**Teoria dos Grafos**

Amanda Carolina Ambrizzi Ramin; 22.00721-0

*Resumo*. Este artigo explora a teoria dos grafos, abordando seus conceitos fundamentais e destacando sua relevância em problemas complexos de computação. São discutidos aspectos como a planaridade dos grafos e a implementação de algoritmos essenciais, como a busca em profundidade (DFS) para encontrar vértices, e os algoritmos de Prim, Kruskal e Dijkstra para problemas de otimização. Também está incluso exemplos de aplicação afim de compreender a relevância dos grafos na resolução de problemas.

Palavras Chaves: Grafos, DFS, Dijkstra, PRIM, Kruskal

*Abstract*. This paper explores graph theory, covering its fundamental concepts and highlighting its relevance in complex computational problems. It discusses aspects such as graph planarity and the implementation of essential algorithms, including Depth-First Search (DFS) for finding vertices, and the Prim, Kruskal, and Dijkstra algorithms for optimization problems. Practical application examples are also included to demonstrate the importance of graphs in problem-solving.

Key Words: Graph, DFS, Dijkstra, PRIM, Kruskal

**Introdução**

A Teoria dos Grafos surge no século XVIII na matemática, porém sua definição formal surge apenas no século XX. A primeira referência a grafo foi na cidade de Königsberg com o Problema das Pontes de Königsberg. A cidade era cortada pelo rio Pregel, que tinha duas ilhas. Para facilitar o transporte e passagem de pessoas foram construídas sete pontes nessa cidade como na figura 1.

|  |
| --- |
| Figura 1 – Mapa de Königsberg |
| Diagrama, Desenho técnico  Descrição gerada automaticamente |

Com o tempo, os habitantes da cidade passaram a se perguntar se havia uma forma de sair de um lugar e voltar para ele passando por todas as pontes sem repeti-las. Então, o matemático Leonhard Euler monta um diagrama da cidade associando cada cidade a um ponto (vértice) e cada ponte a uma reta (aresta), formando o que foi chamado de grafo representado na figura 2.

|  |
| --- |
| Figura 2 – Representação da cidade em forma de grafo |
|  |

Euler percebe que há 2 pontos com uma quantidade ímpar de retas, tornando impossível que uma pessoa saia de um desses pontos e retorne a ele sem repetir nenhuma reta. A ideia seria que é necessário um caminho para “sair” e um para “entrar”, porém tendo três caminhos seria preciso repetir algum deles na volta. Um dos vértices com três arestas deve ser o início enquanto o outro é o final. E assim surgiu o primeiro grafo da história.

O conceito de grafo permitiu a modelagem de situações como redes de computadores, rotas de transporte, redes sociais, árvores genealógicas, química orgânica etc. Também pode ser associado a relações binárias como de parentesco.

**Desenvolvimento**

*Definição de Grafo*

Um grafo pode ser definido informalmente como um conjunto de vários pontos, chamados de vértices, e ligações entre esses pontos, chamados de arestas.

Formalmente, tem-se que um grafo simples G consiste em um conjunto finito e não vazio V(G) de objetos chamados vértices, juntamente com um conjunto E(G) de pares não ordenados de vértices, no qual seus elementos são chamados de arestas. Pode-se representá-lo por G = (V; E).

É importante definir mais alguns conceitos antes mostrar aplicações e implementações de algoritmo:

Arestas Paralelas: se duas ou mais arestas têm os mesmo vértices-extremidade, então essas arestas são paralelas.

Vértice Isolado: um vértice é isolado se não é extremidade de qualquer aresta.

Vértices Vizinhos: dois vértices unidos por uma aresta são chamados de adjacentes ou vizinhos.

Arestas Adjacentes: duas arestas são adjacentes se elas têm vértice comum.

Conjunto Vizinhança: denotado por N(v), é o conjunto de todos os vizinhos de um vértice fixo v.

Grau de um vértice: denotado por d(v), é o número de arestas que incidente em um vértice. Sendo assim, grau 0 é um vértice isolado e grau 1 é um vértice final.

Vértice par: quando o grau do vértice é par.

Vértice ímpar: quando o grau do vértice é ímpar.

Sequência de graus: são todos os graus do grafo escritos em ordem crescente podendo haver repetições.

Grafo simples: é um grafo que não tem loops nem arestas paralelas. O produto entre a quantidade de vértices e o grau deve ser par. Além disso, a soma dos graus de todos os vértices é igual ao dobro do número de aresta: e a quantidade de vértices ímpares é sempre par.

Grafo completo: denotado por , é um grafo que possui n vértices e tem arestas.

Grafos Isomorfos: dois grafos são isomorfos se eles possuírem a mesma quantidade de vértices e arestas, além de quantidade iguais de vértices de um mesmo grau.

Subgrafo e Supergrafo: sejam dois grafos G1 e G2, diz-se que G2 é subgrafo de de G1 se

Grafo Bipartido: um grafo é dito como bipartido se seus vértices puderem ser separados em dois subconjuntos não vazios que não se interceptam, isso é, não deve haver arestas entre vértices do mesmo subconjunto.

Grafo Regular: é um grafo onde cada vértice tem o mesmo número de adjacência, ou seja, o mesmo grau. Um grafo regular com vértices de grau k é chamado de grafa k-regular.

Passeio: um passeio ocorre quando é possível chegar em um vértice a partir de outro seguindo uma sequência finita de vértices e arestas.

Grafo Conexo: um grafo é dito como conexo se existir um asseio entre dois quaisquer vértices.

Trilha: é um passeio em que não se repetem arestas. Toda trilha é um passeio, mas nem todo passeio é uma trilha. Uma trilha na qual o vértice final é igual ao inicial é chamado de circuito.

Caminho: um caminho é um passeio no qual não se repete vértices. Todo caminho é um passeio, mas nem todo passeio é um caminho. Um caminho no qual o vértice final é igual ao inicial é chamado de ciclo.

Grafo Euleriano: um grafo é dito como euleriano se possuir um passeio fechado que usa cada aresta do vértice uma única vez. Uma condição necessária e suficiente para o grafo conexo seja euleriano é que todos os vértices tenham grau par.

Grafo Hamiltoniano: um grafo é dito como hamiltoniano se existir um ciclo que passa por todos os vértices do grafo. Não há teoremas necessários para ser um hamiltoniano, mas há condições suficientes.

Planaridade de Grafos;

Implementação de Busca em Profundidade de Grafos;

Implementação do Algoritmo de PRIM;

Implementação do Algoritmo de Kruskal;

Implementação do Algoritmo de Dijkstra.

**Conclusão**

**Referências**

HOLANDA, B. **Teoria dos GrafosOBM**. [s.l: s.n.]. Disponível em: <https://www.obm.org.br/content/uploads/2017/01/Nivel1\_grafos\_bruno.pdf>. Acesso em: nov. 2024.

SOARES DE MELO, G. **Introdução à Teoria dos GrafosRepositório Institucional da UFPB**. [s.l: s.n.]. Disponível em: <https://repositorio.ufpb.br/jspui/bitstream/tede/7549/5/arquivototal.pdf>. Acesso em: nov. 2024.