Capagrafos$_c$$eset, SumarioC, cap1_introducaoC, cap2_subgrafos, cap3_arvores, cap4_distancia, cap5_ham$

Universidade Estadual de Campinas - UNICAMP

Centro Superior de Educação Tecnológica – CESET

TEORIA DOS GRAFOS — UMA INTRODUÇÃO

Prof.: **Marco Antonio Garcia de Carvalho**
Rua Paschoal Marmo, 1888
Jd. Nova Itália
13484-370 — Limeira/SP

Fevereiro 2005
Limeira, SP - Brasil

Sumário

Capítulo 1

Introdução aos Grafos

1.1 Definições básicas

- **Grafo** – Um grafo G é definido por $G = (V, E)$, sendo que V representa o conjunto de nós e E, o conjunto de arestas (i, j), onde $i, j \in V$. Dois nós i, j são vizinhos, denotado por $i \sim j$, se eles estão conectados por uma aresta. A Figura 1.1 mostra dois exemplos de grafos: o grafo G_1 consiste dos conjuntos $V = \{a, b, c, d, e\}$ e $E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$; G_2 possui o nó 1 que não é conectado com nenhum outro nó do grafo.

Figura 1.1: Exemplos de grafos.

  Quando o grafo possue mais nós não-adjacentes do que nós adjacentes, ele é chamado de *grafo esparso*. Arestas associadas a um mesmo nó constitui um laço ou *loop*.

- **Grafo direcional** – Um grafo é dito direcional, ou dígrafro, quando é necessário ser estabelecido um sentido (orientação) para as arestas. O sentido da aresta é indicado através de uma seta, como ilustrado na Figura 1.2. Nesta situação, a aresta passa a ser denominada de arco.

- **Complemento de um grafo** – O complemento de um grafo G, representado por $\overline{G}$, é o grafo com o mesmo conjunto de vértices de G e tal que $i \sim j$ em $\overline{G}$ se eles não forem vizinhos em G. A Figura 1.3 apresenta um exemplo de grafo complementar.

- **Grafo completo** – Se todos os vértices de G são mutuamente adjacentes, o grafo é dito completo, como mostra o exemplo da Figura 1.4.

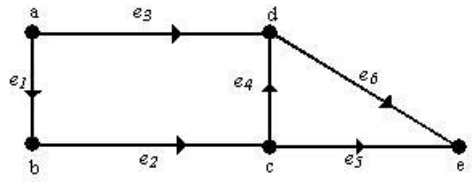


Figura 1.2: Exemplo de um dígrafo.

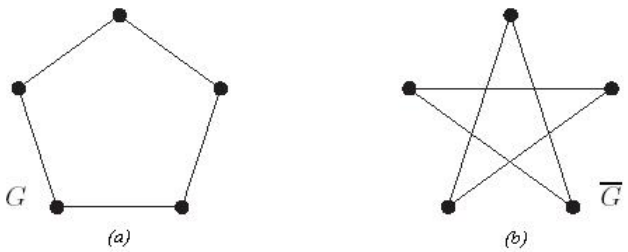


Figura 1.3: Um grafo G e seu complemento $\overline{G}$.

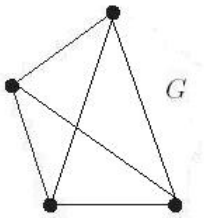


Figura 1.4: G é um grafo completo.

- **Grafo Ponderado**– Em um grafo ponderado, um peso ou conjunto de pesos é associado a cada aresta, representado da forma $w(i, j)$, ou seja, $w\,(1, 2)$ é o peso associado a aresta que une os nós 1 e 2. Já em um grafo com atributos A, definido por $G = (V, E, A)$, os valores são associados aos nós de G.

- **Árvore** – Um grafo conexo sem ciclos é chamado de árvore. Árvores serão estudadas no Capítulo 3.

1.2 Breve histórico

Os primeiros trabalhos em teoria dos grafos surgiram no século XVIII. Vários autores publicaram artigos neste período, com destaque para o problema descrito por Euler, conhecido como *As Pontes de Königsberg*.

- **As Pontes de Königsberg** – Este problema foi proposto em um artigo publicado em 1736. Numa cidade chamada Königsberg, sete pontes estabelecem ligações entre as margens do Rio Pregel e duas de suas ilhas. O problema consiste em, a partir de um determinado ponto, passar por todas as pontes somente uma vez e retornar ao ponto inicial. Esse problema ficou conhecido como Problema do caminho Euleriano e será estudado mais adiante. A Figura 1.5 ilustra o problemas das Pontes de Konigsberg.

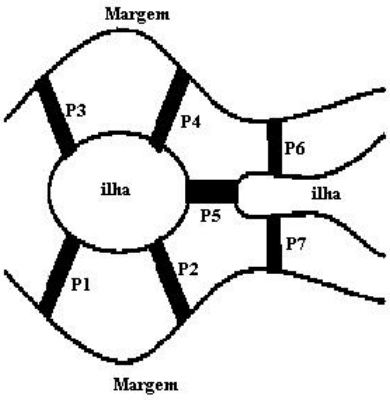


Figura 1.5: As Pontes de Königsberg.

Outro grande pesquisador foi o alemão Kirchhoff que utilizou grafos para modelagem de circuitos elétricos, já no século XIX.

1.3 Representações de Grafos

1.3.1 Matriz de adjacência

Uma das formas mais utilizadas para representar grafos é via a matriz de adjacência. Seja $A = [a_{ij}]$ uma matriz $n \times n$, onde n é o numero de nós de um grafo $G = (V, E)$ qualquer. A matriz de adjacência A é construída da seguinte forma:

$$A(i,j) = \begin{cases} 1 & \text{se } i \sim j \\ 0 & \text{caso contrário} \end{cases}$$

A Figura 1.6 ilustra o conceito de matriz de adjacência para um grafo simples.

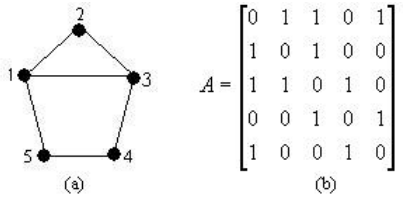


Figura 1.6: (a) Um grafo G e (b) sua matriz de adjacência.

Quando o grafo é ponderado, a representação só fica completa quando também se indica a sua matriz de pesos, construída de maneira semelhante à matriz de adjacência (troca-se o valor do peso pelos 1´s). Para dígrafos, é preciso observar o sentido do caminho entre os nós e adotar um padrão para o sinal dos pesos.

1.3.2 Matriz de incidência

A matriz de incidência $B = [b_{ij}]$ de um grafo $G = (V, E)$, com $V = (v_1, v_2, \ldots, v_n)$ e $E = (e_1, e_2, \ldots, e_m)$, é definida da seguinte forma:

$$B(i,j) = \begin{cases} 1 & \text{se } v_i \in e_j \\ 0 & \text{caso contrário} \end{cases}$$

A matriz de incidência do grafo na Figura 1.6(a) é dada pela Figura 1.7.

Se G é um dígrafo, então $b_{ij} = +1$ se v_i está no início da seta e $b_{ij} = -1$, caso v_i esteja na cabeça da seta. Para grafos ponderados, vale também a mesma observação no que diz respeito à escolha de sinais para representar os arcos e seus pesos.

1.3.3 Lista indexada

Trata-se de uma representação que favorece a recuperação mais rápida de informação nos grafos, basicamente por não possuir informações de não adjacência (os zeros na matriz de adjacência). Usa-se duas tabelas α e β. A tabela β lista os sucessores do nó i, iniciando da

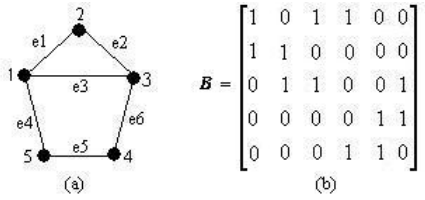


Figura 1.7: Matriz de incidência do grafo na Figura 1.6(a). Observe que as arestas foram rotuladas. As linhas da matriz correspondem aos nós e as colunas correspondem às arestas.

posição referenciada em $\alpha(i)$. Toda informação relacionada ao vértice i está contida entre as posições $\alpha(i)$ e $\alpha(i+1)-1$ da tabela β. Veja um exemplo na Figura 1.8 a seguir.

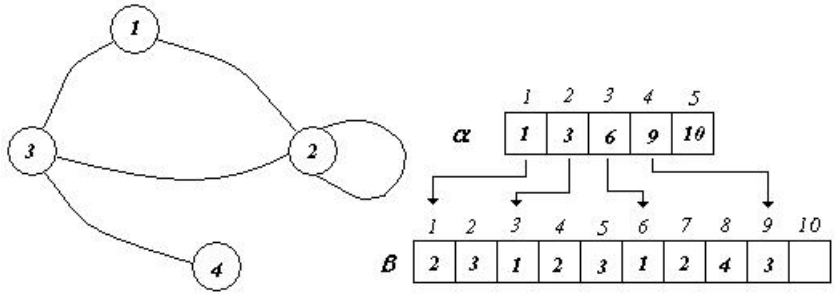


Figura 1.8: Lista indexada de um grafo G.

Se o grafo for ponderado, então a informação dos pesos das arestas deve ser armazenada em uma outra tabela ρ em uma correspondência um-a-um com β.

