

UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE MESTRADO PROFISSIONAL EM MATEMÁTICA

TEORIA DOS GRAFOS E APLICAÇÕES

AUDEMIR LIMA DE SOUZA

MANAUS

2013

UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE MESTRADO PROFISSIONAL EM MATEMÁTICA

AUDEMIR LIMA DE SOUZA

TEORIA DOS GRAFOS E APLICAÇÕES

Dissertação apresentada ao Programa de Mestrado Profissional em Matemática da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Mestre em Matemática.

Orientador: Prof. Dr. Disney Douglas de Lima Oliveira

MANAUS

2013

Ficha Catalográfica
(Catalogação realizada pela Biblioteca Central da UFAM)

Souza, Audemir Lima de

S729t Teoria dos grafos e aplicações / Audemir Lima de Souza, 2014.

.... il. color.

Dissertação (Mestrado em Matemática) – Universidade Federal do Amazonas.

Orientador: Prof. Dr. Disney Douglas de Lima Oliveira

1. Teoria dos grafos 2. Matemática 3. Algoritmos I. Oliveira, Disney Douglas de Lima (Orient.) II. Universidade Federal do Amazonas III. Título

CDU(1997) 519.171(043.3)

AUDEMIR LIMA DE SOUZA

TEORIA DOS GRAFOS E APLICAÇÕES

Dissertação apresentada ao Programa de Mestrado Profissional em Matemática da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Mestre em Matemática.

Aprovado em 22 de agosto de 2013.

BANCA EXAMINADORA

Prof. Dr. Disney Douglas de Lima Oliveira
Presidente

Prof. Dr. Victor Augusto Giraldo
Membro

Prof. Dr. Mário Salvatierra Junior
Membro

AGRADECIMENTOS

Em primeiro lugar agradeço a Deus, por ter me concedido força, perseverança e coragem na dedicação dos meus estudos e para concluir este trabalho.

A minha família: esposa e filhos pelo apoio e compreensão; aos meus pais e irmãos, pela presença, incentivo e ajuda principalmente nas horas mais difíceis pelas quais passei.

Ao meu orientador, Prof. Dr. Disney Douglas, por sua paciência, postura profissional e pelo tempo dedicado a mim sempre que precisei.

A todos os meus professores do Departamento de Matemática da UFAM que muito me incentivaram e apoiaram, pois além de doarem conhecimentos estavam sempre dispostos a nos ajudar.

Aos meus colegas do curso, que mesmo sem nos conhecermos muito bem, agiram como verdadeiros amigos e contribuíram muito para eu chegar ao término deste trabalho.

À banca examinadora pelas sugestões dadas, que contribuíram para a melhoria deste trabalho.

Finalmente agradeço a todos meus amigos que, direta ou indiretamente, contribuíram para que eu concluísse esse curso.

RESUMO

Neste trabalho procuramos fazer uma abordagem simples sobre os conceitos de grafos e torná-los mais conhecidos, pois embora tenha uma grande variedade de aplicações, é um assunto pouco conhecido no ensino básico. No intuito de divulgar modelagem usando a teoria dos grafos, serão mostrados conceitos e definições, modelos e exemplos clássicos aplicados pelos primeiros estudiosos dessa teoria, com a intenção de motivar o raciocínio lógico de nossos alunos para auxiliá-los nas resoluções de outros problemas. Será mostrado como tais informações podem ser representadas no computador e como decidir por qual representação optar. Também apresentaremos algoritmos que podem nos trazer resultados automáticos ou informatizados, pois determinados problemas só resolveremos com o auxílio da máquina. Acreditamos que esta forma de modelar problemas pode contribuir para a melhoria do ensino-aprendizagem e servir como elemento motivador para alunos e professores que buscam melhorar seus conhecimentos sobre a teoria dos grafos e suas aplicações.

Palavras-chave: Grafos bipartido, grafo Euleriano, grafo Hamiltoniano, árvore, coloração, caminho mínimo.

ABSTRACT

In this paper we take a simple approach about the concepts of graphs so they can be better known, although it has a wide variety of applications, it is a subject little known in basic education. In order to disseminate modeling using graph theory, concepts and definitions, models and classic examples applied by early scholars of this theory will be shown, with the intention of motivating the logical reasoning of our students to assist them in the resolution of other problems. It will be shown how such information can be represented in the computer and how to decide which representation to choose. We will also present algorithms that can bring automatic or computerized results because certain problems will be solved only with the help of the machine. We believe that this way of modeling problems can contribute to the improvement of teaching and learning and serve as motivator for students and teachers who seek to improve their knowledge of graph theory and its applications.

Keywords: bipartite Graphs, Eulerian graph, Hamiltonian graph, tree, coloring, minimum path.

Sumário

Introdução	1
1 Teoria dos grafos	3
1.1 Um breve histórico	3
1.1.1 A teoria dos grafos no Brasil	4
1.2 Conceitos básicos	4
1.3 Árvore	11
1.4 Outras definições	12
1.5 Modelos de grafos	13
1.5.1 Grafos de superposição de nichos em ecologia	13
1.5.2 Grafos de relacionamento	13
1.5.3 Grafos de influência	14
1.5.4 Grafos de torneios Round-Robin	14
1.5.5 Grafo da web	15
1.5.6 PageRank	15
1.5.7 Grafos de mapas rodoviários	18
1.6 Problemas clássicos de grafos	18
1.6.1 Grafos eulerianos e as sete pontes de Königsberg	18
1.6.2 Grafos e ciclos hamiltonianos	21
2 Representação computacional de grafos	24
2.1 Matriz de adjacência	24
2.2 Matriz de incidência	26
2.3 Listas de adjacência	27
3 Algoritmos de grafos	28
3.1 Algoritmo de busca	28
3.1.1 Algoritmo de busca em profundidade - DFS	28
3.1.2 Algoritmo de busca em largura	34
3.2 Ordenação topológica	35
3.3 Caminho mínimo	36

3.4	Árvores espalhadas mínimas	37
4	Planaridade	41
4.1	Definições e aplicações	41
4.2	Fórmula de Euler	44
4.3	Coloração de grafos	46
5	Aplicações	50
5.1	Caminho mínimo	50
5.1.1	Problema 1	50
5.1.2	Problema 2	55
5.1.3	Problema 3	64
5.2	Coloração	73
5.2.1	Problema 1	73
5.2.2	Problema 2	74
5.2.3	Problema 3	75
	Referências bibliográficas	78

LISTA DE SÍMBOLOS

$G = (V, E)$	Grafo com V vértices e E arestas.
$V(G)$	Conjunto de vértices do grafo G .
$E(G)$	Conjunto de arestas do grafo G .
$e = \{u, v\}$	Uma aresta que liga os vértices u e v .
K_n	Grafo completo com n vértices.
$g(v)$	Grau de um vértice v .
$ V $	Número de vértices do conjunto V .
$ E $	Número de arestas do conjunto E .
$K_{m,n}$	Grafo bipartido completo.
(u, v)	aresta orientada.
$N(v)$	Vizinhança aberta do vértice v .
$N[v]$	Vizinhança fechada do vértice v .
$\delta(G)$	Grau mínimo do grafo G .
$\Delta(G)$	Grau máximo do grafo G .
\bar{G}	Grafo complementar de G .
$a_{i,j}$	Elemento de uma matriz.
$O(V ^2)$	Ordem/Tempo de $ V ^2$.
$d[v]$	Distância da origem ao vértice v .
$\chi[G]$	Número cromático de um grafo.

Introdução

Este trabalho foi elaborado com o intuito de contribuir para um estudo inicial de grafos, especialmente voltado aos alunos do ensino médio integrado ou iniciantes do curso de graduação na área de ciências exatas, uma vez que a maioria desses alunos, considerando minha experiência como docente de matemática, sempre ansiou saber mais acerca do tema aqui proposto, suas aplicações e que tipos de problemas podem ser resolvidos.

É importante ressaltar que, na matemática, as resoluções de problemas são os primeiros passos dados pelo estudante, entretanto, no decorrer de sua vida escolar/acadêmica, deverá aprender que a interpretação pode ser colocada no papel através de uma modelagem do problema, antes mesmo de pensar em sua solução. Esta forma de interpretar, modelando o problema, ou seja, representando-o através de um desenho, enriquece o conhecimento matemático do aluno que poderá tomar decisões e usar técnicas ou ferramentas adequadas para concluir tal solução. E a teoria dos grafos oferece formas diversificadas de representar objetos matemáticos, modela o problema em um desenho composto por vértices e arestas e encontra uma maneira de resolvê-lo manualmente ou com o auxílio do computador.

Assim sendo, no ano de 2010, e por intermédio do PROFMAT (Mestrado Profissional em Matemática), encontrei a opção para alavancar meus conhecimentos e foi por intermédio de muita leitura, pesquisas e experiências que alcancei meu objetivo. Aliás, no decorrer do curso, pensava tão somente na necessidade de meus alunos, e esse foi o diferencial deste trabalho, pois além do apoio de meus colegas, professores e orientador na compreensão e amadurecimento na teoria dos grafos, também consultei os discentes, informando o que pretendia fazer. Colhi dados e cheguei à conclusão de que deveria montar um material composto de noções básicas e gerais sobre a teoria dos grafos e algumas aplicações para mostrar aos estudantes a grande aplicabilidade deste assunto que no decorrer dos anos ganha mais espaço nas resoluções de problemas.

Desta forma, o trabalho aqui representado acerca da teoria dos grafos teve como objetivo principal demonstrar situações que podem, por um lado, ajudar a melhorar a compreensão sobre o referido assunto, e por outro, incentivar a resolução de problemas, a modelagem de situações, o desenvolvimento da comunicação e das formas de pensamento matemático. Além deste objetivo, outro alvo traçado na pesquisa foi o leitor interessado no tema. Este (o leitor) deve ser preparado no mundo real, verificando que a teoria

dos grafos tem aplicações em diferentes áreas e que é possível desenvolver técnicas de visualizações, raciocínios e habilidades manuais e computacionais, estimulando aplicações interdisciplinares como forma de construção do conhecimento em situações cotidianas.

Diante disso, esta monografia está dividida em cinco capítulos, distribuídos da seguinte forma: o primeiro traz um breve histórico acerca da teoria dos grafos e como esta é vista no Brasil nos dias atuais; também serão abordados conceitos básicos, modelos e exemplos clássicos de grafos. Já o segundo capítulo apresentará como os problemas poderão ser representados em um computador por meio de matrizes ou listas. Em continuidade, o terceiro capítulo abordará alguns algoritmos para resolver problemas modelados por grafos. O quarto tratará das definições e aplicações de planaridade e coloração de mapas, por intermédio de grafos a eles associados. Por fim, no quinto capítulo serão feitas as aplicações dos algoritmos para resolução de problemas modelados por grafos, levando a uma solução satisfatória.

Tendo em vista o exposto, espera-se que este trabalho capacite os alunos do ensino médio e superior a expressar suas ideias e desenvolvê-las para resolver problemas, justificando e provando afirmações sempre que possível por meio da teoria dos grafos; e, quanto ao leitor interessado no assunto, propõe-se que este amplie a ideia apresentada neste trabalho que não esgota o conhecimento em torno do assunto aqui referido.

Capítulo 1

Teoria dos grafos

Neste Capítulo, mostraremos um breve histórico, as definições e conceitos básicos de grafos. Também serão citados modelos de grafos e alguns problemas clássicos.

Os grafos são estruturas que consistem em vértices e arestas que ligam estes vértices. Problemas em várias disciplinas podem ser resolvidos usando modelos de grafos. Pode-se resolver problemas tais como de espécies diferentes em um nicho ecológico, representar quem influencia quem em uma organização, resultados em torneios, modelar conveniência entre pessoas, colaboração entre pesquisadores, chamadas telefônicas, links entre websites, mapas rodoviários, designar tarefas a empregados de uma organização, determinar se é possível percorrer as ruas de uma cidade sem passar por uma mesma rua duas ou mais vezes, encontrar o número necessário de cores para pintar as regiões de um mapa, determinar se um circuito pode ser implementado em uma placa de circuito plano, tais como distinguir dois compostos químicos com a mesma forma molecular de estruturas diferentes. Pode-se também determinar se dois computadores estão ligados por um link de comunicação, planejar exames e associar canais de televisão às estações. Este capítulo introduzirá conceitos básicos da teoria dos grafos e apresentará alguns modelos e exemplos diferentes dele.

1.1 Um breve histórico

Leonhard Euler¹ (1708 - 1783) foi o primeiro matemático a escrever sobre a abstração dos grafos. Com 28 anos de idade ganhou fama mundial ao resolver um problema que por décadas foi um desafio para os matemáticos da época, conhecido como problema da Basileia²[8].

Em 1736, outro problema resolvido por Euler foi o das sete pontes de Königsberg que

¹É atualmente considerado um dos maiores matemáticos de todos os tempos, pois produziu mais de 1100 artigos e livros, durante os últimos 17 anos de sua vida ficou cego, quando produziu quase que a metade de seus trabalhos.

²Série infinita da soma dos inversos dos quadrados.

cruzam o rio Pregel, estabelecendo ligações entre duas ilhas e as margens opostas deste rio, conforme ilustrado na Figura 1.19 do Capítulo 1. Neste problema, ele representou cada faixa de terra por um ponto e as pontes por ligações entre esses pontos [5].

Apesar da beleza do problema das sete pontes, a solução foi um detalhe na imensidão de suas contribuições. E tal solução não apresentava ser de grande relevância para a ciência, pois seu método de abstração ficou oculto por aproximadamente 150 anos. De Euler a última década do século XIX, poucos trabalhos foram propostos com tal abstração.

Em 1847 Gustav Robert Kirchhoff³ utilizou modelos de grafos no estudo dos circuitos elétricos, criando a teoria das árvores.

Com isso, outros cientistas começaram a notar a provável aplicabilidade desta teoria e dez anos mais tarde Arthur Cayley⁴ (1821 a 1895) utilizou a ideia de árvores para outras aplicações, tais como a enumeração dos isômeros de hidrocarbonetos alifáticos saturados, em química orgânica.

A teoria dos grafos contou com o importante auxílio do irlandês William Rowan Hamilton⁵ (1805 a 1865) que ao inventar um jogo simples que consistia na busca de um percurso fechado envolvendo todos os vértices de um dodecaedro regular, de tal modo que cada um deles fosse visitado uma única vez, deu origem ao estudo dos grafos Hamiltonianos que tem por definição, segundo Rabuske (1992, p. 45), “encontrar um caminho fechado, passando uma única vez por todos os vértices”.

A partir de 1970, a teoria dos grafos teve um grande salto com o desenvolvimento acelerado dos computadores. Foi, então, que surgiram publicações referentes a algoritmos de grafos, abrindo, assim, possibilidades para utilização aplicada desta teoria.

1.1.1 A teoria dos grafos no Brasil

No Brasil, a teoria dos grafos chegou, segundo Boaventura [3], no ano de 1968 com a apresentação de alguns trabalhos sobre esta teoria no I Simpósio Brasileiro de Pesquisa Operacional. Desde então, algumas universidades como UFRJ, UFF, USP, UNESP e UNICAMP começaram a realizar trabalhos de pesquisa sobre a teoria dos grafos, de modo que hoje as várias universidades brasileiras, incluindo as citadas, possuem em seus quadros de docentes, pesquisadores na referida teoria e aplicações.

1.2 Conceitos básicos

Definição 1.2.1. *Um grafo $G = (V, E)$ consiste de V , um conjunto não vazio de vértices (ou nós) e de E , um conjunto de arestas. Cada aresta tem um ou dois vértices associados*

³Cientista nascido na cidade de Königsberg.

⁴Britânico nascido em Richmond.

⁵Matemático, físico e astrônomo.

a ela, chamados de suas extremidades. Dizemos que cada aresta liga ou conecta suas extremidades. [11]

Outras notações para denotar um grafo: $G = (V(G), E(G))$, $G(V, E)$ ou simplesmente G .

Uma aresta $e \in E(G)$ é representada por $e = \{u, v\}$ sempre que interliga dois vértices u e v de V . Dois vértices ligados por uma mesma aresta são denominados *adjacentes* e pode-se dizer que uma aresta e é incidente em u , se u for uma extremidade de e . Exemplos de grafos na Figura 1.1.

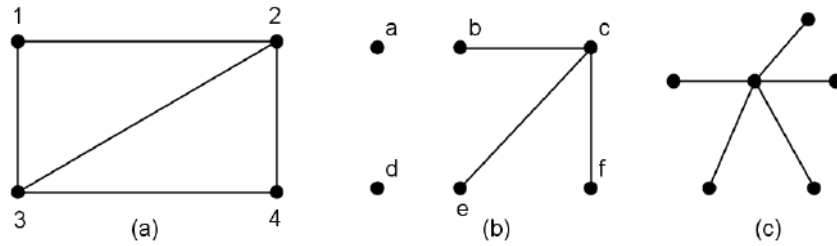


Figura 1.1: Grafos.

Um grafo de aresta do tipo $e = \{u, u\}$, ou seja, com extremidades iguais da aresta, é denominada *laço*. A existência de uma ou mais arestas entre o mesmo par de vértices, são chamadas de *arestas paralelas* ou *arestas múltiplas*. Um grafo que possui arestas paralelas denomina-se *multigrafo*. Na Figura 1.2(a) é apresentado um grafo com laços, e na Figura 1.2(b) um grafo com arestas paralelas. Quando o grafo possui laços e arestas múltiplas denomina-se *pseudografo*. [11]

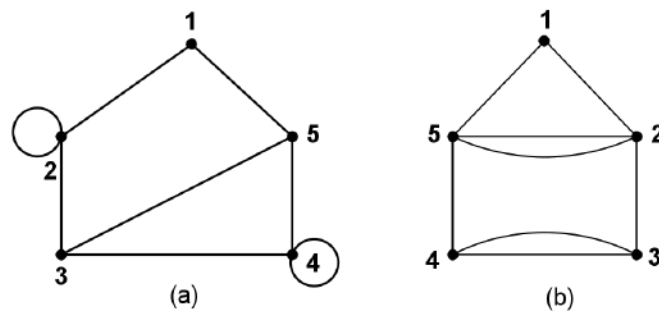


Figura 1.2: (a) Grafos com laços. (b) Grafo com arestas múltiplas.

Definição 1.2.2. Um grafo G é dito *simple* se não possui laço ou aresta múltipla. É chamado de *trivial* quando o grafo possui apenas um vértice. [11]

Os grafos da Figura 1.1 são grafos simples, já os grafos da Figura 1.2 não são.

Definição 1.2.3. Um grafo G é dito completo se todo vértice em G está conectado a qualquer outro vértice em G . [11]

Um grafo completo com n vértices é denotado por K_n . Na Figura 1.3 são apresentados exemplos dos grafos K_3 , K_4 e K_5 .

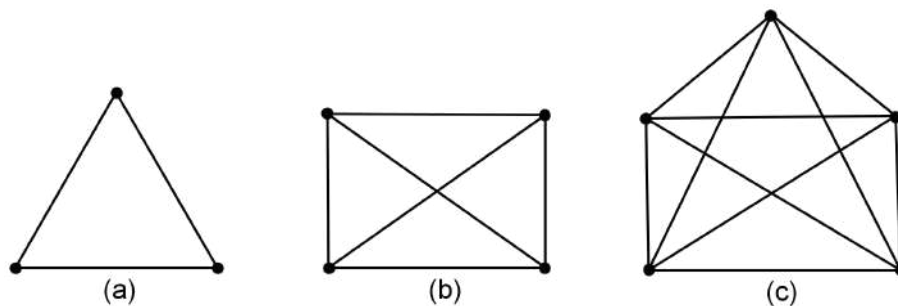


Figura 1.3: Grafos completos.

Dado um grafo $G(V, E)$, o grau de um vértice $v \in V$, denotado por $g(v)$, é igual ao número de arestas que incidem nele. Na Figura 1.3(b), o grau de cada vértice é 3. Considerando que em cada vértice $v \in V$ incidem $g(v)$ arestas e que cada aresta incide em 2 vértices, tem-se:

Teorema 1.2.1. Para todo grafo $G(V, E)$

$$\sum_{v \in V(G)} g(v) = 2 \cdot |E|,$$

onde $|E|$ representa o número de arestas do conjunto E .

Demonstração: Quando contamos os graus dos vértices de um grafo estamos contando o número de arestas que incidem em cada vértice. Como cada aresta está sempre ligada a dois vértices, cada aresta foi contada duas vezes.

Quando todos os vértices de um grafo possuem o mesmo grau, ele é chamado de *grafo regular* de grau r . Todos os grafos da Figura 1.3 são exemplos de grafos regulares. A Figura 1.3(b) possui grau $r = 3$.

Definição 1.2.4. Um passeio é uma sequência de arestas $v_0v_1, v_1v_2, v_2v_3, \dots, v_{s-1}v_s$, s é o comprimento do passeio. Se todas as arestas do passeio são distintas, o passeio é chamado *trilha*; se $v_0 = v_s$, o passeio é uma *trilha fechada*. Se, além das arestas, todos os vértices são distintos, então, tem-se um *caminho* e se $v_0 = v_s$, tem-se um *ciclo*. [11]

Corolário 1. Todo grafo G possui um número par de vértices de grau ímpar.

Demonstração: Se tivéssemos um número ímpar de vértices de grau ímpar, a soma dos graus seria ímpar. Mas a soma dos graus é o dobro do número de arestas e, portanto, é um número par.

Dado um grafo $G(V, E)$, $H(V', E')$ é dito *subgrafo* de G se $V' \subseteq V$ e $E' \subseteq E$. A Figura 1.4 mostra um exemplo de um grafo e subgrafos de G .

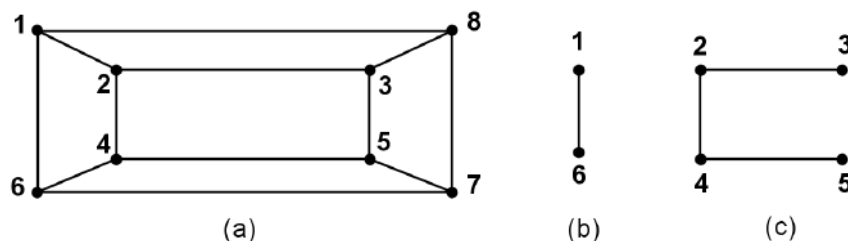


Figura 1.4: Grafos e subgrafos.

Definição 1.2.5. Um grafo G é chamado *conexo* se existe um caminho para cada par de vértices de G . Caso contrário, é chamado *desconexo*.

Um grafo desconexo possui partes conexas que são chamadas de *componentes*. Um grafo conexo possui um único componente conexo enquanto um grafo desconexo possui vários componentes conexas. A Figura 1.5(a) é um exemplo de grafo conexo, enquanto a Figura 1.5(b) é um grafo desconexo com o número de componentes igual a 3.

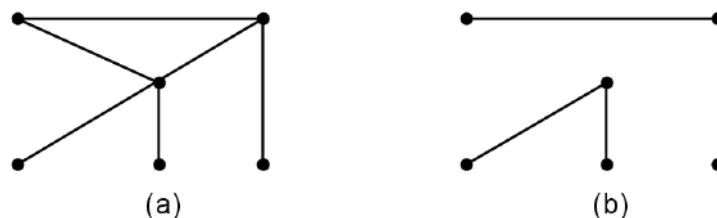


Figura 1.5: (a) Grafo conexo. (b) Grafo desconexo.

Definição 1.2.6. Um grafo $G(V, E)$ é chamado *bipartido* quando seu conjunto de vértices V pode ser particionado em dois subconjuntos V_1 e V_2 , tal que toda aresta de G faz a ligação de um vértice de V_1 a um vértice de V_2 .

Um grafo bipartido de G pode ser escrito como $G(V_1 \cup V_2, E)$ e é dito *grafo bipartido completo* quando existe uma aresta para todo par de vértices u, v , sendo $u \in V_1$ e $v \in V_2$, denotado por $K_{m,n}$ sendo $m = |V_1|$ e $n = |V_2|$. A Figura 1.6 mostra exemplos de grafos bipartidos.

Teorema 1.2.2. Um grafo G é bipartido se, e somente se, não contém ciclos de comprimento ímpar.

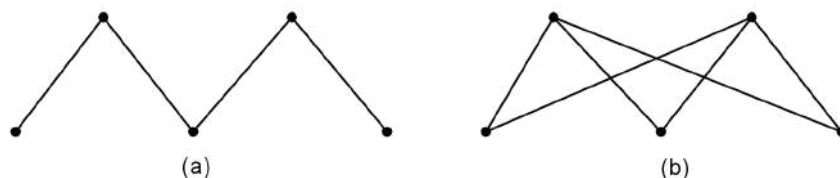


Figura 1.6: (a) Grafo bipartido. (b) Grafo bipartido completo.

Demonstração:

(\Rightarrow) Seja G bipartido. Se não houver um ciclo em G , não há o que mostrar. Se há um ciclo em G , este alterna vértices de V_1 e V_2 , dois subconjuntos disjuntos. Partindo de V_1 , por exemplo, para retornar ao ponto de partida teremos que utilizar um número par de arestas. O ciclo é, portanto, de comprimento par.

