Projeto e Análise de Algoritmos

Conceitos fundamentais de grafos

Atílio G. Luiz

Primeiro Semestre de 2024

Conceitos de grafos

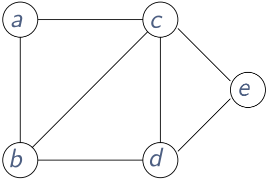
Definição de grafo

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:

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

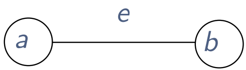


- Poderemos escrever $V(G)$ ou $E(G)$ para quando houver dúvida.

Adjacência e incidência

Considere uma aresta $e = (a, b)$

- note que para pares não ordenados, temos $(a, b) = (b, a)$
- desenhamos a aresta como uma linha ligando os vértices
- dizemos que os vértices a e b são os extremos de e
- e também que a e b são vértices adjacentes



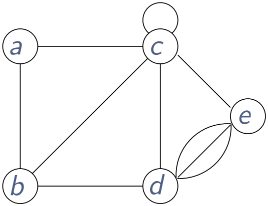
Para enfatizar a relação entre arestas e vértices

- dizemos que a aresta e incide nos vértices a e b
- e que os vértices a e b incidem na aresta e

Multigrafo

Um multigrafo é um generalização de grafos que pode conter

- **laço**: aresta com extremos idênticos. Ex.: (c, c)
- **arestas múltiplas**: duas ou mais arestas com o mesmo par de extremos

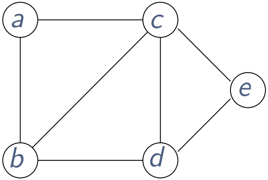


Um grafo é simples se ele não tiver laços ou arestas múltiplas

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 $|V| = 5$ e $|E| = 7$

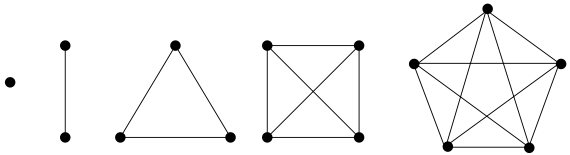


O tamanho do grafo G é dado por $|V| + |E|$

Grafos completos

Um grafo é completo se tiver uma aresta (u, v) para todo par de vértices u, v

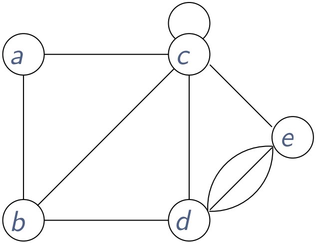
Exemplos de grafos completos:



- se o número de vértices é n, então ele tem $\binom{n}{2}$ arestas
- portanto, um grafo simples tem no máximo $\binom{n}{2}$ arestas
- Um grafo completo com três vértices é chamado de triângulo

Grau de um vértice

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



$d(a) = 2$

$d(b) = 3$

$d(c) = 6$

$d(d) = 5$

$d(e) = 4$

Handshaking Lemma (Euler 1736)

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

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

Alguns nomes

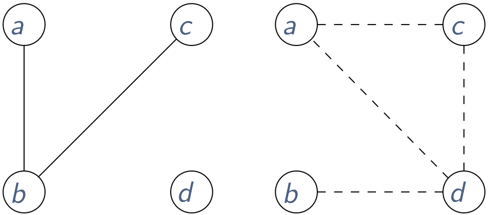
Considere um vértice v de um grafo $G = (V, E)$

- se $d_G(v) = |V| - 1$, dizemos que v é um vértice universal
- se $d_G(v) = 0$, dizemos que v é um vértice isolado

Grafo complementar

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 $(u, v) \in E(\overline{G})$ se e somente se $(u, v) \notin E(G)$

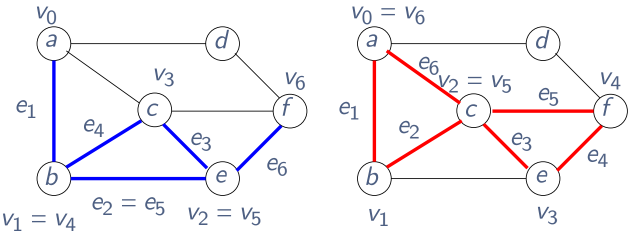


Note que $d_{\overline{G}}(v) = (|V| - 1) - d_G(v)$.

Caminhos e ciclos em grafos

Um caminho 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 $\langle v_0, v_1, \ldots, v_k \rangle$ tal que $(v_{i-1}, v_i) \in E$ para $1 \leq i \leq k$.