1.4 Grau de um grafo

- **Grau** – Grau de um nó i (ou *valência*), denotado por $deg(i)$ é o número $|E(i)|$ de arestas em i. Um nó de grau 0 é um nó isolado. A equação a seguir dá o grau médio de um grafo.

$$deg(G) = \frac{1}{|V|} \sum_{i \in V} deg(i) \tag{1.1}$$

- **Grafo regular** – É o grafo no qual seus nós possuem o mesmo grau.

1.5 Exercícios

1. **(Questão)** Na representação via lista de arcos são necessárias duas tabelas indicando

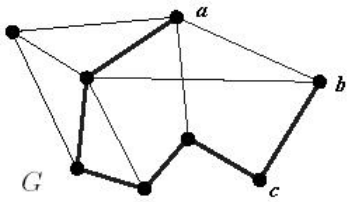


Figura 1.9: Exemplo de grau dos nós de um grafo: *deg(a)=4*, *deg(b)=3* e *deg(c)=2*. Qual o grau médio de G?

os pontos iniciais (Tabela α) e finais (Tabela β) de cada arco. Dada as tabelas abaixo, desenhe o grafo correspondente.

1	3	1	2	2	3	2	2

Tabela 1.1: Tabela α

3	1	3	1	1	2	3	2

Tabela 1.2: Tabela β

2. **(Questão)** A Figura 1.10 representa as moléculas químicas do metano ($\mathbf{CH_4}$) e propano ($\mathbf{C_3H_8}$).
(a) Interpretando esses diagramas como grafos, o que podemos dizer sobre os nós que representam os átomos de carbono (**C**) e os átomos de hidrogênio (**H**)?.
(b) Existe duas moléculas diferentes com fórmula $\mathbf{C_4H_{10}}$. Desenhe os grafos correspondentes a essas moléculas.

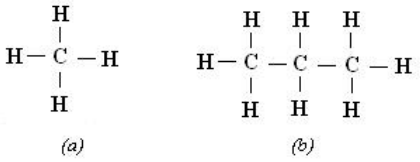


Figura 1.10: Moléculas do metano (a) e do propano (b).

3. **(Questão)** Dada a matriz de adjacência da Figura 8.3 a seguir, desenhe o grafo correspondente.

4. **(Questão)** Dado o dígrafo ponderado da Figura 1.12, determine: (a) matriz de adjacência; (b) lista indexada.

5. **(Questão)** Para os grafos da Figura 1.13 escreva o número de nós, arcos e o grau de cada vértice. Para o desenho da Figura 1.13(b), desenhe antes sua representação sob forma de grafo.

$$
\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array}
\begin{array}{c}
\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\
\left[\begin{array}{cccccccc}
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{array}\right]
\end{array}
$$

Figura 1.11: Matriz de adjacência A.

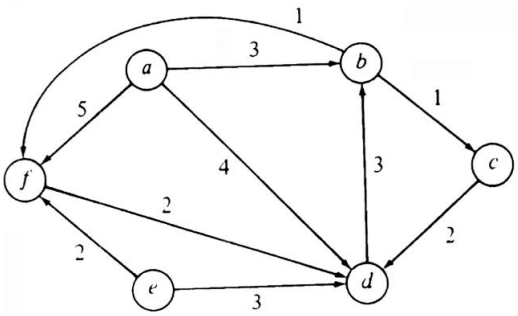


Figura 1.12: Um dígrafo ponderado.

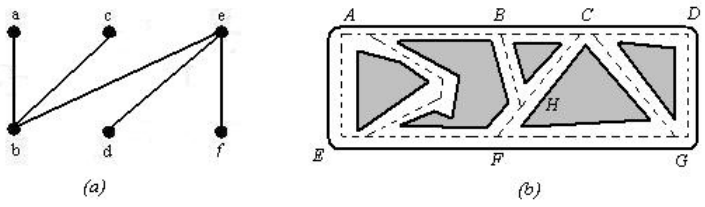


Figura 1.13: Grafos.

6. **(Questão)** Se G é um grafo sem *loops*, o que pode-se afirmar sobre a soma das entradas em: (a) qualquer linha ou coluna da matriz de adjacência de G? (b) qualquer linha da matriz de incidência de G? (c) qualquer coluna da matriz de incidência de G?

7. **(Questão)** Apresente a definição de:
   (a) Grafo simples;
   (b) Ponte;
   (c) Vértice de corte.

8. **(Questão)** (i) Desenhe um grafo com 6 vértices, cuja quantidade de vizinhos de cada nó é (3, 3, 5, 5, 5, 5); Existe um grafo *simples* com esta configuração?
   (ii) Se a quantidade de vizinhos passa a ser (2, 3, 3, 4, 5, 5), o que muda nas respostas do item anterior?

9. **(Questão)** Um grafo completo com n vértices é denotado por K^n. Qual o número de arestas em K^n?

Capítulo 2

Subgrafos, caminhos e conectividade

2.1 Caminho

- **Caminho** – Um caminho entre i_1 e i_n é a lista $(i_1, i_2, \ldots, i_{n-1}, i_n)$, onde $i_k \sim i_{k+1}$, $k = 1, 2, \ldots, n-1$. Dois nós i, j são conectados se existe ao menos um caminho entre i e j. Um caminho onde $i_1 = i_n$ é chamado de *ciclo*. Um exemplo de caminho entre os nós a e e do grafo na Figura 1.1(a) é a lista $(e3, e6)$

- **Grafo conexo** – Um grafo G é dito conexo se existe um caminho para qualquer par de nós (i, j) pertencente à G.

- **Comprimento de um caminho** – O número de arestas de um caminho determina o seu comprimento L e o caminho de comprimento k é denotado por P^k.

- **Caminhos independentes** – Dois ou mais caminhos são independentes se nenhum deles contém ao menos um vértice interno do outro.

- **Componente conexa** – Componente conexa é o máximo subgrafo conexo de G. Observe que uma componente conexa é sempre não-vazia; Um *grafo vazio*, portanto, não tem componentes conexas.

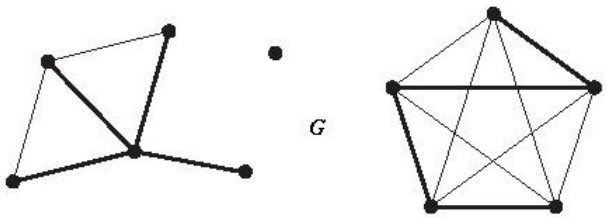


Figura 2.1: Um grafo G com 3 componentes conexas.

2.1.1 Enumeração de caminhos elementares

Será utilizado o Algoritmo de Warshall. Este algoritmo permite a determinação de todos os caminhos elementares entre um par de vértices. Nesta seção, estaremos interessados na determinação de caminhos elementares em dígrafos. A nomenclatura é explicada a seguir.

Seja G um grafo com um conjunto X de n nós. A função K determina todos os caminhos elementares e dada pelas relações abaixo:

- Para $x, y \in X \Rightarrow K[x, y, 0] = J[x, y]$
- Para $i = 1, 2, \ldots, n$ e $x, y \in X$:

$$K[x, y, i] = K[x, y, i-1] \bigcup (K[x, z_i, i-1] \odot K[z_i, y, i-1]) \tag{2.1}$$

se $x \neq z_i$ e $y \neq z_i$

$$K[z_i, y, i] = K[z_i, y, i-1] \tag{2.2}$$

se $x = z_i$

$$K[x, z_i, i] = K[x, z_i, i-1] \tag{2.3}$$

se $z_i = y$

Os vértices do grafo devem estar ordenados como $z_1, z_2, \ldots, z_n$.
O 3o. parâmetro da função K é usado como índice, fazendo que cada nó seja experimentado como possível ponto intermediário.
A função $J[x, y]$ corresponde ao peso do arco (x, y), se este existe, ou NULL, caso contrário.

- Os exemplos abaixo mostram o uso dos operadores $\bigcup$ e $\odot$.
Para $A = \{1-2-4, 1-3-4\}$ e $B = \{1-4, 1-2-3-4\}$,
$A \bigcup B = \{1-2-4, 1-3-4, 1-4, 1-2-3-4\}$.
Para $A = \{1-3, 1-2-3\}$ e $B = \{3-4-6, 3-5-6, 3-6\}$,
$A \odot B = \{1-3-4-6, 1-2-3-4-6, 1-3-5-6, 1-2-3-5-6,$
$1-3-6, 1-2-3-6\}$.

- Determinar todo os caminhos entre os nós 1 e 3 no grafo abaixo.

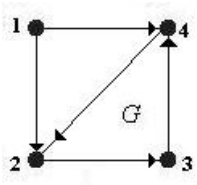


Figura 2.2: Grafo G para aplicação do algoritmo de Warshall.

2.2 Subgrafo

- **Subgrafo** – Um subgrafo de um grafo G é o grafo H tal que $V(H) \subseteq V(G)$ e $E(H) \subseteq E(G)$. Podemos representar na forma $H \subseteq G$ e dizer que G contém H. Os grafos da Figura 2.3(b),(c) são subgrafos do grafo mostrado na Figura 2.3(a).

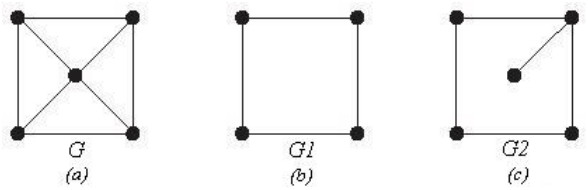


Figura 2.3: Um grafo G em (a) com dois subgrafos $G1$ (b) e $G2$ (c). $G1$ é um subgrafo induzido de G.