(\Leftarrow) Considerando apenas grafos conexos, seja G um grafo sem ciclos ímpares, participando em dois subconjuntos V_1 e V_2 , independentes e disjuntos. Tomando-se primeiramente um vértice qualquer v , o subconjunto V_1 será formado por todos os vértices w tais que exista um caminho de comprimento par entre v e w . O subconjunto V_2 será formado por todos os vértices w tais que exista um caminho de comprimento ímpar entre v e w . Os conjuntos V_1 e V_2 são disjuntos, pois se w estivesse em V_1 e V_2 ao mesmo tempo, haveria um caminho de comprimento par e um comprimento de tamanho ímpar ligando v a w . Esses caminhos podem se cruzar (ou não) antes de chegar em w , produzindo alguns ciclos, como o exemplo da Figura 2.7. Como o número usado de arestas nos caminhos é ímpar (a soma do número de arestas dos dois caminhos) isso produziria pelo menos um ciclo ímpar em G , contrariando a hipótese.

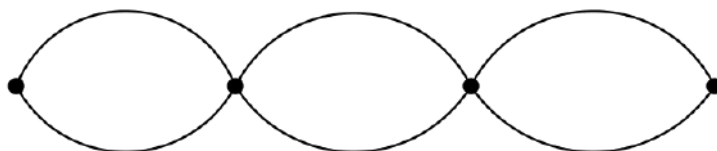


Figura 1.7: Caminhos e ciclos.

Definição 1.2.7. *Dois grafos simples G_1 e G_2 são ditos isomorfos se existir uma bijeção f de $V(G_1)$ em $V(G_2)$ tal que dois vértices v e w são adjacentes em G_1 se, e somente se $f(v)$ e $f(w)$ são adjacentes em G_2 .*

As representações geométricas dos grafos da Figura 1.8 se tornam coincidentes ao se aplicar a função f indicada na figura. Logo, G_1 e G_2 são isomorfos entre si, já os grafos G_1 e G_3 , não o são, pois não existe uma função f que realize esse mapeamento entre os vértices e que faça com que as representações tornem-se coincidentes.

Exemplo: Os grafos $G = (V, E)$ e $H = (W, F)$ apresentados na Figura 1.9, são isomorfos.

De fato, a função f , com $f(u_1) = v_1$, $f(u_2) = v_4$, $f(u_3) = v_3$ e $f(u_4) = v_2$, é uma correspondência biunívoca entre V e W . Para ver que esta correspondência preserva a adjacência, observe que os vértices adjacentes em G são u_1 e u_2 , u_1 e u_3 , u_2 e u_4 e u_3 e u_4 , e cada um dos pares $f(u_1) = v_1$ e $f(u_2) = v_4$, $f(u_1) = v_1$ e $f(u_3) = v_3$, $f(u_2) = v_4$ e $f(u_4) = v_2$, e $f(u_3) = v_3$ e $f(u_4) = v_2$ são adjacentes em H .

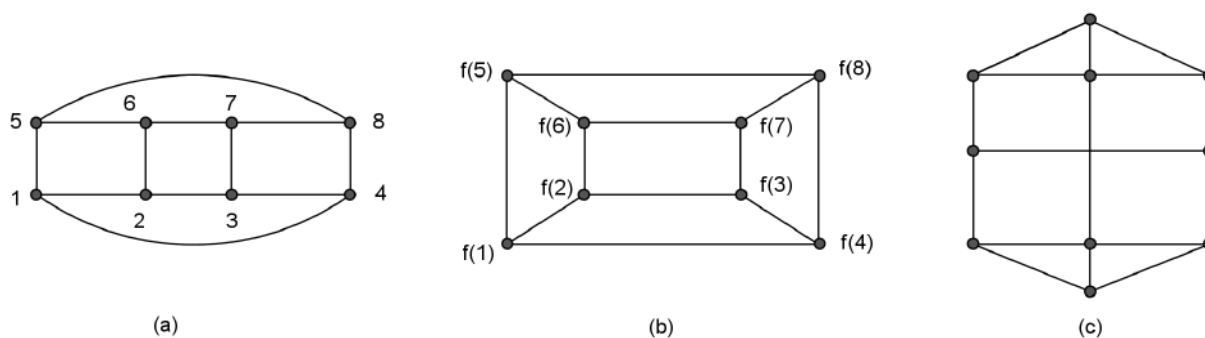


Figura 1.8: Grafo isomorfos e não isomorfos .

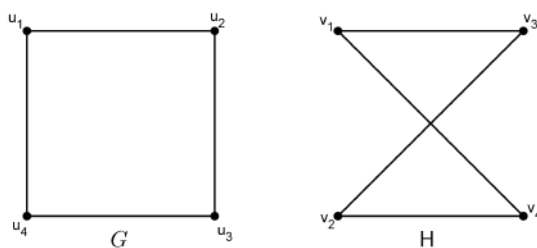


Figura 1.9: Grafo isomorfos.

Não é fácil determinar se dois grafos simples são isomorfos. Existem $n!$ correspondências biunívocas possíveis de dois grafos simples com n vértices. Se n for grande, testar cada uma dessas correspondências para ver se ela preserva a adjacência e a não adjacência não é prático. Para facilitar, analise: grafos simples isomorfos devem ter o mesmo número de arestas, pois a correspondência biunívoca entre os vértices estabelece uma correspondência biunívoca entre as arestas. Além disso, os graus dos vértices em grafos simples isomorfos devem ser os mesmos, isto é, um vértice de grau d em G deve corresponder a um vértice $f(v)$ de grau d em H , pois um vértice w em G é adjacente a v se $f(v)$ e $f(w)$ forem adjacentes em H .

Exemplo: Os grafos da Figura 1.10 não são isomorfos.

G e H têm, ambos, cinco vértices e seis arestas. H tem um vértice de grau 1, enquanto G não tem nenhum vértice de grau 1. Logo, G e H não são isomorfos.

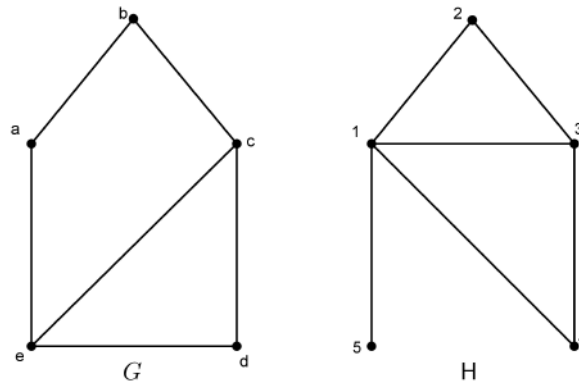


Figura 1.10: Grafo não isomorfos.

Definição 1.2.8. Um grafo é dito ponderado quando são atribuídos pesos às suas arestas.

Esses pesos podem representar custo ou distâncias. Tem-se um exemplo de grafo ponderado na Figura 1.11.

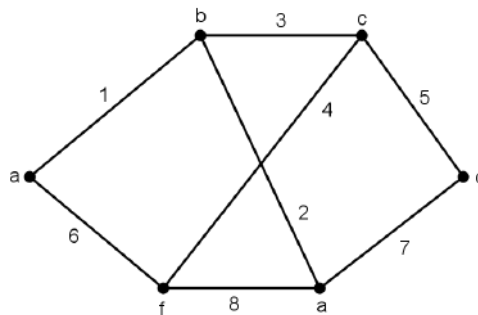


Figura 1.11: Grafo ponderado

Os grafos vistos até agora são grafos não direcionados (ou orientados), pois suas arestas não possuem orientação (aponta de um vértice para outro). Um grafo direcionado $D(V, E)$ pode ser chamado de grafo de orientado ou também chamado de *dígrafo*.

Definição 1.2.9. Um grafo orientado $G = (V, E)$ consiste de um conjunto de vértices V não vazio e de um conjunto de arestas orientadas. Cada aresta está associada a um par ordenado de vértices. Uma aresta orientada associada a um par ordenado (u, v) começa em u e termina em v .

Quando se descreve um grafo orientado, usa-se uma flecha apontando de u para v , indicando que começa em u e termina em v . Um grafo orientado pode conter laços e arestas orientadas múltiplas que começam e terminam nos mesmos vértices. Pode conter também arestas orientadas que conectam u e v em ambos os sentidos. Quando um grafo orientado não tem laços nem arestas orientadas múltiplas, ele é chamado de grafo orientado simples. Veja a Figura 1.12. Para alguns modelos, pode-se precisar de um grafo em que algumas arestas sejam não orientadas, enquanto outras sejam orientadas. Um grafo composto de arestas orientadas e não orientadas é chamado de *grafo misto*.

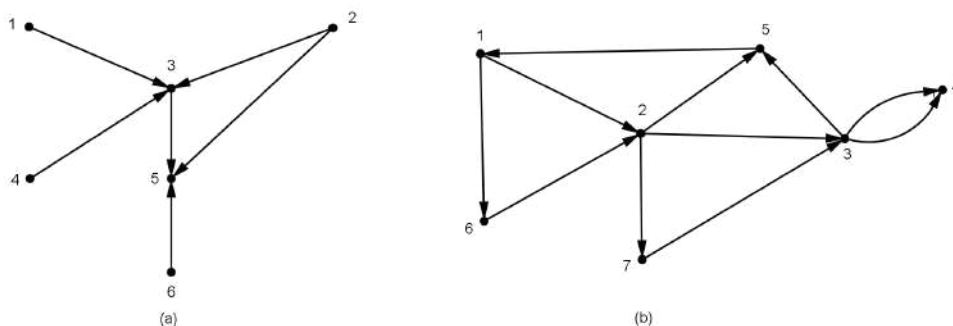


Figura 1.12: Grafos orientados.

1.3 Árvore

Um grafo $T = (V, E)$ que não possui ciclos e é conexo é chamado de *árvore*. Toda árvore possui as seguintes características:

- Seja $v \in V$, se v possui grau menor ou igual a 1, então, v é uma folha. Caso o grau seja maior que 1, v é um vértice interno.
- Uma árvore T com n vértices possui $n - 1$ arestas.
- Um grafo G é uma árvore somente se existir um único caminho entre cada par de vértices de G .
- Um conjunto de árvores é chamado de floresta.[4]

Pode-se dizer que grafos que não contêm nenhum ciclo simples, mas que não são necessariamente conexos são chamados de *florestas* e, têm a propriedade: cada uma de seus componentes conexos é uma árvore.

A terminologia para árvores tem origem na botânica e genealogia. Sendo T uma *árvore com raiz* ou *árvore enraizada*, T é uma árvore que contém vértices designados r , chamado de *raiz da árvore*. Como existe um único caminho simples da raiz r para qualquer outro

vértice v em T , isso determina uma direção das arestas de T . Portanto, T pode ser visto como um grafo orientado. Qualquer árvore pode ser transformada em uma árvore com raiz pela simples seleção de um vértice como raiz.[11]

Se v for um vértice em T diferente do raiz, o *pai* de v é o único vértice u tal que existe uma aresta orientada de u para v . Quando u for pai de v , v é chamado *filho* de u . Vértices com mesmo pai são chamados *irmãos*. Os *ancestrais* de um vértice são os vértices no caminho da raiz até o pai deste vértice. Os *descendentes* de um vértice v são os vértices que tem v como um ancestral. Um vértice de uma árvore é chamado de *folha* se não tiver filhos. Os vértices que têm filhos são chamados de vértices *internos*. A raiz é um vértice interno, a menos que seja o único vértice do grafo cujo caso é também chamado de folha.

Dado um grafo $G(V(G), E(G))$, denomina-se *subgrafo gerador* o grafo $H(V(H), E(H))$ que é subgrafo de G , tal que $V(G) = V(H)$. Se o subgrafo H é uma árvore, então é chamado de *árvore geradora*. A Figura 1.13 mostra exemplos de árvores.

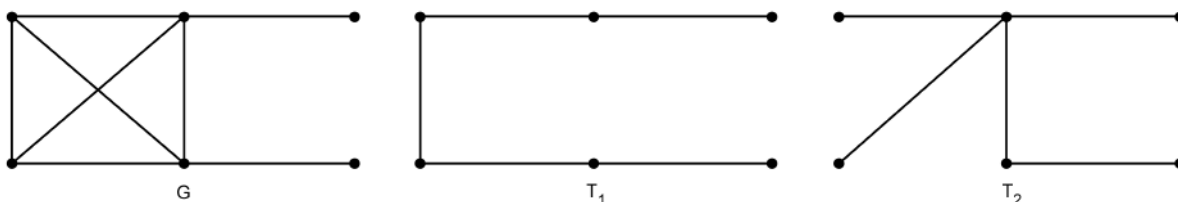


Figura 1.13: Grafo G e árvores geradoras T_1 e T_2 .

Supondo que G é um grafo conexo ponderado, cada aresta de G está associada a um número não negativo chamado de peso da aresta. Então, qualquer árvore geradora T de G está associada a um peso total obtido pela soma dos pesos das arestas em T . Uma *árvore minimal geradora* ou *Árvore geradora mínima* de G é a árvore geradora cujo peso total é o menor possível.

1.4 Outras definições

O conjunto de vértices adjacentes de v , é chamado *vizinhança aberta* de v , denotado por $N(v)$. A *vizinhança fechada* de v é denotada e definida por $N[v] = N(v) \cup \{v\}$. Pode-se estender esta definição para conjuntos de vértices ($N(S)$ e $N[S]$), sendo S um conjunto de vértices dado. Por exemplo, no grafo da figura 1.4(a) tem-se $N(5) = \{3, 4, 7\}$ e $N[5] = \{3, 4, 5, 7\}$.

Um grafo G é chamado de *Grafo nulo ou vazio* quando o conjunto de arestas $A(G)$ é vazio. São *grafos esparsos* aqueles que tem poucas arestas, muito menos do que o número máximo possível $n(n - 1)/2$, onde n é o número de vértices.

Grafo complementar, denotado por \bar{G} é um grafo em que $V(G) = V(\bar{G})$ e que $A(G) \cup A(\bar{G})$ inclui todas as arestas de G . A Figura 1.14 mostra o exemplo de dois grafos complementares.

Algoritmo é conjunto ordenado de passos computacionais que toma um conjunto de valores como entrada e produz um conjunto de valores como saída.

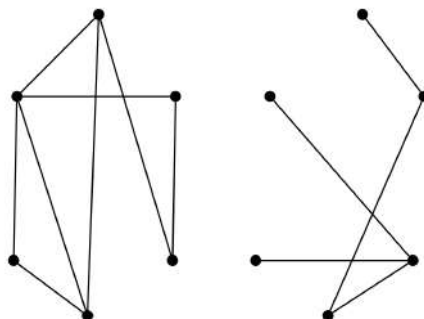


Figura 1.14: Grafo complementares.

1.5 Modelos de grafos

Os grafos são usados em grande variedade de modelos. Serão apresentados alguns modelos de grafos em diversos campos.

1.5.1 Grafos de superposição de nichos em ecologia

Os grafos são usados em muitos modelos que envolvem a interação entre diferentes espécies de animais. Por exemplo, a competição entre espécies em um ecossistema pode ser modelada usando um *grafo de superposição de nicho*, onde cada espécie é representada por um vértice e uma aresta não orientada conectada a dois vértices se duas espécies representadas por esses vértices competem (fontes de alimentos que usam são as mesmas). Um grafo de superposição de nicho é um grafo simples, pois não são necessários nem laços e nem arestas múltiplas. O grafo da Figura 1.15 modela o ecossistema de uma floresta.

1.5.2 Grafos de relacionamento

Pode-se usar modelos de grafos para representar diversas relações entre pessoas. Por exemplo, um grafo simples representa se duas pessoas se conhecem, ou seja, se elas têm um relacionamento. Cada pessoa é representada por um vértice e uma aresta não orientada é usada para ligar duas pessoas quando elas se conhecem. Não há necessidade de laços e arestas múltiplas. Um pequeno grafo de relacionamento é mostrado na Figura 1.16.

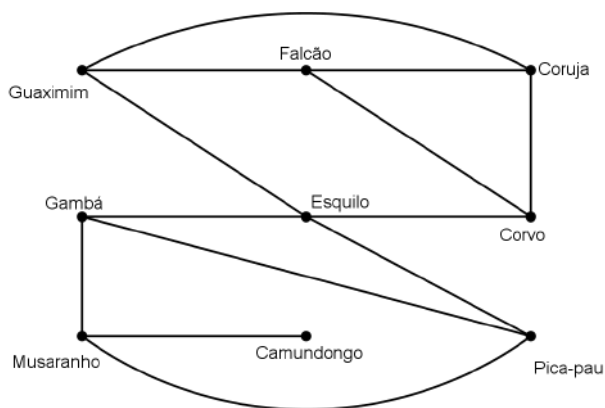


Figura 1.15: Ecossistema de uma floresta.

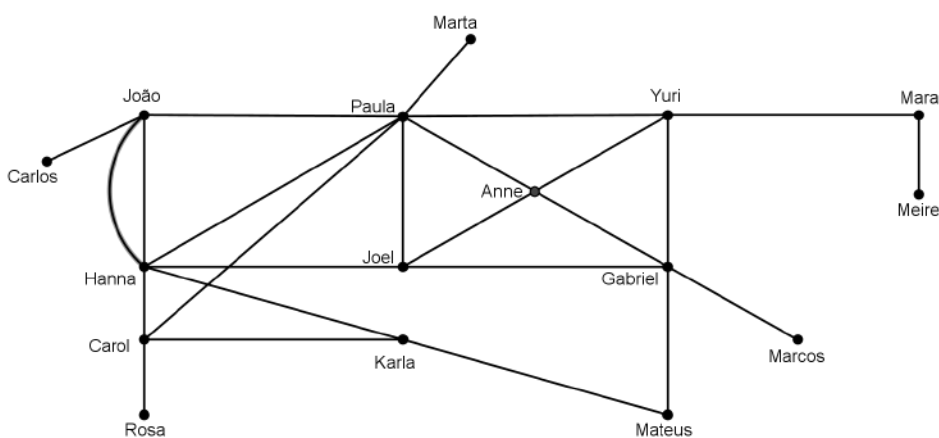


Figura 1.16: Gráfico de relacionamento.

1.5.3 Grafos de influência

Observando o comportamento de grupo, certas pessoas podem influenciar o pensamento de outras. Um grafo orientado chamado de *grafo de influência* pode ser usado para modelar este comportamento, onde cada pessoa do grupo é representada por um vértice e uma aresta orientada mostra quando uma pessoa representada pelo vértice a influencia outra pessoa representada pelo vértice b , pois o vértice a aponta para o vértice b . Um exemplo de um grafo de influência é mostrado na Figura 1.17.

1.5.4 Grafos de torneios Round-Robin

Neste torneio um time joga com cada outro time exatamente uma vez e podem ser modelados usando grafos orientados em que cada time é representado por um vértice. Observa-se que (T_1, T_3) é uma aresta se o time T_1 ganhou do time T_3 . Este grafo é um grafo orientado simples que não contém nem laços nem arestas múltiplas. Tal modelo está

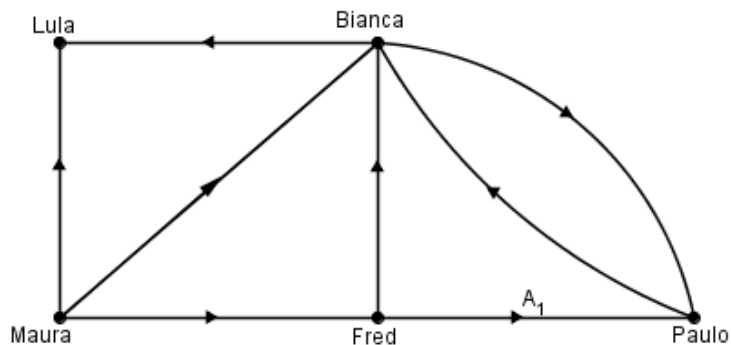


Figura 1.17: Grafo de influência.

exemplificado na Figura 1.18.

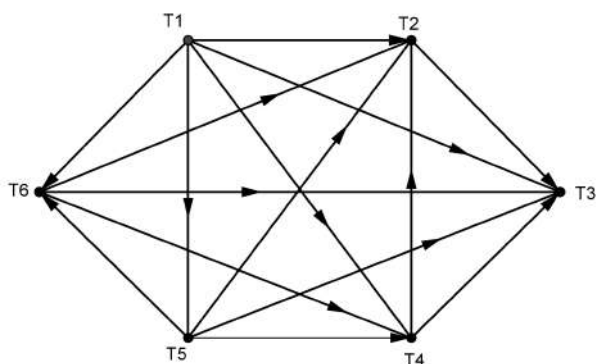


Figura 1.18: Grafo de Torneio.

1.5.5 Grafo da web

A World Wide Web pode ser modelada como um grafo orientado, no qual cada página é representada por um vértice e uma aresta começa na página a da web e termina na página b , se existir um link em a que direcione para b . Como novas páginas da web são criadas e outras removidas, em alguma parte dela, a quase todo segundo, este grafo muda continuamente e têm mais de três milhões de vértices e vinte bilhões de arestas. Muitas pessoas estão estudando as propriedades do grafo da web para entender melhor sua natureza.

1.5.6 PageRank

A internet é uma ferramenta poderosa para obtenção de informações. Digitando uma palavra em um buscador, milhares de resultados relacionados com a mesma, são obtidos instantaneamente.

Atualmente, o Google é a máquina de busca mais estudada e conhecida mundialmente. Quando se digita uma palavra-chave de algo que se deseja obter informações no google, obtém-se uma lista de páginas web que possuem assuntos correlacionados. Mas, para a lista gerada ser apresentada na ordem que interessa ao usuário são necessários cálculos de álgebra linear acompanhados de uma modelagem em grafos. Critérios usados no algoritmo do PageRank são empregados na ordenação dos sites quando se realiza uma busca.

O PageRank é um mecanismo de “votos”. Cada site é representado por um vértice de um grafo orientado e cada link em que uma página aponta para outra, é um "voto", representado por uma aresta (u, v) , onde u indica a página v . Quanto mais alto for o PageRank da página de onde sai o link, mais importante é o voto dado à página de destino. O Google possui milhares de páginas cadastradas cujo cálculo das importâncias se torna cada vez mais complexo. Será mostrado um exemplo simples de grafo e sua aplicação algébrica para ilustrar a importância dos sites de uma determinada rede da internet na figura a seguir.

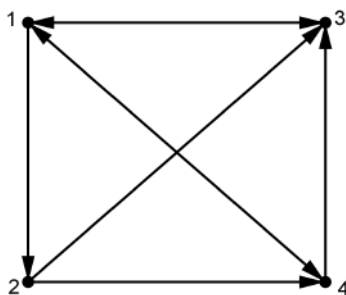


Figura 1.19: Grafo simples.

A rede é composta de quatro sites (vértices). Cada aresta indica que existe um link (u, v) , onde u indica a página v , e cada página possui pelo menos um link para outra página. Há vários fatores adicionais que influenciam na ordenação dos sites. Mas será dada ênfase no processo usado pelo Google.

Seja x_i o índice de importância do site i , $x_i \geq 0$ para qualquer página i . Observe que o site 1 possui link para os sites 2, 3 e 4. O índice de importância da página 1 deve ser dividido por 3. O índice de importância da página 2 possui link para as páginas 3 e 4 e deve ser dividido por 2. Portanto, a importância da página 4 é: $x_4 = x_1/3 + x_2/2$. A importância dos demais sites obedecem às equações análogas, chegando ao sistema:

$$\begin{cases} x_1 = x_3 + x_4/2 \\ x_2 = x_1/3 \\ x_3 = x_1/3 + x_2/2 + x_4/2 \\ x_4 = x_1/3 + x_2/2 \end{cases}$$

Como a resolução deste sistema é feita com recursos computacionais, pode-se encontrar

a solução por escalonamento, onde uma das soluções será $\{s(12, 4, 9, 6), s \in R\}$. Para qualquer s positivo, observa-se que $x_1 > x_3 > x_4 > x_2$. Assim, o site 1 tem importância maior que outros sites e o site 2 tem a menor importância, segundo a ordenação do Google.

O sistema linear relacionado ao grafo pode também ser escrito utilizando matrizes, conduzindo a um problema de autovalores e autovetores, isto é, $Av = v$, com

$$A = \begin{pmatrix} 0 & 0 & 1 & 1/2 \\ 1/3 & 0 & 0 & 0 \\ 1/3 & 1/2 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{pmatrix} \text{ e } v = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

A matriz $A = (a_{ij})$, chamada matriz dos links, é tal que $a_{ij} = 0$ se não houver link de j para i e $a_{ij} = 1/n_j$ se houver link de j para i , sendo n_j o número de links que partem de j .

Qualquer que seja a matriz A , o sistema $Av = v$ é um sistema homogêneo, tem pelo menos uma solução, que é a solução nula. Além disso, pode-se mostrar que, quando A é uma matriz de links em que cada página aponta para pelo menos uma outra página, ele tem uma solução não nula.

O sistema homogêneo citado tem infinitas soluções, mas o que importa é você encontrar uma solução não nula para poder informar a ordenação de importância. Observe a matriz A , a soma de cada coluna da matriz é igual a 1, para calcular a importância, pode-se impor a condição adicional: $x_1 + x_2 + \dots + x_n = 1$, onde no exemplo tem-se: $x_1 = 12/31$, $x_2 = 4/31$, $x_3 = 9/31$ e $x_4 = 6/31$.

