

Projeto e Análise de Algoritmos*

Algoritmos e conceitos fundamentais de grafos

Segundo Semestre de 2019

*Criado por C. de Souza, C. da Silva, O. Lee, F. Miyazawa et al.

A maior parte deste conjunto de slides foi inicialmente preparada por Cid Carvalho de Souza e Cândida Nunes da Silva para cursos de Análise de Algoritmos. Além desse material, diversos conteúdos foram adicionados ou incorporados por outros professores, em especial por Orlando Lee e por Flávio Keidi Miyazawa. Os slides usados nessa disciplina são uma junção dos materiais didáticos gentilmente cedidos por esses professores e contêm algumas modificações, que podem ter introduzido erros.

O conjunto de slides de cada unidade do curso será disponibilizado como guia de estudos e deve ser usado unicamente para revisar as aulas. Para estudar e praticar, leia o livro-texto indicado e resolva os exercícios sugeridos.

Lehilton

Agradecimentos (Cid e Cândida)

- ▶ Várias pessoas contribuíram **direta ou indiretamente** com a preparação deste material.
- ▶ Algumas destas pessoas cederam gentilmente seus arquivos digitais enquanto outras cederam gentilmente o seu tempo fazendo correções e dando sugestões.
- ▶ Uma lista destes “colaboradores” (**em ordem alfabética**) é dada abaixo:
 - ▶ Célia Picinin de Mello
 - ▶ Flávio Keidi Miyazawa
 - ▶ José Coelho de Pina
 - ▶ Orlando Lee
 - ▶ Paulo Feofiloff
 - ▶ Pedro Rezende
 - ▶ Ricardo Dahab
 - ▶ Zanoni Dias

Conceitos de grafos

Definição de Grafo

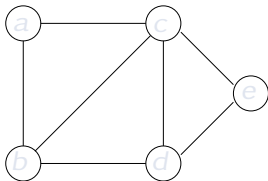
Um *grafo* é um par $G = (V, E)$ onde

- ▶ V é um conjunto finito de elementos chamados *vértices* e
- ▶ E é um conjunto finito de pares *não ordenados* de vértices chamados *arestas*.

▶ **Exemplo:**

$$V = \{a, b, c, d, e\}$$

$$E = \{(a, b), (a, c), (b, c), (b, d), (c, d), (c, e), (d, e)\}$$



Definição de Grafo

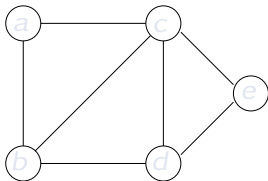
Um *grafo* é um par $G = (V, E)$ onde

- ▶ V é um conjunto finito de elementos chamados *vértices* e
- ▶ E é um conjunto finito de pares *não ordenados* de vértices chamados *arestas*.

▶ **Exemplo:**

$$V = \{a, b, c, d, e\}$$

$$E = \{(a, b), (a, c), (b, c), (b, d), (c, d), (c, e), (d, e)\}$$



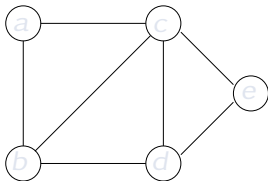
Definição de Grafo

Um *grafo* é um par $G = (V, E)$ onde

- ▶ V é um conjunto finito de elementos chamados *vértices* e
- ▶ E é um conjunto finito de pares **não ordenados** de vértices chamados *arestas*.
- ▶ **Exemplo:**

$$V = \{a, b, c, d, e\}$$

$$E = \{(a, b), (a, c), (b, c), (b, d), (c, d), (c, e), (d, e)\}$$



Definição de Grafo

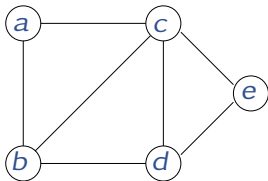
Um *grafo* é um par $G = (V, E)$ onde

- ▶ V é um conjunto finito de elementos chamados *vértices* e
- ▶ E é um conjunto finito de pares **não ordenados** de vértices chamados *arestas*.

▶ **Exemplo:**

$$V = \{a, b, c, d, e\}$$

$$E = \{(a, b), (a, c), (b, c), (b, d), (c, d), (c, e), (d, e)\}$$



Definição de Grafo

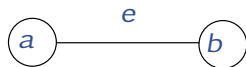
- ▶ Dada uma aresta $e = (a, b)$, dizemos que os vértices a e b são os *extremos* da aresta e e que a e b são vértices *adjacentes*.
- ▶ Dizemos também que a aresta e é *incidente* aos vértices a e b e que os vértices a e b são incidentes à aresta e .



- ▶ Note que para pares não ordenados, temos $(a, b) = (b, a)$.

Definição de Grafo

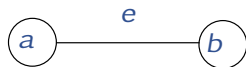
- ▶ Dada uma aresta $e = (a, b)$, dizemos que os vértices a e b são os *extremos* da aresta e e que a e b são vértices *adjacentes*.
- ▶ Dizemos também que a aresta e é *incidente* aos vértices a e b e que os vértices a e b são incidentes à aresta e .



- ▶ Note que para pares não ordenados, temos $(a, b) = (b, a)$.

Definição de Grafo

- ▶ Dada uma aresta $e = (a, b)$, dizemos que os vértices a e b são os *extremos* da aresta e e que a e b são vértices *adjacentes*.
- ▶ Dizemos também que a aresta e é *incidente* aos vértices a e b e que os vértices a e b são incidentes à aresta e .

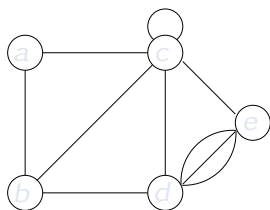


- ▶ Note que para pares não ordenados, temos $(a, b) = (b, a)$.

Multigrafo

Um **multigrafo** é um generalização de grafos que pode conter:

- ▶ *laço*: uma aresta com extremos idênticos
- ▶ *arestas múltiplas*: duas ou mais arestas com o mesmo par de extremos

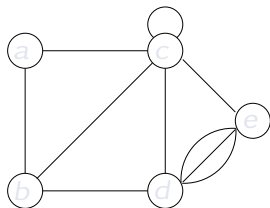


Dizemos que um grafo é *simples* quando ele não possui laços ou arestas múltiplas.

Multigrafo

Um **multigrafo** é um generalização de grafos que pode conter:

- ▶ **laço**: uma aresta com extremos idênticos
- ▶ **arestas múltiplas**: duas ou mais arestas com o mesmo par de extremos

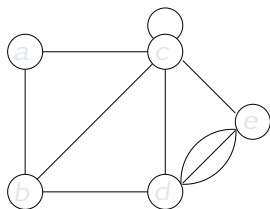


Dizemos que um grafo é **simples** quando ele não possui laços ou arestas múltiplas.

Multigrafo

Um **multigrafo** é um generalização de grafos que pode conter:

- ▶ **laço**: uma aresta com extremos idênticos
- ▶ **arestas múltiplas**: duas ou mais arestas com o mesmo par de extremos

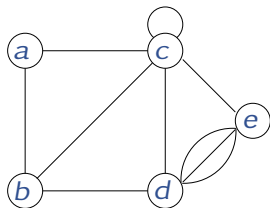


Dizemos que um grafo é **simples** quando ele não possui laços ou arestas múltiplas.

Multigrafo

Um **multigrafo** é um generalização de grafos que pode conter:

- ▶ **laço**: uma aresta com extremos idênticos
- ▶ **arestas múltiplas**: duas ou mais arestas com o mesmo par de extremos

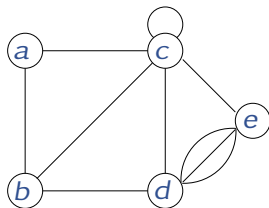


Dizemos que um grafo é **simples** quando ele não possui laços ou arestas múltiplas.

Multigrafo

Um **multigrafo** é um generalização de grafos que pode conter:

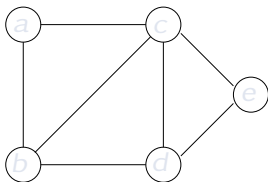
- ▶ **laço**: uma aresta com extremos idênticos
- ▶ **arestas múltiplas**: duas ou mais arestas com o mesmo par de extremos



Dizemos que um grafo é **simples** quando ele não possui laços ou arestas múltiplas.

Tamanho do Grafo

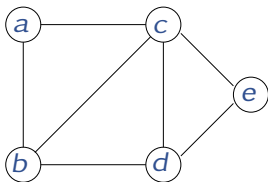
- ▶ Denotamos por $|V|$ e $|E|$ a cardinalidade dos conjuntos de vértices e arestas de um grafo $G = (V, E)$, respectivamente.
- ▶ No exemplo abaixo temos $|V| = 5$ e $|E| = 7$.



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

Tamanho do Grafo

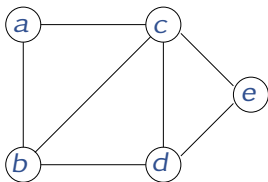
- ▶ Denotamos por $|V|$ e $|E|$ a cardinalidade dos conjuntos de vértices e arestas de um grafo $G = (V, E)$, respectivamente.
- ▶ No exemplo abaixo temos $|V| = 5$ e $|E| = 7$.



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

Tamanho do Grafo

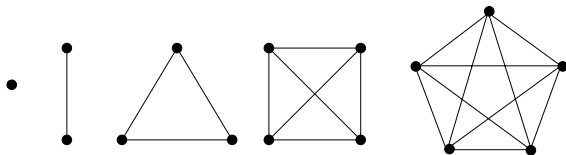
- ▶ Denotamos por $|V|$ e $|E|$ a cardinalidade dos conjuntos de vértices e arestas de um grafo $G = (V, E)$, respectivamente.
- ▶ No exemplo abaixo temos $|V| = 5$ e $|E| = 7$.



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

Grafos completos

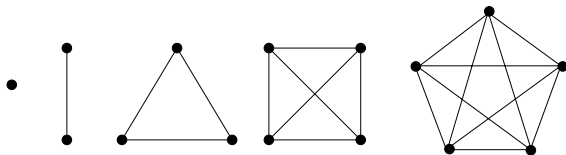
- ▶ *Grafo completo*: para todo par de vértices u, v , a aresta (u, v) pertence ao grafo.



- ▶ O número de arestas de um *grafo completo* com n vértices é $\binom{n}{2}$.
- ▶ O número de arestas de um *grafo simples* com n vértices é no máximo $\binom{n}{2}$.

Grafos completos

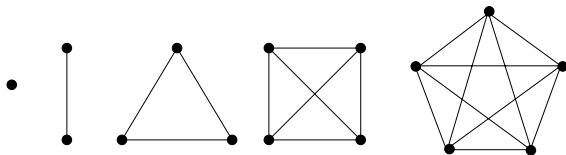
- ▶ *Grafo completo*: para todo par de vértices u, v , a aresta (u, v) pertence ao grafo.



- ▶ O número de arestas de um *grafo completo* com n vértices é $\binom{n}{2}$.
- ▶ O número de arestas de um *grafo simples* com n vértices é no máximo $\binom{n}{2}$.

Grafos completos

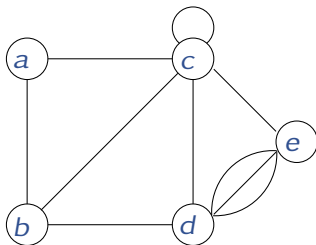
- ▶ *Grafo completo*: para todo par de vértices u, v , a aresta (u, v) pertence ao grafo.



- ▶ O número de arestas de um *grafo completo* com n vértices é $\binom{n}{2}$.
- ▶ O número de arestas de um *grafo simples* com n vértices é no máximo $\binom{n}{2}$.