Seja H um subgrafo de G. Se $H \subseteq G$ e H contém todas as arestas $(i, j) \in E$, dado que i, j são nós de H, então H é um subgrafo induzido de G. A representação adotada é $H = G[S]$ e lê-se da forma *H é o subgrafo de G induzido por S*, onde S é o conjunto de nós de H.

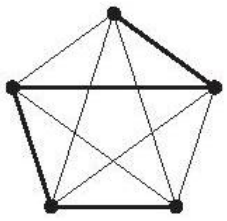
Pode-se obter subgrafos deletando arestas ou vértices de um grafo. Se v é uma aresta, o subgrafo $G - v$ é obtido removando-se o nó e as arestas incidentes a v.

2.3 Exercícios

1. **(Questão)** Descreva um algoritmo que determine todas as componentes conexas de um grafo.

2. **(Questão)** Para o grafo da Figura 2.3, desenhe todos os seus subgrafos conexos.

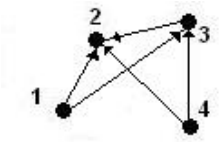
3. **(Questão)** Para o grafo da figura abaixo, desenhe, caso existam, um subgrafo completo com 3 nós e um subgrafo completo com 2 nós.



4. **(Questão)** Para o grafo da Figura anterior, desenhe seu maior subgrafo completo.

5. **(Questão)** Desenhe um grafo G com 5 nós cujo maior subgrafo completo tenha cardinalidade (número de nós) igual a 3. Quantos subgrafos com essa característica podem ser formados a partir de G?

6. **(Questão)** Enumeração de caminhos elementares — Algoritmo de Warshall. Dado o grafo orientado da Figura abaixo, calcule todos os caminhos entre os nós 1 e 2 através do algoritmo de Warshall.



Capítulo 3

Árvores

3.1 Árvores e florestas

- **Árvore** – Um grafo conexo sem ciclos é chamado de árvore (exemplo na Figura 3.1).

  Pode-se designar um nó para ser a raiz da árvore, o que demonstra uma relação lógica entre os nós. Essas árvores são ditas hierárquicas e a distância entre cada nó e a raiz é denominada de *nível*. Em uma árvore hierárquica os nós podem ser rotulados de acordo com a denominação de uma árvore genealógica: filhos, pais e ancestrais, no sentido literal das palavras. Uma árvore hierárquica onde cada nó dá origem a dois outros nós de nível inferior é chamada de *árvore binária*.

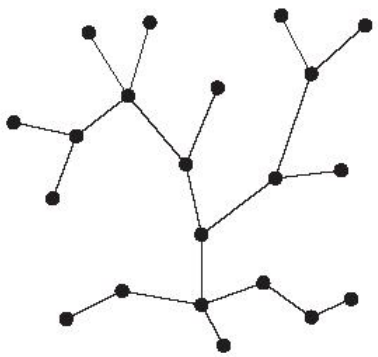


Figura 3.1: Exemplo de árvore.

  Os vértices de grau 1 em uma árvore são chamados de *folhas*.

- A seguir são listadas algumas propriedades de uma árvore T.

- Seja T um grafo com n vértices. Então, as seguintes proposições são válidas:

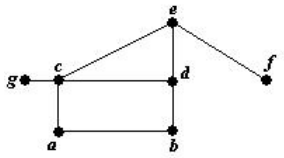
  1. T não contém ciclos e possui $n - 1$ arestas.

2. T é conexa e possui $n - 1$ arestas.
3. T é conexa e cada aresta é uma *ponte*.
4. Qualquer dois vértices de T é conectado por somente um caminho.
5. T não contém ciclos, mas a adição de qualquer nova aresta cria exatamente um ciclo.

- **Floresta** – Uma floresta é um grafo onde as componentes são árvores.

3.2 Excentricidade

- Denomina-se excentricidade de um vértice $v \in V$, ao valor da distância máxima entre v e w, para todo $w \in V$. O *centro* de G é o subconjunto dos vértices de excentricidade mínima. A Figura 3.2 apresenta um exemplo. Para este grafo, o centro é o subconjunto $\{c, d, e\}$.



VÉRTICE	EXCENTRICIDADE
a	3
b	3
c	2
d	2
e	2
f	3
g	3

Figura 3.2: Excentricidade de vértices.

3.3 Árvore geradora

É uma árvore T, subgrafo de G, que contém todos os nós de G. Uma árvore geradora cuja a soma dos pesos de seus arcos seja menor do que em qualquer outra situação é chamada de *árvore geradora mínima*.

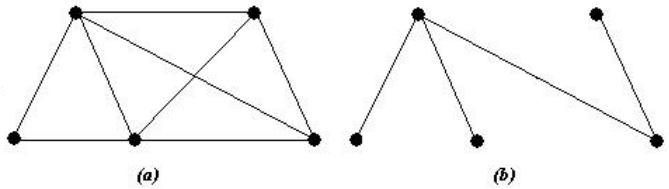


Figura 3.3: (b) é um exemplo de árvore geradora de (a).

Apresentamos a seguir dois métodos para a obtenção de AGMs a partir de um grafo.

3.3.1 Algoritmo de Kruskal

- Determina uma árvore geradora mínima de um grafo $G = (V, E)$.
- Um grafo pode ter mais de uma árvore geradora mínima.
- Descreveremos de maneira informal o algoritmo de Kruskal:

  1. O conjunto T de arestas está inicialmente vazio. COnforme o andamento do algoritmo, arestas vão sendo adicionadas.
  2. A cada instante, o grafo parcial formado pelos nós de G e as arestas em T consistem de várias componentes conexas (inicialmente, quando T está vazio, cada nó de G forma uma componente conexa distinta).
  3. Para construir componentes conexas cada vez maiores, examinam-se as arestas de G em ordem crescente de comprimento. Se uma aresta junta dois nós em uma diferente componente conexa, adiciona-se ela a T e, conseqüentemente, as duas componente conexas transformam-se me uma. Caso contrário, a aresta é rejeitada: ela une dois nós na mesma componente conexa e não pode ser adicionada a T sem formar um ciclo.
  4. O algoritmo pára somente quando uma componente conexa é determinada.

- Vejamos um exemplo de determinação de AGM usando o algoritmo Kruskal para o grafo da Figura 3.4 e Tabela 3.1.

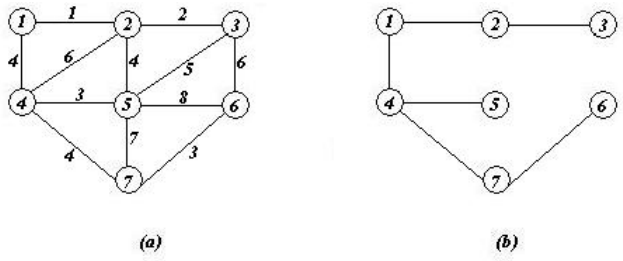


Figura 3.4: (a) Grafo G e sua (b) árvore geradora mínima T.

3.3.2 Algoritmo Prim

- Determina uma árvore geradora mínima T de um grafo $G = (V, E)$.
- No algoritmo de Prim (Algoritmo 1), a AGM cresce naturalmente a partir de um determinado nó denominado de *raiz*. Em cada estágio, adiciona-se um novo ramo à árvore e o algoritmo pára quando todos os nós tenham sido visitados.

Passo	Aresta	Componentes conexas
inicialização	—	$\{1\}$ $\{2\}$ $\{3\}$ $\{4\}$ $\{5\}$ $\{6\}$ $\{7\}$
1	$\{1,2\}$	$\{1,2\}$ $\{3\}$ $\{4\}$ $\{5\}$ $\{6\}$ $\{7\}$
2	$\{2,3\}$	$\{1,2,3\}$ $\{4\}$ $\{5\}$ $\{6\}$ $\{7\}$
3	$\{4,5\}$	$\{1,2,3\}$ $\{4,5\}$ $\{6\}$ $\{7\}$
4	$\{6,7\}$	$\{1,2,3\}$ $\{4,5\}$ $\{6,7\}$
5	$\{1,4\}$	$\{1,2,3,4,5\}$ $\{6,7\}$
6	$\{2,5\}$	*rejeitada*
7	$\{4,7\}$	$\{1,2,3,4,5,6,7\}$

Tabela 3.1: Acompanhamento do exemplo para o algoritmo Kruskal.

- Inicialmente, o conjunto de nós B contém somente um nó e o conjunto T está vazio. Em cada passo, o algoritmo olha para uma possível aresta de menor comprimento $\{u, v\}$ tal que $u \in V \setminus B$ e $v \in B$. Então, adiciona-se u a B e $\{u, v\}$ a T.

Algoritmo 1 Algoritmo de Prim — Determina uma AGM de um grafo $G = (V, E)$.

$T \leftarrow \oslash$
$B \leftarrow$ *um nó arbitrário de* V
Enquanto $B \neq V$ **Faça**
  Determine $\{u, v\}$ de menor comprimento tal que $u \in V \setminus B$ e $v \in B$
  $T = T \cup \{u, v\}$
  $B = B \cup \{u\}$
Fim Enquanto

- Teste o Algoritmo 1 para o grafo da Figura 3.4.

3.4 Árvores binárias

- Uma árvore hierárquica onde cada nó dá origem a dois outros nós de nível inferior é chamada de *árvore binária*.

- Árvores binárias tem diversas aplicações em computação (busca, classificação etc).

3.4.1 Número de nós em árvores binárias

- **Problema:** Provar que uma árvore binária completa com k níveis tem exatamente $2^k - 1$ nós.

