 3$$

$$d(c) = 6$$

$$d(d) = 5$$

$$d(e) = 4$$

Contando graus dos vértices



$$d(a) = 2$$

$$d(b) = 3$$

$$d(c) = 6$$

$$d(d) = 5$$

$$d(e) = 4$$

Teorema 1 [Euler, 1735]

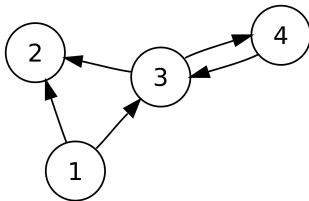
Se $G = (V, E)$ é um grafo não direcionado, então

$$\sum_{v \in V} d(v) = 2|E|.$$

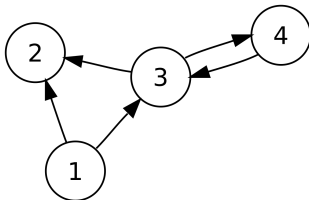


Euler

Contando graus dos vértices



Contando graus dos vértices



Teorema 2

Se $G = (V, E)$ é um grafo direcionado, então

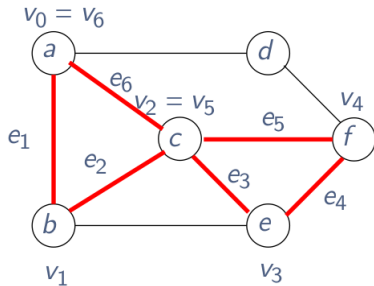
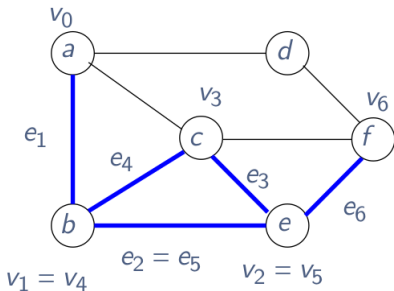
$$\sum_{v \in V} d_{out}(v) = \sum_{v \in V} d_{in}(v) = |E|.$$

Passeando pelo grafo



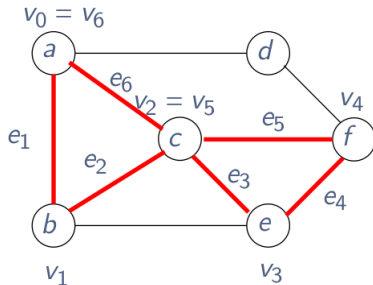
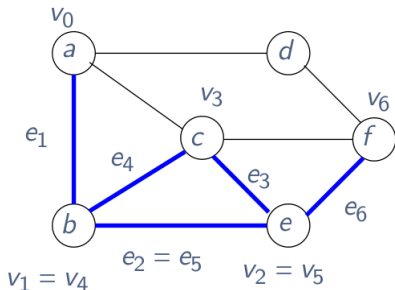
Passeios em grafos

Um **passeio** P de um vértice v_0 a um vértice v_k em um grafo $G = (V, E)$ é uma sequência finita e não vazia de vértices (v_0, v_1, \dots, v_k) tal que $(v_{i-1}, v_i) \in E$ para $1 \leq i \leq k$.



Passeios em grafos

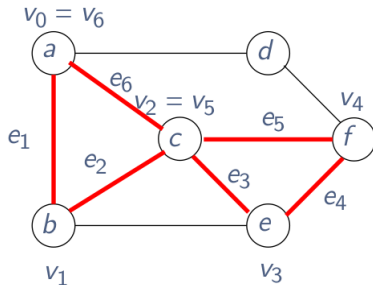
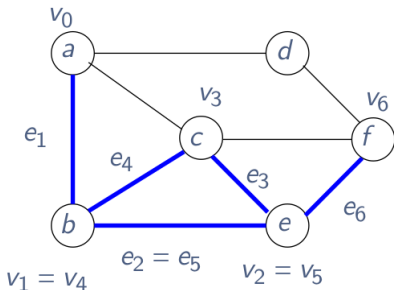
Um **passeio** P de um vértice v_0 a um vértice v_k em um grafo $G = (V, E)$ é uma sequência finita e não vazia de vértices (v_0, v_1, \dots, v_k) tal que $(v_{i-1}, v_i) \in E$ para $1 \leq i \leq k$.



