Estrutura de Dados II

Grafos

Profa. Juliana de Santi
Prof. Rodrigo Minetto
Universidade Tecnológica Federal do Paraná
Material compilado de: Cormen e material de Cid et al. IC-UNICAMP.

Sumário

Motivação: 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.

Motivação: 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:

- **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.

- **Árvore geradora de peso mínimo:** dado um conjunto de computadores, onde cada par de computadores pode ser ligado usando uma quantidade de fibra óptica, encontrar uma rede interconectando-os que use a menor quantidade de fibra óptica possível.

Introdução

- **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 pode ser ocupada por uma pessoa, encontrar um modo de empregar o maior número possível de pessoas.
- **Problema do caixeiro viajante:** dado um conjunto de cidades, encontrar um passeio que sai de uma cidade, passa por todas as cidades e volta para a cidade inicial tal que a distância total a ser percorrida seja menor possível.

Sumário

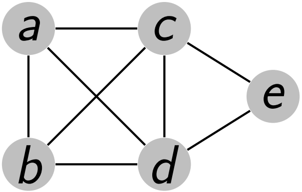
Definição de grafo (*do inglês graphs*): um **grafo** é um par $G = (V, E)$ onde:

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

Exemplo:

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

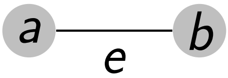
$$E = \{(a, b), (a, c), (a, d), (b, c)\\ (b, d), (c, d), (c, e), (d, e)\}$$



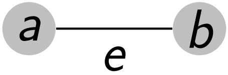
Sumário

Conceitos

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.

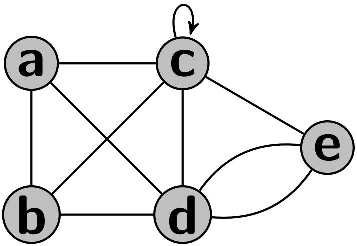


Dizemos também que a aresta e é incidente aos vértices a e b, e que os vértices a e b são incidentes à aresta e.

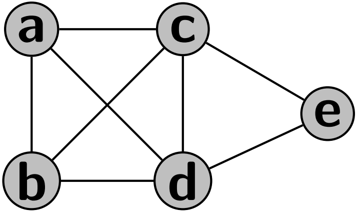


Conceitos

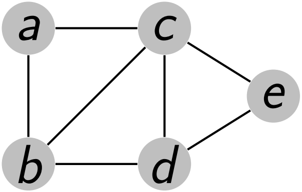
Um **laço** é uma aresta com extremos idênticos e **arestas múltiplas** são duas ou mais arestas com o mesmo par de vértices como extremos.



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



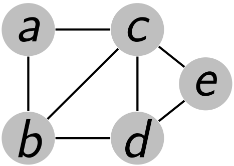
Denotamos por $|V|$ e $|E|$ a cardinalidade dos conjuntos de vértices e arestas de um grafo G, respectivamente. No exemplo abaixo temos $|V| = 5$ e $|E| = 7$.



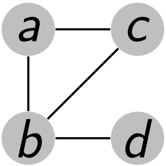
Tamanho do grafo $G = |V| + |E|$.

Conceitos

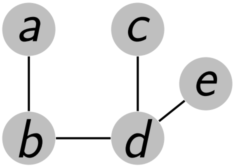
Um **subgrafo** $H = (V', E')$ de um grafo $G = (V, E)$ é um grafo tal que $V' \subset \mathbf{V}$ e $E' \subset \mathbf{E}$. Um **subgrafo gerador** de **G** é um subgrafo H com $V' = \mathbf{V}$.



Grafo **G**

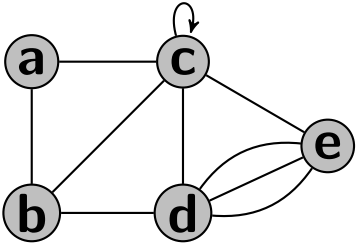


Subgrafo não gerador



Subgrafo gerador

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



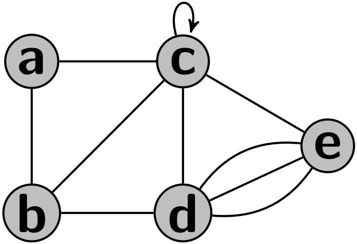
d(a) = 2
d(b) = 3
d(c) = 6
d(d) = 5
d(e) = 4

Conceitos

Teorema (*Handshaking lemma*)

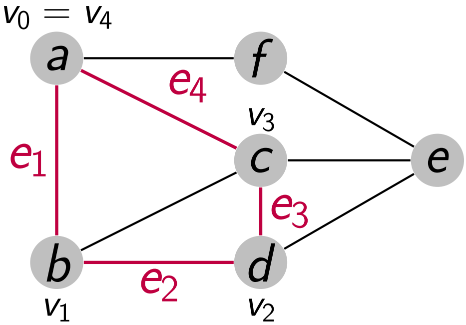
Para todo grafo $G = (V, E)$ temos:

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

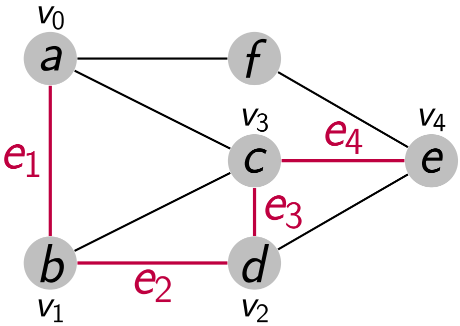


d(a) = 2
d(b) = 3
d(c) = 6
d(d) = 5
d(e) = 4

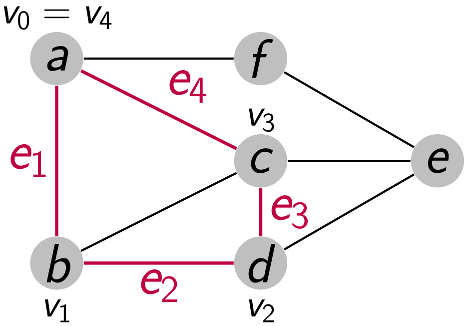
Um **caminho** P de v_0 a v_n no grafo G é uma sequência finita e não vazia $(v_0, e_1, v_1, \ldots, e_n, v_n)$ cujos elementos são alternadamente vértices e arestas e tal que, para todo $1 \leq i \leq n$, v_{i-1} e v_i são os extremos de e_i.



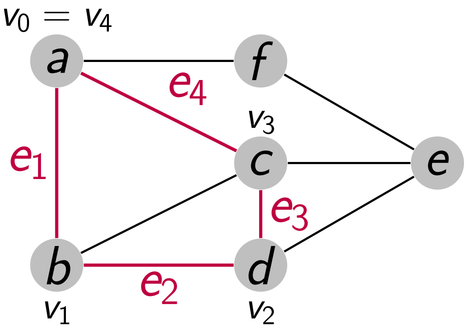
Um **caminho simples** é um caminho em que não há repetição de vértices e nem de arestas na sequência. Exemplo:



Um **ciclo** ou **caminho fechado** é um caminho em que $v_0 = v_n$. Exemplo:

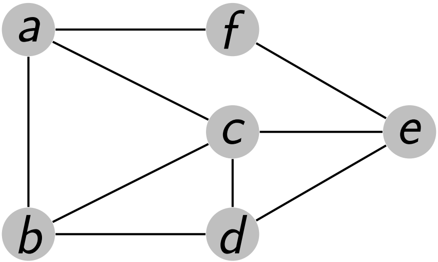


O **comprimento** do caminho P é dado pelo seu número de arestas, ou seja, n.



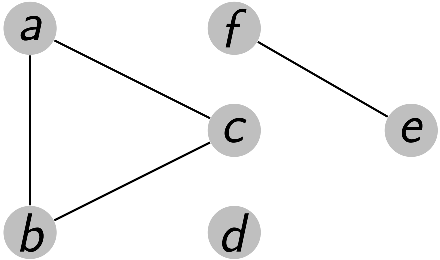
Dizemos que um grafo é **conexo** se, para qualquer par de vértices u e v de $\mathbf{G}$, existe um caminho de u a v em $\mathbf{G}$.

Exemplo:



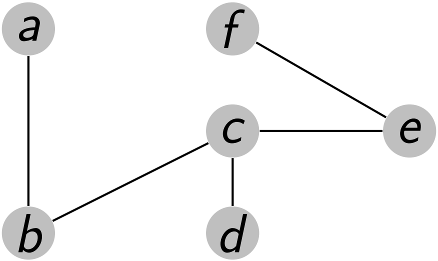
Grafo conexo

Quando o grafo **G não é conexo**, podemos particionar em **componentes conexos**. Dois vértices u e v de **G** estão no mesmo componente conexo de **G** se há caminho de u a v em **G**. Exemplo:



Grafo não-conexo com 3 componentes conexas

Um grafo G é uma **árvore** se é conexo e não possui ciclos (**acíclico**).



Conceitos

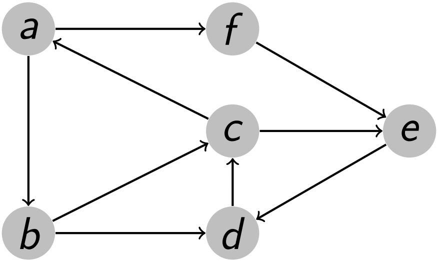
Se G é uma árvore as seguintes afirmações são equivalentes:

- G é conexo e possui exatamente $|\mathbf{V}| - 1$ arestas.
- G é conexo e a remoção de qualquer aresta desconecta o grafo (**minimal** conexo).
- Para todo par de vértices u, v de G, existe um único caminho de u a v em G.