- Indução matemática é usada como uma técnica de prova primária, mas eficaz.

- Seja T um teorema que se deseja provar e suponha que T possue um parâmetro n, positivo e inteiro. Ao invés de provar que T é válida para qualquer valor de n, pode-se provar as seguintes condições:
  1. Provar que é válida para $n = 1$.
  2. Provar que $\forall n > 1$, se a propriedade é válida para n, então ela é válida para $n+1$.
  Existem variações desta prova. Por exemplo, provar que $\forall n > 1$, se a propriedade é válida para $n - 1$, então ela é válida para n.

3.4.2 Construção ascendente de árvores hierárquicas binárias

Fundamentos

- A distância entre os nós (objetos) é calculada previamente.

- Procede-se através de etapas sucessivas, reunindo-se dois-a-dois os objetos mais próximos. Ao fim de cada etapa, recalcula-se a distância entre o novo objeto formado e o restante dos objetos.

- Uma das dificuldades está na escolha da fórmula para recalcular as distâncias entre objetos.

- Este processo gera uma *hierarquia indexada*. Diferentes formas de recalcular a distância produz diferentes tipos de hierarquias. As hierarquias mais comuns são as do tipo Ligação Simples e Ligação Completa.

Definições básicas

Definição 1 (Partição) *Uma partição P de um conjunto f é um conjunto de partes disjuntas R_i, $i = 1, 2, \ldots, n$, onde a união das partes forma o conjunto completo.*

$$P = \{R_1, R_2, \ldots, R_n\} \tag{3.1}$$

onde $\bigcup R_i = f$, $R_i \cap R_j = \emptyset$, $i \neq j$.

Definição 2 (Hierarquia) *Seja P_k um conjunto de partições $P_1, P_2, \ldots, P_n$, de uma imagem f. P_k é uma hierarquia, também chamada de seqüência de partições aninhadas, se para $i < j$, $P_i \supset P_j$. Significa dizer que uma partição em um dado nível é obtida misturando-se regiões de uma partições em um nível imediatamente inferior.*

Definição 3 (Dendrograma Indexado) *Um Dendrograma indexado é uma representação gráfica de uma seqüência aninhada de partições, onde os índices correspondem aos níveis em que as partições foram formadas.*

Ligação Simples

- Designa-se i e j os dois objetos que deseja-se agrupar e k um outro objeto do conjunto. A equação abaixo indica a nova distância a ser calculada no processo de Ligação Simples.

$$d\left(i \cup j, k\right) = min\left(d\left(i,k\right), d\left(j,k\right)\right) \tag{3.2}$$

- Um exemplo de aplicação desta técnica é dado nas Figuras 3.5(a) e (b).

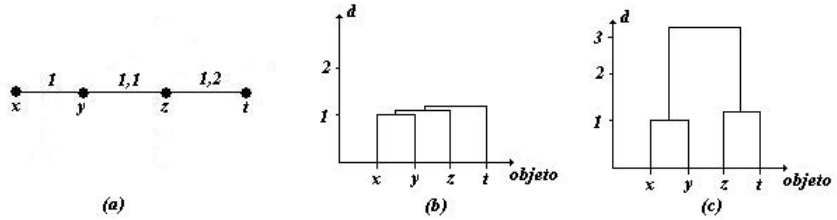


Figura 3.5: Comparação de hierarquias do tipo Ligação (b) Simples e (c) Completa para os dados mostrados em (a). A representação gráfica em (b) e (c) são os correspondentes dendrogramas de cada hierarquia.

- A partir dos dados da Figura 3.5(a), tem-se a distância entre cada objeto:
  $d\left(x,y\right) = 1$; $d\left(x,z\right) = 2,1$; $d\left(x,t\right) = 3,3$;
  $d\left(y,z\right) = 1,1$; $d\left(y,t\right) = 2,3$; $d\left(z,t\right) = 1,2$

- O primeiro grupo formado é o $x \cup y$, com distância 1. Utilizando a equação da distância para a ligação simples, tem-se $d\left(x \cup y, z\right) = 1,1$ e $d\left(x \cup y, t\right) = 2,3$. Portanto, agrupa-se $x \cup y$ com o objeto z. Por fim, agrupa-se o objeto t.

Ligação Completa

- Designa-se i e j os dois objetos que deseja-se agrupar e k um outro objeto do conjunto. A equação abaixo indica a nova distância a ser calculada no processo de Ligação Completa.

$$d\left(i \cup j, k\right) = max\left(d\left(i,k\right), d\left(j,k\right)\right) \tag{3.3}$$

- Um exemplo de aplicação desta técnica é dado nas Figuras 3.5(a) e (c).

- A partir dos dados da Figura 3.5(a), tem-se a distância entre cada objeto:
  $d\left(x,y\right) = 1$; $d\left(x,z\right) = 2,1$; $d\left(x,t\right) = 3,3$;
  $d\left(y,z\right) = 1,1$; $d\left(y,t\right) = 2,3$; $d\left(z,t\right) = 1,2$

- O primeiro grupo formado é o $x \cup y$, com distância 1. Utilizando a equação da distância para a ligação completa, tem-se $d(x \cup y, z) = 2,1$ e $d(x \cup y, t) = 3,3$. Portanto, agrupa-se z com o objeto t, cuja distância é somente 1,2. Por fim, agrupa-se o objeto $x \cup y$ com $z \cup t$.

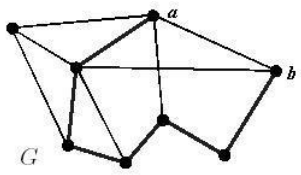
- Observa-se que o método da ligação simples tem a tendência de comprimir os níveis de fusão de objetos, enquanto o método da ligação completa, tende a separá-los ou distanciá-los. A característica do método de ligação simples de aproximar os objetos mais distantes é denominado de *efeito cadeia*.

3.5 Exercícios

1. Um dado usado pelo IBGE para medir o grau de espalhamento das cidades e/ou regiões brasileiras é determinado com o auxílio da árvore geradora mínima (AGM) de um grafo. Calcule a AGM do grafo da Figura 4.3 do Capítulo anterior usando o Algoritmo de Prim ou Kruskal. Apresente os passos do desenvolvimento e desenhe a AGM. Qual a soma dos pesos das arestas? (essa é a medida usada pelo IBGE).
   O que podemos concluir dessa medida (grau de espalhamento) da região sul e sudeste?

2. Uma aplicação do uso de árvores binárias consiste em um algoritmo de codificação denominado de Código de Huffmann. Monte a árvore binária que produz o código de Huffmann para o seguinte problema:
   Um espião americano enviou uma mensagem codificada para seu parceiro no Iraque que dizia: PERTO PORTA PÁTIO. Essa mensagem significa que o objetivo da missão está próximo de ser cumprido. Codifique a mensagem através do Algoritmo de Huffman (apresente o desenvolvimento). Calcule também o número de bits para os códigos de Huffman, ASCII (8 bits por caracter) e um código de tamanho fixo (determine o número mínimo de bits).
   Ignore a acentuação (Ã=A, por exemplo), vírgulas, pontos e espaço em branco.

3. Dado o grafo da Figura 3, construa uma árvore que apresente todos os possíveis caminhos entre os nós a e b indicados.



4. Qual a definição de árvore geradora máxima de um grafo? Que modificação precisa ser efetuada no algoritmo de Prim para determinar a árvore geradora máxima?

5. Os dados usados nesta questão foram tirados do livro *Algorithmes de Classification*, de M. Roux. Na Tabela 3.2 são dadas as estatíticas relativas às causas de morte (S - suicídio e H - homicído) para 5 diferentes países ocidentais. Segundo Mr. Todd, esse tipo de análise é característico da saúde mental de uma sociedade.

	S	H
Áustria	241	16
França	156	9
Portugal	85	19
Bélgica	156	10
Finlândia	251	26

Tabela 3.2: Tabela das causas de morte em 5 diferentes países ocidentais.

Construa o dendrograma que representa esses dados, segundo o método da Ligação Simples. O atributo que representa cada país é o número total de mortes NTM. A distância entre dois países x e y é calculada através da fórmula $|NTM_x - NTM_y|$. Analise o dendrograma, fazendo comentários sobre os países mais semelhantes segundo o número total de mortes.

Capítulo 4

Distâncias e caminho mínimo

4.1 Matriz distância em um grafo

- Em diversas situações, deseja-se calcular a distância, ou o comprimento do caminho entre dois nós. O conceito de distância deve respeitar algumas restrições, como por exemplo, ser associada a números não negativos, além de satisfazer as propriedades:
(a) $d(u, v) \geq 0$ e $d(u, v) = 0$ se e somente se $u = v$
(b) $d(u, v) = d(v, u)$
(c) $d(u, v) + d(v, w) \geq d(u, w)$
onde u, v, w são nós de um grafo. A Figura 4.1 apresenta um grafo e sua respectiva matriz distância. A cada arco é associado um peso de valor 1.

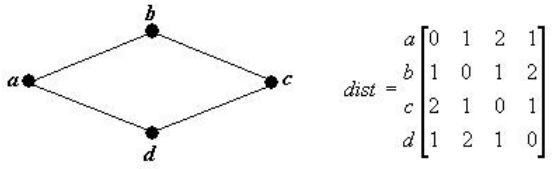


Figura 4.1: Um grafo e sua matriz distância.