Grau de um vértice

- O *grau* de um vértice v , denotado por $d(v)$ é o número de arestas incidentes a v , com laços contados duas vezes.



$$d(a) = 2$$

$$d(b) = 3$$

$$d(c) = 6$$

$$d(d) = 5$$

$$d(e) = 4$$

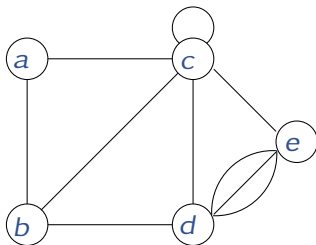
Teorema (*Handshaking lemma*)

Para todo grafo $G = (V, E)$ temos:

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

Grau de um vértice

- ▶ O *grau* de um vértice v , denotado por $d(v)$ é o número de arestas incidentes a v , com laços contados duas vezes.



$$d(a) = 2$$

$$d(b) = 3$$

$$d(c) = 6$$

$$d(d) = 5$$

$$d(e) = 4$$

Teorema (Handshaking lemma)

Para todo grafo $G = (V, E)$ temos:

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

Grafo complementar

Seja $G = (V, E)$ um grafo simples. O *complemento* de G é o grafo \bar{G} com conjunto de vértices V tal que $(u, v) \in E(\bar{G})$ se e somente se $(u, v) \notin E(G)$.

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

Exercício. Mostre que em uma festa com pelo menos $n \geq 6$ pessoas, existem três pessoas que se conhecem mutuamente ou três pessoas que não se conhecem mutuamente.

Exercício. Suponha que em um grupo S de n pessoas, com $n \geq 4$, vale o seguinte: em qualquer grupo $X \subseteq S$ de 4 pessoas, existe uma que conhece as demais pessoas de X .

Mostre que existe uma pessoa em S que conhece todas as demais pessoas de S .

Grafo complementar

Seja $G = (V, E)$ um grafo simples. O *complemento* de G é o grafo \bar{G} com conjunto de vértices V tal que $(u, v) \in E(\bar{G})$ se e somente se $(u, v) \notin E(G)$.

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

Exercício. Mostre que em uma festa com pelo menos $n \geq 6$ pessoas, existem três pessoas que se conhecem mutuamente ou três pessoas que não se conhecem mutuamente.

Exercício. Suponha que em um grupo S de n pessoas, com $n \geq 4$, vale o seguinte: em qualquer grupo $X \subseteq S$ de 4 pessoas, existe uma que conhece as demais pessoas de X .

Mostre que existe uma pessoa em S que conhece todas as demais pessoas de S .

Grafo complementar

Seja $G = (V, E)$ um grafo simples. O *complemento* de G é o grafo \bar{G} com conjunto de vértices V tal que $(u, v) \in E(\bar{G})$ se e somente se $(u, v) \notin E(G)$.

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

Exercício. Mostre que em uma festa com pelo menos $n \geq 6$ pessoas, existem três pessoas que se conhecem mutuamente ou três pessoas que não se conhecem mutuamente.

Exercício. Suponha que em um grupo S de n pessoas, com $n \geq 4$, vale o seguinte: em qualquer grupo $X \subseteq S$ de 4 pessoas, existe uma que conhece as demais pessoas de X .

Mostre que existe uma pessoa em S que conhece todas as demais pessoas de S .

Grafo complementar

Seja $G = (V, E)$ um grafo simples. O *complemento* de G é o grafo \bar{G} com conjunto de vértices V tal que $(u, v) \in E(\bar{G})$ se e somente se $(u, v) \notin E(G)$.

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

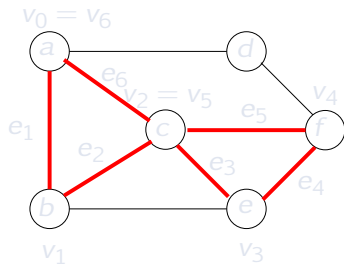
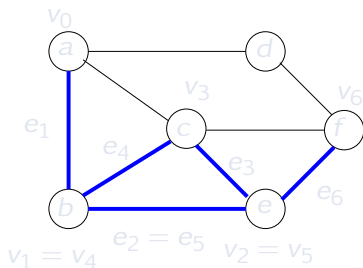
Exercício. Mostre que em uma festa com pelo menos $n \geq 6$ pessoas, existem três pessoas que se conhecem mutuamente ou três pessoas que não se conhecem mutuamente.

Exercício. Suponha que em um grupo S de n pessoas, com $n \geq 4$, vale o seguinte: em qualquer grupo $X \subseteq S$ de 4 pessoas, existe uma que conhece as demais pessoas de X .

Mostre que existe uma pessoa em S que conhece todas as demais pessoas de S .

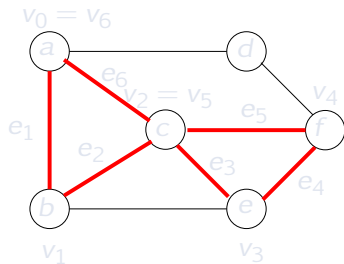
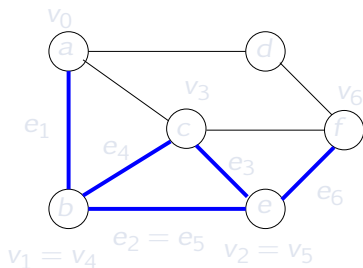
Passeios em Grafos

- ▶ Um *passeio* P de um vértice v_0 a um vértice v_k em um grafo G é uma sequência finita e não vazia $(v_0, e_1, v_1, \dots, e_n, v_k)$ cujos elementos são alternadamente vértices e arestas e tal que, para todo $1 \leq i \leq k$, v_{i-1} e v_i são os extremos de e_i .
- ▶ Dizemos que P é um passeio de v_0 a v_k e que v_k é *alcançável* a partir de v_0 através de P .
- ▶ Dizemos que P é *fechado* se $v_0 = v_k$.



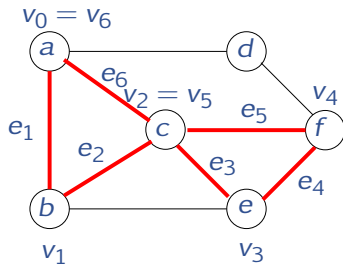
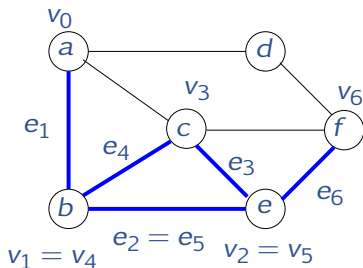
Passeios em Grafos

- ▶ Um *passeio* P de um vértice v_0 a um vértice v_k em um grafo G é uma sequência finita e não vazia $(v_0, e_1, v_1, \dots, e_n, v_k)$ cujos elementos são alternadamente vértices e arestas e tal que, para todo $1 \leq i \leq k$, v_{i-1} e v_i são os extremos de e_i .
- ▶ Dizemos que P é um passeio de v_0 a v_k e que v_k é *alcançável* a partir de v_0 através de P .
- ▶ Dizemos que P é *fechado* se $v_0 = v_k$.



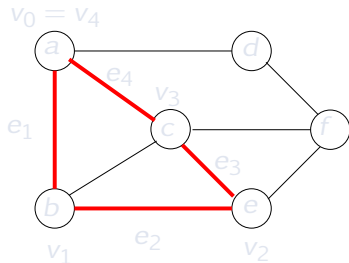
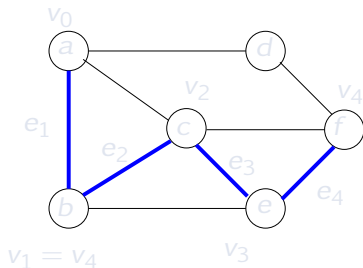
Passeios em Grafos

- ▶ Um *passeio* P de um vértice v_0 a um vértice v_k em um grafo G é uma sequência finita e não vazia $(v_0, e_1, v_1, \dots, e_n, v_k)$ cujos elementos são alternadamente vértices e arestas e tal que, para todo $1 \leq i \leq k$, v_{i-1} e v_i são os extremos de e_i .
- ▶ Dizemos que P é um passeio de v_0 a v_k e que v_k é *alcançável* a partir de v_0 através de P .
- ▶ Dizemos que P é *fechado* se $v_0 = v_k$.



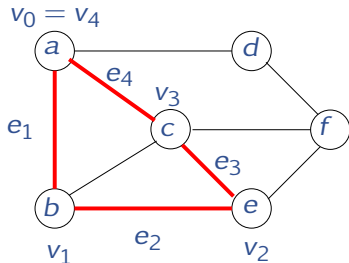
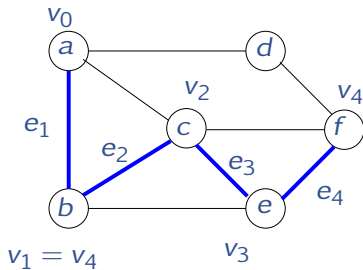
Caminhos e Ciclos

- ▶ O *comprimento* do passeio P é igual ao seu número de arestas, ou seja, k .
- ▶ Um *caminho* é um passeio em que todos seus vértices são distintos.
- ▶ Um *ciclo* é um passeio fechado que possui pelo menos uma aresta e tal que v_2, \dots, v_k são distintos e todas as arestas são distintas.



Caminhos e Ciclos

- ▶ O *comprimento* do passeio P é igual ao seu número de arestas, ou seja, k .
- ▶ Um *caminho* é um passeio em que todos seus vértices são distintos.
- ▶ Um *ciclo* é um passeio fechado que possui pelo menos uma aresta e tal que v_2, \dots, v_k são distintos e todas as arestas são distintas.



Exercícios

- (1) Sejam 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 .

Por que isto é um resultado interessante?

- (2) Sejam 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 .

- (3) É verdade que todo passeio fechado contém um ciclo?

Exercícios

- (1) Sejam 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 .

Por que isto é um resultado interessante?

- (2) Sejam 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 .

- (3) É verdade que todo passeio fechado contém um ciclo?

Exercícios

- (1) Sejam 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 .

Por que isto é um resultado interessante?

- (2) Sejam 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 .

- (3) É verdade que todo passeio fechado contém um ciclo?

Exercícios

- (1) Sejam 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 .

Por que isto é um resultado interessante?

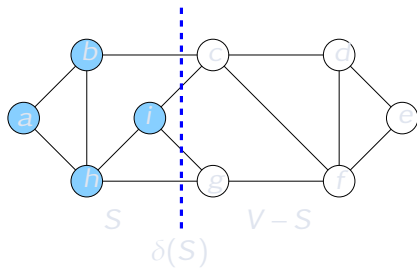
- (2) Sejam G um grafo e u, v, w vértices de G . Mostre que se em G existem um caminho de u a v e um caminho de v a w então existe um caminho de u a w em G .

- (3) É verdade que todo passeio fechado contém um ciclo?

Seja $G = (V, E)$ um grafo e seja $S \subset V$.

Denote por $\delta_G(S)$ o conjunto de arestas de G com um extremo em S e outro em $V - S$. Dizemos que $\delta_G(S)$ é um *corte*.

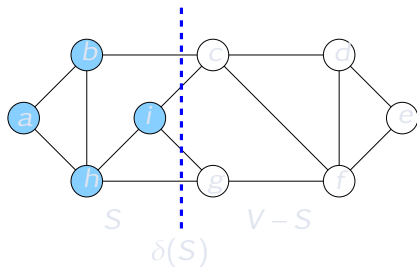
Se $s \in S$ e $t \in V - S$ dizemos que $\delta_G(S)$ *separa* s de t .



Seja $G = (V, E)$ um grafo e seja $S \subset V$.

Denote por $\delta_G(S)$ o conjunto de arestas de G com um extremo em S e outro em $V - S$. Dizemos que $\delta_G(S)$ é um *corte*.