- dizemos que v_k é **alcançável** a partir de v_0 através de P

Passeios em grafos

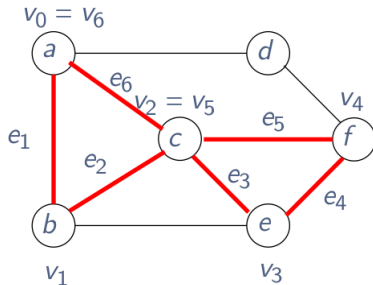
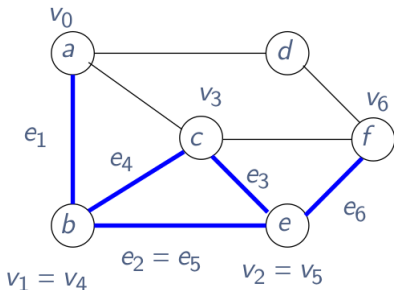
Um **passeio** P de um vértice v_0 a um vértice v_k em um grafo $G = (V, E)$ é uma sequência finita e não vazia de vértices (v_0, v_1, \dots, v_k) tal que $(v_{i-1}, v_i) \in E$ para $1 \leq i \leq k$.



- dizemos que v_k é **alcançável** a partir de v_0 através de P
- se $v_0 = v_k$, então dizemos que P é um **passeio fechado**

Passeios em grafos

Um **passeio** P de um vértice v_0 a um vértice v_k em um grafo $G = (V, E)$ é uma sequência finita e não vazia de vértices (v_0, v_1, \dots, v_k) tal que $(v_{i-1}, v_i) \in E$ para $1 \leq i \leq k$.



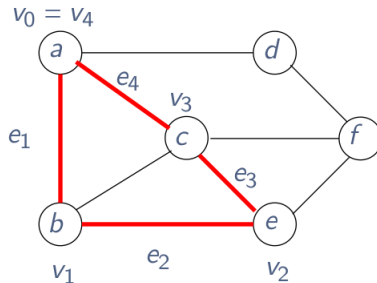
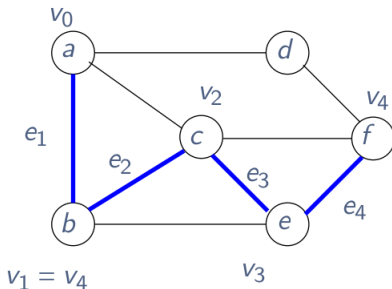
- dizemos que v_k é **alcançável** a partir de v_0 através de P
- se $v_0 = v_k$, então dizemos que P é um **passeio fechado**
- o **comprimento** de P é o seu número de arestas, ou seja, k

Caminhos e ciclos

- Um **caminho** é um passeio (v_0, v_1, \dots, v_k) cujos vértices são todos distintos.

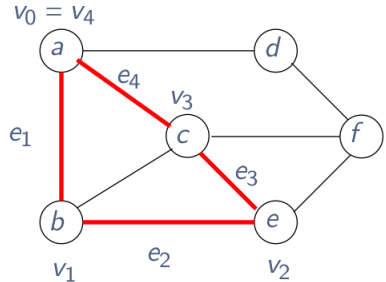
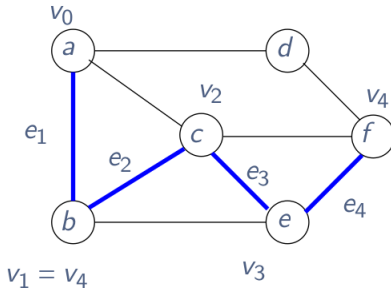
Caminhos e ciclos

- Um **caminho** é um passeio (v_0, v_1, \dots, v_k) cujos vértices são todos distintos.



Caminhos e ciclos

- Um **caminho** é um passeio (v_0, v_1, \dots, v_k) cujos vértices são todos distintos.



- Um **ciclo** é um passeio fechado (v_0, v_1, \dots, v_k) com $k > 0$ que possui pelo menos uma aresta e tal que v_1, \dots, v_k são todos distintos.

Refletindo sobre as definições

- Seja G um grafo e u, v vértices de G . Mostre que se existe um passeio de u a v em G , então existe um caminho de u a v em G .

Refletindo sobre as definições

- Seja G um grafo e u, v vértices de G . Mostre que se existe um passeio de u a v em G , então existe um caminho de u a v em G .
- Seja G um grafo e u, v, w vértices de G . Mostre que se em G existem um caminho de u a v e um caminho de v a w então existe um caminho de u a w em G .

Conexidade em grafos não direcionados

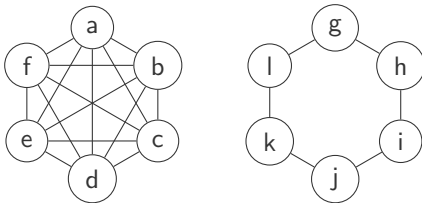
- Dizemos que dois vértices u e v de um grafo G estão **conectados** se existe um caminho de u a v em G .
- Um grafo G não direcionado é **conexo** quando quaisquer dois de seus vértices estão conectados.