4.1.1 Algoritmo Dijkstra

- Como se observa na Figura 4.1, a matriz distância é construída com base na menor distância entre dois vértices. O Algoritmo 2 determina o comprimento do menor caminho entre dois nós a e z, em um grafo conectado. Esse algoritmo foi desenvolvido por Edsger W. Dijkstra, na metade do século passado.

- Um exemplo da implementação desse algoritmo é dado na Figura 4.2. A Figura 4.2(a) apresenta um grafo G com pesos nos arcos. Deseja-se encontrar a menor distância entre os nós a e z. As Figuras 4.2(b)-4.2(f) mostram os passos realizados na implementação do Algoritmo 2 até se alcançar o resultado desejado.

Algoritmo 2 Algoritmo Dijkstra — Determina o menor caminho entre dois nós (custo)

Entrada: Grafo conectado com pesos nos arcos (matriz w), nós a e z.
Saída: L(z) - comprimento do menor caminho entre a e z.

$L(a) \leftarrow 0$
Para todo nó $x \neq a$ **Faça**
  $L(x) \leftarrow \infty$
Fim Para
$T \leftarrow$ conjunto de todos os nós cuja menor distância até a ainda não foi calculada.
Enquanto $z \in T$ **Faça**
  Escolha $v \in T$ com menor $L(v)$
  $T = T - \{v\}$
  Para $x \in T$ vizinho a v **Faça**
    $L(x) \leftarrow min\left\{L\left(x\right), L\left(v\right) + w\left(v, x\right)\right\}$
  Fim Para
Fim Enquanto

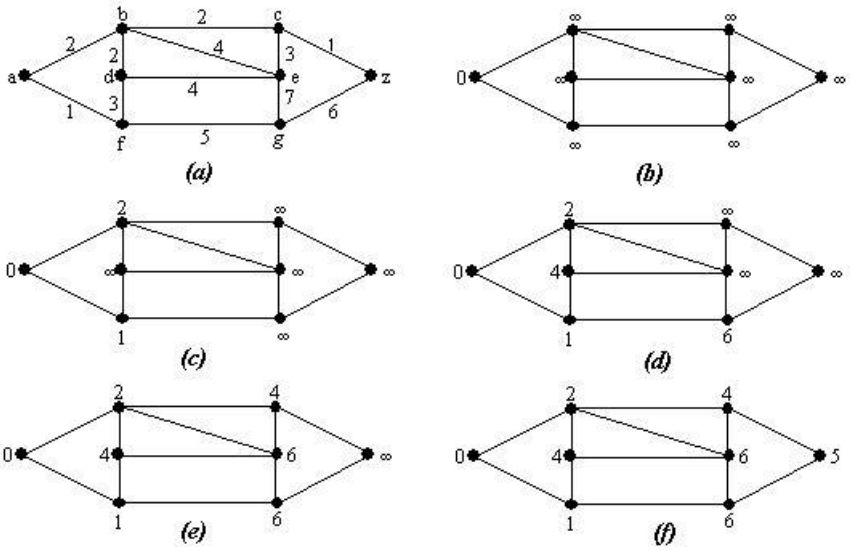


Figura 4.2: (a) Um grafo G e (b)-(f) os passos necessários para se obter a menor distância entre os nós a e z segundo o Algoritmo 2.

4.2 Caminho mínimo

- O caminho mais curto entre dois vértices é aquele cujo custo é o menor possível. O custo é dado pela soma dos pesos das arestas pertencentes ao caminho.
- Podemos assumir que caminhos mais curtos não possuem ciclos.
- A determinação do caminho é realizada através do uso de uma estrutura denominada de *Antecessor*. Para cada vértice $v \in V$ o *Antecessor[v]* é um outro nó $u \in V$ (ou zero para o nó origem) cujo menor caminho da origem até v passa por ele.
- De fato, a estrutura *Antecessor* contém uma árvore de caminhos mais curtos.
- Algumas modificações do Algoritmo de Dijkstra são dadas no Algoritmo 3.

Algoritmo 3 Algoritmo Dijkstra — caminho mínimo entre dois nós

$Antecessor(a) \leftarrow 0$
....
Enquanto $z \in T$ **Faça**
  Escolha $v \in T$ com menor $L(v)$
  $T = T - \{v\}$
  Para $x \in T$ vizinho a v **Faça**
    Se $L(x) > L(v) + w(v, x)$ **Então**
      $L(x) = L(v) + w(v, x)$
      $Antecessor(x) = v$
    Fim Se
  Fim Para
Fim Enquanto

4.3 Exercícios

1. Suponha que a distância entre dois nós é realizada por uma função de nome *Dijkstra(origem, destino, G)*, onde G é um grafo dado por $G = (V, E, W)$. Esta função retorna um valor real. Descreva uma solução (um algoritmo) para o cálculo da *matriz distância* do grafo G.
2. No Algoritmo 3 deste Capítulo, a estrutura *Antecessor* armazena uma árvore de caminhos mínimos que contém o caminho de menor custo entre dois nós a e z . Descreva, em forma de algoritmo, uma solução para determinar a lista de nós pertencentes ao caminho mínimo entre os nós origem e destino.
3. Os caminhos que levam à Região Norte do Brasil são sempre problemáticos em função da quantidade e qualidade das estradas. A Região também possui uma grande bacia

hidrográfica, o que impede a construção de estradas em alguns trechos. Um atleta do Unisal deseja fazer o percurso de bicicleta da Cidade de Goiás até a cidade de Boa Vista, em Roraima, ambas indicadas na Figura 4.3. Determine a menor distância que esse aventureiro deverá percorrer para atingir o seu objetivo através do Algoritmo de Dijkstra. Apresente os passos do desenvolvimento do Algoritmo através de uma tabela ou de figuras.

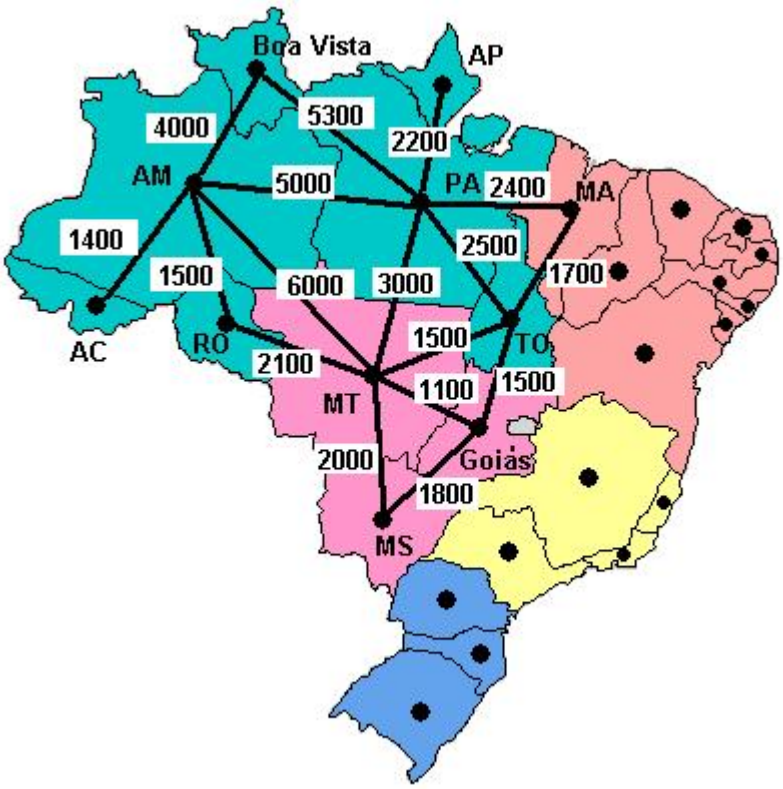


Figura 4.3: Mapa do Brasil.

4. Qual o caminho mínimo da questão anterior? Apresente um teste de mesa com o status da estrutura *Antecessor* a cada passo.

5. Em uma aplicação na área de biologia, tem-se a imagem mostrada na Figura 4.4(a) como sendo as células de um vírus ofensivo ao ser humano. Um parâmetro importante aos pesquisadores é a distância máxima entre células. Ela fornece uma idéia da velocidade de reprodução e transmissão do vírus. Após a modelagem desta imagem através de grafos, obteve-se a imagem da Figura 4.4(b). Os pesos das arestas foram calculados segundo dados estatísticos de cada uma das células, proximidade, tipo de célula etc. Na tentativa de descobrir a maior distância entre células existentes na imagem, os pesquisadores precisam calcular a distância entre os nós 10 e 6. Através

do algoritmo de Dijkstra, calcule essa distância. Apresente os passos de sua resolução através de desenhos ou de uma tabela.