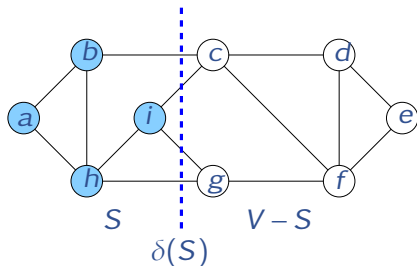
Se $s \in S$ e $t \in V - S$ dizemos que $\delta_G(S)$ *separa* s de t .



Seja $G = (V, E)$ um grafo e seja $S \subset V$.

Denote por $\delta_G(S)$ o conjunto de arestas de G com um extremo em S e outro em $V - S$. Dizemos que $\delta_G(S)$ é um *corte*.

Se $s \in S$ e $t \in V - S$ dizemos que $\delta_G(S)$ *separa* s de t .



Caminhos versus Cortes

Lema. *Seja G um grafo e sejam s, t vértices distintos de G . Então exatamente um dos seguintes ocorre:*

- (a) *existe um caminho de s a t em G , ou*
- (b) *existe um corte $\delta_G(S)$ que separa s de t tal que $\delta_G(S) = \emptyset$.*

Prova: Claramente (a) e (b) não podem valer simultaneamente (Por quê?).

Suponha que (a) não vale. Seja S o conjunto dos vértices que são alcançáveis por s em G . Obviamente, $t \in V - S$ e $\delta_G(S) = \emptyset$. □

Caminhos versus Cortes

Lema. *Seja G um grafo e sejam s, t vértices distintos de G . Então exatamente um dos seguintes ocorre:*

- (a) *existe um caminho de s a t em G , ou*
- (b) *existe um corte $\delta_G(S)$ que separa s de t tal que $\delta_G(S) = \emptyset$.*

Prova: Claramente (a) e (b) não podem valer simultaneamente (Por quê?).

Suponha que (a) não vale. Seja S o conjunto dos vértices que são alcançáveis por s em G . Obviamente, $t \in V - S$ e $\delta_G(S) = \emptyset$. □

Caminhos versus Cortes

Lema. *Seja G um grafo e sejam s, t vértices distintos de G . Então exatamente um dos seguintes ocorre:*

- (a) *existe um caminho de s a t em G , ou*
- (b) *existe um corte $\delta_G(S)$ que separa s de t tal que $\delta_G(S) = \emptyset$.*

Prova: Claramente (a) e (b) não podem valer simultaneamente (Por quê?).

Suponha que (a) não vale. Seja S o conjunto dos vértices que são alcançáveis por s em G . Obviamente, $t \in V - S$ e $\delta_G(S) = \emptyset$. □

Caminhos versus Cortes

Lema. *Seja G um grafo e sejam s, t vértices distintos de G . Então exatamente um dos seguintes ocorre:*

- (a) *existe um caminho de s a t em G , ou*
- (b) *existe um corte $\delta_G(S)$ que separa s de t tal que $\delta_G(S) = \emptyset$.*

Prova: Claramente (a) e (b) não podem valer simultaneamente (Por quê?).

Suponha que (a) não vale. Seja S o conjunto dos vértices que são alcançáveis por s em G . Obviamente, $t \in V - S$ e $\delta_G(S) = \emptyset$. □

Caminhos versus Cortes

Lema. *Seja G um grafo e sejam s, t vértices distintos de G . Então exatamente um dos seguintes ocorre:*

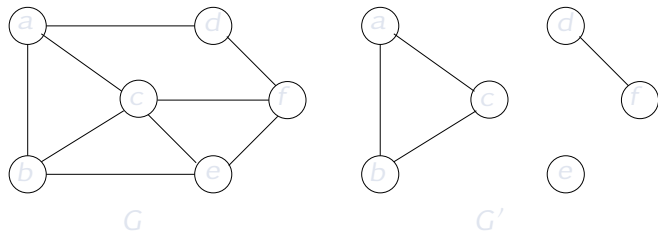
- (a) *existe um caminho de s a t em G , ou*
- (b) *existe um corte $\delta_G(S)$ que separa s de t tal que $\delta_G(S) = \emptyset$.*

Prova: Claramente (a) e (b) não podem valer simultaneamente (Por quê?).

Suponha que (a) não vale. Seja S o conjunto dos vértices que são alcançáveis por s em G . Obviamente, $t \in V - S$ e $\delta_G(S) = \emptyset$. □

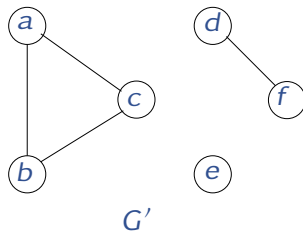
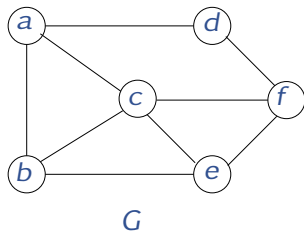
Conexidade

- ▶ Dizemos que um grafo é *conexo* se, para qualquer par de vértices u e v de G , existe um caminho de u a v em G . Caso contrário, dizemos que é *desconexo*.
- ▶ Quando o grafo G não é conexo, podemos particioná-lo em *componentes*. Dois vértices u e v de G estão no mesmo componente de G se existe um caminho de u a v em G .



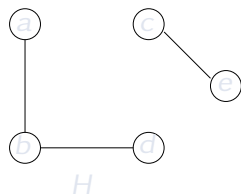
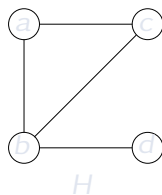
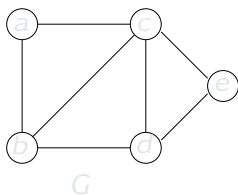
Conexidade

- ▶ Dizemos que um grafo é *conexo* se, para qualquer par de vértices u e v de G , existe um caminho de u a v em G . Caso contrário, dizemos que é *desconexo*.
- ▶ Quando o grafo G não é conexo, podemos particioná-lo em *componentes*. Dois vértices u e v de G estão no mesmo componente de G se existe um caminho de u a v em G .



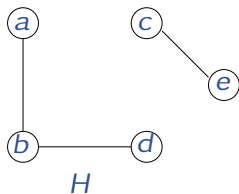
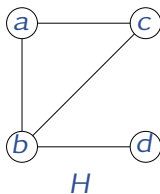
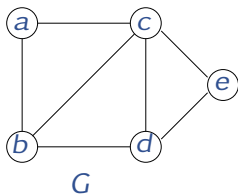
Subgrafo e Subgrafo Gerador

- ▶ Um *subgrafo* $H = (V', E')$ de um grafo $G = (V, E)$ é um grafo tal que $V' \subseteq V, E' \subseteq E$.
- ▶ Um *subgrafo gerador* de G é um subgrafo H com $V' = V$.



Subgrafo e Subgrafo Gerador

- ▶ Um *subgrafo* $H = (V', E')$ de um grafo $G = (V, E)$ é um grafo tal que $V' \subseteq V, E' \subseteq E$.
- ▶ Um *subgrafo gerador* de G é um subgrafo H com $V' = V$.



Grafos obtidos a partir de outros grafos

Seja $G = (V, E)$, e uma aresta de G e v um vértice de G .

Então

- ▶ $G - e$ é o grafo obtido de G removendo-se e . Formalmente,

$$G - e = (V, E - \{e\}).$$

- ▶ $G - v$ é o grafo obtido de G removendo-se v e todas as arestas que incidem em v . Formalmente,

$$G - v = (V - \{v\}, E - \delta(\{v\})).$$

Grafos obtidos a partir de outros grafos

Seja $G = (V, E)$, e uma aresta de G e v um vértice de G .

Então

- ▶ $G - e$ é o grafo obtido de G removendo-se e . Formalmente,

$$G - e = (V, E - \{e\}).$$

- ▶ $G - v$ é o grafo obtido de G removendo-se v e todas as arestas que incidem em v . Formalmente,

$$G - v = (V - \{v\}, E - \delta(\{v\})).$$

Subgrafo induzido

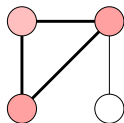
Seja $G = (V, E)$ e S um subconjunto de vértices.

Então $G[S]$ é o subgrafo de G induzido por S , que é formado por S e todas as arestas entre vértices S .

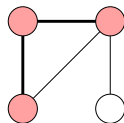
Formalmente,

$$G[S] = (S, \{(uv) \in E \mid u, v \in S\}).$$

Exemplo de um subgrafo que é induzido e de outro subgrafo que não é induzido em um grafo G :



INDUZIDO



NÃO INDUZIDO

Subgrafo induzido

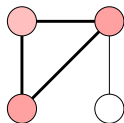
Seja $G = (V, E)$ e S um subconjunto de vértices.

Então $G[S]$ é o subgrafo de G induzido por S , que é formado por S e todas as arestas entre vértices S .

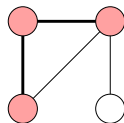
Formalmente,

$$G[S] = (S, \{(uv) \in E \mid u, v \in S\}).$$

Exemplo de um subgrafo que é induzido e de outro subgrafo que não é induzido em um grafo G :



INDUZIDO



NÃO INDUZIDO

Subgrafo induzido

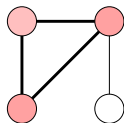
Seja $G = (V, E)$ e S um subconjunto de vértices.

Então $G[S]$ é o subgrafo de G induzido por S , que é formado por S e todas as arestas entre vértices S .

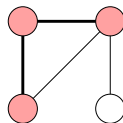
Formalmente,

$$G[S] = (S, \{(uv) \in E \mid u, v \in S\}).$$

Exemplo de um subgrafo que é induzido e de outro subgrafo que não é induzido em um grafo G :



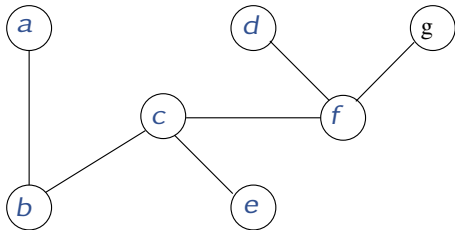
INDUZIDO



NÃO INDUZIDO

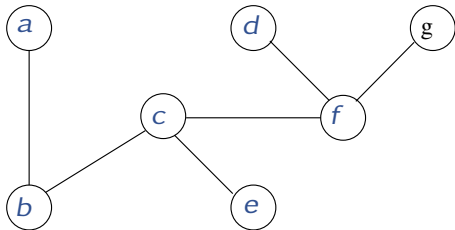
Fatos básicos de grafos

- ▶ Um grafo $G = (V, E)$ é uma *árvore* se é conexo e não possui ciclos (acíclico).



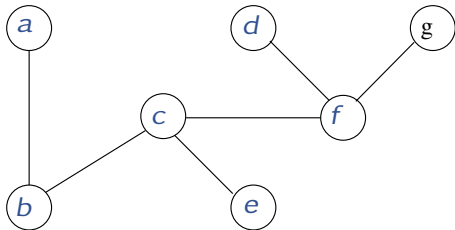
- ▶ Uma *folha* de uma árvore G é um vértice de grau 1.
- ▶ *Toda árvore com pelo menos dois vértices possui uma folha.* (Por quê?)

- ▶ Um grafo $G = (V, E)$ é uma *árvore* se é conexo e não possui ciclos (acíclico).



- ▶ Uma *folha* de uma árvore G é um vértice de grau 1.
- ▶ *Toda árvore com pelo menos dois vértices possui uma folha. (Por quê?)*

- ▶ Um grafo $G = (V, E)$ é uma *árvore* se é conexo e não possui ciclos (acíclico).



- ▶ Uma *folha* de uma árvore G é um vértice de grau 1.
- ▶ *Toda árvore com pelo menos dois vértices possui uma folha.* (Por quê?)