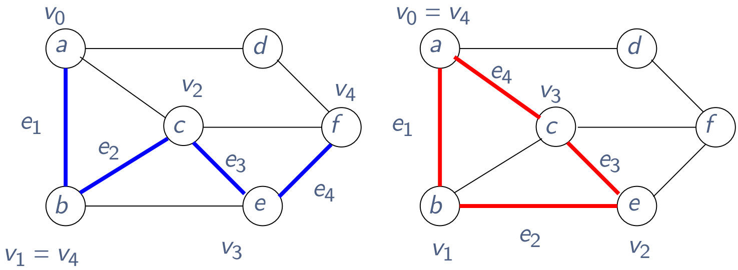
- dizemos que v_k é alcançável a partir de v_0 através de P



- Um ciclo é um caminho $\langle v_0, v_1, \ldots, v_k \rangle$ com $k > 0$, $v_0 = v_k$, e tal que todas as arestas do caminho são distintas.
- O comprimento de um caminho ou ciclo é o seu número de arestas.

Caminho simples e Ciclos simples

- Um caminho é simples se seus vértices são todos distintos.
- Um ciclo $\langle v_0, v_1, \ldots, v_k \rangle$ é simples se os vértices $v_1, \ldots, v_k$ são distintos.



Refletindo sobre as definições

Responda as seguintes questões

1. Seja G um grafo e u, v vértices de G. Mostre que se existe um caminho de u a v em G, então existe um **caminho simples** de u a v em G. Por que isto é um resultado interessante?

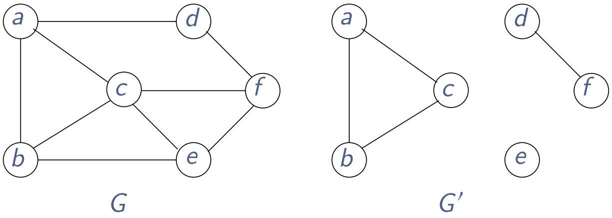
2. Seja G um grafo e u, v, w vértices de G. Mostre que se em G existem um caminho simples de u a v e um caminho simples de v a w então existe um caminho simples de u a w em G.

3. É verdade que todo ciclo contém um ciclo simples?

Conexidade

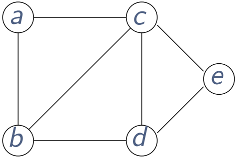
Dizemos que um grafo G é conexo se, para qualquer par de vértices u e v de G, existir um caminho de u a v em G.

- caso contrário, dizemos que G é desconexo
- podemos particionar o grafo em componentes
  - componentes são as classes de equivalência dos vértices sob a relação "é alcançável a partir de"
- u e v estão na mesma componente se há caminho de u a v

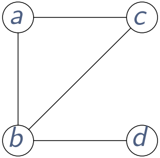


Subgrafo e subgrafo gerador

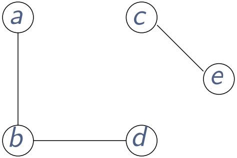
- Um subgrafo $H = (V', E')$ de um grafo $G = (V, E)$ é um grafo tal que $V' \subseteq V$ e $E' \subseteq E$.
- Um subgrafo gerador de G é um subgrafo com $V' = V$.



G



H



H

Grafos obtidos a partir de outros grafos

Considere um grafo $G = (V, E)$, uma aresta e e um vértice v

- $G - e$ é o grafo obtido de G removendo-se e:

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

- $G - v$ é o grafo obtido de G removendo-se v e todas as arestas que incidem em v:

$$G - v = (V \setminus \{v\}, E \setminus \{vw \colon vw \in E, w \in V\})$$

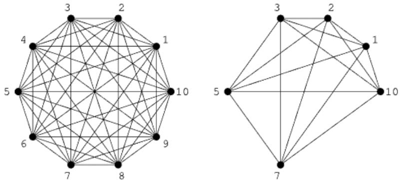
Subgrafo induzido

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

- O subgrafo de G induzido por S, denotado por $G[S]$, é o grafo formado por S e todas as arestas entre vértices de S:

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

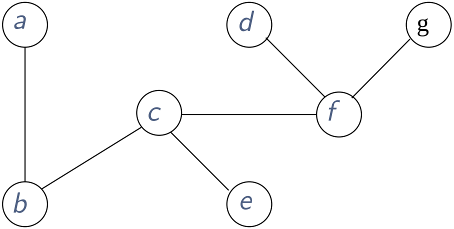
Exemplos:



Árvores

Árvores

- Um grafo sem ciclos simples é chamado de acíclico
- Uma árvore é um grafo conexo e acíclico.



- uma folha de uma árvore G é um vértice de grau 1
- toda árvore com dois ou mais vértices 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 simples de u a v em G.
3. G é conexo e a remoção de qualquer aresta desconecta o grafo, i.e, ele é conexo minimal.
4. G é conexo e possui exatamente $|V|-1$ arestas.

Árvore geradora

Fato 1

Todo grafo conexo contém uma árvore geradora.