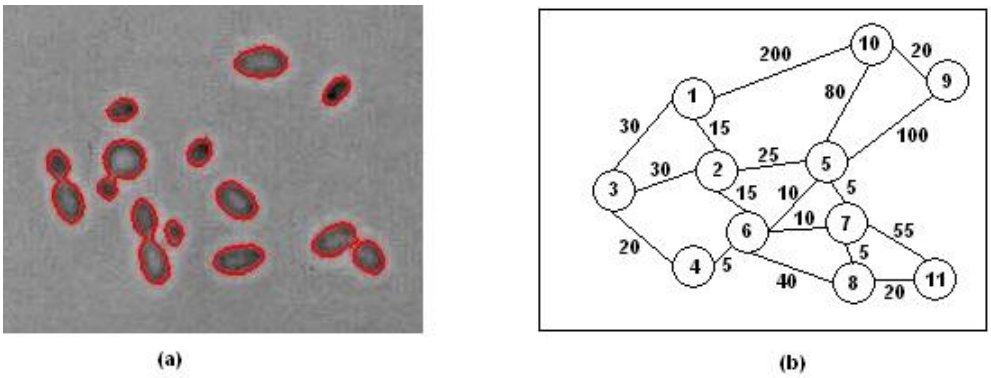
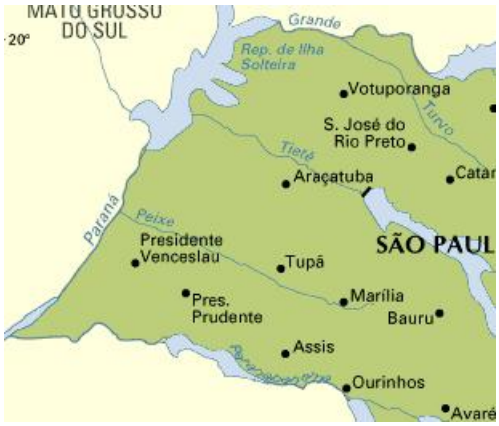


Figura 4.4: Aplicação do Algoritmo de Dijkstra na Biologia.

6. Seja o mapa com algumas das cidades do estado de São Paulo mostrado abaixo e uma tabela de distâncias.



(a) Calcule a distância e o caminho entre as cidades de Araçatuba e Ourinhos. Apresente o desenvolvimento da solução.
(b) Com a privatização das estradas, praças de pedágio foram construídas em todas as rodovias. Nas rodovias de comprimento inferior a 100km o pedágio custa R$4,00; caso contrário, o pedágio tem valor de R$5,00. Apresente o Algoritmo de Dijkstra modificado tal que além do cálculo da distância e o caminho mínimo, também seja determinado o gasto com pedágio. Apresente o acompanhamento do algoritmo para o item anterior.

	Ara	**P. Venc**	**P. Prud**	**Tupã**	**Marí.**	**Assis**	**Bauru**	**Ourinhos**
Ara	-	150	-	90	110	-	160	-
P. Venc	150	-	40	95	-	-	-	-
P. Prud	-	40	-	85	-	105	-	-
Tupã	90	95	85	-	60	80	-	-
Marí.	110	-	-	60	-	85	65	80
Assis	-	-	105	80	85	-	-	75
Bauru	160	-	-	-	65	-	-	105
Ourinhos	-	-	-	-	80	75	105	-

Capítulo 5

Grafos hamiltonianos e Eulerianos

5.1 Grafos Bipartites

- Um grafo $G = (V, E)$ é bipartite quando o seu conjunto de vértices V puder ser particionado em dois subconjuntos V_1 e V_2, tais que toda aresta de G une um vértice de V_1 a outro de V_2. Exemplos de grafos bipartites são dados na Figura 5.1.

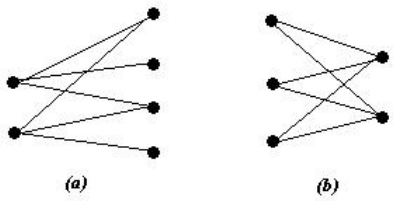


Figura 5.1: Exemplo de grafos bipartites.

5.2 Grafos Hamiltonianos

- É um grafo que possui caminho ou ciclo hamiltoniano. Um *caminho hamiltoniano* é aquele que contém cada nó do grafo exatamente uma vez. A Figura 5.2 apresenta um exemplo de grafo hamiltoniano, pois o mesmo contém o ciclo $1, 2, 3, 4, 5, 1$.

- **(Aplicação — Problema do caixeiro viajante:)** Neste problema, um caixeiro viajante deseja visitar várias cidades e retornar ao ponto de partida, de forma que a distância percorrida seja a menor possível. Esse problema pode ser modelado através de um grafo ponderado, como mostra o exemplo da Figura 5.3.
  Neste caso, deseja-se encontrar um ciclo Hamiltoniano de menor peso total. Um possível algoritmo é calcular todos os possíveis ciclos Hamiltonianos, o que é *pesado* quando existe mais do que 5 cidades (para 20 cidades, esse número é 19!/2). Existe

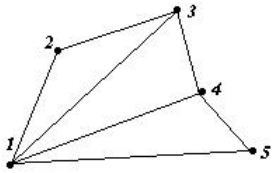


Figura 5.2: Exemplo de grafo hamiltoniano.

diversos algoritmos que usam heurísticas para resolver rapidamente esse problema, resultando *quase sempre* na menor distância.

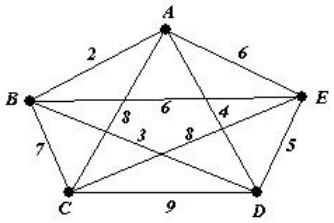


Figura 5.3: Figura exemplo — Problema do caixeiro viajante.

5.3 Grafos Eulerianos

- É um grafo que possui caminho ou ciclo euleriano. Um *caminho euleriano* é aquele que contém cada aresta do grafo exatamente uma vez. A Figura 5.4 apresenta um exemplo de grafo euleriano.

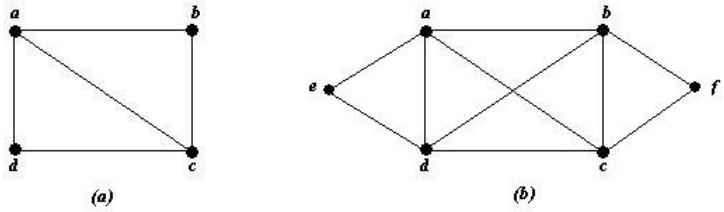


Figura 5.4: Exemplos de grafos euleriano.

- Problemas que usam grafos eulerianos são comuns em jogos matemáticos. Um problema típico é quando pretende-se desenhar algo (ou passar por um labirinto) sem retirar o lápis do papel e sem passar pelas linhas já desenhadas.

- **(Aplicação — Problema do carteiro chinês:)** Este problema foi discutido pelo matemático chinês Mei-Ku Kwan. Suponha que um carteiro deseja entregar cartas fazendo uma menor distância possível e retornar ao ponto de partida. Obviamente ele deve passar por cada estrada em sua rota no mínimo uma vez, mas deve evitar passar pela mesma estrada mais de uma vez.
  O problema pode ser modelado em termos de um grafo ponderado. Deve-se determinar, portanto, um caminho fechado de peso total mínimo que inclua cada aresta pelo menos uma vez. Se o grafo é Euleriano, qualquer percurso Euleriano é um caminho fechado, como requerido (e pode ser determinado pelo Algoritmo de Fleury). Caso o grafo não seja Euleriano, o problema é muito mais difícil de ser resolvido (veja Figura 5.5).

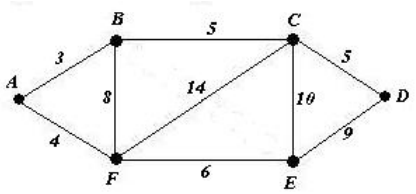


Figura 5.5: Figura exemplo — Problema do carteiro chinês.

5.4 Exercícios

1. O grafo da Figura 5.6 representa um trecho do bairro Parque Taquaral, na altura da Av. Pe. Vieira. Identifique, se houver, um ciclo hamiltoniano e outro euleriano.

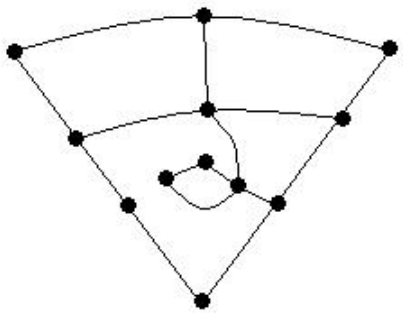


Figura 5.6: Descobrir ciclos hamiltonianos e eulerianos.

Capítulo 6

Planaridade

6.1 Grafos Planares

- Um grafo $G = (V, E)$ é planar quando puder ser desenhado em um plano, sem que ocorra cruzamento de arestas, ou seja, duas ou mais arestas não se intersectam geometricamente exceto nos vértices em que são incidentes. Exemplos de grafos planares são dados na Figura 6.1(b) e (c).

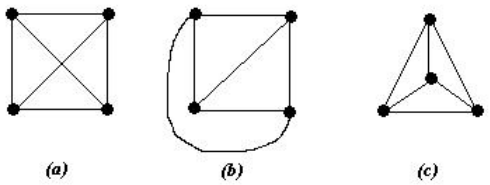


Figura 6.1: Grafos K_4. Somente os grafos (b) e (c) são planares.

- Seja G um grafo planar e R uma representação plana de G em um plano P. Então, as linhas de R dividem P em regiões, as quais são denominadas *faces* de R. Existe somente uma face não limitada, chamada de *face externa* ou *face infinita*.

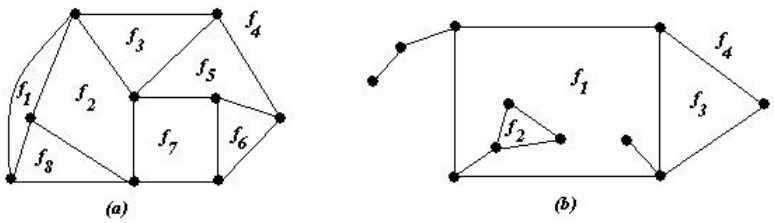


Figura 6.2: O grafo em (a) tem 8 faces e o grafo em (b), tem 4.

- Duas representações planas de um grafo possuem sempre o mesmo número de faces. Portanto, este número constitui um invariante de um grafo.