Teorema. *As seguintes afirmações são equivalentes:*

- ▶ *G é uma árvore.*
- ▶ *G é conexo e possui exatamente $|V| - 1$ arestas.*
- ▶ *G é conexo e a remoção de qualquer aresta desconecta o grafo (**conexo minimal**).*
- ▶ *Para todo par de vértices u, v de G , existe um único caminho de u a v em G (e G não tem laços).*

Teorema. *As seguintes afirmações são equivalentes:*

- ▶ *G é uma árvore.*
- ▶ *G é conexo e possui exatamente $|V| - 1$ arestas.*
- ▶ *G é conexo e a remoção de qualquer aresta desconecta o grafo (**conexo minimal**).*
- ▶ *Para todo par de vértices u, v de G , existe um único caminho de u a v em G (e G não tem laços).*

Teorema. *As seguintes afirmações são equivalentes:*

- ▶ *G é uma árvore.*
- ▶ *G é conexo e possui exatamente $|V| - 1$ arestas.*
- ▶ *G é conexo e a remoção de qualquer aresta desconecta o grafo (conexo minimal).*
- ▶ *Para todo par de vértices u, v de G , existe um único caminho de u a v em G (e G não tem laços).*

Teorema. *As seguintes afirmações são equivalentes:*

- ▶ *G é uma árvore.*
- ▶ *G é conexo e possui exatamente $|V| - 1$ arestas.*
- ▶ *G é conexo e a remoção de qualquer aresta desconecta o grafo (**conexo minimal**).*
- ▶ *Para todo par de vértices u, v de G , existe um único caminho de u a v em G (e G não tem laços).*

Teorema. *As seguintes afirmações são equivalentes:*

- ▶ G é uma árvore.
- ▶ G é conexo e possui exatamente $|V| - 1$ arestas.
- ▶ G é conexo e a remoção de qualquer aresta desconecta o grafo (**conexo minimal**).
- ▶ Para todo par de vértices u, v de G , existe um único caminho de u a v em G (e G não tem laços).

Grafos bipartidos

Uma *bipartição* de um conjunto V é um par (A, B) tal que:

- ▶ $A \cap B = \emptyset$ e
- ▶ $A \cup B = V$.

Um grafo $G = (V, E)$ é *bipartido* se existe uma partição (A, B) de V tal que toda aresta de G tem um extremo em A e outro em B .

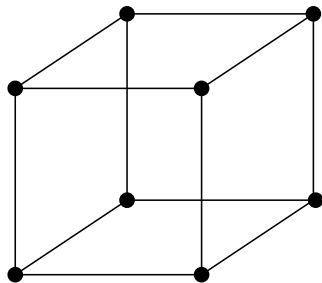
Grafos bipartidos

Uma *bipartição* de um conjunto V é um par (A, B) tal que:

- ▶ $A \cap B = \emptyset$ e
- ▶ $A \cup B = V$.

Um grafo $G = (V, E)$ é *bipartido* se existe uma partição (A, B) de V tal que toda aresta de G tem um extremo em A e outro em B .

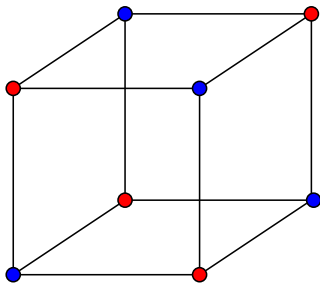
Grafos bipartidos



É bipartido?

Vamos indicar cada parte com uma cor: **vermelho** ou **azul**.

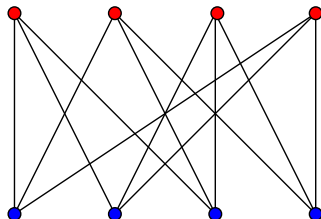
Grafos bipartidos



É bipartido!

Um grafo $G = (V, E)$ é *bipartido* se é possível colorir os vértices de G com **duas cores** de modo que vértices adjacentes tenham cores distintas.

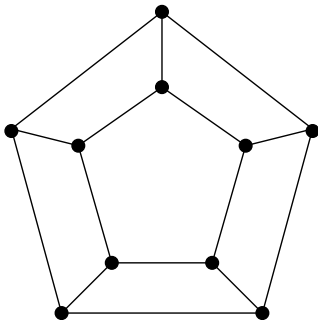
Grafos bipartidos



Isto pode ser visto melhor com outro desenho.

Um grafo $G = (V, E)$ é *bipartido* se é possível colorir os vértices de G com **duas cores** de modo que vértices adjacentes tenham cores distintas.

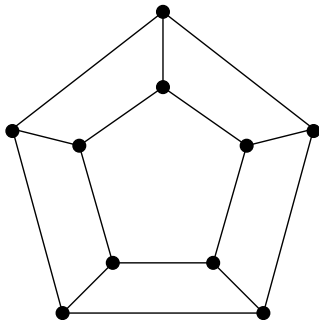
Grafos bipartidos



Este grafo **não** é bipartido.

Você consegue apresentar uma justificativa simples deste fato?

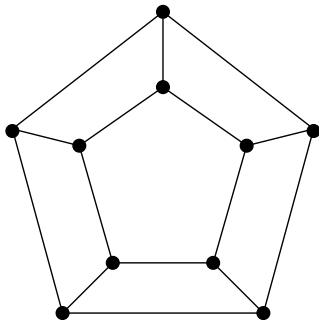
Grafos bipartidos



Um grafo bipartido **não** pode conter *ciclos ímpares* (de comprimento ímpar). Claramente, esta é uma **condição necessária**.

Ela é **suficiente**? Ou seja, é verdade que se G não contém ciclos ímpares então G é bipartido? **SIM!**

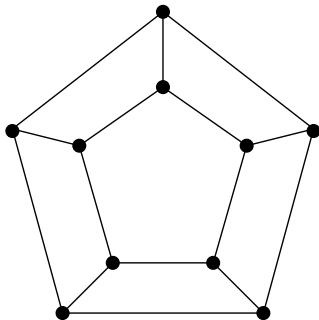
Grafos bipartidos



Um grafo bipartido **não** pode conter **ciclos ímpares** (de comprimento ímpar). Claramente, esta é uma **condição necessária**.

Ela é **suficiente**? Ou seja, é verdade que se G não contém ciclos ímpares então G é bipartido? **SIM!**

Grafos bipartidos



Um grafo bipartido **não** pode conter *ciclos ímpares* (de comprimento ímpar). Claramente, esta é uma **condição necessária**.

Ela é **suficiente**? Ou seja, é verdade que se G não contém ciclos ímpares então G é bipartido? **SIM!**

Grafos bipartidos

Queremos demonstrar o seguinte.

Teorema. *Seja G um grafo. Então G é bipartido se e somente se G não contém um ciclo ímpar.*

Já vimos que se G contém um ciclo ímpar então G não é bipartido provando a implicação " \Rightarrow ".

Falta provar a recíproca " \Leftarrow ". Para provar isto precisaremos de alguns fatos.

Podemos supor que G é conexo. (Por quê?)

Grafos bipartidos

Queremos demonstrar o seguinte.

Teorema. *Seja G um grafo. Então G é bipartido se e somente se G não contém um ciclo ímpar.*

Já vimos que se G contém um ciclo ímpar então G não é bipartido provando a implicação " \Rightarrow ".

Falta provar a recíproca " \Leftarrow ". Para provar isto precisaremos de alguns fatos.

Podemos supor que G é conexo. (Por quê?)

Grafos bipartidos

Queremos demonstrar o seguinte.

Teorema. *Seja G um grafo. Então G é bipartido se e somente se G não contém um ciclo ímpar.*

Já vimos que se G contém um ciclo ímpar então G não é bipartido provando a implicação " \Rightarrow ".

Falta provar a recíproca " \Leftarrow ". Para provar isto precisaremos de alguns fatos.

Podemos supor que G é conexo. (Por quê?)

Grafos bipartidos

Queremos demonstrar o seguinte.

Teorema. *Seja G um grafo. Então G é bipartido se e somente se G não contém um ciclo ímpar.*

Já vimos que se G contém um ciclo ímpar então G não é bipartido provando a implicação " \Rightarrow ".

Falta provar a recíproca " \Leftarrow ". Para provar isto precisaremos de alguns fatos.

Podemos supor que G é conexo. (Por quê?)

Árvore geradora

Fato 1. *Todo grafo conexo contém uma árvore geradora.*

Isto segue facilmente do próximo resultado.

Lema. *Seja G um grafo conexo e seja C um ciclo de G . Se e é uma aresta de C então $G - e$ é conexo. (Exercício!)*

A recíproca também vale.

Lema. *Seja G um grafo conexo e seja e uma aresta de G . Se $G - e$ é conexo então e pertence a algum ciclo de G . (Exercício!)*

Árvore geradora

Fato 1. *Todo grafo conexo contém uma árvore geradora.*

Isto segue facilmente do próximo resultado.

Lema. *Seja G um grafo conexo e seja C um ciclo de G . Se e é uma aresta de C então $G - e$ é conexo. (Exercício!)*

A recíproca também vale.

Lema. *Seja G um grafo conexo e seja e uma aresta de G . Se $G - e$ é conexo então e pertence a algum ciclo de G . (Exercício!)*

Fato 1. *Todo grafo conexo contém uma árvore geradora.*

Isto segue facilmente do próximo resultado.

Lema. *Seja G um grafo conexo e seja C um ciclo de G . Se e é uma aresta de C então $G - e$ é conexo. (Exercício!)*

A recíproca também vale.

Lema. *Seja G um grafo conexo e seja e uma aresta de G . Se $G - e$ é conexo então e pertence a algum ciclo de G . (Exercício!)*

Fato 2. *Toda árvore $T = (V, E)$ é um grafo bipartido.*

A prova é por indução em $|V|$ e do fato de que toda árvore com pelo menos dois vértices contém uma folha. (Exercício!)

Árvores e grafos bipartidos

Fato 2. *Toda árvore $T = (V, E)$ é um grafo bipartido.*

A prova é por indução em $|V|$ e do fato de que toda árvore com pelo menos dois vértices contém uma folha. (**Exercício!**)

Árvore geradora

Fato 3. Seja $T = (V, E')$ uma árvore geradora de um grafo $G = (V, E)$. Então para toda aresta $e \in E - E'$ existe um único ciclo em $T + e := (V, E' \cup \{e\})$.

Prova: Sejam u, v os extremos de e . Como T é uma árvore, existe um único caminho P de u a v em T . Logo, $P + e$ é o único ciclo em $T + e$. □

É comum chamar o único ciclo de $T + e$ de **ciclo fundamental** de $T + e$.

Árvore geradora

Fato 3. Seja $T = (V, E')$ uma árvore geradora de um grafo $G = (V, E)$. Então para toda aresta $e \in E - E'$ existe um único ciclo em $T + e := (V, E' \cup \{e\})$.

Prova: Sejam u, v os extremos de e . Como T é uma árvore, existe um único caminho P de u a v em T . Logo, $P + e$ é o único ciclo em $T + e$. □

É comum chamar o único ciclo de $T + e$ de **ciclo fundamental** de $T + e$.

Árvore geradora

Fato 3. Seja $T = (V, E')$ uma árvore geradora de um grafo $G = (V, E)$. Então para toda aresta $e \in E - E'$ existe um único ciclo em $T + e := (V, E' \cup \{e\})$.

Prova: Sejam u, v os extremos de e . Como T é uma árvore, existe um único caminho P de u a v em T . Logo, $P + e$ é o único ciclo em $T + e$. □

É comum chamar o único ciclo de $T + e$ de **ciclo fundamental** de $T + e$.

Grafos bipartidos

Agora estamos prontos para demonstrar o seguinte.

Teorema. *Seja G um grafo. Então G é bipartido se e somente se G não contém um ciclo ímpar.*

Prova: Já vimos que se G contém um ciclo ímpar então G não é bipartido provando a implicação " \Rightarrow ".