Conexidade em grafos não direcionados

- Dizemos que dois vértices u e v de um grafo G estão **conectados** se existe um caminho de u a v em G .
- Um grafo G não direcionado é **conexo** quando quaisquer dois de seus vértices estão conectados.
- Uma **componente conexa** de um grafo não direcionado G é um subgrafo conexo maximal de G .

Conexidade em grafos não direcionados

- Dizemos que dois vértices u e v de um grafo G estão **conectados** se existe um caminho de u a v em G .
- Um grafo G não direcionado é **conexo** quando quaisquer dois de seus vértices estão conectados.
- Uma **componente conexa** de um grafo não direcionado G é um subgrafo conexo maximal de G .



Um grafo G com 12 vértices e duas componentes conexas.

Famílias de grafos básicas

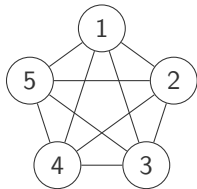


Grafo completo

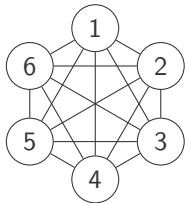
- Um **grafo completo** é um grafo simples no qual quaisquer dois de seus vértices são adjacentes.

Grafo completo

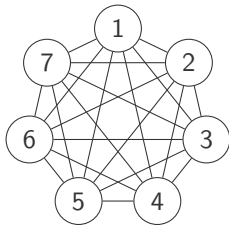
- Um **grafo completo** é um grafo simples no qual quaisquer dois de seus vértices são adjacentes.
- Um grafo completo com n vértices é denotado por K_n .



K_5



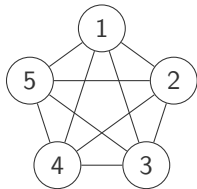
K_6



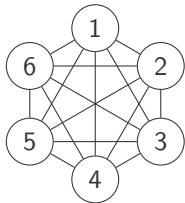
K_7

Grafo completo

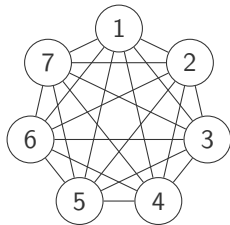
- Um **grafo completo** é um grafo simples no qual quaisquer dois de seus vértices são adjacentes.
- Um grafo completo com n vértices é denotado por K_n .



K_5



K_6

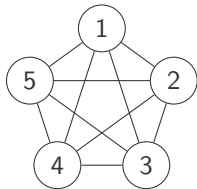


K_7

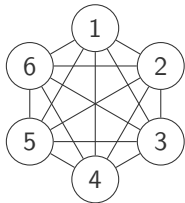
- Quantas arestas tem o grafo K_n ?

Grafo completo

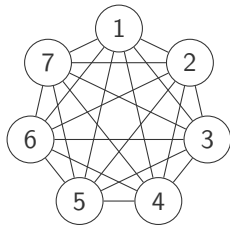
- Um **grafo completo** é um grafo simples no qual quaisquer dois de seus vértices são adjacentes.
- Um grafo completo com n vértices é denotado por K_n .



K_5



K_6

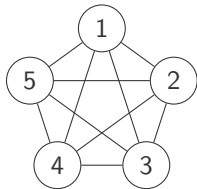


K_7

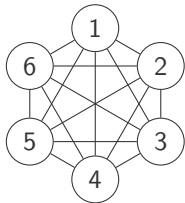
- Quantas arestas tem o grafo K_n ?**
- o grafo K_n tem $\binom{n}{2}$ arestas

Grafo completo

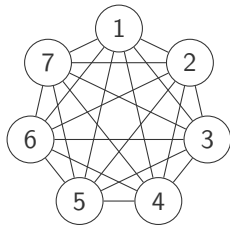
- Um **grafo completo** é um grafo simples no qual quaisquer dois de seus vértices são adjacentes.
- Um grafo completo com n vértices é denotado por K_n .