  Teorema 1 (Euler, 1750) *Seja G um grafo conexo planar e R uma representação plana de G. Seja n, m e f o número de vértices, arestas e faces de G. Então,*

  $$n - m + f = 2$$

  Esse teorema é conhecido com Fórmula de Euler.

- Observe que quanto maior é o número de arestas de um grafo em relação ao seu número de vértices, mais difícil intuitivamente se torna a obtenção de uma representação geométrica plana. De fato, há um limite máximo de arestas de um grafo planar, dado por $m \leq 3n - 6$.

- Uma outra questão interessante é: dado um grafo G planar, existe uma representação plana onde todas as linhas são retas? Foi provado por K. Wagner (1936) e I. Fáry (1948) que *todo grafo planar pode ser desenhado a partir de linhas retas*(mudando disposição dos pontos).

- Aplicação da planaridade de grafos: circuitos impressos!!! (ver Figura 6.3)

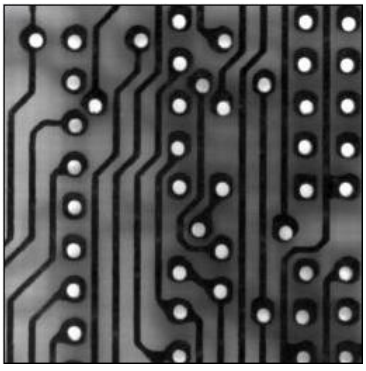


Figura 6.3: Aplicação - circuito impresso.

- **(Exemplo 1)** Mostre, desenhando, que os seguintes grafos são planares:
  (a) o grafo W_5;
  (b) o grafo do octaedro.

- **(Exemplo 2)** Mostre como o grafo abaixo pode ser desenhado em um plano sem cruzamentos:

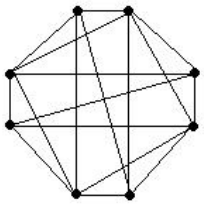


Figura 6.4: Grafo do exemplo 2.

- **(Exemplo 3)** Três vizinhos inimigos usam a mesma água, óleo e gás. A fim de evitar encontros e problemas, eles desejam construir caminhos que não se cruzem de suas casas a cada uma das 3 fontes. Isto pode ser feito? Justifique.

- **(Exemplo 4)** Prove que o limite máximo de arestas de um grafo planar é dado por $m \leq 3n - 6$.

- **(Exemplo 5)** Redesenhe o grafo da Figura 6.2(b) com:
(i) f_1 como face infinita;
(ii) f_2 como face infinita.

Capítulo 7

Isomorfismo

7.1 Definição

- Dois grafos G_1 e G_2 são ditos isomórficos se existir uma correspondência um-a-um entre os nós de G_1 para com os nós de G_2, tal que o número de arestas unindo quaisquer 2 nós de G_1 é igual ao número de arestas unindo os correspondentes nós de G_2. Um exemplo de grafos isomórficos é dado na Figura 7.1.

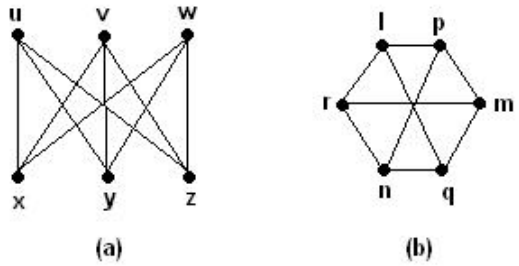


Figura 7.1: (a)Grafo G_1; (b) Grafo $G_{2.}$. G_1 G_2 são isomórficos.

- O isomorfismo é um problema NP-completo e não existem algoritmos polinomiais para resolvê-lo[1].

- Na representação de grafos por matrizes de adjacência, pode-se concluir que dois grafos G_1 e G_2 são isomórficos se, para alguma ordem de seus nós, suas matrizes forem iguais. Uma maneira simples, mas eficiente, de testar o isomorfismo é provar exatamente o contrário através do conceito de Ivariante. Invariante é uma propriedade que é preservada pelo isomorfismo, como por exemplo, número de nós, grau e determinante da matriz de adjacência.

[1]R. C. Read, D. G. Corneil. *The Graph Isomorphism Disease.* Journal of Graph Theory, vol.1, pp. 339-363, 1977.

- **Exemplo 1** — Através de uma adequada rotulação dos vértices, mostre que os dois grafos da Figura 7.2 são isomórficos.

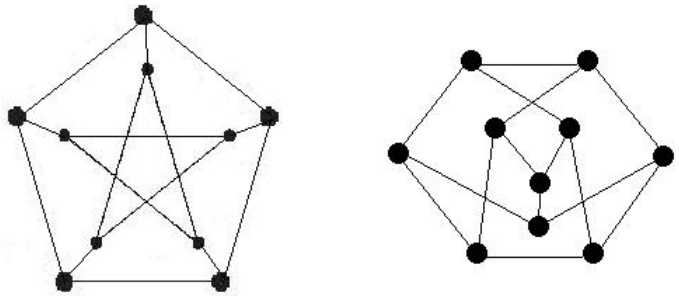


Figura 7.2: Grafos G_1 e G_2 do Exemplo 1.

- **Exemplo 2** — Explique porque os dois grafos da figura abaixo não são isomórficos.

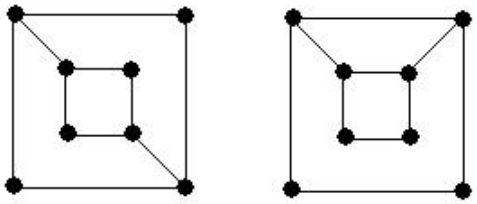


Figura 7.3: Grafos G_1 e G_2 do Exemplo 2.

- **Exemplo 3** — Classifique como verdadeiro ou falso:
  (a) Quaisquer dois grafos isomórficos têm a mesma seqüência de graus de vértices.
  (b) Dois grafos quaisquer com a mesma seqüência de graus de vértices são isomórficos.

Capítulo 8

Casamento de grafos

8.1 Introdução

- O uso de estruturas relacionais é menos restritiva do que procurar isomorfismo de grafos, já que nós ou arestas podem estar ausentes nos grafos que se deseja corresponder. Portanto, casamento é mais genérico do que isomorfismo de grafos, pois os grafos não precisam ser exatamente iguais.

- A estrutura relacional mais conhecida é o *grafo associativo*.

8.1.1 Grafo Associativo

- O grafo associativo é uma estrutura relacional construída a partir de dois outros grafos a fim de se determinar a correspondência entre eles.

- Dados dois grafos $G_1 = (V_1, E_1)$ e $G_2 = (V_2, E_2)$, um grafo associativo GA é construído como segue. Para cada $v_1 \in V_1$ e $v_2 \in V_2$, construa um nó de GA com rótulo (v_1, v_2). Depois, conecte dois nós (v_1, v_2) e $\left(v_1^{'}, v_2^{'}\right)$ se os mesmos forem *compatíveis*.

- Uma das primeiras restrições no estabelecimento de arestas entre nós é dada pelo *princípio da exclusão*. Este princípio estabelece a unicidade do casamento entre nós, *i.e.*, um nó de um grafo pode casar com somente um outro nó do grafo seguinte. A Figura 8.1 ilustra um grafo associativo construído a partir de duas árvores de 3 nós cada e respeitando o princípio da exclusão.

8.1.2 Clique

- Um casamento entre T_1 e T_2 na Figura 8.1(c) é simplesmente um conjunto de nós que são mutuamente compatíveis e o melhor (mais completo) casamento é aquele conjunto com maior número de nós compatíveis.

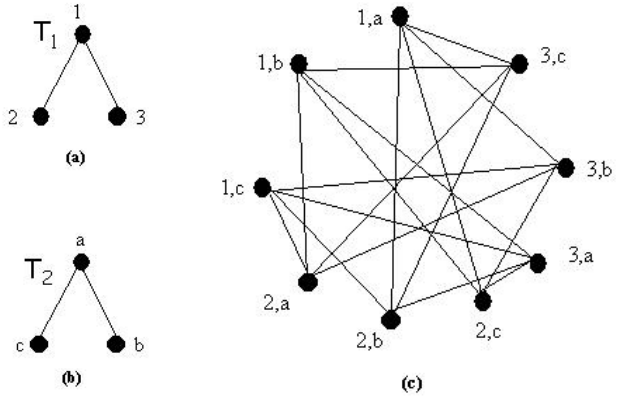


Figura 8.1: (a)Árvore T_1; (b) Árvore T_2; (c) Grafo associativo GA.

- Em outras palavras, devemos procurar no grafo associativo o subgrafo de maior cardinalidade cujos nós são mutuamente adjacentes. Este subgrafo é denominado de *clique máximo*.
- O Algoritmo 4, desenvolvido por Pardalos[1] apresenta um algoritmo simples para determinar o clique máximo em um grafo associativo. Usando este algoritmo, encontre o clique máximo para o grafo associativo da Figura 8.1(c).
- O Algoritmo 4 apresenta um resultado que pode não ser o exato, ou seja, nem sempre os nós que têm maior grau de incidência fazem parte do clique máximo.
- Pode ser resolvido o problema da correspondência em grafos associativos através da busca do maior componente k-conexo, e não do clique máximo como feito anteriormente. *Um componente k-conexo* é um conjunto de nós tal que cada nó é conectado com no mínimo k outros nós. A Figura 8.2 ilustra esse conceito.