Quando uma rede é fortemente conectada, isto é, quando se pode passar de um site arbitrário para outro qualquer apenas clicando nos links, o conjunto de soluções do sistema tem dimensão 1. Obtém-se, então, uma solução única impondo a restrição adicional de que a soma das importâncias seja igual a 1. Quando não é fortemente conectada, as soluções do sistema $Av = v$ não são todas proporcionais entre si, encontra-se problemas na ordenação das importâncias.

Segundo Almeida e Celeman [1], existe um truque que pode ser aplicado para qualquer página web admissível, bastante útil para o caso em que a Web não é fortemente conectada. Faz-se uma pequena modificação na matriz dos links A , substituindo-a por uma matriz M definida por $M = (1 - m)A + mS$, como $0 < m < 1$ e $S_{ij} = 1/n$, para todo i e j , sendo S uma matriz $n \times n$. O valor utilizado pelo Google é $m = 0,15$. Quanto menor for o valor de m , mais peso se dá à matriz A e menos peso se dá à matriz S . Se relacionarmos S a um grafo direcionado, esse representaria uma web onde todos os sites teriam links para todos os outros, inclusive para si mesmos, pois S seria uma matriz neutra que estaria fazendo uma média ponderada com a matriz A . O ponto é que o conjunto de soluções de $Mv = v$ tem dimensão 1 e, assim, é possível classificar as páginas. Devido à neutralidade de S , a

mudança não afetaria a ordenação intuitiva da importância dos sites.

1.5.7 Grafos de mapas rodoviários

Usando grafos para modelar mapas rodoviários, os vértices representam as intersecções e as arestas representam estradas. Arestas não orientadas representam estradas de mão dupla, e as arestas orientadas representam estradas de mão única. Arestas não orientadas múltiplas representam estradas de mão dupla múltiplas que conectam as mesmas duas intersecções. Arestas orientadas múltiplas representam estradas de mão única múltiplas que começam em uma intersecção e terminam em outra intersecção. Os grafos mistos são necessários para representar mapas rodoviários que incluem tanto estradas de mão única como de mão dupla.

1.6 Problemas clássicos de grafos

1.6.1 Grafos eulerianos e as sete pontes de Königsberg

O primeiro e mais famoso problema resolvido por Euler foi o das sete pontes da cidade Königsberg que cruzam o rio Pregel, estabelecendo ligações entre duas ilhas e entre as margens opostas do rio, conforme mostra a figura 1.20 a seguir. Surgiu a pergunta: Será possível fazer um passeio pela cidade, começando e terminando no mesmo lugar, cruzando cada ponte exatamente uma única vez?

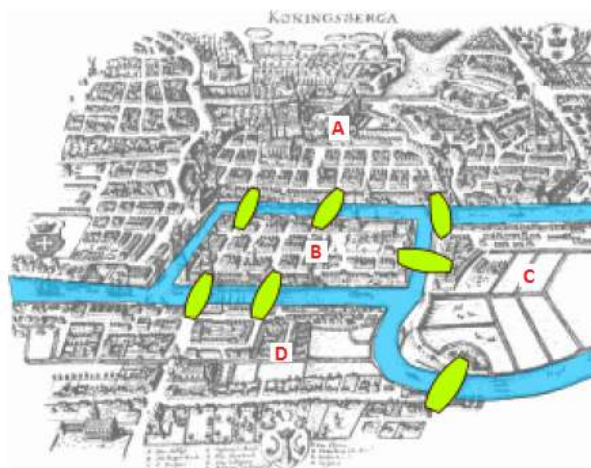


Figura 1.20: As sete pontes de Königsberg.

Algumas definições e exemplos serão necessárias para resolver o problema.

Definição 1.6.1. Um grafo G de m arestas é dito eulériano se existir uma trilha fechada de comprimento m em G .

Se o grafo não é eulériano, mas tem uma trilha aberta de m arestas, é dito *semieulériano*. O grafo 1 da Figura 1.21 é o desenho de um grafo eulériano, o grafo 2 é semieulériano e grafo 3 não é eulériano nem semieulériano.

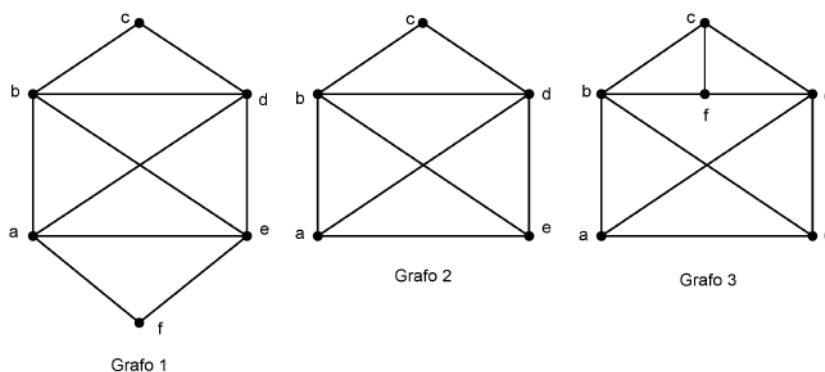


Figura 1.21: Grafos eulériano, semieulériano e não eulériano.

Lema 1.6.1. Se todo vértice de um grafo G (não necessariamente simples) tem grau maior ou igual a 2, então, G contém um ciclo.

Demonstração: Se G tem laços ou arestas múltiplas, não há o que provar, pois G já contém um ciclo. Considerando um grafo simples, partindo de um vértice inicial qualquer, inicia-se uma trilha. Quando chegamos a outro vértice, estamos visitando-o pela primeira vez; continuando o percurso, ao chegar a um vértice já visitado, produzimos um ciclo. Como o número de vértices é finito, o lema está provado.

Teorema 1.6.1. (Euler) Um grafo conexo (não necessariamente simples) G é eulériano se, e somente se, todos seus vértices tem grau par.

Demonstração: (\Rightarrow) Supondo que G tenha uma trilha fechada m . Cada vez que a trilha passa por um vértice, utiliza duas novas arestas, um para entrar outro para sair. Logo, o grau de cada vértice é obrigatoriamente par.

(\Leftarrow) Seja G com todos os vértices de grau par. Usaremos indução sobre o número de arestas m do grafo. Para $m=0$ é verdadeiro. Supondo que seja válido para todos os grafos com menos do que m arestas, sendo G conexo, todos os vértices tem grau maior do que 2, pois os graus são pares. Pelo lema anterior, G contém um ciclo. Dentre todas as trilhas fechadas de G escolhemos uma trilha T com o comprimento máximo. Se T tem comprimento m o teorema está provado. Caso contrário, consideramos o grafo H resultante da retirada das arestas de T . Como retiramos um número par de arestas de cada vértice de T , e todos os vértices tem grau par (pela hipótese), pelo menos uma das

componentes de H tem um vértice em comum com T e tem todos os vértices de grau par. Pela hipótese de indução, H tem uma trilha que passa por todos os vértices de H , e podemos formar uma trilha fechada maior, conectando T com a trilha em H . Mas isto contraria a maximalidade na escolha de T .

Voltemos ao problema. Euler o modelou em um grafo. Chamou as ilhas de vértices e as pontes de arestas. Verificou que partindo de um vértice, passando por todas as arestas e voltando para o mesmo vértice, precisaria de, no mínimo, duas arestas (pontes), uma para chegada e outra para saída. Ver Figura 1.22.

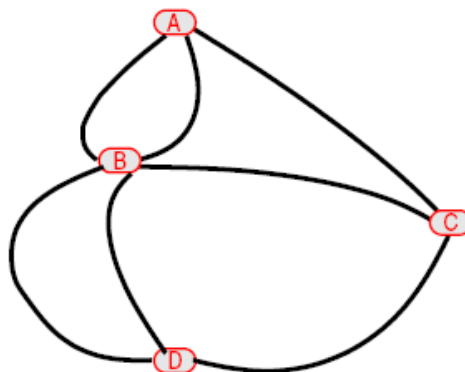


Figura 1.22: Grafo das sete pontes de Königsberg.

Concluiu, então, que para achar uma rota que começa e termina em um mesmo vértice, esse número total de chegadas e saídas deve ser um múltiplo de 2. Isto é, condição necessária para existir um ciclo. Portanto, o grafo que representa o problema não é euleriano. Por isto, não tem solução.

No problema das pontes, se fosse construída uma nova ponte seria possível fazer um passeio pela cidade, começando e terminando no mesmo lugar, cruzando cada ponte exatamente uma única vez?

Precisamos de mais informações:

Corolário 2. *Um grafo conexo (não necessariamente simples) G é semieuleriano se, e somente se, dois vértices têm grau ímpar.*

Demonstração: (\Rightarrow) Suponhamos que G possua um caminho semieuleriano E começando num vértice v e terminando num vértice w . Como $v \neq w$, v e w têm ambos grau ímpar. Pois cada vez que um dos demais vértices aparecem em E tem duas arestas incidentes. Como cada aresta ocorre uma vez em E , o grau desses vértices é par.

(\Leftarrow) Suponhamos que G seja conexo e possui dois vértices v (inicial) e w (final) de grau ímpar, pois em outra situação não teríamos um grafo. Consideremos o grafo H que

se obtém de G por junção de uma nova aresta ligando v a w . A este novo grafo podemos aplicar o Teorema de Euler e concluir que admite um caminho euleriano. Apagando deste caminho a aresta previamente adicionada a G obtemos um caminho semi-euleriano ligando v a w , com desejado.

Portanto, este problema também não tem solução, mas teríamos um grafo semieuleriano. Seria possível sair de um ponto, caminhar uma única vez por todas as pontes, mas não poderia voltar ao ponto de origem.

Observação: Sendo G um grafo semi-euleriano, se adicionarmos uma nova aresta, incidente aos dois vértices de grau ímpar, obtém-se um grafo euleriano. Esse processo chama-se *eulerizar um grafo*.

1.6.2 Grafos e ciclos hamiltonianos

Semelhante ao problema das pontes, o ciclo hamiltoniano consiste em verificar se, dado um grafo, é possível construir uma trilha que passe por todos os nós uma única vez. Uma trilha desse tipo teria que ser necessariamente um ciclo. Tal ciclo é chamado de *ciclo hamiltoniano*, devido ao matemático irlandês do século XIX, Sir Willian R. Hamilton.

Definição 1.6.2. *Um caminho que contém todos os vértices de um grafo G é dito caminho hamiltoniano.*

Definição 1.6.3. *Um ciclo que contém todos os vértices de um grafo G é dito ciclo hamiltoniano.*

Se um grafo G contém um ciclo hamiltoniano, dizemos que G é um grafo hamiltoniano.

Um exemplo de grafo hamiltoniano é o grafo que foi base de um jogo, "O Dodecaedro do Viajante", concebido por Hamilton e comercializado sem muito sucesso. O nome dodecaedro decorre do fato de que o grafo provém do poliedro sólido com doze faces pentagonais, em que cada um dos 20 vértices foram nomeados como cidades importantes. O objetivo do jogo era, utilizando as 30 arestas do dodecaedro, passar por cada uma das cidades apenas uma vez, começando e terminando na mesma cidade.

Para formar o grafo, imagine que o sólido é elástico e uma das faces é esticada pelos seus vértices até que o resto do sólido possa ser projetado dentro dela. Não são conhecidas condições necessárias e suficientes para que um grafo genérico contenha um ciclo hamiltoniano, nem tampouco métodos para construir tal ciclo, caso exista.

Um problema de otimização correspondente, conhecido como o Problema do caixeiro viajante, é o de achar um ciclo de custo mínimo (sair de uma cidade, passar todas as outras uma única vez e voltar a cidade inicial, com custo mínimo), onde o custo de um ciclo é a soma dos custos dos arcos (pesos atribuídos as arestas) pertencentes ao ciclo. Possivelmente é um dos problemas mais famosos na teoria dos grafos. Ainda não são

conhecidos algoritmos eficientes para resolvê-lo e conjecturam-se que tais algoritmos de fato não existam.

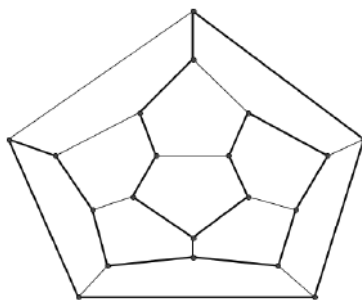


Figura 1.23: Dodecaedro planarizado.

Este tipo de problema é resolvido através de tentativas e erros, da seguinte maneira: comece por algum vértice e tente qualquer caminho escolhendo suas arestas. Se o caminho resultante tiver um vértice repetido, ele não é um ciclo, descarte e tente um caminho diferente. Se o caminho puder ser completado como um ciclo, verifique se visitou todos os vértices; se não, descarte e tente um novo caminho. Continue o procedimento até que todos os caminhos possíveis tenham sido testados ou até que se tenha obtido um ciclo hamiltoniano. Embora semelhante ao problema de ciclo euleriano, achar um ciclo hamiltoniano é muito mais complexo.

Algumas propriedades dos ciclos hamiltonianos podem ajudar a construí-los.

1. Se um vértice $v \in V$ tem grau 2, então, as duas arestas incidentes a v fazem parte de qualquer ciclo hamiltoniano.
2. Na construção de um ciclo hamiltoniano, nenhum ciclo se pode formar antes de percorrer todos os vértices.
3. Na construção de um ciclo hamiltoniano não pode conter arestas múltiplas nem laços.
4. Grafos simples completos são hamiltonianos.

Definição 1.6.4. *Um grafo G simples chama-se grafo maximal não hamiltoniano se não é um grafo hamiltoniano mas a adição de qualquer aresta que ligue dois vértices não adjacentes forma um grafo hamiltoniano.*

O problema do caixeiro-viajante é muito estudado em pesquisa operacional. Sua formulação é simples: Dado um grafo valorado G , determinar o menor ciclo hamiltoniano de G . Tal problema será exemplificado, usando um grafo k_5 com sua representação matricial, que será visto no próximo capítulo.

	a	b	c	d	e
a	0	200	170	250	250
b	200	0	320	480	480
c	170	320	0	180	180
d	250	480	180	0	90
e	250	480	180	90	0

Uma solução óbvia seria examinar todas as permutações entre os vértices que teríamos $4! = 24$ permutações (circulares) que será fácil, pois contém poucos vértices, se resolvido por computador. Mas geralmente problemas dessa natureza envolvem uma quantidade muito grande de vértices que até mesmo um computador levaria tempo demais para resolvê-lo. Observe que na matriz acima a, c, b, e, d, a formam um ciclo com o valor 1.310. O algoritmo parte de um ponto e procura sempre o ponto da vez, procura-se um ciclo que não precisa ser sequencial. Encontramos um ciclo, mas que nem sempre é a solução ótima.

Capítulo 2

Representação computacional de grafos

Neste Capítulo mostraremos como as informações sobre grafos podem ser representadas no computador.

Quando se trabalha com grafos, é útil ser capaz de escolher uma representação conveniente. Este capítulo mostrará como representar grafos de maneiras diferentes. Algumas vezes dois grafos têm a mesma forma, no sentido que existe uma correspondência biunívoca entre seus conjuntos de vértices que preserva as arestas. Em tal caso, esses grafos são isomorfos, e determinar se dois grafos são isomorfos é muito importante na teoria dos grafos.

Uma das formas de representar um grafo sem arestas múltiplas é listar todas as arestas desse grafo. Outra forma é por meio de listas de adjacência, na qual especificam os vértices que são adjacentes a cada vértice do grafo. Um grafo $G(V, E)$ com $|V| = n$ pode ser representado por meio de matrizes ou listas.

2.1 Matriz de adjacência

Seja $G(V, E)$ um grafo simples em que $|V| = n$ e supondo que os vértices de G sejam listados por v_1, v_2, \dots, v_n , a *matriz de adjacência* A é uma matriz de ordem n , com (i, j) igual a 1 quando v_i e v_j forem adjacentes e igual a 0 quando (i, j) não forem adjacentes. A matriz de adjacência é $A = [a_{i,j}]$, em que:

$$a_{ij} = \begin{cases} 1, & \text{se } (i, j) \in E(G) \\ 0, & \text{se } (i, j) \notin E(G) \end{cases}$$

Uma matriz de adjacência de um grafo é baseada na ordem escolhida para os seus vértices. Por essa razão, há $n!$ matrizes de adjacência diferentes para um grafo com n vértices, pois existem $n!$ ordenações diferentes de n vértices.

A matriz de adjacência de um grafo simples é simétrica, isto é, $a_{i,j} = a_{j,i}$, pois ambos os

elementos são 1 quando v_i e v_j são adjacentes e ambos os elementos são 0 caso contrário, além disso, cada elemento $a_{i,i}$, $i = 1, 2, \dots, n$ também é 0, pois não possui laços. Na Figura 2.1 temos um exemplo de grafo simples e sua representação por matriz de adjacência obedecendo a ordem alfabética dos vértices.



Figura 2.1: Grafo simples e Matriz de adjacência.

Matriz de adjacência também pode representar grafos não orientados com laços e arestas múltiplas. Um laço em um vértice a_i é representado por 1 na posição (i, i) . Na posição (i, j) , será representado por 2 se duas arestas ligam os vértices a_i e a_j desta matriz, ou melhor, será igual a n , se n arestas ligam os vértices a_i e a_j desta matriz. Todos os grafos não orientados, incluindo multigrafos e pseudografos, possuem matrizes de adjacência simétricas. A Figura 2.2 mostra um pseudografo e sua representação por matriz de adjacência (com vértices ordenados).

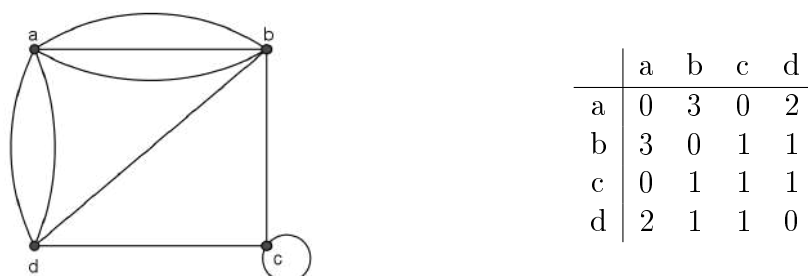


Figura 2.2: Pseudografo e Matriz de adjacência.

Para representar grafos orientados $G(V, E)$, atribui-se 1 em sua posição (i, j) se existir uma aresta de v_i para v_j , em que v_1, v_2, \dots, v_n é uma lista dos vértices. Esta matriz não necessariamente tem que ser simétrica, pois pode não haver uma aresta de v_j para v_i . As matrizes de adjacência também podem representar multigrafos orientados, sempre levando em consideração a quantidade de arestas entre dois vértices e o sentido destas arestas, como mostra a Figura 2.3. O espaço reservado para o armazenamento das informações nessa matriz é da ordem $O(|V|^2)$. Seu uso é indicado para tratar de grafos densos, em que o número de arestas é próximo de $|V|^2$. Torna-se desvantajosa para grafos esparsos em que o número de arestas é bem menor que $|V|^2$.



Figura 2.3: Grafo orientado e Matriz de adjacência.

2.2 Matriz de incidência

Outra forma de representar grafos é usar *matriz de incidência*.

Seja $G = (V, E)$ de n vértices e m arestas, um grafo não orientado e suponha que v_1, v_2, \dots, v_n sejam os vértices e e_1, e_2, \dots, e_m sejam as arestas de G , então a *matriz de incidência* com relação a esta ordem de V e E é a matriz $n \times m$, $M = [m_{ij}]$, em que:

$$m_{ij} = \begin{cases} 1, & \text{quando } e_j \text{ for incidente a } v_i, \\ 0, & \text{caso contrário.} \end{cases}$$

A Figura 2.4 mostra um grafo simples representado por uma matriz de incidência.

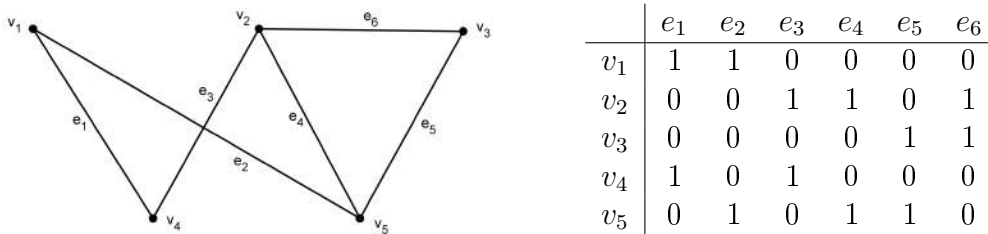


Figura 2.4: Grafo simples e Matriz de incidência.

Matrizes de incidência também podem ser usadas para representar grafos com arestas múltiplas e laços. As arestas múltiplas são representadas na matriz de incidência usando colunas como elementos idênticos, pois estas arestas são incidentes ao mesmo par de vértices. Os laços são representados usando uma coluna com exatamente um elemento igual a 1, correspondendo ao vértice que é incidente deste laço.

A Figura 2.5 mostra um pseudografo e sua representação por matriz de incidência.

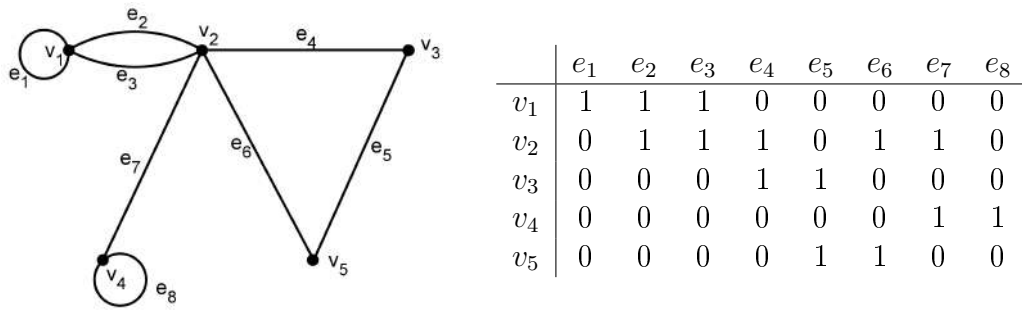


Figura 2.5: Pseudografo e Matriz de incidência.

2.3 Listas de adjacência

Uma lista de adjacências para um grafo $G(V, E)$ consiste em um vetor Adj com $n = |V|$ entradas, uma para cada vértice do grafo. Cada entrada $Adj[v]$ possui uma lista encadeada de vértices a v em G . A Figura 2.6 mostra um exemplo de grafo não direcionado e sua representação por lista de adjacências.

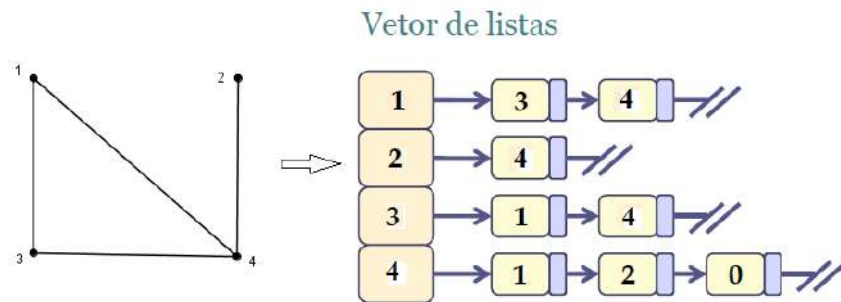


Figura 2.6: Lista de adjacência.

A lista de adjacência consiste de n elementos (vértices) contendo $2m$ elementos, onde m é o número de arestas do grafo. O espaço utilizado pela lista é da ordem de $O(n + m)$.

Capítulo 3

Algoritmos de grafos

Neste Capítulo veremos alguns procedimentos de busca para resolver problemas com o auxílio do computador e serão citados alguns algoritmos.

Os grafos são estruturas de dados muito usadas na ciência da computação e seus algoritmos são fundamentais nesta área. Muitos problemas são definidos em termos de grafos e resolvidos com recursos computacionais. Neste capítulo serão mostradas algumas dessas aplicações. No capítulo 2 foi mostrado como um grafo pode ser representado na máquina e agora veremos como resolver problemas utilizando algoritmos.

3.1 Algoritmo de busca

Os algoritmos de busca são os procedimentos mais utilizados na teoria de grafos, com objetivo de mostrar uma sistemática de como caminhar pelos vértices e arestas de um grafo. Para tanto, existem dois métodos de busca: DFS - Depth First Search (Busca em Profundidade) e BFS - Breadth First Search (Busca em Largura).

3.1.1 Algoritmo de busca em profundidade - DFS

A ideia básica da *DFS* é buscar mais “fundo” no grafo quando possível. Este procedimento atende a um critério de explorar um vértice marcado (ou visitado), dentre os vários marcados e incidentes a alguma aresta ainda não explorada, escolher aquele vértice mais recentemente alcançado na busca. A lista de espera usada para busca em profundidade será uma *pilha*, onde o último que é visitado é o primeiro a ser finalizado. Um exemplo de pilha é colocar vários livros em uma determinada ordem um sobre outro (formando uma pilha de livros), para ter acesso a todos os livros, o processo deverá começar pelo último que entrou na pilha e o último a ser acessado será o primeiro que entrou na pilha.

A ordem de visitação das arestas incidentes a um vértice depende da ordem em que seus vizinhos foram inseridos na lista de adjacências, que no caso é arbitrária.