K_5



K_6



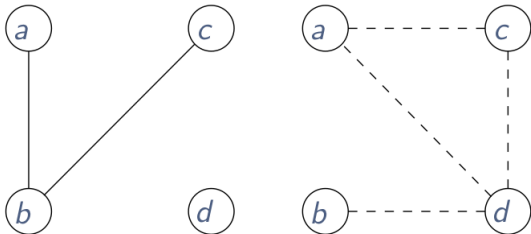
K_7

- Quantas arestas tem o grafo K_n ?**
- o grafo K_n tem $\binom{n}{2}$ arestas
- portanto, um grafo simples tem **no máximo** $\binom{n}{2}$ arestas

Grafo complemento

O **complemento** de um grafo simples G é o grafo simples \overline{G}

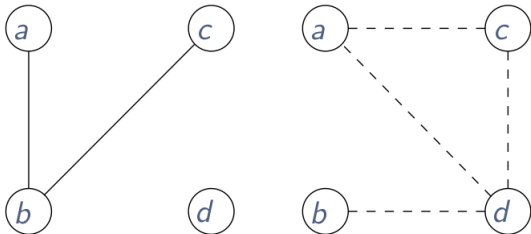
- cujo conjunto de vértices é $V(\overline{G}) = V(G)$
- e com $uv \in E(\overline{G})$ se e somente se $uv \notin E(G)$



Grafo complemento

O **complemento** de um grafo simples G é o grafo simples \overline{G}

- cujo conjunto de vértices é $V(\overline{G}) = V(G)$
- e com $uv \in E(\overline{G})$ se e somente se $uv \notin E(G)$



Note que $d_G(v) + d_{\overline{G}}(v) = |V(G)| - 1$

Grafo bipartido

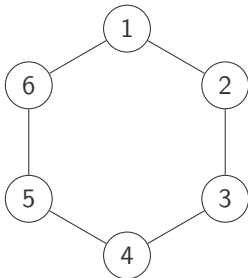
- Um grafo é **bipartido** se o seu conjunto de vértices pode ser particionado em dois subconjuntos X e Y de modo que toda aresta tenha um extremo em X e o outro extremo em Y .

Grafo bipartido

- Um grafo é **bipartido** se o seu conjunto de vértices pode ser particionado em dois subconjuntos X e Y de modo que toda aresta tenha um extremo em X e o outro extremo em Y .
 - Tal partição (X, Y) é chamada uma **bipartição** do grafo, e X e Y são suas **partes**.

Grafo bipartido

- Um grafo é **bipartido** se o seu conjunto de vértices pode ser particionado em dois subconjuntos X e Y de modo que toda aresta tenha um extremo em X e o outro extremo em Y .
 - Tal partição (X, Y) é chamada uma **bipartição** do grafo, e X e Y são suas **partes**.
 - Nós denotamos um grafo bipartido G com bipartição (X, Y) por $G[X, Y]$.



Caminhos

- Um **caminho** é um grafo simples cujos vértices podem ser dispostos em uma sequência linear de modo que dois vértices são adjacentes se e somente se eles são consecutivos na sequência.
- Um caminho com n vértices é denotado por P_n .



Caminho P_6

Ciclos

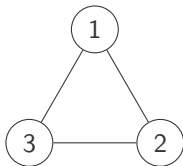
- Um **ciclo** com três ou mais vértices é um grafo simples cujos vértices podem ser dispostos em uma sequência cíclica de modo que dois vértices são adjacentes se e somente se eles são consecutivos na sequência.

Ciclos

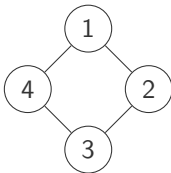
- Um **ciclo** com três ou mais vértices é um grafo simples cujos vértices podem ser dispostos em uma sequência cíclica de modo que dois vértices são adjacentes se e somente se eles são consecutivos na sequência.
- Um ciclo com n vértices é denotado por C_n .

Ciclos

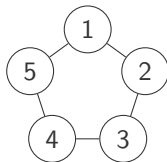
- Um **ciclo** com três ou mais vértices é um grafo simples cujos vértices podem ser dispostos em uma sequência cíclica de modo que dois vértices são adjacentes se e somente se eles são consecutivos na sequência.
- Um ciclo com n vértices é denotado por C_n .



Ciclo C_3



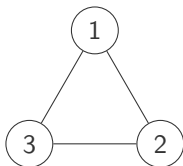
Ciclo C_4



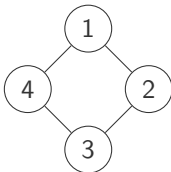
Ciclo C_5

Ciclos

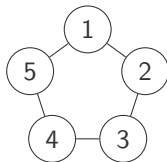
- Um **ciclo** com três ou mais vértices é um grafo simples cujos vértices podem ser dispostos em uma sequência cíclica de modo que dois vértices são adjacentes se e somente se eles são consecutivos na sequência.
- Um ciclo com n vértices é denotado por C_n .



Ciclo C_3



Ciclo C_4



Ciclo C_5

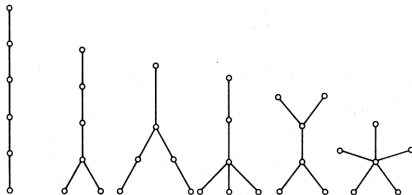
- O **comprimento** de um caminho ou ciclo é o seu número de arestas.