Agora provaremos a recíproca " \Leftarrow ". Suponha que G não contém um ciclo ímpar. Construiremos uma bipartição (A, B) de V tal que toda aresta de G tem um extremo em A e outro em B .

Podemos supor que G é conexo.

Grafos bipartidos

Agora estamos prontos para demonstrar o seguinte.

Teorema. *Seja G um grafo. Então G é bipartido se e somente se G não contém um ciclo ímpar.*

Prova: Já vimos que se G contém um ciclo ímpar então G não é bipartido provando a implicação " \Rightarrow ".

Agora provaremos a recíproca " \Leftarrow ". Suponha que G não contém um ciclo ímpar. Construiremos uma bipartição (A, B) de V tal que toda aresta de G tem um extremo em A e outro em B .

Podemos supor que G é conexo.

Grafos bipartidos

Agora estamos prontos para demonstrar o seguinte.

Teorema. *Seja G um grafo. Então G é bipartido se e somente se G não contém um ciclo ímpar.*

Prova: Já vimos que se G contém um ciclo ímpar então G não é bipartido provando a implicação " \Rightarrow ".

Agora provaremos a recíproca " \Leftarrow ". Suponha que G não contém um ciclo ímpar. Construiremos uma bipartição (A, B) de V tal que toda aresta de G tem um extremo em A e outro em B .

Podemos supor que G é conexo.

Grafos bipartidos

Agora estamos prontos para demonstrar o seguinte.

Teorema. *Seja G um grafo. Então G é bipartido se e somente se G não contém um ciclo ímpar.*

Prova: Já vimos que se G contém um ciclo ímpar então G não é bipartido provando a implicação " \Rightarrow ".

Agora provaremos a recíproca " \Leftarrow ". Suponha que G não contém um ciclo ímpar. Construiremos uma bipartição (A, B) de V tal que toda aresta de G tem um extremo em A e outro em B .

Podemos supor que G é conexo.

Pelo Fato 1, G contém uma árvore geradora $T = (V, E')$.

Pelo Fato 2, T possui uma bipartição (A, B) de V tal que toda aresta de T tem um extremo em A e outro em B .

Mostraremos que toda aresta de $E - E'$ tem um extremo em A e outro em B , o que implica que G é bipartido.

Pelo Fato 1, G contém uma árvore geradora $T = (V, E')$.

Pelo Fato 2, T possui uma bipartição (A, B) de V tal que toda aresta de T tem um extremo em A e outro em B .

Mostraremos que toda aresta de $E - E'$ tem um extremo em A e outro em B , o que implica que G é bipartido.

Grafos bipartidos

Pelo Fato 1, G contém uma árvore geradora $T = (V, E')$.

Pelo Fato 2, T possui uma bipartição (A, B) de V tal que toda aresta de T tem um extremo em A e outro em B .

Mostraremos que toda aresta de $E - E'$ tem um extremo em A e outro em B , o que implica que G é bipartido.

Seja e uma aresta de $E - E'$. Pelo Fato 3, existe um único ciclo C em $T + e$ (que contém e).

- ▶ Se e tem extremos em partes distintas, então nada há a fazer.
- ▶ Se os extremos de e pertencem a mesma parte (A ou B), então C é um ciclo ímpar (contradição)!

O resultado segue. □

Seja e uma aresta de $E - E'$. Pelo Fato 3, existe um único ciclo C em $T + e$ (que contém e).

- ▶ Se e tem extremos em partes distintas, então nada há a fazer.
- ▶ Se os extremos de e pertencem a mesma parte (A ou B), então C é um ciclo ímpar (contradição)!

O resultado segue. □

Seja e uma aresta de $E - E'$. Pelo Fato 3, existe um único ciclo C em $T + e$ (que contém e).

- ▶ Se e tem extremos em partes distintas, então nada há a fazer.
- ▶ Se os extremos de e pertencem a mesma parte (A ou B), então C é um ciclo ímpar (contradição)!

O resultado segue. □

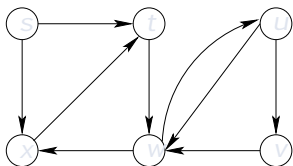
Seja e uma aresta de $E - E'$. Pelo Fato 3, existe um único ciclo C em $T + e$ (que contém e).

- ▶ Se e tem extremos em partes distintas, então nada há a fazer.
- ▶ Se os extremos de e pertencem a mesma parte (A ou B), então C é um ciclo ímpar (contradição)!

O resultado segue. □

Grafo direcionado

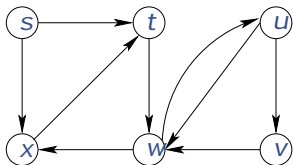
- ▶ As definições que vimos até agora são para grafos *não direcionados*.
- ▶ Um *grafo direcionado* é definido de forma semelhante, com a diferença que as *arestas* (às vezes chamadas de *arcos*) consistem de *pares ordenados* de vértices.



- ▶ Às vezes, para enfatizar, dizemos *grafo não direcionado* em vez de simplesmente *grafo*.

Grafo direcionado

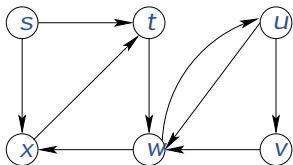
- ▶ As definições que vimos até agora são para grafos *não direcionados*.
- ▶ Um *grafo direcionado* é definido de forma semelhante, com a diferença que as *arestas* (às vezes chamadas de *arcos*) consistem de *pares ordenados* de vértices.



- ▶ Às vezes, para enfatizar, dizemos *grafo não direcionado* em vez de simplesmente *grafo*.

Grafo direcionado

- ▶ As definições que vimos até agora são para grafos *não direcionados*.
- ▶ Um *grafo direcionado* é definido de forma semelhante, com a diferença que as *arestas* (às vezes chamadas de *arcos*) consistem de *pares ordenados* de vértices.



- ▶ Às vezes, para enfatizar, dizemos *grafo não direcionado* em vez de simplesmente *grafo*.

Grafo direcionado

- ▶ Se $e = (u, v)$ é uma aresta de um grafo direcionado G , então dizemos que e *sai* de u e *entra* em v , u é a *cauda* de e e v é *cabeça* de e .
- ▶ O *grau de saída* $g^+(v)$ de um vértice v é o número de arestas que saem de v . O *grau de entrada* $g^-(v)$ de v é o número de arestas que entram em v .

Teorema.

Para todo grafo direcionado $G = (V, E)$ temos:

$$\sum_{v \in V} g^+(v) = \sum_{v \in V} g^-(v) = |E|.$$

Grafo direcionado

- ▶ Se $e = (u, v)$ é uma aresta de um grafo direcionado G , então dizemos que e *sai* de u e *entra* em v , u é a *cauda* de e e v é *cabeça* de e .
- ▶ O *grau de saída* $g^+(v)$ de um vértice v é o número de arestas que saem de v . O *grau de entrada* $g^-(v)$ de v é o número de arestas que entram em v .

Teorema.

Para todo grafo direcionado $G = (V, E)$ temos:

$$\sum_{v \in V} g^+(v) = \sum_{v \in V} g^-(v) = |E|.$$

Grafo direcionado

- ▶ Se $e = (u, v)$ é uma aresta de um grafo direcionado G , então dizemos que e *sai* de u e *entra* em v , u é a *cauda* de e e v é *cabeça* de e .
- ▶ O *grau de saída* $g^+(v)$ de um vértice v é o número de arestas que saem de v . O *grau de entrada* $g^-(v)$ de v é o número de arestas que entram em v .

Teorema.

Para todo grafo direcionado $G = (V, E)$ temos:

$$\sum_{v \in V} g^+(v) = \sum_{v \in V} g^-(v) = |E|.$$

Grafo direcionado

- ▶ Se $e = (u, v)$ é uma aresta de um grafo direcionado G , então dizemos que e *sai* de u e *entra* em v , u é a *cauda* de e e v é *cabeça* de e .
- ▶ O *grau de saída* $g^+(v)$ de um vértice v é o número de arestas que saem de v . O *grau de entrada* $g^-(v)$ de v é o número de arestas que entram em v .

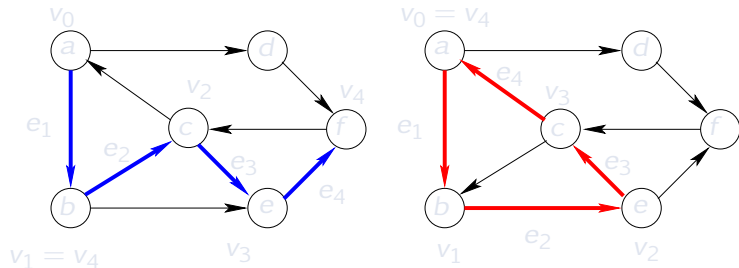
Teorema.

Para todo grafo direcionado $G = (V, E)$ temos:

$$\sum_{v \in V} g^+(v) = \sum_{v \in V} g^-(v) = |E|.$$

Passeios em grafos direcionados.

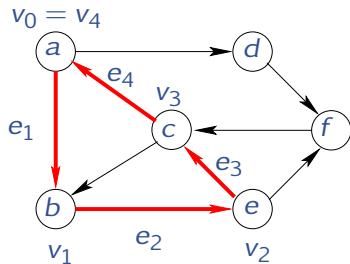
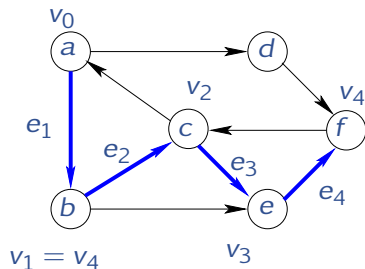
- Supõe-se que passeios, caminhos e ciclos de grafos direcionados são **direcionados** (todas as arestas “seguem o mesmo sentido”).



- Podemos definir subgrafo (gerador) de um grafo direcionado de modo análogo ao caso não direcionado. Há também um conceito de **conexidade** para grafos direcionados que veremos mais tarde.

Passeios em grafos direcionados.

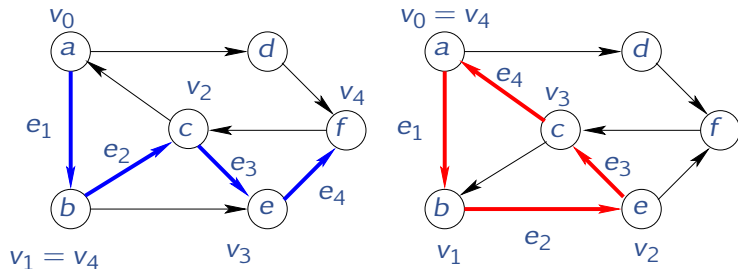
- Supõe-se que passeios, caminhos e ciclos de grafos direcionados são **direcionados** (todas as arestas “seguem o mesmo sentido”).



- Podemos definir subgrafo (gerador) de um grafo direcionado de modo análogo ao caso não direcionado. Há também um conceito de **conexidade** para grafos direcionados que veremos mais tarde.

Passeios em grafos direcionados.

- Supõe-se que passeios, caminhos e ciclos de grafos direcionados são **direcionados** (todas as arestas “seguem o mesmo sentido”).



- Podemos definir subgrafo (gerador) de um grafo direcionado de modo análogo ao caso não direcionado. Há também um conceito de **conexidade** para grafos direcionados que veremos mais tarde.

Exercícios

- (1) Sejam G um grafo direcionado 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 .
- (2) Sejam G um grafo direcionado 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 .
- (3) É verdade que todo passeio fechado em um grafo direcionado contém um ciclo (direcionado)?

Exercícios

- (1) Sejam G um grafo direcionado 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 .
- (2) Sejam G um grafo direcionado 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 .
- (3) É verdade que todo passeio fechado em um grafo direcionado contém um ciclo (direcionado)?