Dado um vértice v de um grafo, encontrar o conjunto de todos os vértices adjacentes ligados a v , é fazendo uma busca ao vértice mais profundo no grafo sempre que possível. As arestas são exploradas a partir do vértice v , recentemente descoberto, que ainda possui arestas não exploradas partindo dele. Quando todas as arestas adjacentes a v tiverem sido exploradas, a busca desloca-se para trás a fim de explorar vértices que saem do vértice do qual v foi descoberto. O processo continua até que sejam descobertos todos os vértices que são alcançáveis a partir do vértice inicial.

O algoritmo a seguir descreve uma busca em profundidade (DFS):

Algoritmo 3.1: Busca em profundidade

```

1 DFS(GRAFO(V, E))
2 PARA todo vértice i do GRAFO FAÇA
3   cor[i] ← BRANCO
4   pai[i] ← nulo
5 FIM_PARA
6 Tempo ← 0
7 PARA todo vértice i do GRAFO FAÇA
8   Se cor [i] = BRANCO ENTÃO
9     DFS-VISITA(i)
10  FIM-SE
11 FIM_PARA

```

Comentário: Na linha 2, inicializam-se as variáveis associadas a cada vértice i ; em 3 e 4, marca-se i como não descoberto (BRANCO) e seu ancestral (pai) como inexistente; já na linha 6, inicializa-se a variável tempo e examina-se cada vértice i do GRAFO; por fim, de 7 a 11, temos um laço que verifica se todos os vértices do grafo foram visitados, caso contrário visita.

Análise: Nas linhas de 2 a 5 o custo é $O(|V|)$. Já a linha 6 tem custo $O(1)$. As linhas de 7 a 11 de DFS gastam $O(|V| + |E|)$, pois conta o tempo de DFS-VISITA que só é chamado uma única vez para cada vértice BRANCO. DFS-VISITA executa em $O(|E|)$. Logo, o custo total do algoritmo é $O(|V| + |E|)$.

Algoritmo 3.2: DFS-VISITA

```

1 DFS-VISITA(u)
2 cor[u] ← CINZA
3 tempo ← tempo + 1
4 inicio[u] ← tempo
5 PARA todo vértice i de Adj(u) FAÇA
6   Se cor [i] = BRANCO ENTÃO
7     pai[i] ← u
8     DFS-VISITA(i)
9   FIM-SE
10 FIM_PARA

```

```
11 cor[u] ← PRETO
12 tempo ← tempo + 1
13 fim[u] ← tempo
```

A seguir faremos uma aplicação da DFS no grafo da Figura 3.1.

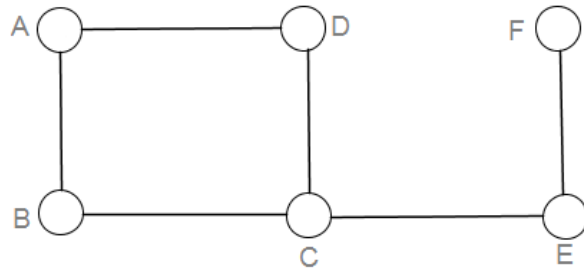


Figura 3.1: DFS.

Começamos a busca a partir do vértice A , marcando-o como visitado (cinza), e denominando-o de raiz da DFS. Têm-se dois caminhos a seguir: (A, B) e (A, D) , assim, escolhe-se (A, B) e marca-se B de cinza (visitado), conforme mostra a Figura 3.2.

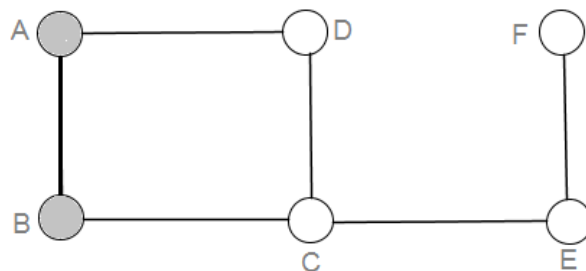


Figura 3.2: DFS.

A partir de B , têm-se duas escolhas: (B, A) e (B, C) . Como o vértice A já foi visitado, escolhe-se (B, C) e marca-se C de cinza (visitado), conforme mostra a Figura 3.3.

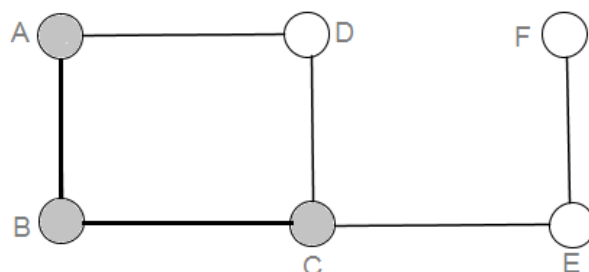


Figura 3.3: DFS.

A partir de C , têm-se três escolhas: (C, B) , (C, D) e (C, E) . O vértice B já foi visitado. Pode-se escolher (C, D) ou (C, E) . Escolhendo (C, D) , marca-se D de cinza (visitado), como mostra a Figura 3.4.

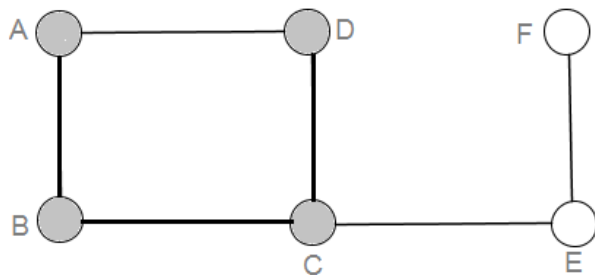


Figura 3.4: DFS.

A partir de D , têm-se: (D, C) e (D, A) . Os vértices D e A já foram visitados, logo, marca-se D de preto (finalizado). Nota-se que não se tem outro vértice a explorar, desta forma, volta-se para o pai de D , que é C , como mostra a Figura 3.5.

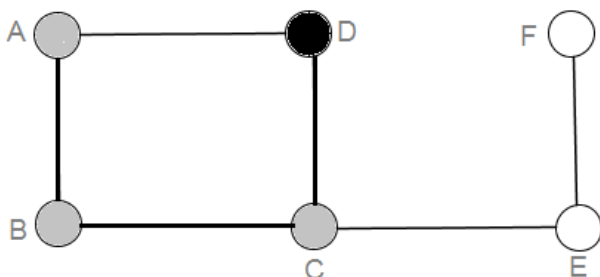


Figura 3.5: DFS.

A partir de C , têm-se: (C, B) , (C, D) e (C, E) . Os vértices B e D já foram visitados. Então escolhe-se (C, E) e marca-se E de cinza (visitado), como mostra a Figura 3.6.

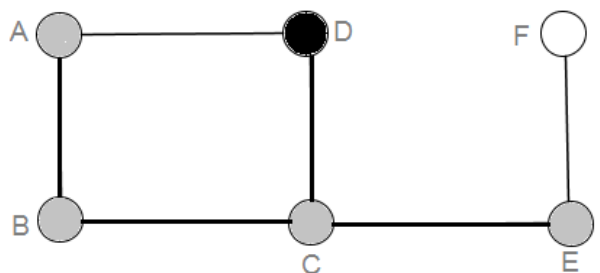


Figura 3.6: DFS.

A partir de E , têm-se: (E, C) e (E, F) . O vértice C já foi visitado. Assim, escolhe-se (E, F) e marca-se F de cinza (visitado), como mostra a Figura 3.7.

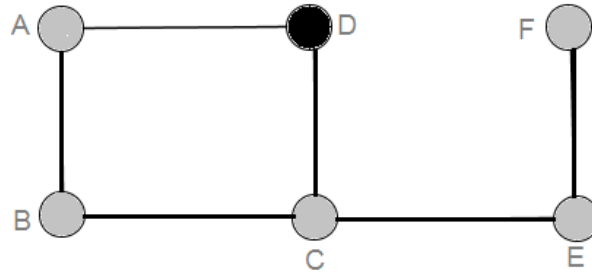


Figura 3.7: DFS.

A partir de F , não tem outra aresta a ser explorada, então, marca-se F de preto (finalizado), escreve-se pai de F é o vértice E e retorna-se ao vértice E , conforme mostra a Figura 3.8.

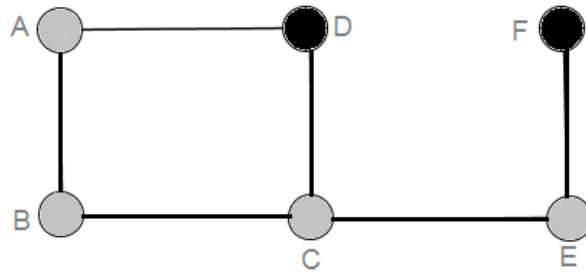


Figura 3.8: DFS.

A partir de E , não tem outra aresta a ser explorada, então marca-se E de preto (finalizado), escreve-se que o pai de E é o vértice C e retorna-se ao vértice C , como mostra a Figura 3.9.

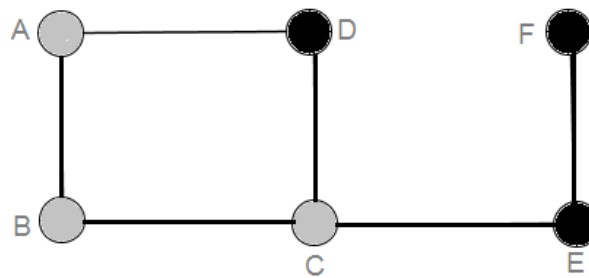


Figura 3.9: DFS.

A partir de C , não há outra aresta a ser explorada, então, marca-se C de preto (finalizado), escreve-se que o pai de C é o vértice B e retorna-se ao vértice B , conforme mostra a Figura 3.10.

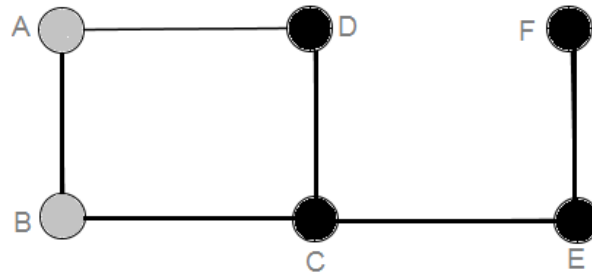


Figura 3.10: DFS.

A partir de B , não tem outra aresta a ser explorada, então, marca-se B de preto (finalizado), escreve-se que o pai de B é o vértice A e retorna-se ao vértice A , conforme mostra a Figura 3.11.

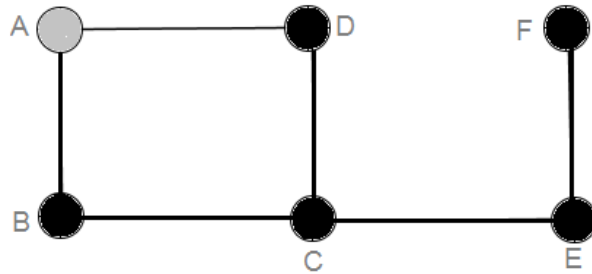


Figura 3.11: DFS.

A partir de A , não tem outra aresta a ser explorada, então, marca-se A de preto (finalizado), escreve-se o pai de A como nulo e fim do algoritmo, como mostra a Figura 3.12.

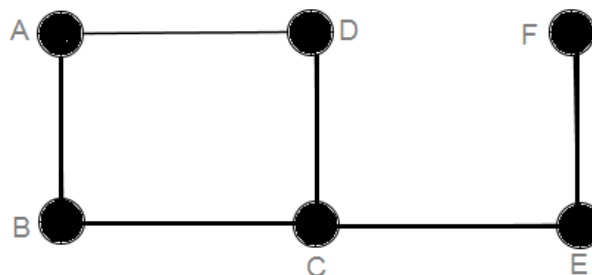


Figura 3.12: DFS.

A busca em profundidade procura acessar todos os vértices em um grafo $G(V, E)$, onde o número de vértices $|V| = V$ e o número de arestas $|E| = E$ do grafo. Para acessar todos os possíveis vértices é varrida a lista de arestas de cada vértice com o tempo de $O(|V| + |E|)$.

3.1.2 Algoritmo de busca em largura

Na busca em largura (BFS) os vértices do grafo são visitados nível a nível.

Segundo Cormen [4], dado um grafo $G(V, E)$ e um vértice de origem v , a busca em largura explora as arestas de G até todos os vértices alcançáveis a partir de v . Além disso, o algoritmo calcula a menor distância em número de arestas de v até todos os vértices acessíveis a ele. A busca em largura recebe esse nome porque descobre todos os vértices que estão a uma distância k de v , antes de descobrir os vértices que se encontram a uma distância $k + 1$. Neste procedimento, usa-se uma estrutura de *fila* na qual as novas chegadas são incluídas ao fim e as saídas tem lugar na frente. Um exemplo é a fila de um banco - um novo cliente entra no fim da fila e os clientes da frente saem dela à medida que um caixa fica livre. A seguir descreveremos um algoritmo para o método BFS.

Algoritmo 3.3: Busca em largura

```
1 BFS (GRAFO(V, E), v)
2 PARA todo vértice i do GRAFO FAÇA
3   cor[i] ← BRANCO
4   d_arestas[i] ← ∞
5   pai[i] ← nulo
6 FIM_PARA
7 d_arestas[v] ← 0
8 cor[v] ← CINZA
9 FILA ← ∅
10 INSERE(FILA, v)
11 ENQUANTO FILA ≠ ∅ FAÇA
12   u ← REMOVE(FILA)
13   PARA todo vértice i de Adj(u) FAÇA
14     Se cor[i] = BRANCO ENTÃO
15       cor[i] ← CINZA
16       d_arestas[i] ← d_arestas[u] + 1
17       pai[i] ← u
18       INSERE(FILA, i)
19   FIM-SE
20 FIM_PARA
21 cor[u] ← PRETO
22 FIM_ENQUANTO
```

Comentário do algoritmo:

A busca em largura inicializa todos os vértices, marcando-os como não visitados. A distância de todos os outros vértices à origem é “infinita”, o que indica que ainda não foi calculada ou que ainda não é possível alcançá-la a partir da origem. Essa inicialização gasta o tempo $O(|V|)$. A medida que os vértices vão sendo descobertos, são colocados em uma fila. As operações de inserção e remoção na fila gastam tempo constante, $O(1)$, e como todos os vértices são analisados, o tempo total gasto é $O(|V|)$. Como cada vértice

só é colocado na fila uma vez, sua lista de vizinhos é analisada apenas uma vez. Todas as listas de todos os vértices são analisadas, gastando o tempo proporcional a $O(|E|)$. Logo, o tempo de execução da busca em largura é $O(|V| + |E|)$.

Considerando o grafo da Figura 3.13 e sua representação por lista de adjacências, mostra-se que cada aresta tem valor igual a 1. Desta forma, será explorado brevemente como este procedimento acontece.

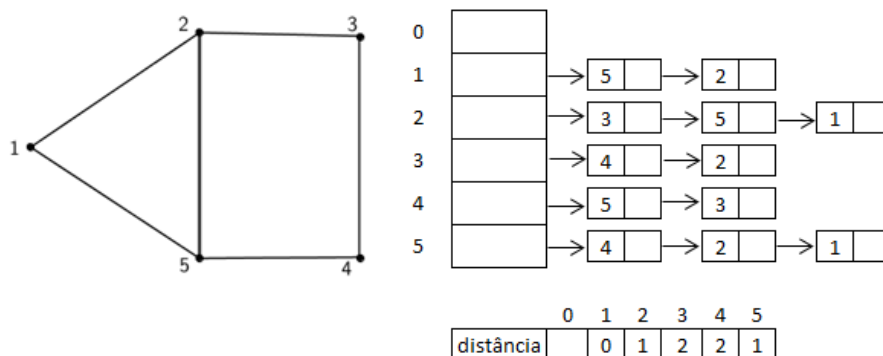


Figura 3.13: Lista de adjacência.

Inicialmente, marca-se o vértice 1, vértice de origem, colocando-o na fila com distância $d[1] = 0$; em seguida, marcam-se os vértices adjacentes de 1, o vértice 5 com $d[5] = 1$ e o vértice 2 com $d[2] = 1$. Então, continua-se a busca dos adjacentes dos vértices 5 e 2 que terão $d[4] = d[5] + 1 = 1 + 1 = 2$ até marcar todos os adjacentes dos vértices 5 e 2 depois todos os adjacentes do vértice 2, pois ambos têm $d[u] = 2$. Este processo se repete até o último vértice do grafo a ser marcado ou finalizado. Também informa o vértice de origem e a distância desse vértice para qualquer vértice do grafo. No Capítulo 6, veremos aplicações de busca em largura usando o algoritmo de Dijkstra.

Análise de custo do algoritmo BFS: Nas linhas de 2 a 6, o tempo gasto é $O(|V|)$; de 7 a 10 o tempo é $O(1)$; e nas linhas de 11 a 22 é $O(|V| + |E|)$; logo, o custo total é de $O(|V| + |E|)$.

3.2 Ordenação topológica

Os grafos direcionados são usados para indicar precedência entre eventos.

A ordenação topológica de um grafo direcionado acíclico pode ser vista como uma ordenação de seus vértices ao longo de uma linha horizontal, de tal forma que todas as arestas estão direcionadas da esquerda para direita.

A busca em profundidade pode ser usada para executar ordenações topológicas de grafos acíclicos orientados. Este processo se desenvolve da seguinte maneira: recebe um

grafo orientado acíclico e retorna uma lista ordenada de vértices, de modo que, se houver uma aresta (u, v) no grafo, então v aparecerá antes de u na lista.

Modela-se um problema, montando uma ordem parcial no conjunto de tarefas, de modo que, se a e b são tarefas e $a < b$, então, b não pode começar até que a tenha sido completada.

Definição 3.2.1. *Uma ordenação topológica de um grafo acíclico orientado $G = (V, E)$ é uma ordenação linear de todos os vértices, tal que se G tiver a aresta (u, v) , então u aparece antes de v na ordenação.*

Se o grafo não é acíclico, então não é possível uma ordenação linear. Grafos acíclicos orientados são usados em muitas aplicações para indicar precedências entre eventos.

Exemplo: Uma pessoa vai se vestir para ir a uma formatura. Ela deve vestir certas peças antes que outras, por exemplo meias antes dos sapatos. Outros itens podem ser colocados em qualquer ordem. Uma aresta (u, v) indica que a peça de roupa u deve ser vestida antes da peça v . Uma ordenação topológica desse grafo fornece uma ordem para o processo de se vestir. Algumas tarefas devem ser realizadas apenas quando outras tiverem sido terminadas.

Pode-se executar uma ordenação topológica no tempo $O(|V|+|E|)$, pois é uma aplicação de busca em profundidade.

3.3 Caminho mínimo

O problema do caixeiro-viajante é um problema de caminho ponderado mínimo com fortes restrições sobre a natureza do caminho, de modo que, este caminho pode vir a não existir, pois percorre todos os vértices e tem que voltar ao vértice de origem. No problema de caminho mínimo, não colocamos restrições, além do peso na natureza do caminho, pois o grafo deverá ser conexo (sempre existe um caminho) e por esta razão temos algoritmos eficientes para resolver problemas de caminho mínimo.

O algoritmo de Dijkstra encontra o caminho mínimo em um grafo conexo, com arestas de pesos não negativos. É um algoritmo guloso ¹. O quadro a seguir apresenta este algoritmo na forma de pseudocódigo. A entrada é uma matriz de adjacência de um grafo simples conexo com pesos positivos, com o objetivo de encontrar o caminho mínimo (menor custo) do vértice raiz para aos demais vértices do grafo.

Algoritmo 3.4: Dijkstra

```
1 Dijkstra (GRAFO(V, E), w, s)
2 PARA todo vértice i do GRAFO FAÇA
3   d[i] ← ∞
```

¹É o algoritmo que sempre realiza os passos que parecem melhores, baseado apenas em seu conhecimento imediato.

```

4 FIM_PARA
5 d[s] ← 0
6 S ← ∅
7 Q ← V
8 ENQUANTO Q ≠ ∅ FAÇA
9   u ← ExtraiMin(Q)
10  S ← S ∪ {u}
11  PARA todo vértice i de Adj(u) FAÇA
12    Se d[i] > d[u] + w(u,i) ENTÃO
13      d[i] ← d[u] + w(u,i)
14      pai[i] ← u
15  FIM-SE
16 FIM_PARA
17 FIM_ENQUANTO

```

Este algoritmo funciona, também, para grafos desconexos, orientados e não orientados.

A eficiência do algoritmo: como as distâncias são atualizadas do início ao fim da execução, o laço PARA requer $O(n)$ operações, pois verifica todos os vértices. Cada execução do laço ENQUANTO toma $O(n)$ operações. No pior caso, o último vértice a ser incluído será executado $n - 1$ vezes. Portanto, o número total de operações envolvidas no laço será $O(n(n - 1)) = O(n^2)$ operações em seu pior caso.

No último capítulo serão resolvidos problemas aplicando este algoritmo.

Um outro algoritmo para obtenção de caminhos mínimos, a partir de um vértice de origem para outro vértice qualquer, é o *algoritmo de Bellman-Ford*, que não tem restrições quanto às arestas, pois podem ser de pesos negativos.

3.4 Árvores espalhadas mínimas

Seja $G = (V, E)$ um grafo não orientado com um valor real, chamado custo, associado a cada aresta de G . Uma *árvore espalhada* é uma árvore que conecta todos os vértices $V(G)$. Formalmente, é uma árvore da forma $K = (V(G), E')$ onde $E' \subseteq E$. O custo de uma árvore espalhada é simplesmente a soma dos custos das suas arestas. O problema é a determinação de uma árvore espalhada de custo mínimo para G , ou simplesmente, uma árvore espalhada mínima para G . Portanto, as árvores espalhadas são soluções viáveis deste problema e o custo da árvore espalhada é o valor da função, objetivo que queremos minimizar. Uma árvore espalhada mínima também pode ser chamada de árvore minimal geradora ou simplesmente árvore geradora mínima.

Os algoritmos de Kruskal e Prim nos permitem achar uma árvore minimal geradora T de um grafo conexo ponderado G , onde G tem n vértices. Cada um deles pode ser criado com facilidade para executar no tempo $O(|E|\log|V|)$. Suas elaborações partem de um algoritmo genérico e cada um deles utiliza uma regra específica. Tanto Kruskal

como Prim são algoritmos gulosos. Observe, de forma simples, os passos para se aplicar o algoritmo genérico que determina uma árvore geradoa mínima:

- **Algoritmo:** A entrada é um grafo conexo ponderado G com n vértices.

Passo 1. Ordene as arestas de G em ordem decrescente de peso.

Passo 2. Sequencialmente, delete cada aresta que não desconecta o grafo até que restem $n - 1$ arestas.

Passo 3. Saia.

O custo de uma árvore minimal é único, mas a árvore não é. Árvores geradoras minimais distintas podem ocorrer quando duas ou mais arestas tem o mesmo peso.

Exemplo: Encontrar uma árvore minimal geradora do grafo ponderado da Figura 3.14.

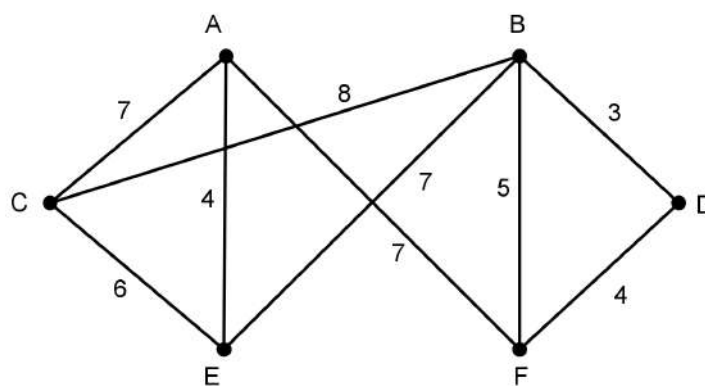


Figura 3.14: Grafo ponderado.

Aplicando o algoritmo genérico: primeiramente, colocam-se as arestas em ordem decrescente de peso; em seguida, deletam-se as arestas sem desconectar o grafo até que restem cinco arestas, pois o grafo tem um total de seis vértices. Resultam nos seguintes dados:

Arestas	BC	AF	AC	BE	CE	BF	AE	DF	BD
Peso	8	7	7	7	6	5	4	4	3
Deletar	Sim	Sim	Sim	Não	Não	Sim	Não	Não	Não

Logo, a árvore minimal geradora do grafo obtida contém as arestas BE , CE , AE , DF , BD . A árvore minimal geradora tem peso 24 mostrado na Figura 3.15

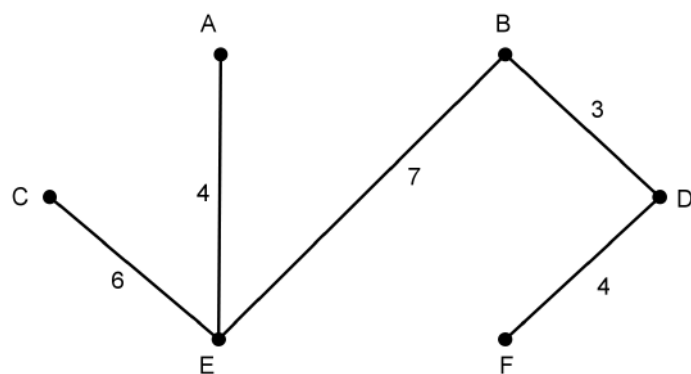


Figura 3.15: Árvore minimal geradora.