Árvore

- Um grafo é **acíclico** se ele não contém ciclos.

Árvore

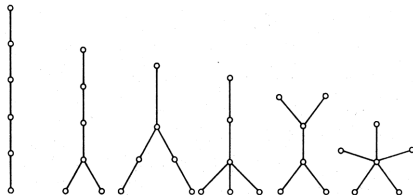
- Um grafo é **acíclico** se ele não contém ciclos.
- Uma **árvore** é um grafo conexo e acíclico.



Todas as árvores com 6 vértices

Árvore

- Um grafo é **acíclico** se ele não contém ciclos.
- Uma **árvore** é um grafo conexo e acíclico.

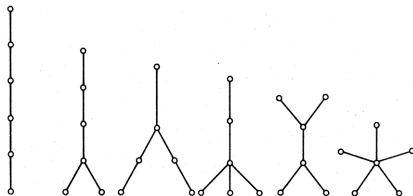


Todas as árvores com 6 vértices

- Uma **floresta** é um grafo acíclico.

Árvore

- Um grafo é **acíclico** se ele não contém ciclos.
- Uma **árvore** é um grafo conexo e acíclico.

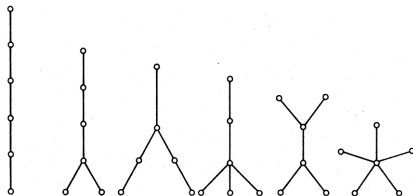


Todas as árvores com 6 vértices

- Uma **floresta** é um grafo acíclico.
- Uma **folha** é um vértice de grau 1 em uma árvore.

Árvore

- Um grafo é **acíclico** se ele não contém ciclos.
- Uma **árvore** é um grafo conexo e acíclico.



Todas as árvores com 6 vértices

- Uma **floresta** é um grafo acíclico.
- Uma **folha** é um vértice de grau 1 em uma árvore.
- toda árvore com pelo menos um vértice tem folha (**por quê?**)

Caracterização de árvores

Teorema

As seguintes afirmações são equivalentes:

1. G é uma árvore.
2. Para todo par de vértices u, v de G , existe um único caminho de u a v em G .
3. G é conexo e a remoção de qualquer aresta desconecta o grafo.
4. G é conexo e possui exatamente $|V| - 1$ arestas.

- Demonstração no livro do Cormen, Apêndice B.

Representação de grafos



Representação interna de grafos

Representamos grafos de duas maneiras principais

Representação interna de grafos

Representamos grafos de duas maneiras principais

1. **matriz de adjacência**

Representação interna de grafos

Representamos grafos de duas maneiras principais

1. **matriz de adjacência**
2. **listas de adjacência**

Representação interna de grafos

Representamos grafos de duas maneiras principais

1. **matriz de adjacência**
2. **listas de adjacência**

Qual estrutura de dados escolher?

Representação interna de grafos

Representamos grafos de duas maneiras principais

1. **matriz de adjacência**
2. **listas de adjacência**

Qual estrutura de dados escolher?

- depende do problema sendo tratado

Representação interna de grafos

Representamos grafos de duas maneiras principais

1. **matriz de adjacência**
2. **listas de adjacência**

Qual estrutura de dados escolher?

- depende do problema sendo tratado
- e das operações realizadas pelo algoritmo

Representação interna de grafos

Representamos grafos de duas maneiras principais

1. **matriz de adjacência**
2. **listas de adjacência**

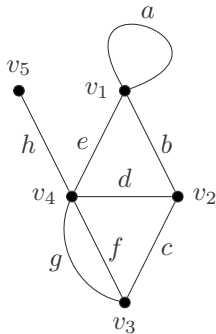
Qual estrutura de dados escolher?

- depende do problema sendo tratado
- e das operações realizadas pelo algoritmo
- a estrutura escolhida afeta a **complexidade do algoritmo**

Matriz de adjacência

grafo não direcionado

- Seja G um grafo com n vértices. A **matriz de adjacência** de G é a matriz $A(G) = (a_{uv})$ de dimensão $n \times n$, tal que a_{uv} é o número de arestas ligando os vértices u e v , cada laço contando como duas arestas.

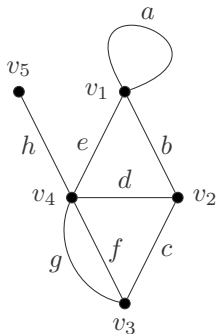


$$A(G) = \begin{matrix} & \begin{matrix} v_1 & v_2 & v_3 & v_4 & v_5 \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{matrix} & \begin{pmatrix} 2 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 2 & 0 \\ 1 & 1 & 2 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{matrix}$$

Matriz de adjacência

grafo não direcionado

- Seja G um grafo com n vértices. A **matriz de adjacência** de G é a matriz $A(G) = (a_{uv})$ de dimensão $n \times n$, tal que a_{uv} é o número de arestas ligando os vértices u e v , cada laço contando como duas arestas.