Exercícios

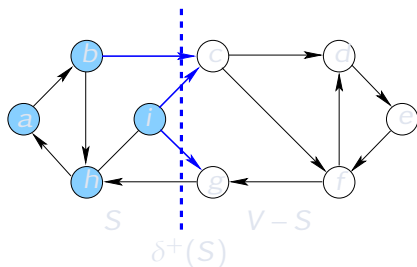
- (1) Sejam G um grafo direcionado e u, v vértices de G . Mostre que se existe um passeio de u a v em G , então existe um caminho de u a v em G .
- (2) Sejam G um grafo direcionado e u, v, w vértices de G . Mostre que se em G existem um caminho de u a v e um caminho de v a w então existe um caminho de u a w em G .
- (3) É verdade que todo passeio fechado em um grafo direcionado contém um ciclo (direcionado)?

Cortes em grafos direcionados

Seja $G = (V, E)$ um grafo direcionado e seja $S \subset V$.

Denote por $\delta_G^+(S)$ o conjunto de arestas de G com cauda em S e cabeça em $V - S$. Dizemos que $\delta_G^+(S)$ é um *corte direcionado*.

Se $s \in S$ e $t \in V - S$ dizemos que $\delta_G^+(S)$ *separa* s de t .



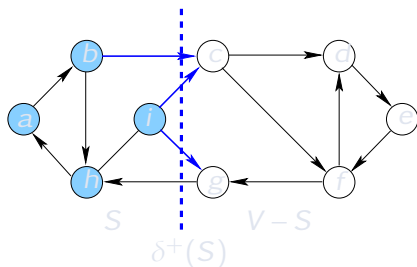
Note que (g, h) não pertence a $\delta_G^+(S)$.

Cortes em grafos direcionados

Seja $G = (V, E)$ um grafo direcionado e seja $S \subset V$.

Denote por $\delta_G^+(S)$ o conjunto de arestas de G com cauda em S e cabeça em $V - S$. Dizemos que $\delta_G^+(S)$ é um *corte direcionado*.

Se $s \in S$ e $t \in V - S$ dizemos que $\delta_G^+(S)$ *separa* s de t .



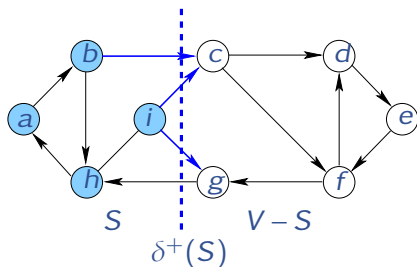
Note que (g, h) não pertence a $\delta_G^+(S)$.

Cortes em grafos direcionados

Seja $G = (V, E)$ um grafo direcionado e seja $S \subset V$.

Denote por $\delta_G^+(S)$ o conjunto de arestas de G com cauda em S e cabeça em $V - S$. Dizemos que $\delta_G^+(S)$ é um *corte direcionado*.

Se $s \in S$ e $t \in V - S$ dizemos que $\delta_G^+(S)$ *separa* s de t .



Note que (g, h) não pertence a $\delta_G^+(S)$.

Caminhos versus Cortes

Lema. *Seja G um grafo direcionado e sejam s, t vértices distintos de G . Então exatamente um dos seguintes ocorre:*

- (a) *existe um caminho de s a t em G , ou*
- (b) *existe um corte direcionado $\delta_G^+(S)$ que separa s de t tal que $\delta_G^+(S) = \emptyset$.*

Prova: Claramente (a) e (b) não podem valer simultaneamente (Por quê?).

Suponha que (a) não vale. Seja S o conjunto dos vértices que são alcançáveis por s em G . Obviamente, $t \in V - S$ e $\delta_G^+(S) = \emptyset$. □

Caminhos versus Cortes

Lema. *Seja G um grafo direcionado e sejam s, t vértices distintos de G . Então exatamente um dos seguintes ocorre:*

- (a) *existe um caminho de s a t em G , ou*
- (b) *existe um corte direcionado $\delta_G^+(S)$ que separa s de t tal que $\delta_G^+(S) = \emptyset$.*

Prova: Claramente (a) e (b) não podem valer simultaneamente (Por quê?).

Suponha que (a) não vale. Seja S o conjunto dos vértices que são alcançáveis por s em G . Obviamente, $t \in V - S$ e $\delta_G^+(S) = \emptyset$.

□

Caminhos versus Cortes

Lema. *Seja G um grafo direcionado e sejam s, t vértices distintos de G . Então exatamente um dos seguintes ocorre:*

- (a) *existe um caminho de s a t em G , ou*
- (b) *existe um corte direcionado $\delta_G^+(S)$ que separa s de t tal que $\delta_G^+(S) = \emptyset$.*

Prova: Claramente (a) e (b) não podem valer simultaneamente (Por quê?).

Suponha que (a) não vale. Seja S o conjunto dos vértices que são alcançáveis por s em G . Obviamente, $t \in V - S$ e $\delta_G^+(S) = \emptyset$. □

Caminhos versus Cortes

Lema. *Seja G um grafo direcionado e sejam s, t vértices distintos de G . Então exatamente um dos seguintes ocorre:*

- (a) *existe um caminho de s a t em G , ou*
- (b) *existe um corte direcionado $\delta_G^+(S)$ que separa s de t tal que $\delta_G^+(S) = \emptyset$.*

Prova: Claramente (a) e (b) não podem valer simultaneamente (Por quê?).

Suponha que (a) não vale. Seja S o conjunto dos vértices que são alcançáveis por s em G . Obviamente, $t \in V - S$ e $\delta_G^+(S) = \emptyset$. □

Caminhos versus Cortes

Lema. *Seja G um grafo direcionado e sejam s, t vértices distintos de G . Então exatamente um dos seguintes ocorre:*

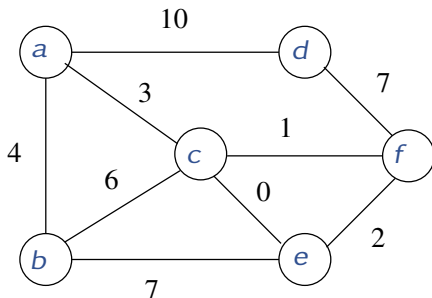
- (a) *existe um caminho de s a t em G , ou*
- (b) *existe um corte direcionado $\delta_G^+(S)$ que separa s de t tal que $\delta_G^+(S) = \emptyset$.*

Prova: Claramente (a) e (b) não podem valer simultaneamente (Por quê?).

Suponha que (a) não vale. Seja S o conjunto dos vértices que são alcançáveis por s em G . Obviamente, $t \in V - S$ e $\delta_G^+(S) = \emptyset$. □

Grafo Ponderado

- Um grafo (direcionado ou não) é *ponderado* se a cada aresta e do grafo está associado um valor real $c(e)$, o qual denominamos *custo (ou peso)* da aresta.



Representação de grafos

Algoritmos em Grafos - Motivação prática

- ▶ Grafos são estruturas abstratas que podem modelar diversos problemas do mundo real.
- ▶ Por exemplo, um grafo pode representar conexões entre cidades por estradas ou uma rede de computadores.
- ▶ O interesse em estudar algoritmos para problemas em grafos é que conhecer um algoritmo para um determinado problema em grafos pode significar conhecer algoritmos para diversos problemas reais.

Algoritmos em Grafos - Motivação prática

- ▶ Grafos são estruturas abstratas que podem modelar diversos problemas do mundo real.
- ▶ Por exemplo, um grafo pode representar conexões entre cidades por estradas ou uma rede de computadores.
- ▶ O interesse em estudar algoritmos para problemas em grafos é que conhecer um algoritmo para um determinado problema em grafos pode significar conhecer algoritmos para diversos problemas reais.

Algoritmos em Grafos - Motivação prática

- ▶ Grafos são estruturas abstratas que podem modelar diversos problemas do mundo real.
- ▶ Por exemplo, um grafo pode representar conexões entre cidades por estradas ou uma rede de computadores.
- ▶ O interesse em estudar algoritmos para problemas em grafos é que conhecer um algoritmo para um determinado problema em grafos pode significar conhecer algoritmos para diversos problemas reais.

Aplicações

- ▶ *Problema do Caminho Mínimo:* dado um conjunto de cidades, as distâncias entre elas e duas cidades A e B , determinar um **caminho (trajeto) mais curto** de A até B .
- ▶ *Problema da Árvore Geradora de Peso Mínimo:* dado um conjunto de computadores, onde cada par de computadores pode ser ligado usando uma quantidade de fibra ótica, encontrar **uma rede interconectando todos os computadores** que use a menor quantidade de fibra ótica possível.
- ▶ *Problema do Emparelhamento Máximo:* dado um conjunto de pessoas e um conjunto de vagas para diferentes empregos, onde cada pessoa é qualificada para certos empregos e cada vaga deve ser ocupada por exatamente uma pessoa, encontrar um **conjunto de associações pessoa-emprego** que tenha o maior número possível de pessoas.

Aplicações

- ▶ *Problema do Caminho Mínimo:* dado um conjunto de cidades, as distâncias entre elas e duas cidades A e B , determinar um **caminho (trajeto) mais curto** de A até B .
- ▶ *Problema da Árvore Geradora de Peso Mínimo:* dado um conjunto de computadores, onde cada par de computadores pode ser ligado usando uma quantidade de fibra ótica, encontrar **uma rede interconectando todos os computadores** que use a menor quantidade de fibra ótica possível.
- ▶ *Problema do Emparelhamento Máximo:* dado um conjunto de pessoas e um conjunto de vagas para diferentes empregos, onde cada pessoa é qualificada para certos empregos e cada vaga deve ser ocupada por exatamente uma pessoa, encontrar um **conjunto de associações pessoa-emprego** que tenha o maior número possível de pessoas.

Aplicações

- ▶ *Problema do Caminho Mínimo:* dado um conjunto de cidades, as distâncias entre elas e duas cidades A e B , determinar um **caminho (trajeto) mais curto** de A até B .
- ▶ *Problema da Árvore Geradora de Peso Mínimo:* dado um conjunto de computadores, onde cada par de computadores pode ser ligado usando uma quantidade de fibra ótica, encontrar **uma rede interconectando todos os computadores** que use a menor quantidade de fibra ótica possível.
- ▶ *Problema do Emparelhamento Máximo:* dado um conjunto de pessoas e um conjunto de vagas para diferentes empregos, onde cada pessoa é qualificada para certos empregos e cada vaga deve ser ocupada por exatamente uma pessoa, encontrar um **conjunto de associações pessoa-emprego** que tenha o maior número possível de pessoas.

Aplicações

- ▶ *Problema do Caminho Mínimo:* dado um conjunto de cidades, as distâncias entre elas e duas cidades A e B , determinar um **caminho (trajeto) mais curto** de A até B .
- ▶ *Problema da Árvore Geradora de Peso Mínimo:* dado um conjunto de computadores, onde cada par de computadores pode ser ligado usando uma quantidade de fibra ótica, encontrar **uma rede interconectando todos os computadores** que use a menor quantidade de fibra ótica possível.
- ▶ *Problema do Emparelhamento Máximo:* dado um conjunto de pessoas e um conjunto de vagas para diferentes empregos, onde cada pessoa é qualificada para certos empregos e cada vaga deve ser ocupada por exatamente uma pessoa, encontrar um **conjunto de associações pessoa-emprego** que tenha o maior número possível de pessoas.

- ▶ *Problema do Caixeiro Viajante:* dado um conjunto de cidades, encontrar um **ciclo que passa por todas as cidades** tal que a distância total percorrida seja menor possível.
- ▶ *Problema Chinês do Correio:* dado o conjunto das ruas de um bairro, encontrar um **passeio fechado que passa por todas as ruas** tal que a distância total percorrida seja menor possível.