Também poderia ser usado o algoritmo de Kruskal, cuja aplicação será mostrada no Capítulo 5. O algoritmo possui a seguinte forma:

Algoritmo 3.5: AGCM-KRUSKAL

```

1 AGCM-KRUSKAL( $G=(V, E), w$ )
2  $A \leftarrow \emptyset$ 
3 PARA todo vértice  $i$  do GRAFO FAÇA
4     Construa_Conjunto( $i$ )
5 FIM_PARA
6 Ordene as arestas de  $E$  por ordem crescente do peso  $w$ 
7 PARA cada aresta  $(u, v) \in E$ , em ordem crescente FAÇA
8     SE Conjunto( $u$ )  $\neq$  Conjunto( $v$ ) ENTÃO
9          $A \leftarrow A \cup \{(u, v)\}$ 
10        Une_arvores( $u, v$ )
11    FIM-SE
12 FIM_PARA
13 RETORNE  $A$ 

```

Funcionamento: primeiro ordenam-se, em ordem crescente, as arestas de G . Considera-se cada vértice de G como pertencendo a um conjunto (árvore) em G . O conjunto de todas as árvores é uma floresta. A partir da primeira aresta ordenada, pode ser verificado se os vértices desta aresta pertencem mesma árvore. Se não pertencem à mesma árvore, faz-se a união dos conjuntos, os dois numa mesma árvore, caso contrário, passe para o próximo vértice. Repete-se o processo até que a floresta fique com uma única árvore.

O tempo de execução do algoritmo de Kruskal para um grafo $G = (V, E)$ depende da implementação de como priorizar as arestas.

O algoritmo gasta o tempo $|V|$ para criar as $|V|$ árvores iniciais. Então ordena as arestas, gastando $O(|E|\log|E|)$ para tal. Cada aresta é verificada uma vez por união das subárvores, gastando $O(|E|)$. Logo, o tempo total de execução do algoritmo é $O(|E|\log|E|)$.

Observação: Esses algoritmos são executados de forma simples quando o grafo é

relativamente pequeno. Em grafos com grandes quantidades de vértices e arestas será necessário o uso de algoritmos de buscas (profundidade ou largura).

Capítulo 4

Planaridade

Neste Capítulo veremos definições e aplicações de planaridade e coloração.

4.1 Definições e aplicações

Analise o problema: É possível ligar três casas a três serviços públicos, como mostrado na Figura 4.1, de modo que nenhuma das conexões se cruze? Este problema pode ser modelado por um grafo bipartido completo, $K_{3,3}$. Nesta seção, estudaremos a possibilidade de um grafo ser desenhado no plano sem que as arestas se cruzem. Também será respondido o problema das casas e dos serviços.

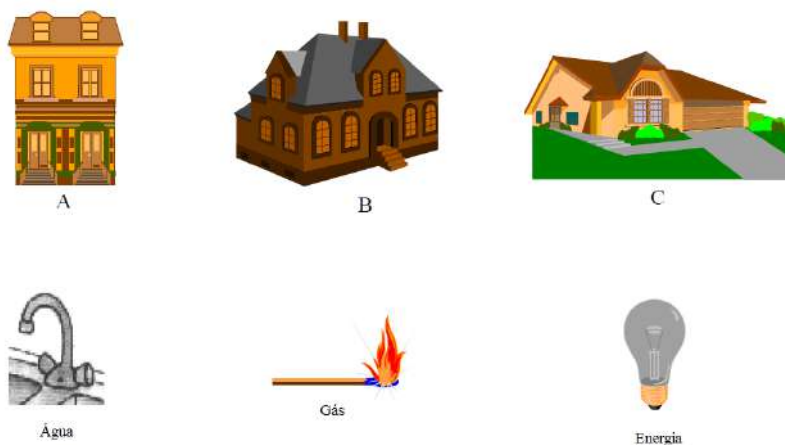


Figura 4.1: Três casas e três serviços públicos.

Definição 4.1.1. *Um grafo é dito planar se puder ser desenhado no plano sem que suas aresta se cruzem.*

Um grafo desenhado com cruzamentos, pode ser planar se for redesenhado de maneira diferente, sem cruzamentos. Vejamos alguns exemplos.

Exemplo 1: O grafo K_4 mostrado na Figura 4.2 é planar.

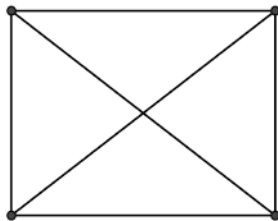


Figura 4.2: O grafo K_4 .

De fato, grafo K_4 é planar, pois pode ser desenhado como mostra a Figura 5.3.

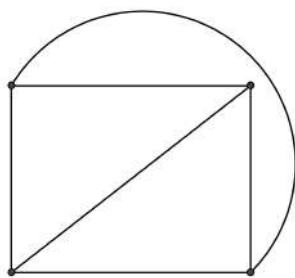


Figura 4.3: O grafo K_4 redesenhado.

Exemplo 2: O grafo mostrado na Figura 4.4 é planar.

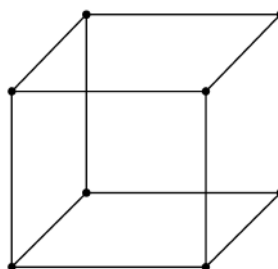


Figura 4.4: Grafo.

De fato, pois pode ser desenhado sem cruzamento de arestas como mostra a Figura 4.5.

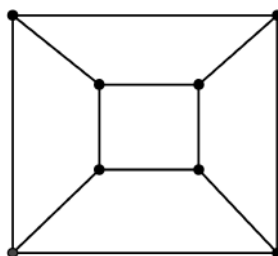


Figura 4.5: Grafo da Figura 5.4 redesenhado.

Exemplo 3: O grafo $K_{3,3}$ mostrado na Figura 4.6 não é planar.

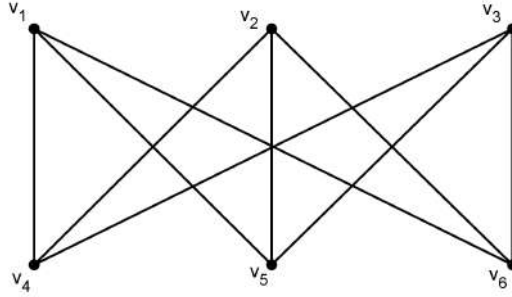


Figura 4.6: Grafo $K_{3,3}$.

De fato, qualquer tentativa de redesenhar o grafo $K_{3,3}$ no plano sem cruzamento de arestas falhará, pois representando $K_{3,3}$, os vértices v_1 e v_2 devem estar conectados tanto a v_4 quanto a v_5 . Esses quatro vértices formam uma curva fechada que divide o plano em duas regiões, R_1 e R_2 , como mostrado na Figura 4.7(a). O vértice v_3 pode estar em R_1 ou pode estar em R_2 . Supondo v_3 em R_2 , o interior da curva fechada, as arestas entre v_3 e v_4 e entre v_3 e v_5 separam R_2 em duas sub-regiões, R_{21} e R_{22} , como mostra a Figura 4.7(b). Logo, não existe nenhuma maneira de colocar v_6 sem que haja um cruzamento, pois se v_6 estiver em R_1 , então a aresta entre v_3 e v_6 não pode ser traçada sem cruzamento. Se v_6 estiver em R_{21} , então a aresta entre v_2 e v_6 não pode ser desenhada sem cruzamento. Se v_6 estiver em R_{22} , então a aresta entre v_1 e v_6 não pode ser desenhada sem cruzamento. E acontece de forma análoga se v_3 estiver em R_1 . Desta forma, pode-se concluir que $K_{3,3}$ não é planar.

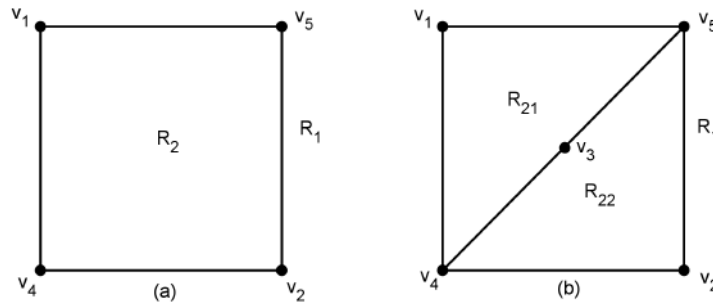


Figura 4.7: Mostrando que $K_{3,3}$ não é planar.

O exemplo 3 mostra que no problema das casas e dos serviços não se pode ligar os três serviços às três casas sem que haja cruzamento de conexões.

O fato de um grafo ser planar desempenha um papel importante no projeto de circuitos eletrônicos. Podemos modelar um circuito com um grafo que representa os componentes por vértices e as conexões por arestas. Pode ser impresso um circuito em uma única placa sem nenhum cruzamento de conexões se o grafo for planar. Se não for planar, deve-se

optar por opções mais caras. Por exemplo, dividir os vértices no grafo que representa o circuito em subgrafos planares. Então, construiremos o circuito original usando fios isolados sempre que as conexões se cruzem, de modo que esse circuito tenha o menor número possível de cruzamentos.

4.2 Fórmula de Euler

Euler mostrou que uma representação planar de um grafo divide o plano em regiões. Conseguiu isto descobrindo uma relação entre o número de regiões, o número de vértices e o número de arestas de um grafo planar. A representação planar do grafo na Figura 4.8 divide o plano em seis regiões.

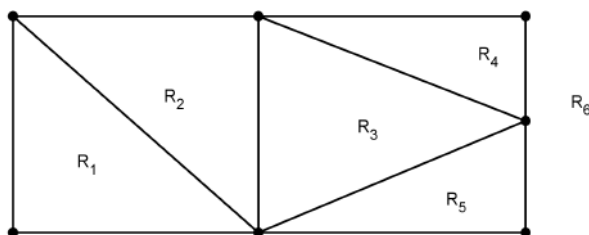


Figura 4.8: Regiões da representação planar.

Teorema 4.2.1. (Euler) *Seja G um grafo simples, planar e conexo com e arestas e v vértices. Seja r o número de regiões em uma representação planar de G . Então, $r - e + v = 2$*

Demonstração: Demonstra-se por indução sobre o número de arestas. Toma-se um grafo conexo qualquer. Se for uma árvore, o número e de arestas é igual ao número de vértices menos um, então temos que $r - e + n = 1 - (n - 1) + n = 2$. Se houver um ciclo, retira-se uma aresta dele, logo o grafo fica com uma face a menos; pela hipótese de indução, a relação vale para um novo grafo. Temos então que $(r - 1) - (e - 1) + v = 2$ e, portanto, $r - e + v = 2$.

Para melhor entendimento faremos uma aplicação deste teorema.

Exemplo: Suponha que um grafo simples planar tenha 20 vértices, cada vértice de grau 3. Em quantas regiões uma representação deste grafo divide o plano?

Como o número de vértices é $|v| = 20$, cada um de grau 3, a soma dos graus dos vértices é $3 \cdot |v| = 3 \cdot 20 = 60$, duas vezes o número de arestas; temos que $2 \cdot e = 60$, logo, $e = 30$. Usando a fórmula de Euler, o número de regiões é $r = e - v + 2 = 30 - 20 + 2 = 12$.

A fórmula de Euler pode ser usada para estabelecer algumas desigualdades que devem ser satisfeitas pelos grafos planares.

Corolário 3. Se G é um grafo simples planar conexo com e arestas e v vértices, em que $v \geq 3$, então $e \leq 3.v - 6$.

Demonstração: Se contarmos as arestas de cada face f (região), contaremos duas vezes cada aresta do grafo. Como cada face tem no mínimo 3 arestas (a igualdade valendo no caso maximal) temos:

$$3.f \leq 2.e$$

Substituindo na fórmula de Euler:

$$\begin{aligned} f - e + v &= 2 \\ 3.f - 3.e + 3.v &= 6 \\ 2.e - 3.e + 3.v &\geq 6 \\ e &\leq 3.v - 6 \end{aligned}$$

Este corolário mostra que K_5 não é planar, pois $e = 10$ e $v = 5$, logo não obedece à relação $e \leq 3.v - 6$ e tem-se que $10 > 3.5 - 6$, é falsa. Portanto, o K_5 não é planar. Também todos os demais grafos completos com mais do que 4 vértices não são planares.

Corolário 4. Num grafo planar bipartido conexo vale $e \leq 2.v - 4$.

Demonstração: Um grafo bipartido só tem ciclos pares. Cada face tem no mínimo quatro arestas.

$$4.f \leq 2.e$$

Substituindo na fórmula de Euler:

$$\begin{aligned} f - e + v &= 2 \\ 4.f - 4.e + 4.v &= 8 \\ 2.e - 4.e + 4.v &\geq 8 \\ e &\leq 2.v - 4 \end{aligned}$$

Em exemplos anteriores vimos que o $K_{3,3}$ e K_5 não são grafos planares. E qualquer grafo que contiver um destes dois grafos como subgrafos, também não será planar. Se um grafo for planar, qualquer grafo obtido da remoção de uma aresta $\{u, v\}$, da adição de um novo vértice w e das arestas $\{u, w\}$ e $\{w, v\}$ também será planar. Tal operação é chamada *subdivisão elementar*. Dois grafos $G_1 = (V_1, E_1)$ e $G_2 = (V_2, E_2)$ são ditos *homeomorfos* se eles puderem ser obtidos do mesmo grafo por uma sequência de subdivisões elementares.

Exemplo: Os grafos G_1 e G_2 da Figura 4.9 são homeomorfos.

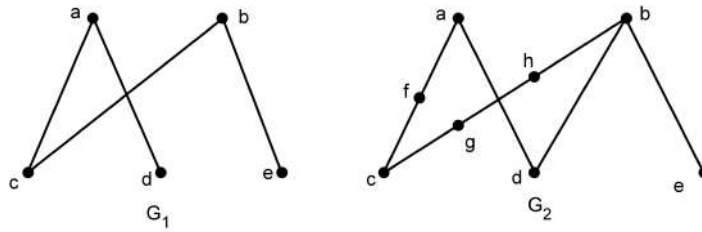


Figura 4.9: Grafos homeomorfos.

De fato, pois ambos podem ser obtidos de G_1 por subdivisões elementares. G_1 pode ser obtido de si mesmo por uma sequência vazia de subdivisões elementares. Para obter G_2 de G_1 , podemos usar uma sequência de subdivisões elementares: (i) remova a aresta $\{a, c\}$ e adicione o vértice f e as arestas $\{a, f\}$ e $\{f, c\}$; (ii) remova a aresta $\{b, c\}$ e adicione o vértice g e as arestas $\{b, g\}$ e $\{g, c\}$; (iii) remova a aresta $\{b, g\}$ e adicione o vértice h e as arestas $\{g, h\}$ e $\{b, h\}$. Logo, G_1 e G_2 são homeomorfos.

Kuratowski ¹ estabeleceu o teorema abaixo, que caracteriza os grafos planares usando conceito de homeomorfismo de grafo.

Teorema 4.2.2. *Um grafo não é planar se e somente se ele contiver um subgrafo homeomorfo a $K_{3,3}$ ou K_5 .*

Um grafo que contenha um subgrafo homeomorfo a $K_{3,3}$ ou a K_5 não é planar. Entretanto, a demonstração da recíproca é complexa e omitiremos sua demonstração. O leitor interessado pode consultar [5] e [8] para maiores detalhes.

4.3 Coloração de grafos

Um dos problemas mais famosos e conhecidos de teoria dos grafos que desafiou muitos matemáticos por muito tempo e que ainda não foi resolvido satisfatoriamente, é um problema sobre grafos planares: *O problema das quatro cores*.

Quando um mapa é colorido, duas regiões com fronteira comum são associadas a cores diferentes, isto garante que duas regiões adjacentes nunca devem ter a mesma cor. Entretanto, um número mínimo de cores deve ser usado sempre que possível. Duas regiões com apenas um ponto em comum não são consideradas adjacentes.

Para colorir, com o menor número de cores, o mapa 1 da Figura 4.10, quatro cores seriam suficientes, mas três cores não seriam. Entretanto, no mapa 2, três cores bastariam, mas duas não.

Cada mapa no plano pode ser representado por um grafo, onde cada região do mapa é representado por um vértice. As arestas conectam dois vértices se as regiões tiverem

¹ Matemático polonês.

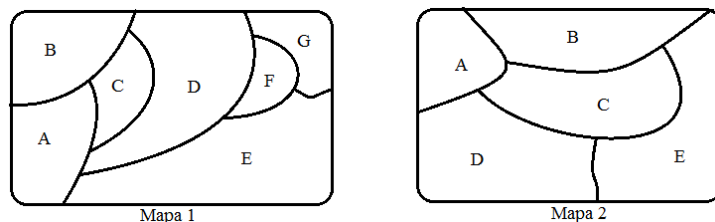


Figura 4.10: Mapas.

uma fronteira em comum. Desta forma, o grafo resultante é chamado *grafo dual* do mapa. Pela maneira na qual os grafos duais de mapas são construídos, qualquer mapa no plano tem um grafo dual planar. A Figura 4.11 mostra os grafos duais que correspondem aos mapas da Figura 4.10.

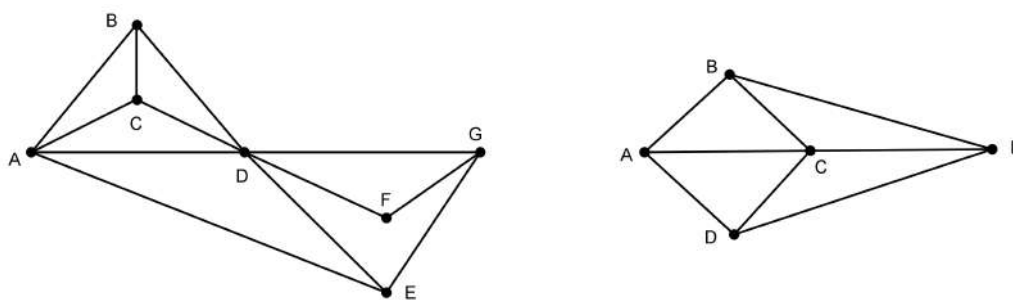


Figura 4.11: Grafos duais dos mapas da Figura 5.10.

Colorir as regiões de um mapa é equivalente a colorir os vértices de um mapa dual, de modo que dois vértices adjacentes não tenham a mesma cor.

Definição 4.3.1. *Uma coloração de um grafo simples é a associação de uma cor a cada vértice do grafo, de modo que dois vértices adjacentes não estejam associados à mesma cor.*

Definição 4.3.2. *O número cromático de um grafo é o menor número de cores necessárias para a coloração deste grafo. O número cromático de um grafo G é indicado por $\chi(G)$.*

Teorema 4.3.1. *O número cromático de um grafo planar não é maior do que quatro.*

O grafo K_4 mostra que 4 cores são necessárias, mas serão suficientes? O problema demorou um século para ser resolvido. Em 1976, Appel, Haken e Koch [11], com o auxílio do computador mais rápido de sua época, trabalhando mais de 1000 horas e executando mais do que 10^{10} operações computacionais, provaram o teorema.

O teorema das quatro cores se aplica apenas a grafos planares, pois grafos não planares podem ter números cromáticos arbitrariamente grandes, como mostrado no segundo exemplo a seguir.

Duas coisas são necessárias para mostrar que o número cromático de um grafo é K .

1º) Precisamos mostrar que o grafo pode ser colorido com K cores. Isso pode ser feito construindo tal coloração.

2º) Devemos mostrar que o grafo não pode ser colorido usando menos que K cores.

Os exemplos a seguir ilustram como números cromáticos podem ser encontrados.

Exemplo 1: Qual é o número cromático do grafo mostrado na Figura 4.12?

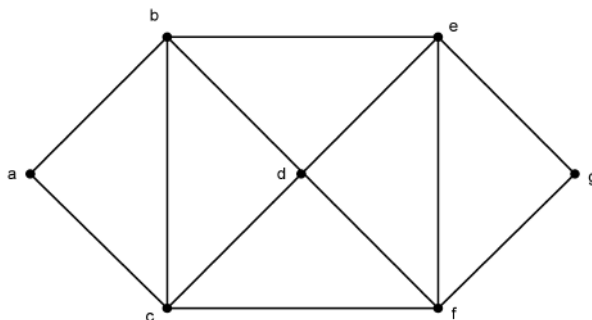


Figura 4.12: Grafos simples.

O número cromático do grafo é pelo menos igual a 3, pois os vértices a , b e c devem ter cores diferentes. Para ver se o grafo pode ser colorido com três cores, associamos vermelho a a , azul a b e verde a c . Então, d pode ser colorido de vermelho, pois é adjacente a b e c . Além disso, e pode ser colorido de verde, pois é adjacente apenas a vértices de cor vermelha e azul, e f pode ser colorido de azul, pois é adjacente apenas a vértices colorido de vermelho e verde. Finalmente, g pode ser colorido de vermelho, pois é adjacente apenas a vértices coloridos de azul e verde. Este grafo não pode ser colorido apenas com duas cores. Portanto, o número cromático desse grafo é igual a 3, como mostra a coloração da Figura 4.13

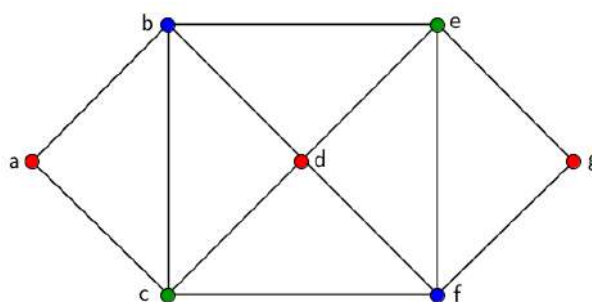


Figura 4.13: Coloração da Figura 5.12.

Exemplo 2: Qual é o número cromático de K_n ?

Uma coloração de K_n pode ser construída usando n cores, associando uma cor diferente a cada vértice, pois não podemos usar menos cores, porque nenhum par de vértices pode

estar associado à mesma cor e no K_n quaisquer dois vértices são adjacentes. Logo, o número cromático $\chi(K_n) = n$.

Observação: O grafo K_n não é planar, pois quando $n \geq 5$ contradiz o teorema das quatro cores. Veja, como exemplo, na Figura 4.14 uma coloração do grafo K_5 .

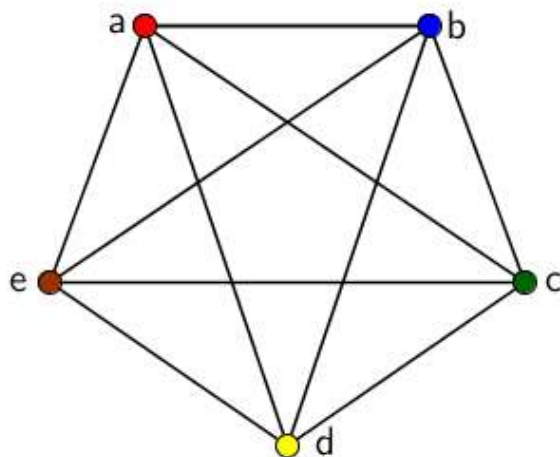


Figura 4.14: Coloração do K_5 .

Exemplo 3: Qual é o número cromático do grafo bipartido completo $K_{m,n}$ em que m e n são inteiros positivos?

Aparentemente esse número depende de m e n . Apenas duas cores são necessárias, pois $K_{m,n}$ é bipartido.

Os melhores algoritmos conhecidos para determinar o número cromático de um grafo têm complexidade no tempo exponencial no pior caso. Mesmo para encontrar uma aproximação para o número cromático de um grafo é uma tarefa difícil. [8]

Capítulo 5

Aplicações

Neste Capítulo resolveremos alguns problemas clássicos para encontrar caminho/custo mínimos e alguns problemas de coloração.

5.1 Caminho mínimo

5.1.1 Problema 1

Deseja-se encontrar o caminho mínimo da cidade A para as demais cidades B, C, D, E e F, de modo que os caminhos existentes entre as cidades com suas respectivas distâncias, em quilômetros, estão sendo indicadas no grafo da Figura 5.1.

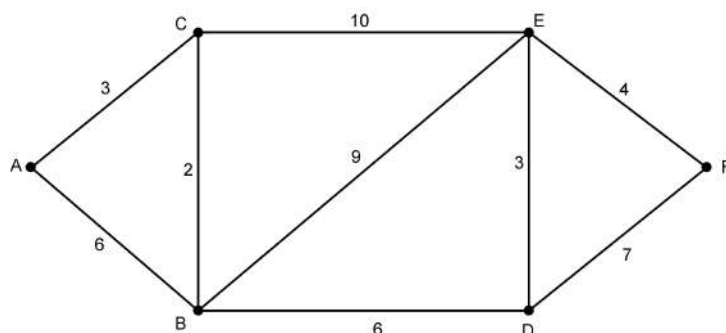


Figura 5.1: Grafo das cidades interligadas.

O algoritmo de Dijkstra visto no Capítulo 3, Seção 3.3 lê: $S = \{\}$, o conjunto que guardará os vértices finalizados, $V = \{A, B, C, D, E, F\}$, o conjunto de todos os vértices do grafo e todas as arestas com seus respectivos pesos w . Atribui as distâncias: $d[A] = 0$ para o vértice de origem e aos demais, distâncias $d[B] = d[C] = d[D] = d[E] = d[F] = \infty$. A execução da iteração é visualizada como mostra a tabela e a Figura 5.2.