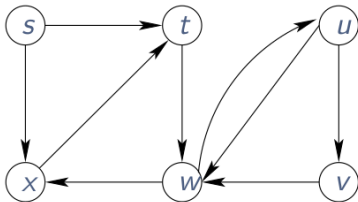


$$A(G) = \begin{matrix} & \begin{matrix} v_1 & v_2 & v_3 & v_4 & v_5 \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{matrix} & \begin{pmatrix} 2 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 2 & 0 \\ 1 & 1 & 2 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{matrix}$$

- se G for não direcionado, então a matriz A é simétrica

Matriz de adjacência

Grafo direcionado



	<i>s</i>	<i>t</i>	<i>u</i>	<i>x</i>	<i>w</i>	<i>v</i>
<i>s</i>	0	1	0	1	0	0
<i>t</i>	0	0	0	0	1	0
<i>u</i>	0	0	0	0	1	1
<i>x</i>	0	1	0	0	0	0
<i>w</i>	0	0	1	1	0	0
<i>v</i>	0	0	0	0	1	0

- se G for direcionado, então a matriz A não é simétrica

Listas de adjacência

Para representar um grafo $G = (V, E)$ por **listas de adjacências**:

Listas de adjacência

Para representar um grafo $G = (V, E)$ por **listas de adjacências**:

- criamos uma lista encadeada $Adj[v]$ para cada vértice v

Listas de adjacência

Para representar um grafo $G = (V, E)$ por **listas de adjacências**:

- criamos uma lista encadeada $Adj[v]$ para cada vértice v
- adicionamos a $Adj[v]$ todos os vértices adjacentes a v

Listas de adjacência

Para representar um grafo $G = (V, E)$ por **listas de adjacências**:

- criamos uma lista encadeada $Adj[v]$ para cada vértice v
- adicionamos a $Adj[v]$ todos os vértices adjacentes a v

Como representamos uma aresta (u, v) ?

Listas de adjacência

Para representar um grafo $G = (V, E)$ por **listas de adjacências**:

- criamos uma lista encadeada $Adj[v]$ para cada vértice v
- adicionamos a $Adj[v]$ todos os vértices adjacentes a v

Como representamos uma aresta (u, v) ?

- se a aresta for direcionada, então v está em $Adj[u]$

Listas de adjacência

Para representar um grafo $G = (V, E)$ por **listas de adjacências**:

- criamos uma lista encadeada $Adj[v]$ para cada vértice v
- adicionamos a $Adj[v]$ todos os vértices adjacentes a v

Como representamos uma aresta (u, v) ?

- se a aresta for direcionada, então v está em $Adj[u]$
- se a aresta for **não** direcionada,

Listas de adjacência

Para representar um grafo $G = (V, E)$ por **listas de adjacências**:

- criamos uma lista encadeada $Adj[v]$ para cada vértice v
- adicionamos a $Adj[v]$ todos os vértices adjacentes a v

Como representamos uma aresta (u, v) ?

- se a aresta for direcionada, então v está em $Adj[u]$
- se a aresta for **não** direcionada,
 1. então v está em $Adj[u]$

Listas de adjacência

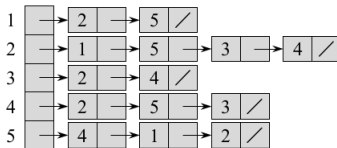
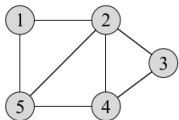
Para representar um grafo $G = (V, E)$ por **listas de adjacências**:

- criamos uma lista encadeada $Adj[v]$ para cada vértice v
- adicionamos a $Adj[v]$ todos os vértices adjacentes a v

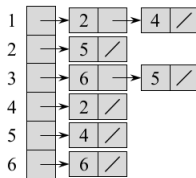
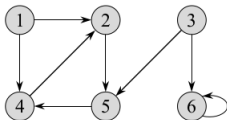
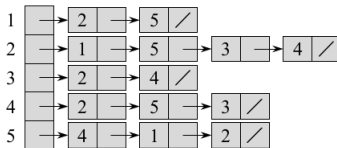
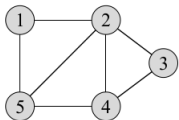
Como representamos uma aresta (u, v) ?