- ▶ *Problema do Caixeiro Viajante:* dado um conjunto de cidades, encontrar um **ciclo que passa por todas as cidades** tal que a distância total percorrida seja menor possível.
- ▶ *Problema Chinês do Correio:* dado o conjunto das ruas de um bairro, encontrar um **passeio fechado que passa por todas as ruas** tal que a distância total percorrida seja menor possível.

- ▶ *Problema do Caixeiro Viajante*: dado um conjunto de cidades, encontrar um **ciclo que passa por todas as cidades** tal que a distância total percorrida seja menor possível.
- ▶ *Problema Chinês do Correio*: dado o conjunto das ruas de um bairro, encontrar um **passeio fechado que passa por todas as ruas** tal que a distância total percorrida seja menor possível.

Representação Interna de Grafos

- ▶ A complexidade dos algoritmos para solução de problemas modelados por grafos depende fortemente da sua representação interna.
- ▶ Existem duas representações canônicas: *matriz de adjacência* e *listas de adjacência*.
- ▶ O uso de uma ou outra num determinado algoritmo depende da natureza das operações que ditam a complexidade do algoritmo.

Representação Interna de Grafos

- ▶ A complexidade dos algoritmos para solução de problemas modelados por grafos depende fortemente da sua representação interna.
- ▶ Existem duas representações canônicas: *matriz de adjacência* e *listas de adjacência*.
- ▶ O uso de uma ou outra num determinado algoritmo depende da natureza das operações que ditam a complexidade do algoritmo.

Representação Interna de Grafos

- ▶ A complexidade dos algoritmos para solução de problemas modelados por grafos depende fortemente da sua representação interna.
- ▶ Existem duas representações canônicas: *matriz de adjacência* e *listas de adjacência*.
- ▶ O uso de uma ou outra num determinado algoritmo depende da natureza das operações que ditam a complexidade do algoritmo.

Matriz de adjacência

- ▶ Seja $G = (V, E)$ um grafo simples (direcionado ou não).
- ▶ A *matriz de adjacência* de G é uma matriz quadrada A de ordem $|V|$, cujas linhas e colunas são indexadas pelos vértices em V , e tal que:

$$A[i, j] = \begin{cases} 1 & \text{se } (i, j) \in E, \\ 0 & \text{caso contrário.} \end{cases}$$

- ▶ Note que se G é não direcionado, então a matriz A correspondente é simétrica.

Matriz de adjacência

- ▶ Seja $G = (V, E)$ um grafo simples (direcionado ou não).
- ▶ A *matriz de adjacência* de G é uma matriz quadrada A de ordem $|V|$, cujas linhas e colunas são indexadas pelos vértices em V , e tal que:

$$A[i, j] = \begin{cases} 1 & \text{se } (i, j) \in E, \\ 0 & \text{caso contrário.} \end{cases}$$

- ▶ Note que se G é não direcionado, então a matriz A correspondente é simétrica.

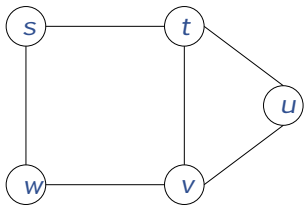
Matriz de adjacência

- ▶ Seja $G = (V, E)$ um grafo simples (direcionado ou não).
- ▶ A *matriz de adjacência* de G é uma matriz quadrada A de ordem $|V|$, cujas linhas e colunas são indexadas pelos vértices em V , e tal que:

$$A[i, j] = \begin{cases} 1 & \text{se } (i, j) \in E, \\ 0 & \text{caso contrário.} \end{cases}$$

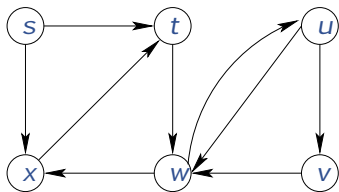
- ▶ Note que se G é não direcionado, então a matriz A correspondente é simétrica.

Matriz de adjacência



	s	t	w	v	u
s	0	1	1	0	0
t	1	0	0	1	1
w	1	0	0	1	0
v	0	1	1	0	1
u	0	1	0	1	0

Matriz de adjacência



	s	t	u	x	w	v
s	0	1	0	1	0	0
t	0	0	0	0	1	0
u	0	0	0	0	1	1
x	0	1	0	0	0	0
w	0	0	1	1	0	0
v	0	0	0	0	1	0

Matriz de adjacência em C

Em C para representar os vértices de um grafo G com n vértices usamos $0, 1, 2, \dots, n-2, n-1$.

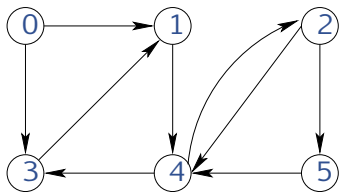
A declaração de uma matriz poderia ser:

```
#define NMAX 100
```

```
unsigned char A[NMAX]; /* matriz estática */
```

ou

```
unsigned char **A; /* matriz dinâmica */
```



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

Listas de adjacência

- ▶ Seja $G = (V, E)$ um grafo simples (direcionado ou não).
- ▶ A representação de G por uma *lista de adjacências* consiste no seguinte.

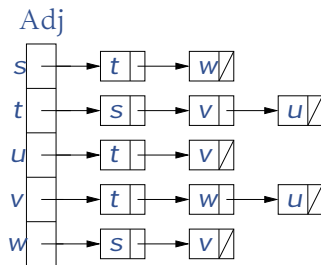
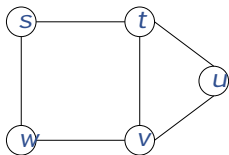
Para cada vértice v , temos uma lista ligada $Adj[v]$ dos vértices *adjacentes* a v , ou seja, w aparece em $Adj[v]$ se (v, w) é uma aresta de G . Os vértices podem estar em qualquer ordem em uma lista.

Listas de adjacência

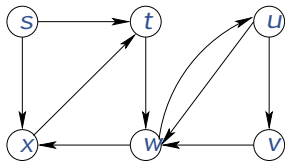
- ▶ Seja $G = (V, E)$ um grafo simples (direcionado ou não).
- ▶ A representação de G por uma *lista de adjacências* consiste no seguinte.

Para cada vértice v , temos uma lista ligada $Adj[v]$ dos vértices *adjacentes* a v , ou seja, w aparece em $Adj[v]$ se (v, w) é uma aresta de G . Os vértices podem estar em qualquer ordem em uma lista.

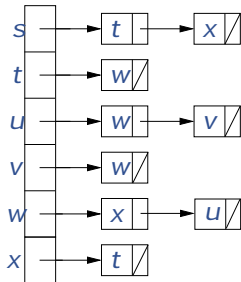
Listas de adjacência



Lista de adjacências

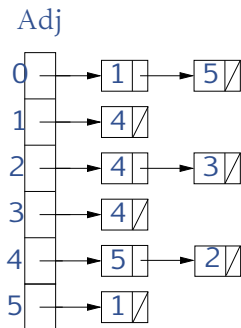
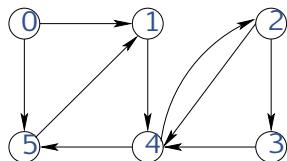


Adj



Listas de adjacências em C

```
#define NMAX 100
typedef struct SVertex SVertex;
struct SVertex {
    int vert;
    SVertex *next;
}
SVertex *Adj[NMAX]; /* Vetor estático */
```



Notação para complexidade de algoritmos

Quando analisarmos a complexidade de um algoritmo envolvendo um grafo $G = (V, E)$ usaremos V e E na **notação assintótica**, em vez de $|V|$ e $|E|$.

Por exemplo, escrevemos $O(E^2 \lg V)$ em vez de $O(|E|^2 \lg |V|)$.

Notação para complexidade de algoritmos

Quando analisarmos a complexidade de um algoritmo envolvendo um grafo $G = (V, E)$ usaremos V e E na **notação assintótica**, em vez de $|V|$ e $|E|$.

Por exemplo, escrevemos $O(E^2 \lg V)$ em vez de $O(|E|^2 \lg |V|)$.

Matriz × Lista de adjacência

O que é melhor? **Depende.**

- ▶ Matriz de adjacência: é fácil verificar se (i, j) é uma aresta de G .
- ▶ Lista de adjacência: é fácil descobrir os vértices adjacentes a um dado vértice v (ou seja, listar $\text{Adj}[v]$).
- ▶ Matriz de adjacência: espaço $\Theta(V^2)$.
Adequada a grafos densos ($|E| = \Theta(V^2)$).
- ▶ Lista de adjacência: espaço $\Theta(V + E)$.
Adequada a grafos esparsos ($|E| = \Theta(V)$).

Matriz × Lista de adjacência

O que é melhor? **Depende.**

- ▶ Matriz de adjacência: é fácil verificar se (i, j) é uma aresta de G .
- ▶ Lista de adjacência: é fácil descobrir os vértices adjacentes a um dado vértice v (ou seja, listar $\text{Adj}[v]$).
- ▶ Matriz de adjacência: espaço $\Theta(V^2)$.
Adequada a grafos densos ($|E| = \Theta(V^2)$).
- ▶ Lista de adjacência: espaço $\Theta(V + E)$.
Adequada a grafos esparsos ($|E| = \Theta(V)$).

Matriz × Lista de adjacência

O que é melhor? **Depende.**

- ▶ Matriz de adjacência: é fácil verificar se (i, j) é uma aresta de G .
- ▶ Lista de adjacência: é fácil descobrir os vértices adjacentes a um dado vértice v (ou seja, listar $\text{Adj}[v]$).
- ▶ Matriz de adjacência: espaço $\Theta(V^2)$.
Adequada a grafos densos ($|E| = \Theta(V^2)$).
- ▶ Lista de adjacência: espaço $\Theta(V + E)$.
Adequada a grafos esparsos ($|E| = \Theta(V)$).

Matriz × Lista de adjacência

O que é melhor? **Depende.**

- ▶ Matriz de adjacência: é fácil verificar se (i, j) é uma aresta de G .
- ▶ Lista de adjacência: é fácil descobrir os vértices adjacentes a um dado vértice v (ou seja, listar $\text{Adj}[v]$).
- ▶ Matriz de adjacência: espaço $\Theta(V^2)$.
Adequada a grafos densos ($|E| = \Theta(V^2)$).
- ▶ Lista de adjacência: espaço $\Theta(V + E)$.
Adequada a grafos esparsos ($|E| = \Theta(V)$).

Matriz × Lista de adjacência

O que é melhor? **Depende.**

- ▶ Matriz de adjacência: é fácil verificar se (i, j) é uma aresta de G .
- ▶ Lista de adjacência: é fácil descobrir os vértices adjacentes a um dado vértice v (ou seja, listar $\text{Adj}[v]$).
- ▶ Matriz de adjacência: espaço $\Theta(V^2)$.
Adequada a grafos densos ($|E| = \Theta(V^2)$).
- ▶ Lista de adjacência: espaço $\Theta(V + E)$.
Adequada a grafos esparsos ($|E| = \Theta(V)$).

Matriz × Lista de adjacência

O que é melhor? **Depende.**

- ▶ Matriz de adjacência: é fácil verificar se (i, j) é uma aresta de G .
- ▶ Lista de adjacência: é fácil descobrir os vértices adjacentes a um dado vértice v (ou seja, listar $\text{Adj}[v]$).
- ▶ Matriz de adjacência: espaço $\Theta(V^2)$.
Adequada a **grafos densos** ($|E| = \Theta(V^2)$).
- ▶ Lista de adjacência: espaço $\Theta(V + E)$.
Adequada a **grafos esparsos** ($|E| = \Theta(V)$).

- ▶ Há outras alternativas para representar grafos, mas matrizes e listas de adjacência são as mais usadas.
- ▶ Elas podem ser adaptadas para representar grafos ponderados, grafos com laços e arestas múltiplas, grafos com pesos nos vértices etc.
- ▶ Para determinados problemas é essencial ter estruturas de dados adicionais para melhorar a eficiência dos algoritmos.