Vértices	A	B	C	D	E	F
distância	0	∞	∞	∞	∞	∞
pai
finaliza

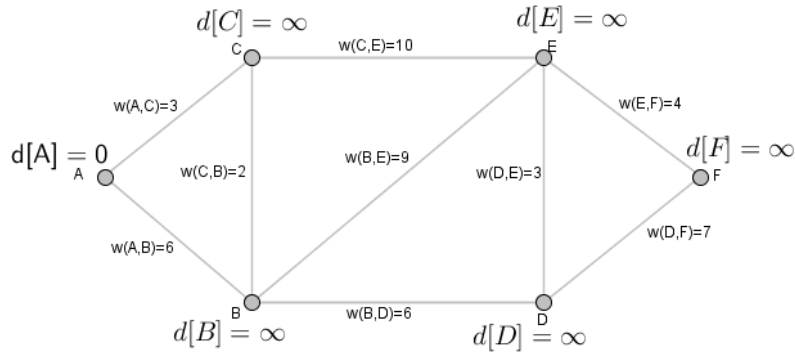


Figura 5.2: Tabela e grafo do problema 1.

Entra no laço ENQUANTO : guarda (ou finaliza) em S , o vértice de menor distância com seu respectivo pai (anterior a ele).

menor distância: $d[A]=0$; $\text{pai}[A]=$ nulo; A foi finalizado

$S = \{ A \}$ e $V = \{ B, C, D, E \text{ e } F \}$. (Figura 5.3)

Vértices	A	B	C	D	E	F
distância	0	∞	∞	∞	∞	∞
pai	nulo
finaliza	sim

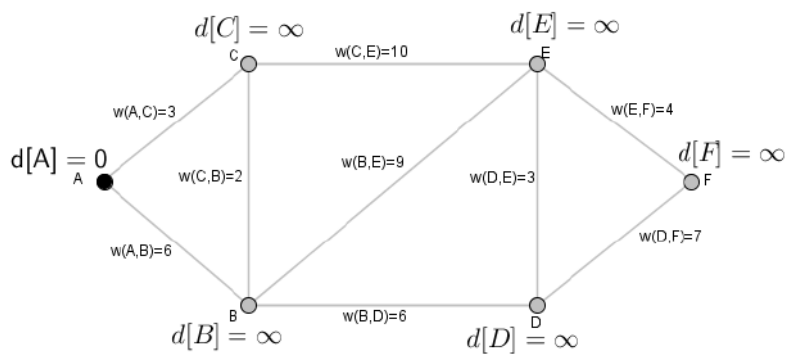


Figura 5.3: Tabela e grafo do problema 1.

Entra no laço PARA: atualiza as distâncias dos vértices adjacentes ao vértice finalizado, até que o conjunto V seja vazio, fazendo:

- Para todo $adj[A]$, tem-se:

$$d[B] = (d[A] + w(A, B)) = 0 + 6 = 6 ; \text{pai}[B] = A$$

$d[C] = (d[A] + w(A, C)) = 0 + 3 = 3$; $\text{pai}[C] = A$
 Guarda menor distância: $d[C]=3$; $\text{pai}[C] = A$; C foi finalizado
 $S = \{A, C\}$ e $V = \{B, D, E \text{ e } F\}$. (Figura 5.4)

Vértices	A	B	C	D	E	F
distância	0	6	3	∞	∞	∞
pai	nulo	A	A
finaliza	sim	...	sim

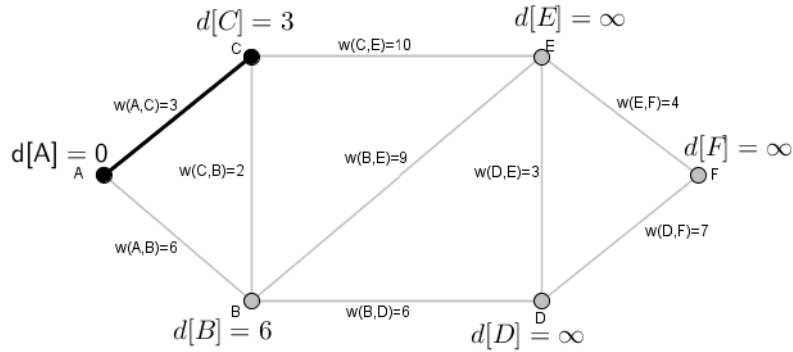


Figura 5.4: Tabela e grafo do problema 1.

- Para todo $\text{adj}[C]$, tem-se:

$d[B] = (d[C] + w(C, B)) = 3 + 2 = 5 < d[B] = 6$; atualiza $d[B]=5$; $\text{pai}[B]=C$
 $d[E] = (d[C] + w(C, E)) = 3 + 10 = 13$; $\text{pai}[E]=C$. (Figura 5.5)

Vértices	A	B	C	D	E	F
distância	0	5	3	∞	13	∞
pai	nulo	C	A	...	C	...
finaliza	sim	...	sim

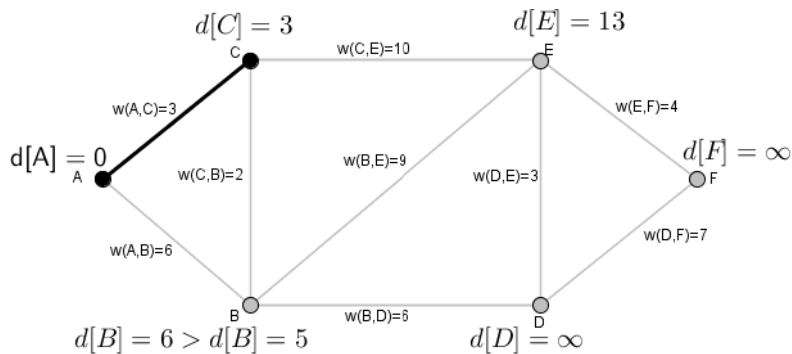


Figura 5.5: Tabela e grafo do problema 1.

$d[B] = (d[C] + w(C, B)) = 5$; $\text{pai}[B]=C$
 $d[E] = (d[C] + w(C, E)) = 13$; $\text{pai}[E]=C$

Guarda menor distância: $d[B]=5$; $\text{pai}[B]= C$; B foi finalizado
 $S =\{ A, B, C\}$ e $V =\{D, E \text{ e } F\}$. (Figura 5.6)

Vértices	A	B	C	D	E	F
distância	0	5	3	∞	13	∞
pai	nulo	C	A	...	C	...
finaliza	sim	sim	sim	

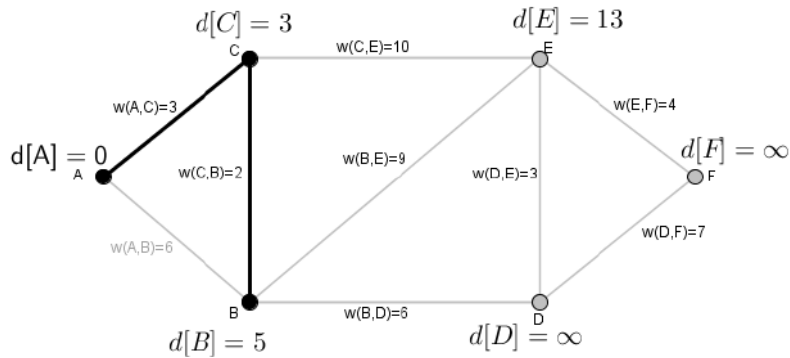


Figura 5.6: Tabela e grafo do problema 1.

- Para todo $\text{adj}[B]$, tem-se:

$$d[D] = (d[B] + w(B, D)) = 5 + 6 = 11; \text{pai}[D] = B$$

$$d[E] = (d[B] + w(B, E)) = 5 + 9 = 14 > d[E] = 13; \text{mantém } d[E] \text{ anterior pois é menor.}$$

Guarda menor distância: $d[D] = 11$; $\text{pai}[D] = B$; D foi finalizado.

$S =\{ A, B, C, D\}$ e $V =\{E, F\}$. (Figura 5.7)

Vértices	A	B	C	D	E	F
distância	0	5	3	11	13	∞
pai	nulo	C	A	B	C	...
finaliza	sim	sim	sim	sim

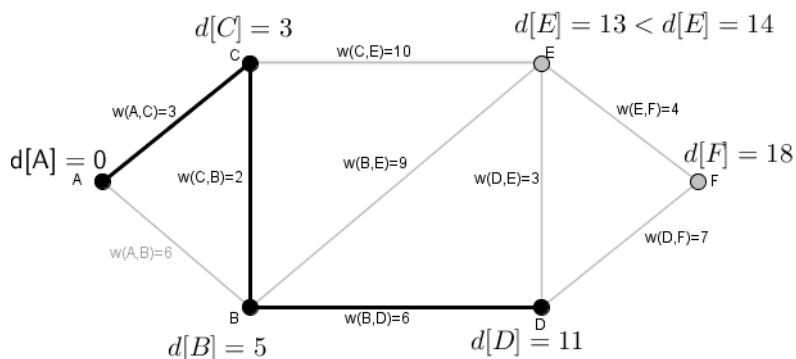


Figura 5.7: Tabela e grafo do problema 1.

- Para todo $adj[D]$, tem-se:

$d[E] = (d[D] + w(D, E)) = 11 + 3 = 14 > d[E] = 13$; mantém $d[E]$ anterior pois é menor.

$$d[F] = (d[D] + w(D, F)) = 11 + 7 = 18; \text{pai}[F] = D$$

Guarda menor distância: $d[E] = 13$; $\text{pai}[E] = C$; E foi finalizado.

$S = \{A, B, C, D, E\}$ e $V = \{F\}$. (Figura 5.8)

Vértices	A	B	C	D	E	F
distância	0	5	3	11	13	18
pai	nulo	C	A	B	C	D
finaliza	sim	sim	sim	sim	sim	...

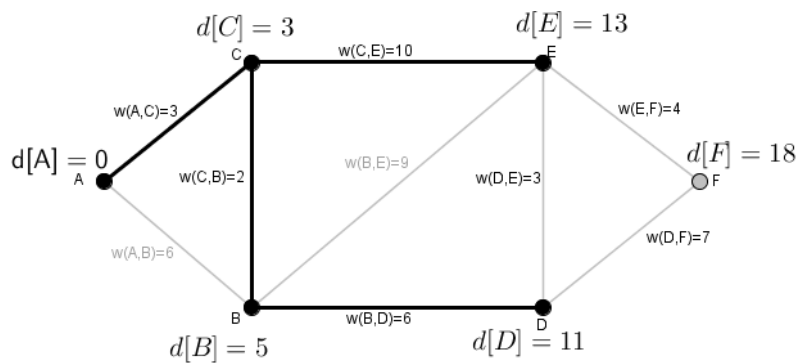


Figura 5.8: Tabela e grafo do problema 1.

- Para todo $adj[E]$, tem-se:

$d[F] = (d[E] + w(E, F)) = 13 + 4 = 17 < d[F] = 18$. atualiza $d[F]$. (Figura 5.9)

Vértices	A	B	C	D	E	F
distância	0	5	3	11	13	17
pai	nulo	C	A	B	C	E
finaliza	sim	sim	sim	sim	sim	sim

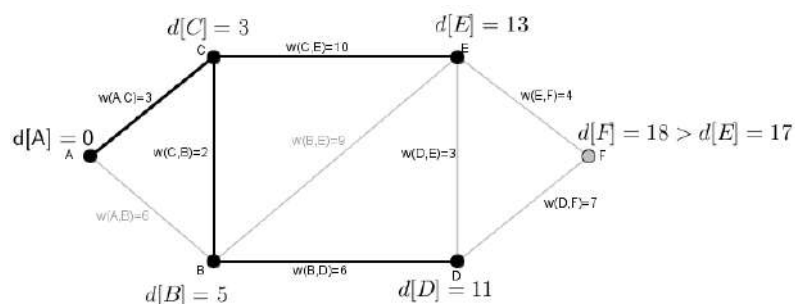


Figura 5.9: Tabela e grafo do problema 1.

Guarda menor distância: $d[F]= 17$; $\text{pai}[F]= E$; F foi finalizado.

$S =\{ A, B, C, D, E, F\}$ e $V =\{ \}$. Como todos os vértices foram finalizados, encontramos a tabela abaixo, com a distância do vértice de origem A aos demais vértices, indicando seus pais. (Figura 5.10)

Vértices	A	B	C	D	E	F
distância	0	5	3	11	13	17
pai	nulo	C	A	B	C	E
finaliza	sim	sim	sim	sim	sim	sim

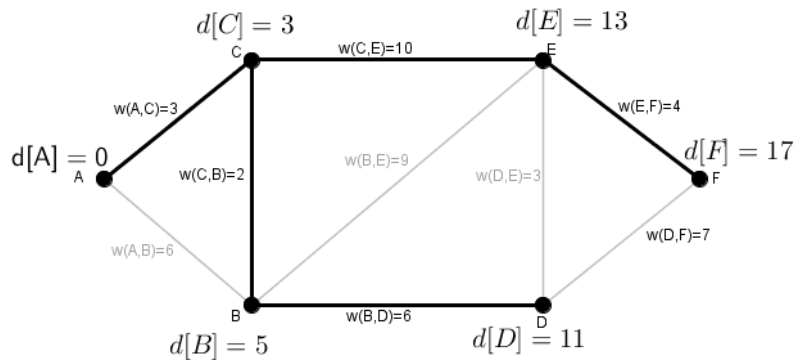


Figura 5.10: Tabela e grafo do problema 1.

5.1.2 Problema 2

Em certa cidade, os bairros: $A, B, C, D, E, F, G, H, I, J, K, L$, estão ligados entre si como mostra o grafo da Figura 5.11, com suas respectivas distâncias em quilômetros. O prefeito resolveu asfaltar uma única rua que ligue dois bairros quaisquer, pois não dava para asfaltar todas as ruas da cidade. Para isso, queria encontrar a menor quantidade de asfalto a ser usada, de forma que duas cidades estejam ligadas por uma rua asfaltada.

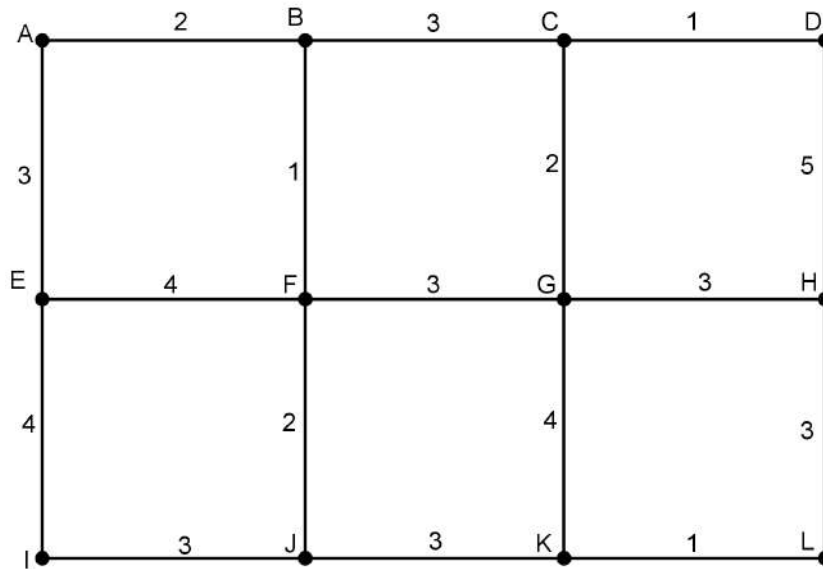


Figura 5.11: Grafo modelando os bairros de uma cidade.

O problema será resolvido usando o algoritmo de Kruskal visto no Capítulo 3, Seção 3.4, pois determina o custo total mínimo. O algoritmo de Kruskal, primeiramente, constrói uma floresta onde cada árvore é composta de um único vértice do grafo, depois ordena as arestas em ordem crescente de pesos w como mostra a tabela a seguir.

Floresta: $\{\{A\},\{B\},\{C\},\{D\},\{E\},\{F\},\{G\},\{H\},\{I\},\{J\},\{K\},\{L\}\}$.

A tabela abaixo mostra as arestas ordenadas em ordem crescente, com seus respectivos pesos, mostrados no grafo da Figura 5.11.

ordem	1	2	3	4	5	6	7	8	9	10	11	12	13
arestas	CD	BF	KL	AB	CG	FJ	HL	JK	FG	GH	IJ	BC	AE
peso	1	1	1	2	2	2	3	3	3	3	3	3	3
insere

Entra no laço PARA: verifica cada aresta ordenada, se os vértices, que são extremos dela, estão na mesma árvore. Se não estiver, faz a união das duas árvores, caso contrário, passa para a próxima aresta, até que a floresta tenha uma única árvore. Isso garante custo mínimo total.

O procedimento acontece da seguinte forma:

Dada a floresta:

$\{\{A\},\{B\},\{C\},\{D\},\{E\},\{F\},\{G\},\{H\},\{I\},\{J\},\{K\},\{L\}\}$.

A execução da iteração é visualizada como mostra tabela e a Figura 5.12.

ordem	1	2	3	4	5	6	7	8	9	10	11	12	13
arestas	<i>CD</i>	<i>BF</i>	<i>KL</i>	<i>AB</i>	<i>CG</i>	<i>FJ</i>	<i>HL</i>	<i>JK</i>	<i>FG</i>	<i>GH</i>	<i>IJ</i>	<i>BC</i>	<i>AE</i>
peso	1	1	1	2	2	2	3	3	3	3	3	3	3
insere

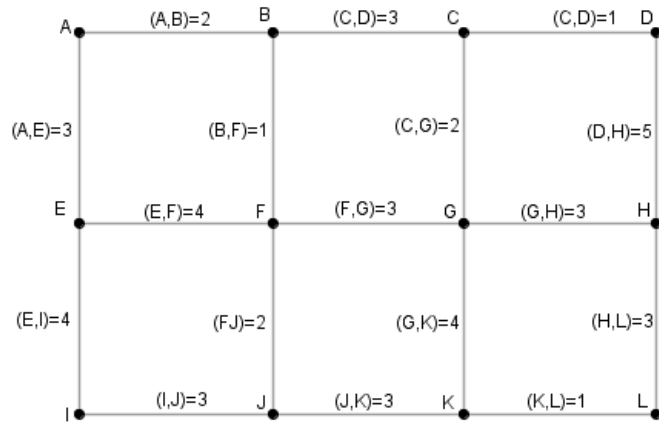


Figura 5.12: Tabela e grafo do problema 2.

1 - Os vértices C e D estão na mesma árvore? Não, então C e D ficam na mesma árvore (faz-se a união dos dois conjuntos). (Figura 5.13).

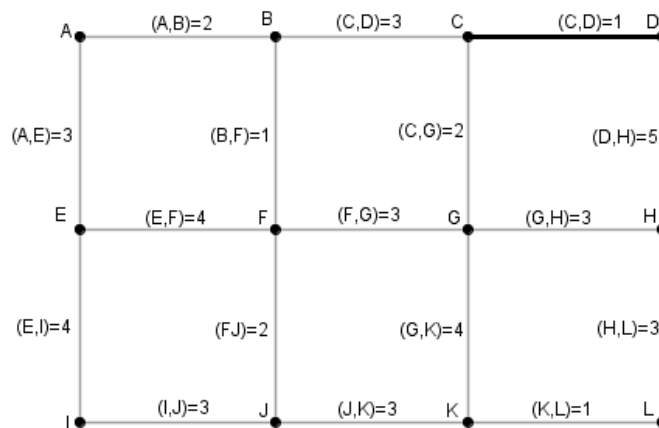


Figura 5.13: Tabela e grafo do problema 2.

Dada a nova floresta e tabela atualizada:

$\{\{A\}, \{B\}, \{C,D\}, \{E\}, \{F\}, \{G\}, \{H\}, \{I\}, \{J\}, \{K\}, \{L\}\}$.

ordem	1	2	3	4	5	6	7	8	9	10	11	12	13
arestas	<i>CD</i>	<i>BF</i>	<i>KL</i>	<i>AB</i>	<i>CG</i>	<i>FJ</i>	<i>HL</i>	<i>JK</i>	<i>FG</i>	<i>GH</i>	<i>IJ</i>	<i>BC</i>	<i>AE</i>
peso	1	1	1	2	2	2	3	3	3	3	3	3	3
insere	sim

2 - Os vértices B e F estão na mesma árvore? Não, então B e F ficam na mesma árvore (faz-se a união dos dois conjuntos). (Figura 5.14).

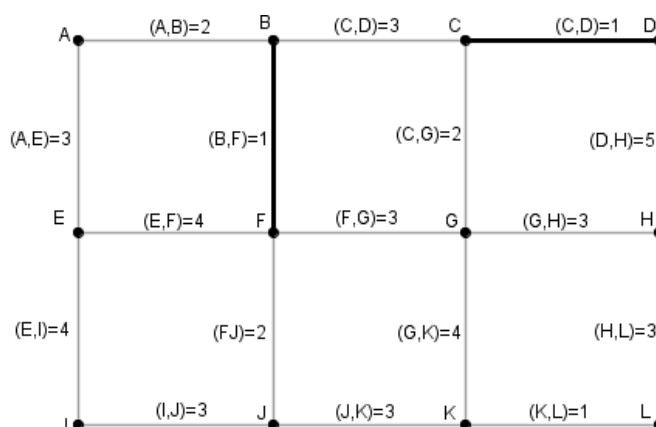


Figura 5.14: Tabela e grafo do problema 2.

Dada a nova floresta e tabela atualizada:

$\{\{A\}, \{B,F\}, \{C,D\}, \{E\}, \{G\}, \{H\}, \{I\}, \{J\}, \{K\}, \{L\}\}$.

ordem	1	2	3	4	5	6	7	8	9	10	11	12	13
arestas	<i>CD</i>	<i>BF</i>	<i>KL</i>	<i>AB</i>	<i>CG</i>	<i>FJ</i>	<i>HL</i>	<i>JK</i>	<i>FG</i>	<i>GH</i>	<i>IJ</i>	<i>BC</i>	<i>AE</i>
peso	1	1	1	2	2	2	3	3	3	3	3	3	3
insere	sim	sim

3 - Os vértices K e L estão na mesma árvore? Não, então K e L ficam na mesma árvore (faz-se a união dos dois conjuntos). (Figura 5.15).

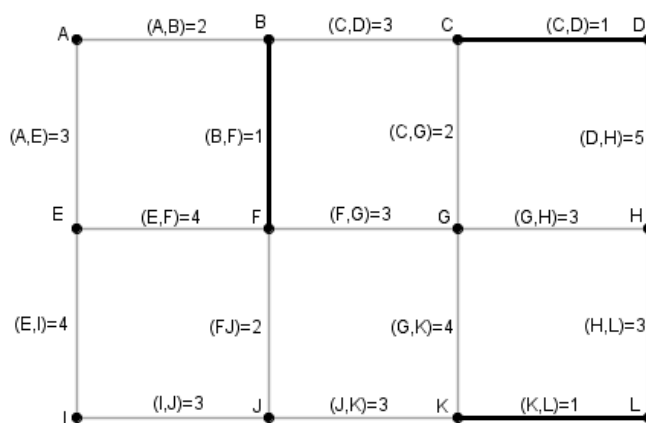


Figura 5.15: Tabela e grafo do problema 2.

Dada a nova floresta e tabela atualizada:

$\{\{A\}, \{B,F\}, \{C,D\}, \{E\}, \{G\}, \{H\}, \{I\}, \{J\}, \{K,L\}\}$.

ordem	1	2	3	4	5	6	7	8	9	10	11	12	13
arestas	<i>CD</i>	<i>BF</i>	<i>KL</i>	<i>AB</i>	<i>CG</i>	<i>FJ</i>	<i>HL</i>	<i>JK</i>	<i>FG</i>	<i>GH</i>	<i>IJ</i>	<i>BC</i>	<i>AE</i>
peso	1	1	1	2	2	2	3	3	3	3	3	3	3
insere	sim	sim	sim

4 - Os vértices A e B estão na mesma árvore? Não, então A e B ficam na mesma árvore (faz-se a união dos dois conjuntos). (Figura 5.16).

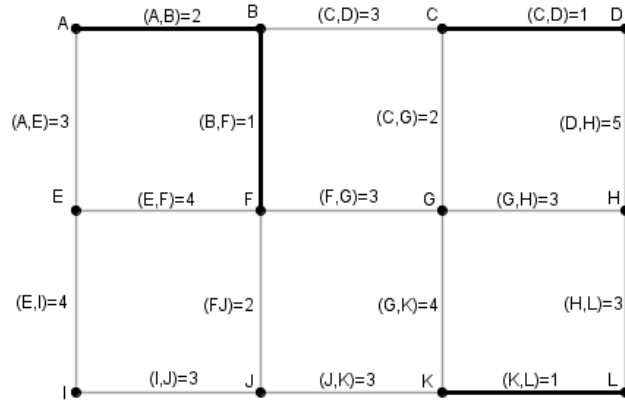


Figura 5.16: Tabela e grafo do problema 2.