Demonstração:

- segue facilmente do próximo resultado
- demonstre-o como exercício

Lema

Seja G um grafo conexo e seja C um ciclo simples de G. Se e é uma aresta de C então $G - e$ é conexo.

A recíproca também vale.

Lema

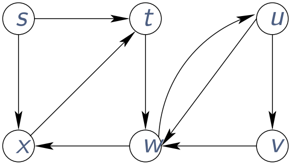
Seja G um grafo conexo e seja e uma aresta de G. Se $G - e$ é conexo então e pertence a algum ciclo simples de G.

Outros tipos de grafos

Grafo direcionado

Um grafo direcionado é definido de forma semelhante, com a diferença que as arestas consistem de pares ordenados de vértices.

- também chamamos essas variantes de digrafos
- muitas vezes chamamos suas arestas de arcos



Adjacência de grafos direcionados

Considere uma aresta $e = (u, v)$ de um grafo direcionado G

- dizemos que e sai de u e entra em v
- o vértice u é a cauda de e
- e o vértice v é cabeça de e

Temos dois tipos de grau para grafos direcionados

- grau de saída $d_{out}(v)$ é o número de arestas que saem de v
- grau de entrada $d_{in}(v)$ é o número de arestas que entram em v

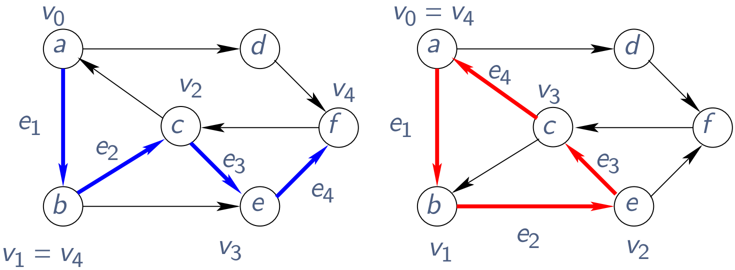
Teorema

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

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

Caminhos em grafos direcionados

Em um caminho direcionado de um grafo direcionado, todas as arestas seguem o mesmo sentido.



- definimos ciclos direcionados analogamente
- assim como os subgrafos de um grafo direcionado
- veremos a noção de conexidade para esses grafos depois

Refletindo sobre as definições

Vamos rever algumas questões, mas para grafos direcionados

1. Seja G um grafo direcionado e u, v vértices de G. Mostre que se existe um caminho de u a v em G, então existe um **caminho simples** de u a v em G.

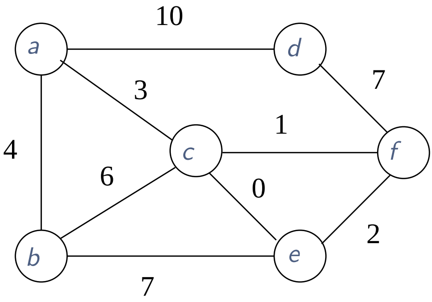
2. Seja G um grafo direcionado e u, v, w vértices de G. Mostre que se em G existem um caminho simples de u a v e um caminho simples de v a w então existe um caminho simples de u a w em G.

3. É verdade que todo ciclo em um grafo direcionado contém um **ciclo simples** direcionado?

Grafo ponderado

Um grafo é ponderado se a cada aresta e do grafo está associado um valor real $w(e)$, denominado peso da aresta.

- o grafo pode ser direcionado ou não
- também dizemos que $w(e)$ é o custo da aresta



Representação de grafos

Motivação prática

Grafos podem modelar diversas estruturas reais:

1. se computadores são representados por vértices, então as conexões entre eles correspondem a arestas
2. se as cidades forem representadas por vértices, então as estradas correspondem a arestas direcionadas
3. etc.

Queremos construir algoritmos genéricos

- vamos estudar algoritmos para grafos de modo abstrato
- mas eles são aplicados em problemas concretos

Aplicações

Problema do caminho mínimo

- dadas as cidades, as distâncias entre elas e duas cidades A e B, determinar um trajeto mais curto de A até B

Problema da árvore geradora mínima

- dados os computadores e o custo de conectar cada par de computadores, projetar uma rede interconectando todos os computadores de menor custo possível

Problema do emparelhamento máximo

- dadas vagas de empregos e uma lista de candidatos para cada vaga, determinar uma lista de associações candidato-emprego de maior tamanho possível

Aplicações

Problema do caixeiro viajante

- dadas cidades e as distâncias entre elas, encontrar um rota de comprimento mínimo que visita todas as cidades exatamente uma vez

Problema do carteiro chinês

- dadas as ruas de um bairro, encontrar uma rota fechada de comprimento mínimo que passa por todas as ruas pelo menos uma vez

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

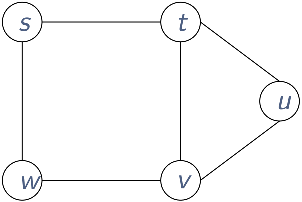
Matriz de adjacência

A matriz de adjacência de um grafo simples G é uma matriz quadrada A de ordem $|V|$ tal que