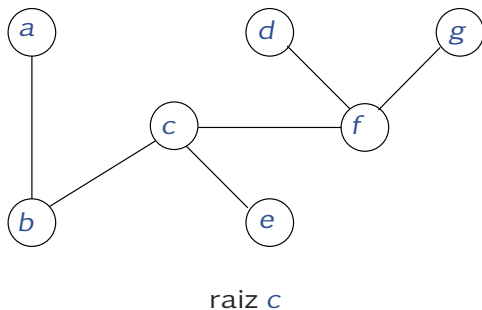
- ▶ Há outras alternativas para representar grafos, mas matrizes e listas de adjacência são as mais usadas.
- ▶ Elas podem ser adaptadas para representar grafos ponderados, grafos com laços e arestas múltiplas, grafos com pesos nos vértices etc.
- ▶ Para determinados problemas é essencial ter estruturas de dados adicionais para melhorar a eficiência dos algoritmos.

- ▶ Há outras alternativas para representar grafos, mas matrizes e listas de adjacência são as mais usadas.
- ▶ Elas podem ser adaptadas para representar grafos ponderados, grafos com laços e arestas múltiplas, grafos com pesos nos vértices etc.
- ▶ Para determinados problemas é essencial ter estruturas de dados adicionais para melhorar a eficiência dos algoritmos.

- ▶ Há outras alternativas para representar grafos, mas matrizes e listas de adjacência são as mais usadas.
- ▶ Elas podem ser adaptadas para representar grafos ponderados, grafos com laços e arestas múltiplas, grafos com pesos nos vértices etc.
- ▶ Para determinados problemas é essencial ter estruturas de dados adicionais para melhorar a eficiência dos algoritmos.

Representação de árvores

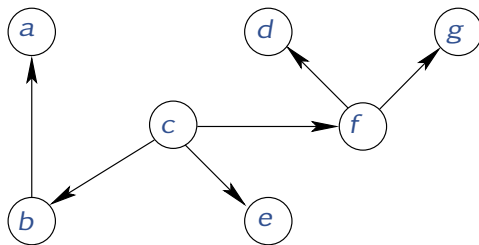
Uma *árvore enraizada* é uma árvore com um vértice especial chamado *raiz*.



Representação de árvores

Uma *árvore direcionada com raiz r* é um grafo direcionado acíclico $T = (V, E)$ tal que:

1. $g^-(r) = 0$,
2. $g^-(v) = 1$ para $v \in V - \{r\}$.



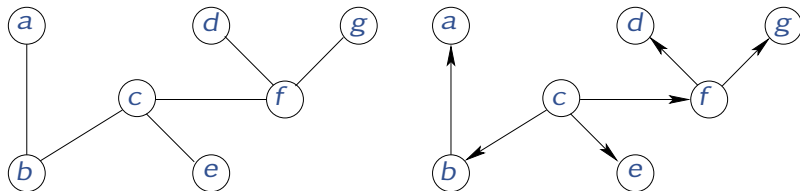
raiz c

Representação de árvores

Vetor de predecessores π (ou outro nome):

v	a	b	c	d	e	f	g
$\pi[v]$	b	c	N	f	c	c	f

N é um símbolo usado para indicar não existência.



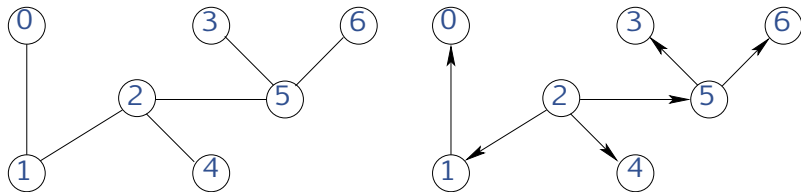
raiz c

Representação de árvores em C

Vetor de predecessores π :

v	0	1	2	3	4	5	6
$\pi[v]$	1	2	N	5	2	2	5

N é um símbolo usado para indicar não existência. Por exemplo -1 , NULL etc.



raiz 2

Alguns detalhes de implementação

- ▶ Nos algoritmos que veremos durante o semestre, usa-se a representação de um grafo (direcionado ou não) por **listas de adjacências**.
- ▶ Em uma implementação de verdade de um algoritmo, o mais provável é que esta representação não seja dada a priori.
- ▶ Em geral é necessário construir tal representação a partir da entrada dada. Como fazer isto depende do formato da entrada.

Alguns detalhes de implementação

- ▶ Nos algoritmos que veremos durante o semestre, usa-se a representação de um grafo (direcionado ou não) por **listas de adjacências**.
- ▶ Em uma implementação de verdade de um algoritmo, o mais provável é que esta representação não seja dada a priori.
- ▶ Em geral é necessário construir tal representação a partir da entrada dada. Como fazer isto depende do formato da entrada.

Alguns detalhes de implementação

- ▶ Nos algoritmos que veremos durante o semestre, usa-se a representação de um grafo (direcionado ou não) por **listas de adjacências**.
- ▶ Em uma implementação de verdade de um algoritmo, o mais provável é que esta representação **não** seja dada a priori.
- ▶ Em geral é necessário construir tal representação a partir da entrada dada. Como fazer isto depende do formato da entrada.

Alguns detalhes de implementação

- ▶ Nos algoritmos que veremos durante o semestre, usa-se a representação de um grafo (direcionado ou não) por **listas de adjacências**.
- ▶ Em uma implementação de verdade de um algoritmo, o mais provável é que esta representação **não** seja dada a priori.
- ▶ Em geral é necessário construir tal representação a partir da entrada dada. Como fazer isto depende do formato da entrada.

Alguns detalhes de implementação

Suponha que a entrada (arquivo texto ou teclado) tenha a seguinte forma:

5 7 0 /* respectivamente, $|V|$, $|E|$, 0/1 (direcionado ou não) */

0 1

0 2

1 2

1 3

2 3

2 4

3 4

Alguns detalhes de implementação

```
leia  $n$ ,  $m$  e  $b$   
enquanto houver arestas faça  
  leia a próxima aresta  $(u, v)$   
  insira  $v$  na lista  $Adj[u]$   
  se  $b = FALSE$  então  
    insira  $u$  na lista  $Adj[v]$ 
```

Exercício. (extremamente fácil) Suponha que a entrada tenha esta mesma forma. Escreva um pseudocódigo para construir a **matriz de adjacências** a partir dela.

Alguns detalhes de implementação

```
leia  $n$ ,  $m$  e  $b$   
enquanto houver arestas faça  
  leia a próxima aresta  $(u, v)$   
  insira  $v$  na lista  $Adj[u]$   
  se  $b = FALSE$  então  
    insira  $u$  na lista  $Adj[v]$ 
```

Exercício. (extremamente fácil) Suponha que a entrada tenha esta mesma forma. Escreva um pseudocódigo para construir a **matriz de adjacências** a partir dela.

Calculando o quadrado

O *quadrado* de um grafo direcionado $G = (V, E)$ é o grafo $G^2 = (V, E^2)$ onde $(u, v) \in E^2$ se existe um caminho de comprimento no máximo 2 de u a v em G .

(1) Dado $G = (V, E)$ representado por uma matriz de adjacência, mostre como calcular G^2 (i.e. sua matriz de adjacência).

(2) Dado $G = (V, E)$ representado por listas de adjacência, mostre como calcular G^2 (i.e. suas listas de adjacência).

Calculando o quadrado

O *quadrado* de um grafo direcionado $G = (V, E)$ é o grafo $G^2 = (V, E^2)$ onde $(u, v) \in E^2$ se existe um caminho de comprimento no máximo 2 de u a v em G .

(1) Dado $G = (V, E)$ representado por uma *matriz de adjacência*, mostre como calcular G^2 (i.e. sua *matriz de adjacência*).

(2) Dado $G = (V, E)$ representado por *listas de adjacência*, mostre como calcular G^2 (i.e. suas *listas de adjacência*).

Calculando o quadrado

O *quadrado* de um grafo direcionado $G = (V, E)$ é o grafo $G^2 = (V, E^2)$ onde $(u, v) \in E^2$ se existe um caminho de comprimento no máximo 2 de u a v em G .

- (1) Dado $G = (V, E)$ representado por uma *matriz de adjacência*, mostre como calcular G^2 (i.e. sua *matriz de adjacência*).
- (2) Dado $G = (V, E)$ representado por *listas de adjacência*, mostre como calcular G^2 (i.e. suas *listas de adjacência*).

Calculando G^2 a partir da matriz de adjacência

Suponha que M é a matriz de adjacência de $G = (V, E)$.

QUADRADO(M)

1 $N \leftarrow M$

2 para cada $u \in V$ faça

3 para cada $v \in V$ faça

4 para cada $w \in V$ faça

5 se $N[u, v] = 0$ então

6 $N[u, v] \leftarrow M[u, w] \cdot M[w, v]$

7 devolva N

Complexidade: $\Theta(V^3)$

Calculando G^2 a partir da matriz de adjacência

Suponha que M é a matriz de adjacência de $G = (V, E)$.

QUADRADO(M)

1 $N \leftarrow M$

2 para cada $u \in V$ faça

3 para cada $v \in V$ faça

4 para cada $w \in V$ faça

5 se $N[u, v] = 0$ então

6 $N[u, v] \leftarrow M[u, w] \cdot M[w, v]$

7 devolva N

Complexidade: $\Theta(V^3)$

Calculando G^2 a partir da matriz de adjacência

Suponha que M é a matriz de adjacência de $G = (V, E)$.

QUADRADO(M)

1 $N \leftarrow M$

2 para cada $u \in V$ faça

3 para cada $v \in V$ faça

4 para cada $w \in V$ faça

5 se $N[u, v] = 0$ então

6 $N[u, v] \leftarrow M[u, w] \cdot M[w, v]$

7 devolva N

Complexidade: $\Theta(V^3)$

Calculando G^2 a partir da lista de adjacências

Suponha que Adj guarda as listas de adjacências de $G = (V, E)$.

QUADRADO(Adj)

- 1 Crie uma cópia Adj' de Adj
- 2 para cada $v \in V$ faça
- 3 para cada $u \in Adj[v]$ faça
- 4 concatene uma cópia de $Adj[u]$ a $Adj'[v]$
- 5 para cada $v \in V$ faça
- 6 ordene $Adj'[v]$ em tempo linear (Counting-Sort)
- 7 elimine as repetições de $Adj'[v]$ em tempo linear
- 8 devolva Adj'

Complexidade: $\Theta(VE)$

Calculando G^2 a partir da lista de adjacências

Suponha que Adj guarda as listas de adjacências de $G = (V, E)$.

QUADRADO(Adj)

- 1 Crie uma cópia Adj' de Adj
- 2 para cada $v \in V$ faça
- 3 para cada $u \in Adj[v]$ faça
- 4 concatene uma cópia de $Adj[u]$ a $Adj'[v]$
- 5 para cada $v \in V$ faça
- 6 ordene $Adj'[v]$ em tempo linear (Counting-Sort)
- 7 elimine as repetições de $Adj'[v]$ em tempo linear
- 8 devolva Adj'

Complexidade: $\Theta(VE)$

Calculando G^2 a partir da lista de adjacências

Suponha que Adj guarda as listas de adjacências de $G = (V, E)$.

QUADRADO(Adj)

- 1 Crie uma cópia Adj' de Adj
- 2 para cada $v \in V$ faça
- 3 para cada $u \in Adj[v]$ faça
- 4 concatene uma cópia de $Adj[u]$ a $Adj'[v]$
- 5 para cada $v \in V$ faça
- 6 ordene $Adj'[v]$ em tempo linear (Counting-Sort)
- 7 elimine as repetições de $Adj'[v]$ em tempo linear
- 8 devolva Adj'

Complexidade: $\Theta(VE)$