Dada a nova floresta e tabela atualizada:

$\{\{A,B,F\}, \{C,D\}, \{E\}, \{G\}, \{H\}, \{I\}, \{J\}, \{K,L\}\}$.

ordem	1	2	3	4	5	6	7	8	9	10	11	12	13
arestas	<i>CD</i>	<i>BF</i>	<i>KL</i>	<i>AB</i>	<i>CG</i>	<i>FJ</i>	<i>HL</i>	<i>JK</i>	<i>FG</i>	<i>GH</i>	<i>IJ</i>	<i>BC</i>	<i>AE</i>
peso	1	1	1	2	2	2	3	3	3	3	3	3	3
insere	sim	sim	sim	sim

5 - Os vértices C e G estão na mesma árvore? Não, então C e G ficam na mesma árvore (faz-se a união dos dois conjuntos). (Figura 5.17).

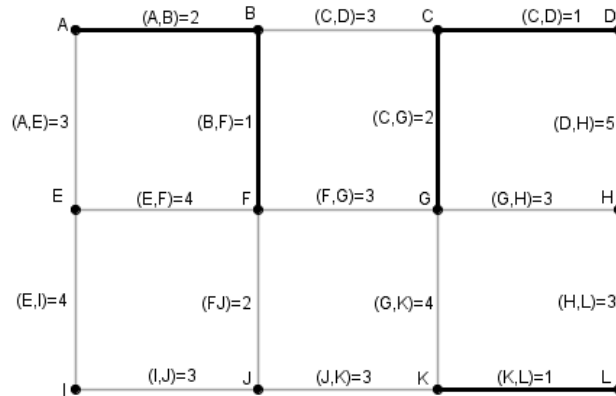


Figura 5.17: Tabela e grafo do problema 2.

Dada a nova floresta e tabela atualizada:
 $\{\{A,B,F\},\{C,D,G\},\{E\},\{H\},\{I\},\{J\},\{K,L\}\}$.

ordem	1	2	3	4	5	6	7	8	9	10	11	12	13
arestas	<i>CD</i>	<i>BF</i>	<i>KL</i>	<i>AB</i>	<i>CG</i>	<i>FJ</i>	<i>HL</i>	<i>JK</i>	<i>FG</i>	<i>GH</i>	<i>IJ</i>	<i>BC</i>	<i>AE</i>
peso	1	1	1	2	2	2	3	3	3	3	3	3	3
insere	sim	sim	sim	sim	sim

6 - Os vértices F e J estão na mesma árvore? Não, então F e J ficam na mesma árvore (faz-se a união dos dois conjuntos). (Figura 5.18).

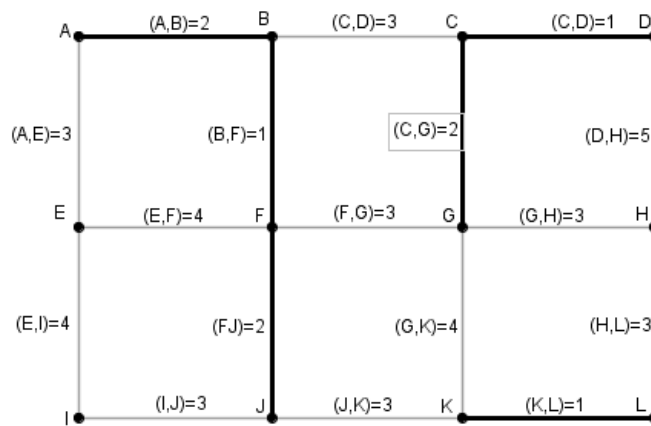


Figura 5.18: Tabela e grafo do problema 2.

Dada a nova floresta e tabela atualizada:
 $\{\{A,B,F,J\},\{C,D,G\},\{E\},\{H\},\{I\},\{K,L\}\}$.

ordem	1	2	3	4	5	6	7	8	9	10	11	12	13
arestas	<i>CD</i>	<i>BF</i>	<i>KL</i>	<i>AB</i>	<i>CG</i>	<i>FJ</i>	<i>HL</i>	<i>JK</i>	<i>FG</i>	<i>GH</i>	<i>IJ</i>	<i>BC</i>	<i>AE</i>
peso	1	1	1	2	2	2	3	3	3	3	3	3	3
insere	sim	sim	sim	sim	sim	sim

7 - Os vértices H e L estão na mesma árvore? Não, então H e L ficam na mesma árvore (faz-se a união dos dois conjuntos). (Figura 5.19).

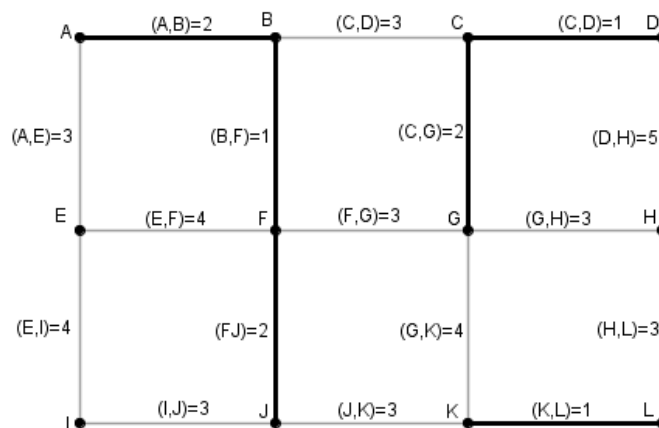


Figura 5.19: Tabela e grafo do problema 2.

Dada a nova floresta e tabela atualizada:

$\{\{A,B,F,J\},\{C,D,G\},\{E\},\{H,K,L\},\{I\}\}$.

ordem	1	2	3	4	5	6	7	8	9	10	11	12	13
arestas	<i>CD</i>	<i>BF</i>	<i>KL</i>	<i>AB</i>	<i>CG</i>	<i>FJ</i>	<i>HL</i>	<i>JK</i>	<i>FG</i>	<i>GH</i>	<i>IJ</i>	<i>BC</i>	<i>AE</i>
peso	1	1	1	2	2	2	3	3	3	3	3	3	3
insere	sim	sim	sim	sim	sim	sim	sim

8 - Os vértices J e K estão na mesma árvore? Não, então J e K ficam na mesma árvore (faz-se a união dos dois conjuntos). (Figura 5.20).

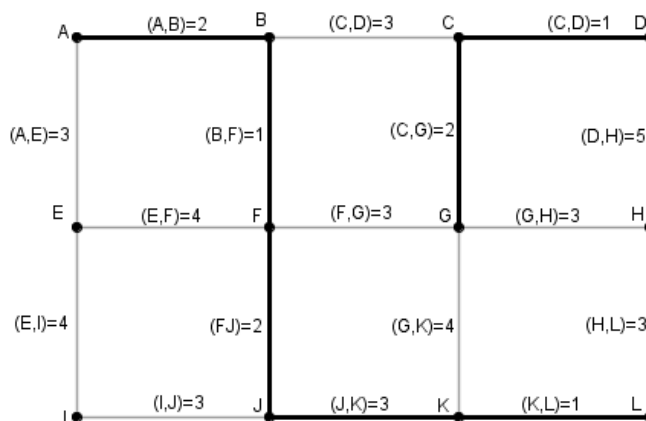


Figura 5.20: Tabela e grafo do problema 2.

Dada a nova floresta e tabela atualizada:

$\{\{A,B,F,H,J,K,L\},\{C,D,G\},\{E\},\{I\}\}$.

ordem	1	2	3	4	5	6	7	8	9	10	11	12	13
arestas	<i>CD</i>	<i>BF</i>	<i>KL</i>	<i>AB</i>	<i>CG</i>	<i>FJ</i>	<i>HL</i>	<i>JK</i>	<i>FG</i>	<i>GH</i>	<i>IJ</i>	<i>BC</i>	<i>AE</i>
peso	1	1	1	2	2	2	3	3	3	3	3	3	3
insere	sim	sim	sim	sim	sim	sim	sim	sim

9 - Os vértices F e G estão na mesma árvore? Não, então F e G ficam na mesma árvore (faz-se a união dos dois conjuntos). (Figura 5.21).

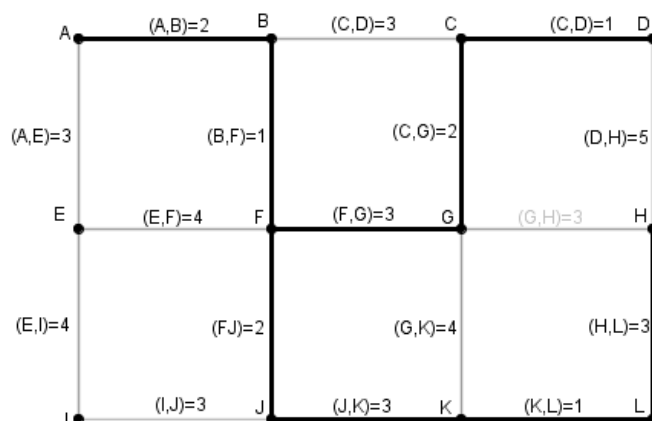


Figura 5.21: Tabela e grafo do problema 2.

Dada a nova floresta e tabela atualizada:

$\{\{A,B,C,D,F,G,H,J,K,L\}, \{E\}, \{I\}\}$.

ordem	1	2	3	4	5	6	7	8	9	10	11	12	13
arestas	<i>CD</i>	<i>BF</i>	<i>KL</i>	<i>AB</i>	<i>CG</i>	<i>FJ</i>	<i>HL</i>	<i>JK</i>	<i>FG</i>	<i>GH</i>	<i>IJ</i>	<i>BC</i>	<i>AE</i>
peso	1	1	1	2	2	2	3	3	3	3	3	3	3
insere	sim	sim	sim	sim	sim	sim	sim	sim	sim

10 - Os vértices G e H estão na mesma árvore? Sim, lê próximos vértices. Não altera a Floresta e atualiza a tabela.

Dada a nova floresta e tabela atualizada:

$\{\{A,B,C,D,F,G,H,J,K,L\}, \{E\}, \{I\}\}$.

ordem	1	2	3	4	5	6	7	8	9	10	11	12	13
arestas	<i>CD</i>	<i>BF</i>	<i>KL</i>	<i>AB</i>	<i>CG</i>	<i>FJ</i>	<i>HL</i>	<i>JK</i>	<i>FG</i>	<i>GH</i>	<i>IJ</i>	<i>BC</i>	<i>AE</i>
peso	1	1	1	2	2	2	3	3	3	3	3	3	3
insere	sim	sim	sim	sim	sim	sim	sim	sim	sim	não

11 - Os vértices I e J estão na mesma árvore? Não, então I e J ficam na mesma árvore (faz-se a união dos dois conjuntos). (Figura 5.22).

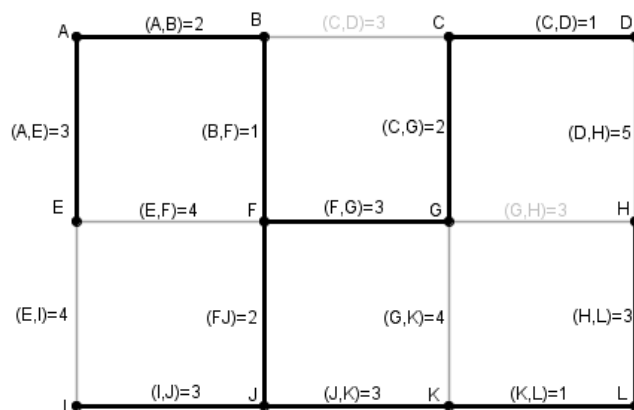


Figura 5.22: Tabela e grafo do problema 2.

Dada a nova floresta e tabela atualizada:

$\{\{A,B,C,D,F,G,H,I,J,K,L\},\{E\}\}$.

ordem	1	2	3	4	5	6	7	8	9	10	11	12	13
arestas	<i>CD</i>	<i>BF</i>	<i>KL</i>	<i>AB</i>	<i>CG</i>	<i>FJ</i>	<i>HL</i>	<i>JK</i>	<i>FG</i>	<i>GH</i>	<i>IJ</i>	<i>BC</i>	<i>AE</i>
peso	1	1	1	2	2	2	3	3	3	0	3	3	3
insere	sim	sim	sim	sim	sim	sim	sim	sim	sim	não	sim

12- Os vértices B e C estão na mesma árvore? Sim, lê próximos vértices. Não altera a Floresta e atualiza a tabela.

ordem	1	2	3	4	5	6	7	8	9	10	11	12	13
arestas	<i>CD</i>	<i>BF</i>	<i>KL</i>	<i>AB</i>	<i>CG</i>	<i>FJ</i>	<i>HL</i>	<i>JK</i>	<i>FG</i>	<i>GH</i>	<i>IJ</i>	<i>BC</i>	<i>AE</i>
peso	1	1	1	2	2	2	3	3	3	0	3	3	3
insere	sim	sim	sim	sim	sim	sim	sim	sim	sim	não	sim	não	...

Dada a nova floresta e tabela atualizada:

$\{\{A,B,C,D,F,G,H,I,J,K,L\},\{E\}\}$.

ordem	1	2	3	4	5	6	7	8	9	10	11	12	13
arestas	<i>CD</i>	<i>BF</i>	<i>KL</i>	<i>AB</i>	<i>CG</i>	<i>FJ</i>	<i>HL</i>	<i>JK</i>	<i>FG</i>	<i>GH</i>	<i>IJ</i>	<i>BC</i>	<i>AE</i>
peso	1	1	1	2	2	2	3	3	3	0	3	0	3
insere	sim	sim	sim	sim	sim	sim	sim	sim	sim	não	sim	não	...

13- Os vértices A e E estão na mesma árvore? Não, então A e E ficam na mesma árvore (faz-se a união dos dois conjuntos). (Figura 5.23).

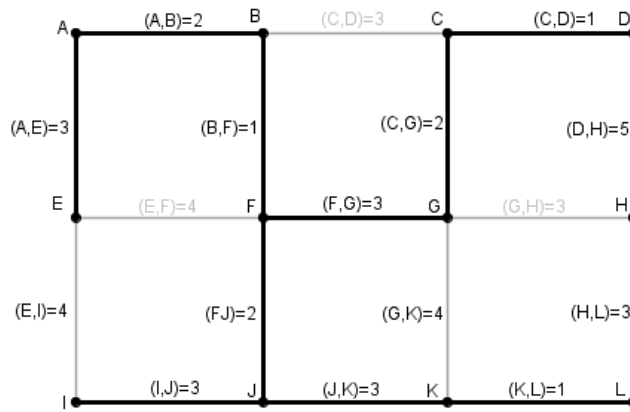


Figura 5.23: Tabela e grafo do problema 2.

14 - Como temos uma única árvore na floresta, não será possível inserir nenhum dos demais vértices, pois neste caso um ciclo será. Temos o custo mínimo total somando as arestas inseridas, que é igual a 27 quilômetros de asfalto, mostrado na Figura 5.24. $\{\{A,B,C,D,E,F,G,H,I,J,K,L\}\}$.

ordem	1	2	3	4	5	6	7	8	9	10	11	12	13
arestas	CD	BF	KL	AB	CG	FJ	HL	JK	FG	GH	IJ	BC	AE
peso	1	1	1	2	2	2	3	3	3	0	3	0	3
insere	sim	sim	sim	sim	sim	sim	sim	sim	sim	não	sim	não	sim

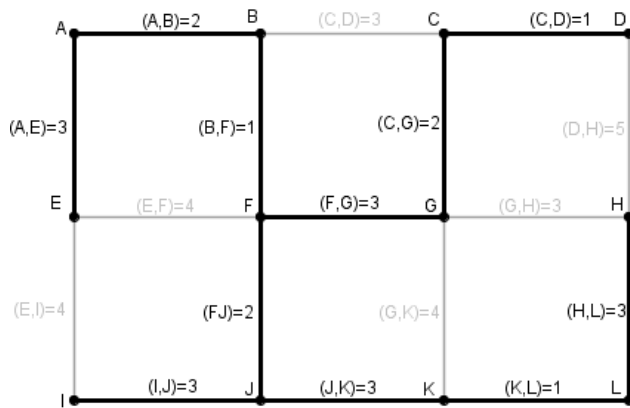


Figura 5.24: Tabela e grafo do problema 2.

5.1.3 Problema 3

Encontrar o caminho mínimo do vértice A (origem) em relação aos demais vértices do grafo do problema 2. (Figura 5.11)

Aplicando o algoritmo de Dijkstra:

Lê todos os vértices: o vértice de origem A com $d[A] = 0$ e os demais vértices inicializam com $d[v] = \infty$. Também, lê todas as arestas do grafo com seus respectivos pesos w . (Figura 5.25)

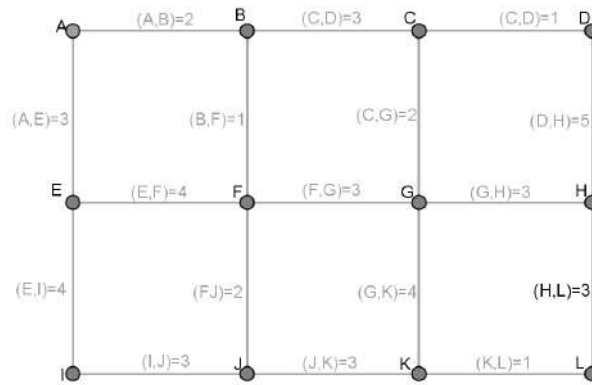


Figura 5.25: Grafo modelando os bairros de uma cidade.

Vértices	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>G</i>	<i>H</i>	<i>I</i>	<i>J</i>	<i>K</i>	<i>L</i>
Distâncias	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
pai
Finaliza

Finaliza o vértice *A* (menor distância); como é o vértice de origem, seu pai é nulo. (Figura 5.26)

$S = \{ A \}$ e $V = \{ B, C, D, E, F, G, H, I, J, K, L \}$.

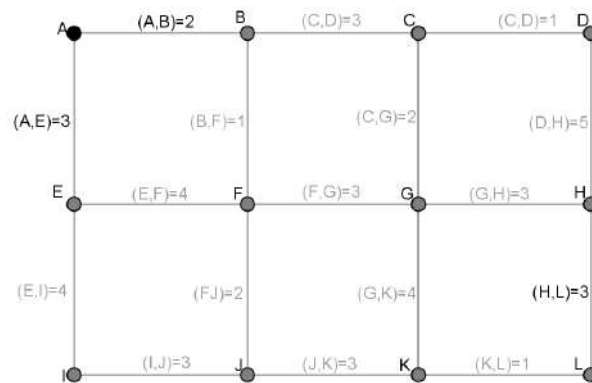


Figura 5.26: Grafo modelando os bairros de uma cidade.

- Para todo adjacente de *A*:

$$d[B] = (d[A] + w(A, B)) = 0 + 2 = 2; \text{ pai}[B] = A$$

$$d[E] = (d[A] + w(A, E)) = 0 + 3 = 3; \text{ pai}[E] = A$$

Vértices	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>G</i>	<i>H</i>	<i>I</i>	<i>J</i>	<i>K</i>	<i>L</i>
Distâncias	0	2	∞	∞	3	∞	∞	∞	∞	∞	∞	∞
pai	nulo	<i>A</i>	<i>A</i>
Finaliza	sim

Guarda menor distância: $d[B]=2$; $\text{pai}[B]=A$; B foi finalizado. (Figura 5.27)
 $S = \{A, B\}$ e $V = \{C, D, E, F, G, H, I, J, K, L\}$.

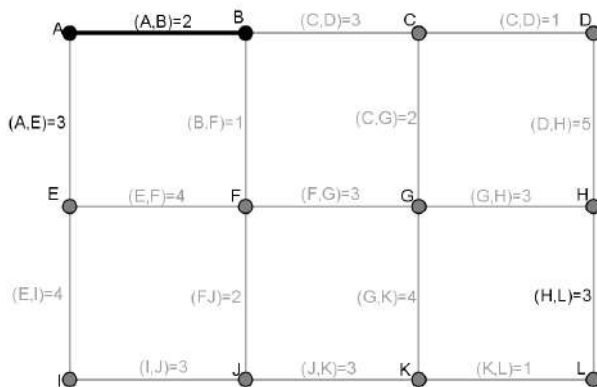


Figura 5.27: Grafo modelando os bairros de uma cidade.

- Para todo adjacente de B:

$$d[C] = (d[B] + w(B, C) = 2 + 3 = 5; \text{pai}[C]=B$$

$$d[F] = (d[B] + w(B, F) = 2 + 1 = 3; \text{pai}[F]=B$$

Vértices	A	B	C	D	E	F	G	H	I	J	K	L
Distâncias	0	2	5	∞	3	3	∞	∞	∞	∞	∞	∞
pai	nulo	A	B	...	A	B
Finaliza	sim	sim

Guarda menor distância: $d[E] = 3$; $\text{pai}[E]=A$; E foi finalizado. (Figura 5.28)
 $S = \{A, B, E\}$ e $V = \{C, D, F, G, H, I, J, K, L\}$.

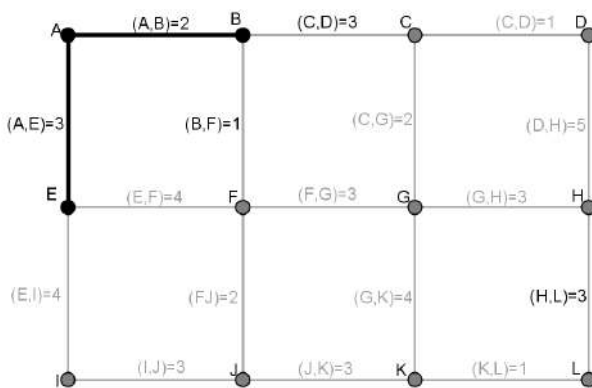


Figura 5.28: Grafo modelando os bairros de uma cidade.

- Para todo adjacente de E:

$$d[F] = (d[E] + w(E, F) = 3 + 4 = 7 > d[F] = 3; \text{pai}[F]=E \text{ (não atualiza).}$$

$$d[I] = (d[E] + w(E, I) = 3 + 4 = 7; \text{pai}[I]=E$$

Vértices	A	B	C	D	E	F	G	H	I	J	K	L
Distâncias	0	2	5	∞	3	3	∞	∞	7	∞	∞	∞
pai	nulo	A	B	...	A	B	E
Finaliza	sim	sim	sim

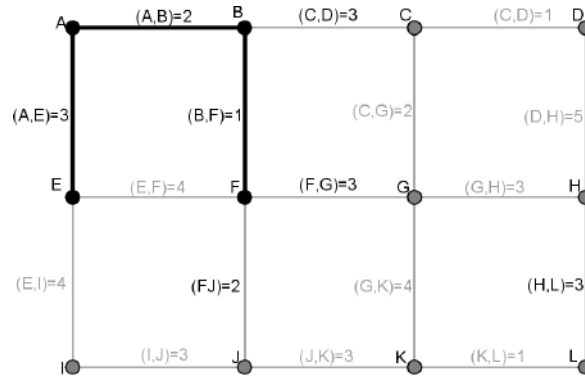


Figura 5.29: Grafo modelando os bairros de uma cidade.

Guarda menor distância: $d[F] = 3$; $\text{pai}[F] = B$; F foi finalizado. (Figura 5.29)
 $S = \{ A, B, E, F \}$ e $V = \{ C, D, G, H, I, J, K, L \}$.

- Para todo adjacente de F:

$$d[G] = (d[F] + w(F, G)) = 3 + 3 = 6; \text{pai}[G] = F$$

$$d[J] = (d[F] + w(F, J)) = 3 + 2 = 5; \text{pai}[J] = F$$

Vértices	A	B	C	D	E	F	G	H	I	J	K	L
Distâncias	0	2	5	∞	3	3	6	∞	7	5	∞	∞
pai	nulo	A	B	...	A	B	F	...	E	F
Finaliza	sim	sim	sim	sim

Guarda menor distância: $d[C] = 5$; $\text{pai}[C] = B$; C foi finalizado. (Figura 5.30)
 $S = \{ A, B, C, E, F \}$ e $V = \{ D, G, H, I, J, K, L \}$.

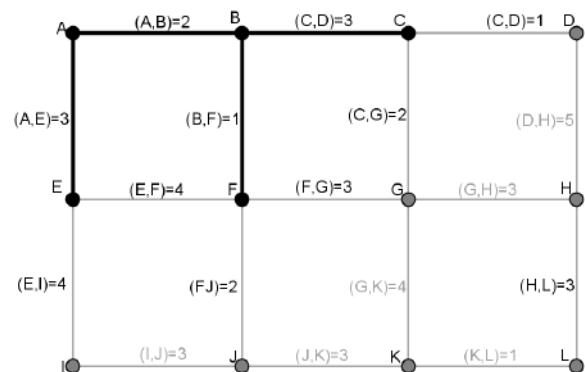


Figura 5.30: Grafo modelando os bairros de uma cidade.

- Para todo adjacente de C:

$$d[D] = (d[C] + w(C, D)) = 5 + 1 = 6; \text{ pai}[D] = C$$

$$d[G] = (d[C] + w(C, G)) = 5 + 2 = 7 > d[G] = 6; \text{ (n\~{a}o atualiza)}$$

Vértices	A	B	C	D	E	F	G	H	I	J	K	L
Distâncias	0	2	5	6	3	3	6	∞	7	5	∞	∞
pai	nulo	A	B	C	A	B	F	...	E	F
Finaliza	sim	sim	sim	...	sim	sim

Guarda menor distância: $d[J] = 5$; $\text{pai}[J] = F$; J foi finalizado. (Figura 5.31)

$S = \{A, B, C, E, F, J\}$ e $V = \{D, G, H, I, K, L\}$.