Alguns exemplos de grafos

- **Floresta:** grafo acíclico (não precisa ser conexo). Cada componente é uma árvore!
- **Grafo completo:** para todo par de vértices u, v a aresta (u, v) pertence ao grafo.
- **Grafo bipartido:** possui uma bipartição (A, B) do conjunto de vértices tal que toda aresta tem um extremo em A e outro em B.
- **Grafo planar:** pode ser desenhado no plano de modo que arestas se interceptam apenas nos extremos.

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



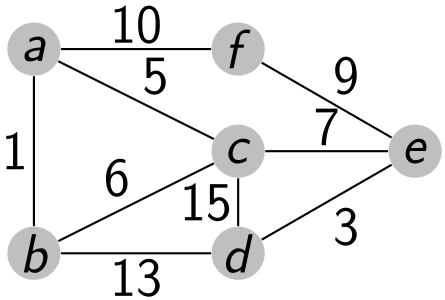
Conceitos

Se $e = (u, v)$ é uma aresta de um grafo orientado G, então dizemos que e sai de u e entra em v. O grau de saída $d^{+}(v)$ de um vértice v é o número de arestas que saem de v. O grau de entrada $d^{-}(v)$ de v é o número de arestas que entram em v.

Teorema: para todo grafo orientado $G = (V, E)$ temos:

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

Um grafo (orientado 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. Exemplo:



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.

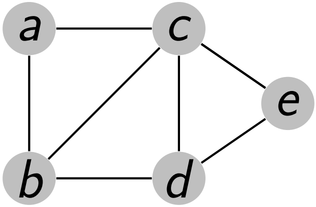
Matriz de adjacência

Seja $G = (V, E)$ um grafo simples (orientado 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-orientado, então a matriz A correspondente é simétrica.

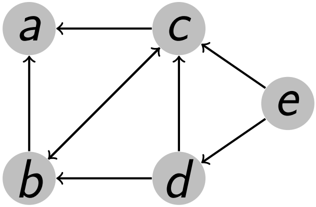
Exemplo de um grafo e a matriz de adjacência correspondente.



	a	b	c	d	e
a	0	1	1	0	0
b	1	0	1	1	0
c	1	1	0	1	1
d	0	1	1	0	1
e	0	0	1	1	0

Matriz de adjacência

Exemplo de um grafo orientado e a matriz de adjacência correspondente.

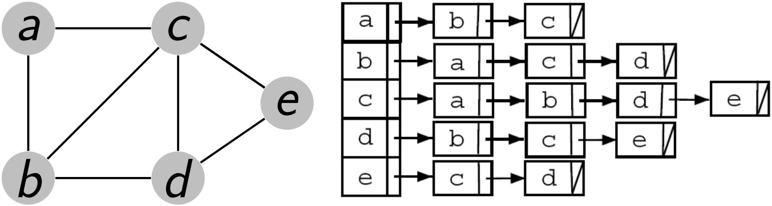


	a	b	c	d	e
a	0	0	0	0	0
b	1	0	1	0	0
c	1	1	0	0	0
d	0	1	1	0	0
e	0	0	1	1	0

Lista de adjacências

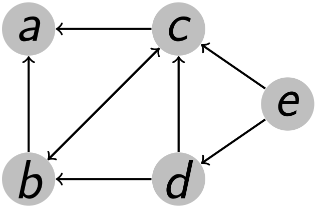
Seja $G = (V, E)$ um grafo simples (orientado ou não). A representação de G por uma **lista de adjacências** consiste no seguinte: para cada vértice v, temos uma lista ligada $Adj[v]$ dos vértices adjacentes a v, ou seja, w aparece em $Adj[v]$ se (v, w) é uma aresta de G. Os vértices podem estar em qualquer ordem em uma lista.

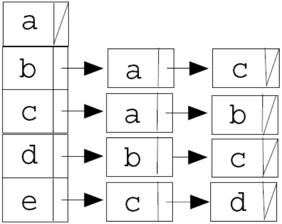
Exemplo de um grafo não-orientado e a lista de adjacências correspondente.



Lista de adjacências

Exemplo de um grafo orientado e a lista de adjacências correspondente.





Matriz de adjacência *vs* Lista de adjacência

- 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 $Adj[v]$).
- Matriz de adjacência: espaço $\Theta(|V|^2)$. Geralmente mais adequada a grafos densos ($|E| = \Theta(|V|^2)$).
- Lista de adjacência: espaço $\Theta(|V| + |E|)$. Geralmente mais adequada a grafos esparsos ($|E| = \Theta(|V|)$).

Extensões

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.