- se a aresta for direcionada, então v está em $Adj[u]$
- se a aresta for **não** direcionada,
 1. então v está em $Adj[u]$
 2. também u está em $Adj[v]$

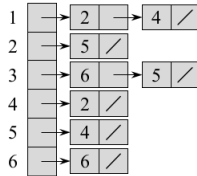
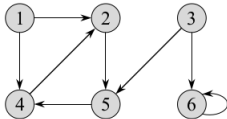
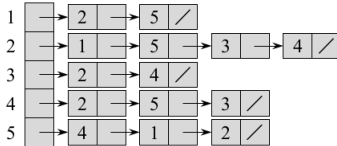
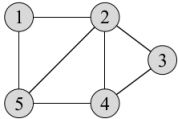
Listas de adjacências



Listas de adjacências



Listas de adjacências



- A representação por **listas de adjacências** geralmente é a mais usada por ser mais eficiente em grafos com poucas arestas (**grafos esparsos**).

Comparação Listas e Matrizes

Espaço para o armazenamento:

Comparação Listas e Matrizes

Espaço para o armazenamento:

- Matriz: $O(|V|^2)$

Comparação Listas e Matrizes

Espaço para o armazenamento:

- Matriz: $O(|V|^2)$
- Listas: $O(|V| + |E|)$

Comparação Listas e Matrizes

Espaço para o armazenamento:

- Matriz: $O(|V|^2)$
- Listas: $O(|V| + |E|)$

Tempo:

Comparação Listas e Matrizes

Espaço para o armazenamento:

- Matriz: $O(|V|^2)$
- Listas: $O(|V| + |E|)$

Tempo:

Operação	Matriz	Listas
Inserir	$O(1)$	$O(1)$
Remover	$O(1)$	$O(d(v))$
Aresta existe?	$O(1)$	$O(d(v))$
Percorrer vizinhança	$O(V)$	$O(d(v))$
Espaço utilizado	$O(V ^2)$	$O(V + E)$

Comparação Listas e Matrizes

Espaço para o armazenamento:

- Matriz: $O(|V|^2)$
- Listas: $O(|V| + |E|)$

Tempo:

Operação	Matriz	Listas
Inserir	$O(1)$	$O(1)$
Remover	$O(1)$	$O(d(v))$
Aresta existe?	$O(1)$	$O(d(v))$
Percorrer vizinhança	$O(V)$	$O(d(v))$
Espaço utilizado	$O(V ^2)$	$O(V + E)$

As duas permitem representar grafos e digrafos

Comparação Listas e Matrizes

Espaço para o armazenamento:

- Matriz: $O(|V|^2)$
- Listas: $O(|V| + |E|)$

Tempo:

Operação	Matriz	Listas
Inserir	$O(1)$	$O(1)$
Remover	$O(1)$	$O(d(v))$
Aresta existe?	$O(1)$	$O(d(v))$
Percorrer vizinhança	$O(V)$	$O(d(v))$
Espaço utilizado	$O(V ^2)$	$O(V + E)$

As duas permitem representar grafos e digrafos

Qual usar?

Comparação Listas e Matrizes

Espaço para o armazenamento:

- Matriz: $O(|V|^2)$
- Listas: $O(|V| + |E|)$

Tempo:

Operação	Matriz	Listas
Inserir	$O(1)$	$O(1)$
Remover	$O(1)$	$O(d(v))$
Aresta existe?	$O(1)$	$O(d(v))$
Percorrer vizinhança	$O(V)$	$O(d(v))$
Espaço utilizado	$O(V ^2)$	$O(V + E)$

As duas permitem representar grafos e digrafos

Qual usar?

- Depende das operações usadas e se o grafo é esparsos

Grafo como matriz de adjacência em C++

```
1 // criando um grafo ciclo C4 e imprimindo
2 #include <iostream> // Exemplo01.cpp
3 #include <vector>
4 using namespace std;
5
6 using Graph = vector<vector<int>>>;
7
8 int main() {
9     Graph G {    {0,1,0,1},
10                  {1,0,1,0},
11                  {0,1,0,1},
12                  {1,0,1,0},
13                };
14
15     for(size_t i = 0; i < G.size(); ++i) {
16         cout << i << ": [ ";
17         for(size_t j = 0; j < G.size(); ++j) {
18             if(G[i][j] == 1)
19                 cout << j << " ";
20         }
21         cout << "]\n";
22     }
23 }
```

Grafo como lista de adjacência em C++

```
1 // criando um grafo ciclo C4 e imprimindo
2 #include <iostream> // Exemplo02.cpp
3 #include <unordered_map>
4 #include <list>
5 using namespace std;
6
7 // usei hash table ao inves de vector
8 using Graph = unordered_map<int, list<int>>;
9
10 int main() {
11     Graph G { {0,{1,3}}, {1,{0,2}},
12               {2,{1,3}}, {3,{2,0}}
13             };
14
15     for(size_t i = 0; i < G.size(); ++i) {
16         cout << i << ": [ ";
17         for(auto& vertex : G[i]) {
18             cout << vertex << " ";
19         }
20         cout << "]\n";
21     }
22 }
```

Extensões

- Há alternativas para representar grafos, mas matrizes e listas de adjacência são as mais usadas.