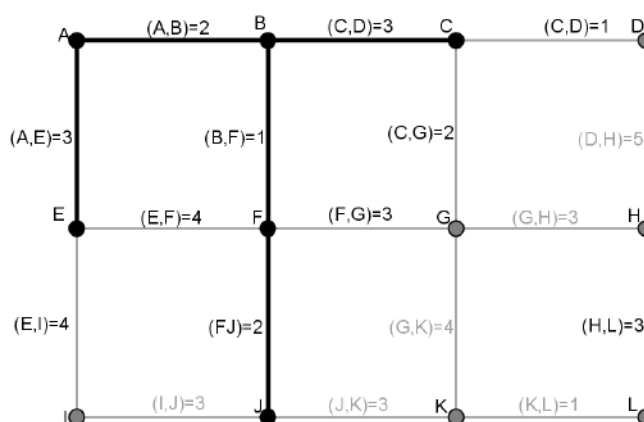


Figura 5.31: Grafo modelando os bairros de uma cidade.

- Para todo adjacente de J:

$$d[I] = (d[J] + w(J, I)) = 5 + 3 = 8 > d[I] = 7; \text{ (n\~{a}o atualiza)}$$

$$d[K] = (d[J] + w(J, K)) = 5 + 3 = 8; \text{ pai}[K] = J \text{ (n\~{a}o atualiza)}$$

Vértices	A	B	C	D	E	F	G	H	I	J	K	L
Distâncias	0	2	5	6	3	3	6	∞	7	5	8	∞
pai	nulo	A	B	C	A	B	F	...	E	F	J	...
Finaliza	sim	sim	sim	...	sim	sim	sim

Guarda menor distância: $d[D] = 5$; $\text{pai}[D] = F$; D foi finalizado. (Figura 5.32)

$S = \{A, B, C, E, D, F, J\}$ e $V = \{G, H, I, K, L\}$.

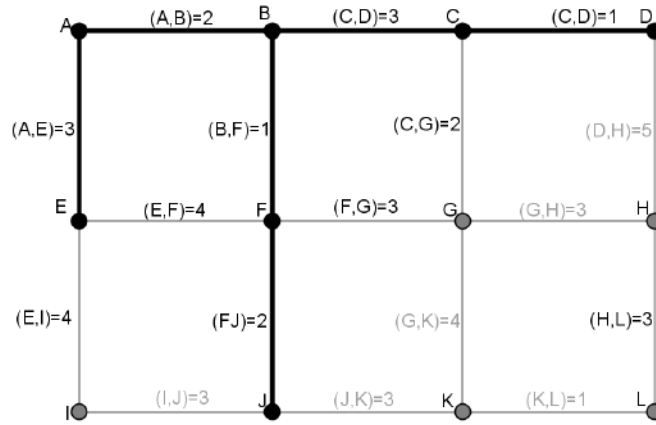


Figura 5.32: Grafo modelando os bairros de uma cidade.

- Para todo adjacente de D:

$$d[H] = (d[D] + w(D, H) = 5 + 5 = 10; \text{pai}[H]=D$$

Vértices	A	B	C	D	E	F	G	H	I	J	K	L
Distâncias	0	2	5	6	3	3	6	10	7	5	8	∞
pai	nulo	A	B	C	A	B	F	D	E	F	J	...
Finaliza	sim	sim	sim	sim	sim	sim	sim

Guarda menor distância: $d[G] = 6$; $\text{pai}[G]= F$; G foi finalizado. (Figura 5.33)

$S = \{ A, B, C, E, D, F, G, J \}$ e $V = \{ H, I, K, L \}$.

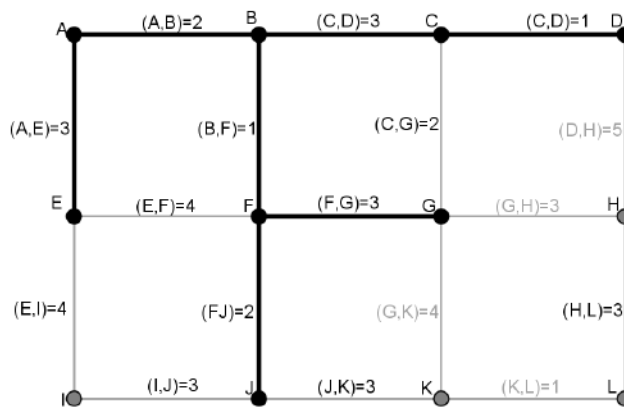


Figura 5.33: Grafo modelando os bairros de uma cidade.

- Para todo adjacente de G:

$$d[H] = (d[G] + w(G, H) = 6 + 3 = 9 < d[H] = 10; \text{pai}[H]= G \text{ (atualiza } d[H] \text{ e } \text{pai}[H]).$$

$$d[K] = (d[G] + w(G, K) = 6 + 4 = 10 > d[K] = 8; \text{ (mantêm } d[K])$$

Vértices	A	B	C	D	E	F	G	H	I	J	K	L
Distâncias	0	2	5	6	3	3	6	9	7	5	8	∞
pai	nulo	A	B	C	A	B	F	G	E	F	J	...
Finaliza	sim	sim	sim	sim	sim	sim	sim	sim

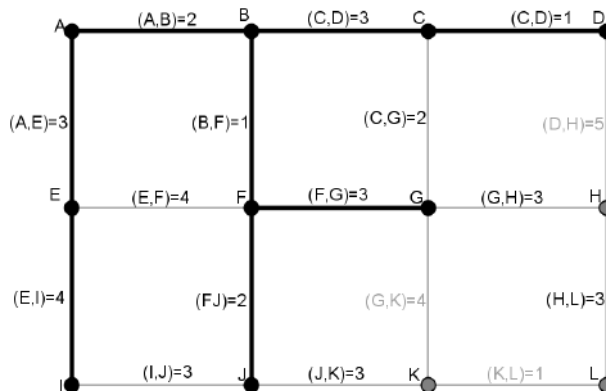


Figura 5.34: Grafo modelando os bairros de uma cidade.

Guarda menor distância: $d[I] = 7$; $\text{pai}[I] = E$; D foi finalizado. (Figura 5.34)
 $S = \{ A, B, C, E, D, F, G, I, J \}$ e $V = \{ H, K, L \}$.

- Para todo adjacente de I:

Não há o que atualizar, pois todos os vértices adjacentes ao vértice I, foram alcançados e finalizados.

Vértices	A	B	C	D	E	F	G	H	I	J	K	L
Distâncias	0	2	5	6	3	3	6	9	7	5	8	∞
pai	nulo	A	B	C	A	B	F	G	E	F	J	...
Finaliza	sim	sim	sim	sim	sim	sim	sim	...	sim	sim

Guarda menor distância: $d[K] = 8$; $\text{pai}[K] = J$; K foi finalizado. (Figura 5.35)
 $S = \{ A, B, C, E, D, F, I, J, K \}$ e $V = \{ H, L \}$.

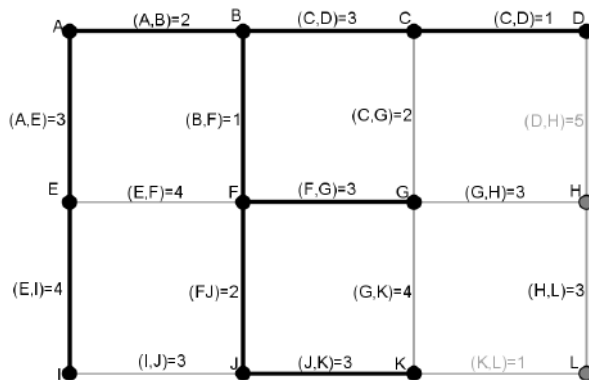


Figura 5.35: Grafo modelando os bairros de uma cidade.

- Para todo adjacente de K:

$$d[L] = (d[K] + w(K, L)) = 8 + 1 = 9; \text{ pai}[L] = K.$$

Vértices	A	B	C	D	E	F	G	H	I	J	K	L
Distâncias	0	2	5	6	3	3	6	9	7	5	8	9
pai	nulo	A	B	C	A	B	F	G	E	F	J	K
Finaliza	sim	sim	sim	sim	sim	sim	sim	...	sim	sim	sim	...

Guarda menor distância: $d[H] = 9$; $\text{pai}[H] = G$; H foi finalizado. (Figura 5.36)

$S = \{ A, B, C, E, D, F, G, H, I, J, K \}$ e $V = \{ L \}$.

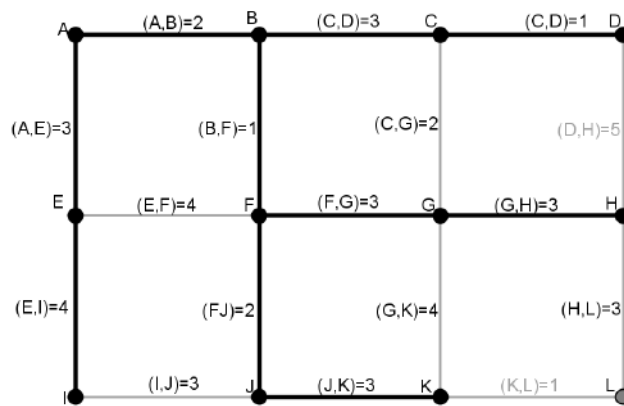


Figura 5.36: Grafo modelando os bairros de uma cidade.

- Para todo adjacente de H:

Não há o que atualizar, pois todos os vértices adjacentes ao vértice H, foram alcançados e finalizados. (Figura 5.37)

Vértices	A	B	C	D	E	F	G	H	I	J	K	L
Distâncias	0	2	5	6	3	3	6	9	7	5	8	9
pai	nulo	A	B	C	A	B	F	G	E	F	J	K
Finaliza	sim	sim	sim	sim	sim	sim	sim	sim	sim	sim	sim	...

Guarda menor distância: $d[L] = 9$; $\text{pai}[L] = K$; L foi finalizado.

$S = \{ A, B, C, E, D, F, G, H, I, J, K, L \}$ e $V = \{ \}$.

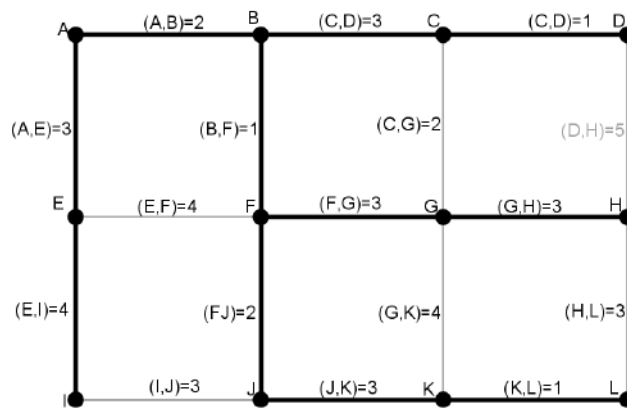


Figura 5.37: Grafo modelando os bairros de uma cidade.

Vértices	A	B	C	D	E	F	G	H	I	J	K	L
Distâncias	0	2	5	6	3	3	6	9	7	5	8	9
pai	nulo	A	B	C	A	B	F	G	E	F	J	K
Finaliza	sim	sim	sim	sim	sim	sim	sim	sim	sim	sim	sim	sim

O caminho mínimo, partindo do vértice A (origem) para qualquer outro vértice do grafo está destacado na Figura 5.38.

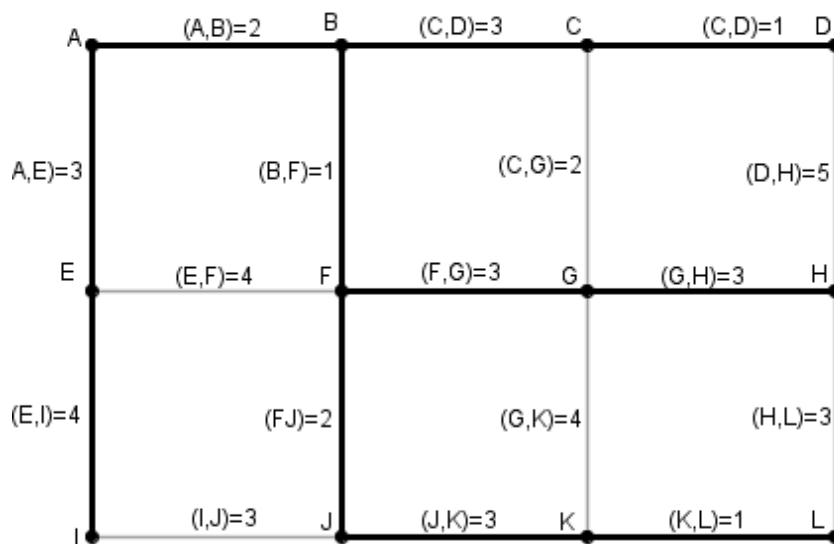


Figura 5.38: Caminho mínimo de A para todos os vértices.

5.2 Coloração

A coloração de grafos tem diversas aplicações a problemas que envolvem planejamento de cronogramas e distribuições. Nos problemas seguintes, serão mostradas algumas aplicações de coloração.

5.2.1 Problema 1

Como os exames finais, em uma universidade, podem ser marcados de modo que nenhum estudante tenha dois exames ao mesmo tempo.

Modelando a situação problema para um grafo, temos os vértices representando cursos e as arestas entre dois vértices, se existir, representando um aluno comum aos cursos. Cada posição na grade horária para um exame final é representada por uma cor diferente. Suponha que existam sete exames finais a serem marcados e que as disciplinas estejam numeradas de 1 a 7 e que os seguintes pares de disciplinas tenham estudantes em comum: 1 e 2, 1 e 3, 1 e 4, 1 e 7, 2 e 3, 2 e 4, 2 e 5, 2 e 7, 3 e 4, 3 e 6, 3 e 7, 4 e 5, 4 e 6, 5 e 6, 5 e 7, 6 e 7. O grafo da Figura 5.39 representa o grafo dos exames finais.

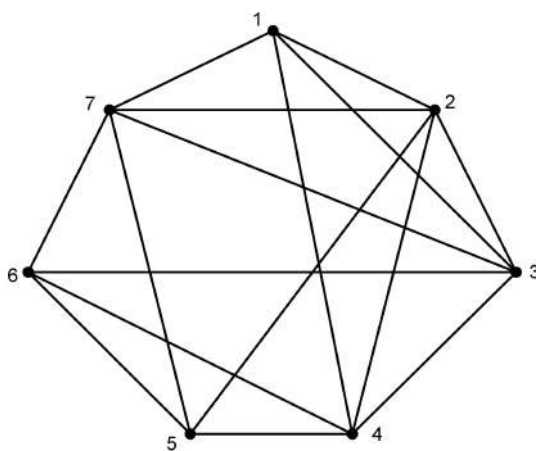


Figura 5.39: Horário dos exames finais.

Um horário consiste em uma coloração do grafo. Pode-se verificar que o número cromático deste grafo é 4 ou são necessárias quatro posições na grade horária. Logo, uma coloração do grafo e o horário associado estão mostrados na Figura 5.40.

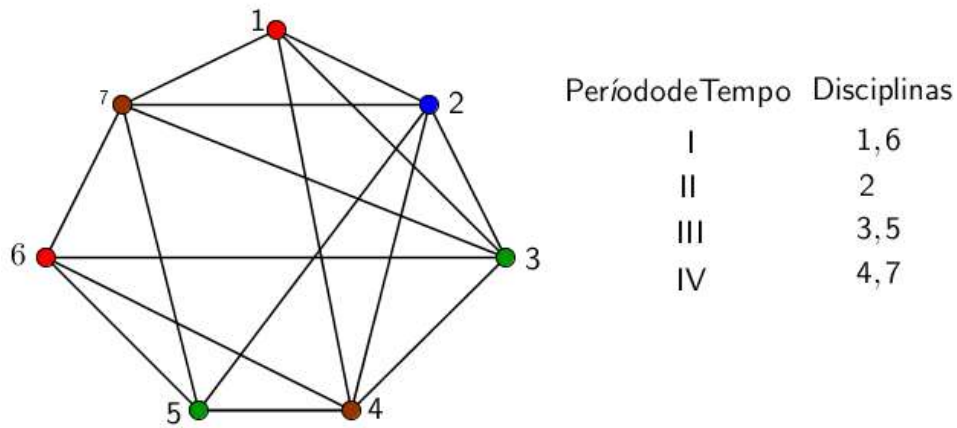


Figura 5.40: Coloração para marcar os exames finais.

5.2.2 Problema 2

Uma companhia industrial deseja armazenar sete produtos farmacêuticos $C_1, C_2, C_3, C_4, C_5, C_6$ e C_7 , mas alguns não pode ser armazenados juntos por motivos de segurança. A tabela seguinte mostra os produtos que não podem estar no mesmo local. Encontrar o número mínimo de locais necessários para armazenar estes produtos.

	C_1	C_2	C_3	C_4	C_5	C_6	C_7
C_1		x				x	x
C_2	x		x	x			
C_3		x		x	x		
C_4		x	x		x	x	
C_5			x	x		x	x
C_6	x			x	x		x
C_7	x				x	x	

Os produtos serão representados por vértices e não podem ser armazenados juntos por arestas no grafo. Faz-se a coloração com o mínimo de cores possíveis, que são quatro cores, e tem-se uma das soluções na figura 5.41.

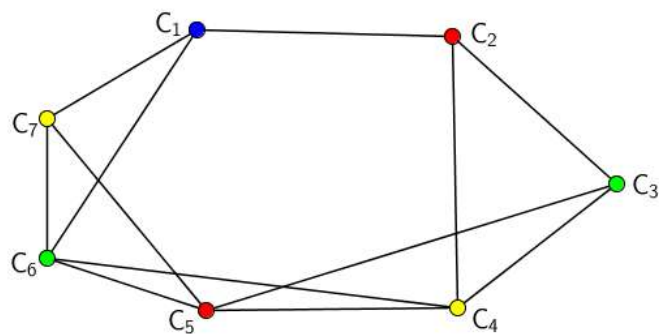


Figura 5.41: Coloração do grafo.

Uma opção de armazenar os produtos será: C_1 , C_2C_5 , C_3C_6 e C_4C_7 .

5.2.3 Problema 3

A herança do califa de Bagdad

Há muitos anos atrás, o califa de Bagdad tinha quatro filhos e tinha muito gosto por eles. Para cada filho mandou construir um palácio. O filho mais velho, Abdul, ficou com o terreno 1, Budal com o terreno 2, Cadaf com o terreno 3 e Dubal com o terreno 4, como mostrado no mapa da Figura 5.42.

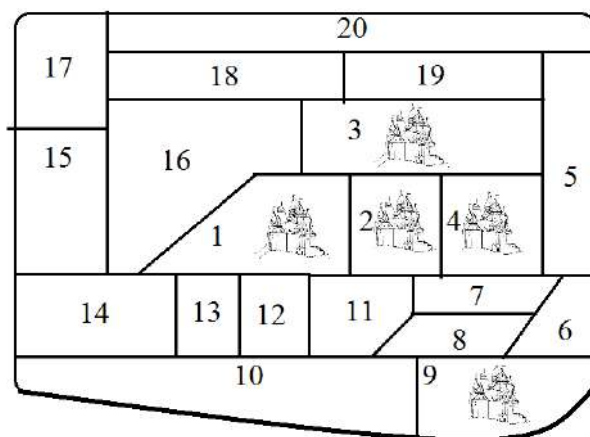


Figura 5.42: Mapa das terras do califa de Bagdad.

Antes de morrer, fez um testamento com indicações de como deveriam ser distribuídas suas ricas terras, num total de 20. Cada filho ficaria com o terreno onde tinha seu palácio. Abdul, herdaria também o terreno 9, onde ficava situado o palácio do califa. Os outros

terrenos seriam distribuídos de modo que, no final, cada filho ficasse com 5 terrenos. Mas impôs uma condição a cada um dos filhos: os seus 5 terrenos não poderiam ter fronteiras comuns. Por exemplo, Cadaf não podia ficar com o terreno 19. Como esses irmãos fariam a divisão das terras entre si?

A divisão das terras será feita por coloração das faixas de terras, como mostra a Figura 5.43

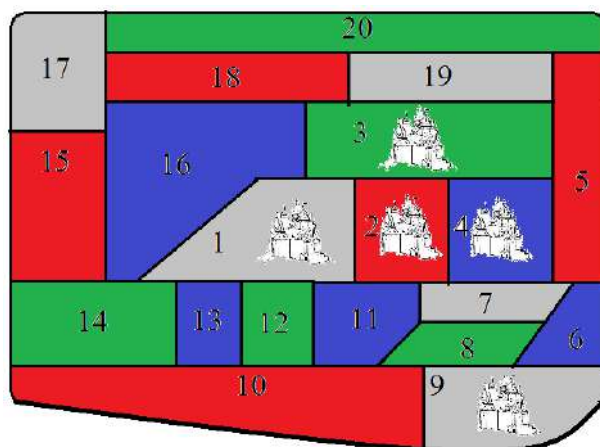


Figura 5.43: Coloração do mapa dos terrenos.

Considerações Finais

Neste trabalho elaboramos um texto introdutório relativo ao estudo dos grafos que pode ser utilizado por alunos do ensino médio/profissionalizante. Para tanto, mostramos os conceitos básicos da referida teoria e sua aplicação na modelagem e resoluções de problemas bem como sua representação no computador. Esses conceitos foram apresentados por meio de exemplos, de forma que mesmo um leigo no assunto consiga entender/identificar quando e como usar grafos na resolução de problemas.

Fizeram-se abordagens a problemas considerados clássicos na teoria dos grafos, como o problema das Pontes de Königsberg que prova que não é possível caminhar por todas as pontes uma única vez e voltar ao ponto inicial; o problema das Três Casas e Três Serviços que desempenha um papel importante na escolha do grafo a ser usado na construção de circuitos eletrônicos; o problema do Caminho Mínimo usando o Algoritmo de Dijkstra que nos ajuda a identificar a menor distância de um ponto de origem em relação aos demais pontos do grafo - sua aplicação prática pode ser observada no GPS; já no problema de Custo Mínimo usando o algoritmo de Kruskal, aprendemos a calcular a quantidade mínima de produtos a ser usada num serviço como ligações de redes de computadores e, por fim, apresentamos problemas de Coloração que nos auxiliam na resolução de problemas que envolvem planejamento e distribuição.

Segundo Piaget [9], os adolescentes se desprendem do real sem precisar se apoiar em fatos, ou seja, começam a pensar e a entender o abstrato e a deduzir mentalmente sobre várias hipóteses que se colocam. Entendemos, portanto, ser uma boa hora para aprenderem sobre grafos, pois são capazes de entender e encontrar soluções de problemas.

A modelagem de um problema em grafos faz exatamente isso, leva o aluno a um raciocínio mais abstrato, tornando-se uma técnica ou ferramenta capaz de ajudá-lo a tomar a decisão mais adequada para resolver problemas.

Atendendo à necessidade dos alunos de ensino secundário, espera-se que o leitor sintase incentivado a conhecer mais sobre o assunto, ao tempo em que deixamos espaço para futuras implementações ou contribuições para que a Teoria dos Grafos faça parte de nosso cotidiano. Como trabalho futuro, pretendemos implementá-lo em um problemas real que será discutido junto aos alunos.

Referências Bibliográficas

- [1] ALMEIDA, M. F. L. B. P. CELEMAN, S. *A Matemática escondida no Google*. Cadernos da RPM-Revista do Professor de Matemática, nº 80, 2013.
- [2] ASCENCIO, Ana Fernanda Gomes e ARAÚJO, Graziela Santos de. *Estrutura de dados: Algoritmos, análise da complexidade e implementações em java e C/C++*. São Paulo: Perarson Prentice Halt, 2010.
- [3] BOAVENTURA, Paulo. *Grafos: teoria, modelos e algoritmos*; São Paulo : Edgard Blucher Ltda.
- [4] CORMEN, Thomas H.; LEIRSON, Charles E; RIVEST, Ronald L; STEIN, Clifford. *Algoritmos: Teoria e Prática*. Tradução da segunda edição [americana] Vandenberg D. de Souza. Rio de Janeiro: Campus, 2002.
- [5] GERSTING, J. L. *Fundamentos Matemáticos para a Ciência da Computação*. Tradução da 3 ed. Rio de Janeiro: LTC - Livros Técnicos e Científicos Editora S.A, 1995.
- [6] JURKIEWICZ, Samuel. *Grafos - Uma introdução*. São Paulo : OBMEP, 2009.
- [7] LIMA, L.; SIMÕES, M.; SELAS, L. *Seminário de Matemática*. Mestrado em Supervisão Pedagógica em Ensino de Matemática, 1999/2000.
- [8] LIPSCHUTZ, S.; LIPSON, M. *Teoria e problemas de matemática discreta*; Tradução: Heloisa Bauzer Medeiros. - 2.ed. - Porto Alegre: Bookman, 2004. (Coleção Schaum)
- [9] PIAGET, J. *O nascimento da inteligência na criança*. Rio de Janeiro: Zahar, 1970.
- [10] REVISTA DO PROFESSOR DE MATEMÁTICA, nº 80, ano 30/2013
- [11] ROSEN, Kenneth H. *Matemática Discreta e suas Aplicações*; [Tradução João Giudice]. São Paulo : McGraw-Hill, 2009.
- [12] ZIVIANI, Nivio. *Projeto de Algoritmos: com Implementações em Pascal e C*. 3 ed. São Paulo: Cengage Learning, 2012.