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

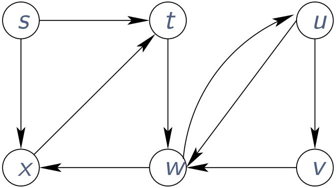
- o grafo pode ser direcionado ou não
- se G for não direcionado, então a matriz A é simétrica

Matriz de adjacência



$$
\begin{array}{c|ccccc|}
 & s & t & w & v & u \\
\hline
s & 0 & 1 & 1 & 0 & 0 \\
t & 1 & 0 & 0 & 1 & 1 \\
w & 1 & 0 & 0 & 1 & 0 \\
v & 0 & 1 & 1 & 0 & 1 \\
u & 0 & 1 & 0 & 1 & 0 \\
\hline
\end{array}
$$

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

Listas de adjacência

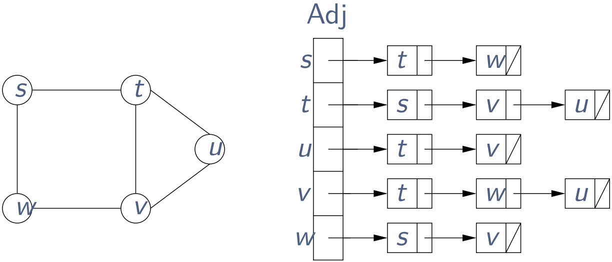
Para representar um grafo $G = (V, E)$ por listas de adjacências:

- criamos uma lista ligada $\text{Adj}[v]$ para cada vértice v
- adicionamos a $\text{Adj}[v]$ todos os vértices adjacentes a v

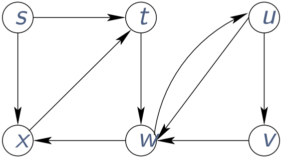
Como representamos uma aresta (u, v)?

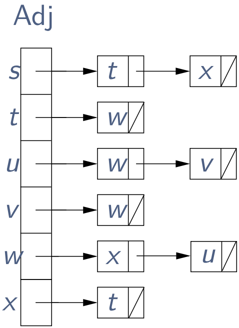
- se a aresta for direcionada, então v está em $\text{Adj}[u]$
- se a aresta for não direcionada,
    1. então v está em $\text{Adj}[u]$
    2. também u está em $\text{Adj}[v]$

Listas de adjacência



Listas de adjacências





Notação para complexidade

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

- vamos simplificar a notação assintótica
- escrevemos V e E ao invés de $|V|$ e $|E|$
- por exemplo, $O(E^2 \lg V)$ ao invés de $O(|E|^2 \lg |V|)$

Matriz versus listas

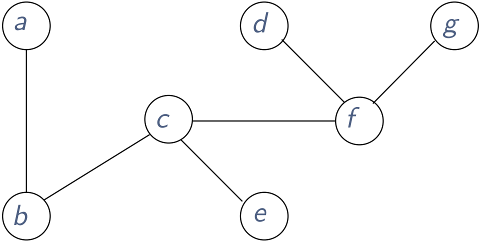
A melhor representação depende do algoritmo

1. matriz de adjacência
   - é fácil verificar se (u, v) é uma aresta de G
   - o espaço utilizado é $\Theta(V^2)$
   - adequada para grafos densos (com $|E| = \Theta(V^2)$)
2. listas de adjacência
   - é fácil listar os vértices adjacentes de um dado vértice v
   - o espaço utilizado é $\Theta(V + E)$
   - adequada a grafos esparsos (com $|E| = \Theta(V)$)

Extensões

- 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.

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

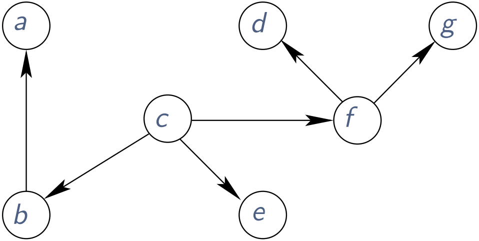


raiz c

Representação de árvores

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

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



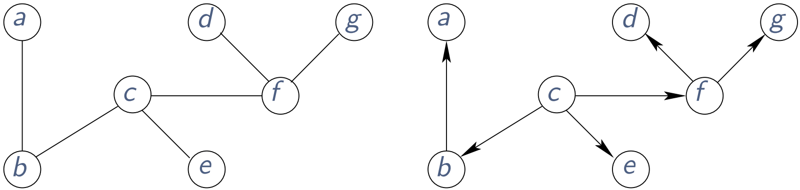
raiz c

Representação de árvores

Representar uma árvore enraizada com um vetor de predecessores π.

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

- usamos o símbolo NIL para indicar a ausência



raiz c