- Há alternativas para representar grafos, mas matrizes e listas de adjacência são as mais usadas.
- Essas representações podem ser usadas para grafos ponderados, grafos com laços e arestas múltiplas, grafos com pesos nos vértices etc.

- Há alternativas para representar grafos, mas matrizes e listas de adjacência são as mais usadas.
- Essas representações podem ser usadas para grafos ponderados, grafos com laços e arestas múltiplas, grafos com pesos nos vértices etc.
- Para determinados algoritmos é importante manter **estruturas de dados adicionais**.

Exercícios



Exercício

- O **grau** de um vértice em um grafo não-direcionado é o número de arestas que nele incidem. Em um grafo direcionado, o **grau de saída** de um vértice é o número de arestas que saem dele, e o **grau de entrada** de um vértice é o número de arestas que entram nele. O **grau** de um vértice em um grafo direcionado é seu grau de entrada mais seu grau de saída.

Dada uma representação por lista de adjacências de um grafo direcionado:

- (a) Qual o tempo necessário para calcular os graus de saída de todos os vértices?
- (b) Qual o tempo necessário para calcular os graus de entrada?

Exercício

- O **grau** de um vértice em um grafo não-direcionado é o número de arestas que nele incidem. Em um grafo direcionado, o **grau de saída** de um vértice é o número de arestas que saem dele, e o **grau de entrada** de um vértice é o número de arestas que entram nele. O **grau** de um vértice em um grafo direcionado é seu grau de entrada mais seu grau de saída.

Dada uma representação por lista de adjacências de um grafo direcionado:

- (a) Qual o tempo necessário para calcular os graus de saída de todos os vértices?
- (b) Qual o tempo necessário para calcular os graus de entrada?

- O **transposto** de um grafo direcionado $G = (V, E)$ é o grafo $G_T = (V, E_T)$, onde $E_T = \{(v, u) \in V \times V : (u, v) \in E\}$. Assim, G_T é G com todas as suas arestas invertidas. Descreva algoritmos eficientes para calcular G_T a partir de G , para a representação por lista de adjacências e também para a representação por matriz de adjacências de G . Analise os tempos de execução de seus algoritmos.
- O **quadrado** de um grafo direcionado $G = (V, E)$ é o grafo $G_2 = (V, E_2)$ em que $(u, v) \in E_2$ se e somente se G contiver um caminho que tenha no máximo duas arestas entre u e v . Descreva algoritmos eficientes para calcular G_2 a partir de G para uma representação por lista de adjacências e para uma representação por matriz de adjacências de G . Analise os tempos de execução de seus algoritmos.

- A maioria dos algoritmos em grafos que adota uma representação por matriz de adjacências como entrada exige o tempo $\Omega(V^2)$, mas há algumas exceções. Mostre como determinar se um grafo direcionado G contém um **sumidouro** (isto é, um vértice com grau de entrada $|V| - 1$ e grau de saída 0) no tempo $O(V)$, dada uma matriz de adjacências para G .

- A **matriz de incidência** de um grafo direcionado $G = (V, E)$ sem nenhum laço é uma matriz $B = (b_{ij})$ de dimensão $|V| \times |E|$ tal que:

$$b_{ij} = \begin{cases} -1, & \text{se a aresta } j \text{ sai do vértice } i; \\ 1, & \text{se a aresta } j \text{ entra no vértice } i; \\ 0, & \text{se caso contrário.} \end{cases}$$

Descreva o que representam as entradas do produto de matrizes BB_T , onde B_T é a transposta de B .

- Dada a matriz de adjacências de um grafo com N vértices, faça um algoritmo que determine se esse grafo é direcionado ou não-direcionado.

- Um **multigrafo** é um grafo que possui arestas paralelas e/ou laços. Dada uma representação por lista de adjacências de um multigrafo $G = (V, E)$, descreva um algoritmo de tempo $O(V + E)$ para calcular a representação por lista de adjacências do grafo não direcionado “equivalente” $G' = (V, E')$, onde E' consiste nas arestas em E onde todas as arestas múltiplas entre dois vértices foram substituídas por uma aresta única e onde todos os laços foram removidos.

FIM