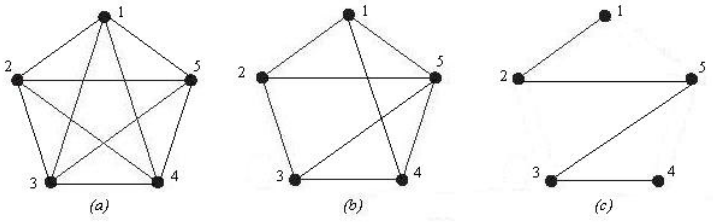


Figura 8.2: (a)Um clique de tamanho 5 (4 conexo); (b) Um grafo 3-conexo de 5 nós; (c) Um grafo 1-conexo de 5 nós.

[1]P. M. Pardalos. An Algorithm for Finding the Maximum Clique on an Arbitrary Graph. In http://www.mai.liu.se/~nigol/MAXCLIQUE

Algoritmo 4 Algoritmo Clique — Determina o clique máximo em um grafo associativo

Entrada: Grafo $GA = (V, E)$, n número de nós
Saída: Clique Máximo C_{GA}de GA, $|C_{GA}|$

Escolha entre todos os nós em V o nó v_m com maior grau de incidência
$V_H \leftarrow \{v_m\}$; $E_H \leftarrow \emptyset$; inicialmente H possui somente um nó.
$C_H \leftarrow \{v_m\}$; $|C_H| \leftarrow 1$

Para i de 2 até n **Faça**
  Escolha entre os nós em $V - V_H$ o nó v_m que possui o maior numero de nós adjacentes em H.
  Se existirem vários nós, escolha aquele com maior grau de incidência.
  Faça A ser o conjunto de nós em V_H adjacentes a v_m em H.
  Faça E_{HA} ser o conjunto de arestas em E_H incidentes aos nós somente em A.
  Faça $B = (A, E_{HA})$
  Se B tem um clique de tamanho $|C_H|$ **Então**
    $C_H \leftarrow C_H \cup \{v_m\}$
    $|C_H| \leftarrow |C_H| + 1$
  Fim Se
  Atualize o conjunto H, adicionando o nó v_m e todos as arestas entre V_H e v_m.
Fim Para
Retorne C_H, $|C_H|$

8.2 Exercícios

1. **(Questão)** Aplique a técnica de casamento de grafos para as duas árvores dadas na Figura 8.3. Construa o grafo associativo via princípio da exclusão e critério de hierarquia dos nós. Determine o clique máximo de maneira intuitiva. Descreva seu raciocínio para determinação do clique.

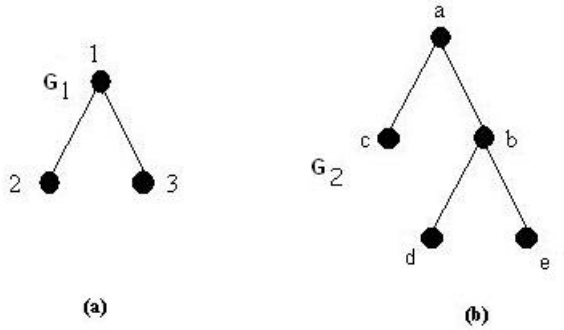


Figura 8.3: Casar os grafos G_1 e G_2.

Capítulo 9

Busca em grafos

9.1 Introdução

- A busca visa resolver uma questão básica: como explorar um grafo? Ou seja, deseja-se obter um processo de como caminhar pelos nós e arestas de um grafo.

- Para árvores, a resposta a essa questão é mais simples, pois pode-se caminhar por ela através do seguinte processo recursivo:

  1. Visite a raiz da árvore;
  2. Caminhe pela sub-árvore mais à esquerda da raiz;
     após, pela 2a. sub-árvore mais à esquerda;
     após, pela 3a. sub-árvore mais à esquerda;
     e assim sucessivamente.

- Algoritmos de busca são bastante utilizados em aplicações da área de inteligência artificial. Problemas de jogos são exemplos clássicos de aplicação.

9.1.1 Busca em profundidade

- Uma busca é dita em profundidade quando o critério de escolha de vértice marcado (visitado) obedecer a:
  dentre todos os vértices marcados e incidentes a alguma aresta ainda não explorada, escolher aquele mais recentemente alcançado na busca.

- O Algoritmo 5 apresenta um algoritmo simples de busca em profundidade[1]. A Figura 9.1 apresenta o grafo utilizado para acompanhamento dos algoritmos de busca em grafos.

[1] R. Johnsonbaugh. *Discrete Mathematics*. Macmillan Publishing Company, 1993.

Algoritmo 5 Algoritmo Busca em Profundidade

Entrada: Grafo conexo $G = (V, E)$ com nós ordenados $v_1, v_2, \ldots, v_n$
Saída: Árvore geradora $T = \left(V^{'}, E^{'}\right)$

$V^{'} \leftarrow \{v_1\}$
$E^{'} \leftarrow \emptyset$
$w \leftarrow v_1$
Enquanto *TRUE* **Faça**
  Enquanto existir uma aresta (w, v) que possa ser adicionada a T sem produzir ciclo **Faça**
    Escolha a aresta (w, v_k), com menor k
    Adicione (w, v_k) a $E^{'}$
    Adicione v_k a $V^{'}$
    $w \leftarrow v_k$
  Fim Enquanto
  Se $w = v_1$ **Então**
    Retorne(T)
  Fim Se
  $w \leftarrow$ pais de w em T (*backtracking*)
Fim Enquanto

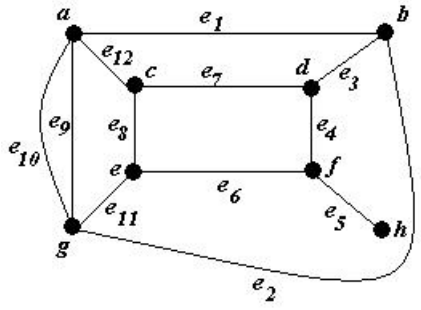


Figura 9.1: Grafo exemplo para acompanhamento das buscas em profundidade e em largura.

9.1.2 Busca em largura

- A idéia da busca em largura consiste em processar todos os nós em um dado nível antes de caminhar para um nível mais alto.
- O Algoritmo 6 implementa esta idéia. A Figura 9.1 apresenta o grafo utilizado para acompanhamento dos algoritmos de busca em grafos.

Algoritmo 6 Algoritmo Busca em Largura

Entrada: Grafo conexo $G = (V, E)$ com nós ordenados $v_1, v_2, \ldots, v_n$
Saída: Árvore geradora $T = \left(V^{'}, E^{'}\right)$

$S \leftarrow v_1$; S *lista ordenada*
$V^{'} \leftarrow v_1$
$E^{'} \leftarrow \emptyset$
Enquanto *TRUE* **Faça**
  Para $x \in S$, em ordem **Faça**
    Para $y \in V - V^{'}$, em ordem **Faça**
      Se (x, y) é uma aresta que não produz ciclo quando adiciionada a T **Então**
        Adicione aresta (x, y) a $E^{'}$
        Adicione y a $V^{'}$
      Fim Se
    Fim Para
  Fim Para
  Se *arestas não foram adicionadas* **Então**
    Retorne(T)
  Fim Se
  $S \leftarrow$filhos de S ordenados consistentemente
Fim Enquanto

9.2 Exercícios

1. Implemente a busca em profundidade para o grafo da Figura 9.2. Inicie a busca a partir do nó **A**. Apresente o desenvolvimento do algoritmo de busca.
2. Para o mesmo grafo acima, implemente a busca em largura partindo do nó **A**. Apresente o desenvolvimento.
3. Dado o grafo da Figura 9.3, implemente a busca em profundidade e em largura. Escolha o nó inicial.

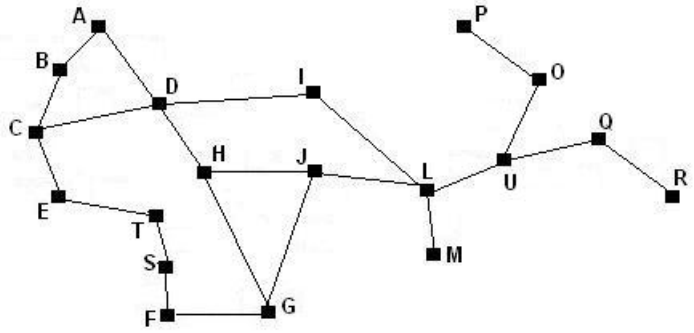


Figura 9.2: Implementar a busca em profundidade.

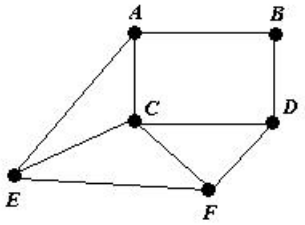


Figura 9.3: Implementar busca em profundidade e em largura no grafo G.

Referências Bibliográficas

[1] R. Diestel. Graph Theory. Springer, 1997.

[2] R. Gould. Graph Theory. The Benjamim/Cummings Publishing Company, 1988.

[3] M. Gondran, M. Minoux. Graphes et Algorithmes. Collection des Études et Recherches d´Électricité de France, Eds. Eyrolles, 1995.

[4] R. Johnsonbaugh. Discrete Mathematics. Macmillan Publishing Company, 1993.

[5] J. L. Szwarcfiter. Grafos e Algoritmos Computacionais. Ed. Campus, 1984.

[6] A. L. Furtado. Teoria dos Grafos: Algoritmos. LTC, 1973.