

Aula 01 – Introdução e Motivação ao Estudo de Teoria dos Grafos

Notas de Aula de Teoria dos Grafos

Prof^a: Patrícia D. L. Machado

UFCG – Unidade Acadêmica de Sistemas e Computação

Muitas situações no mundo real podem ser descritas por meio de um diagrama formado por pontos e linhas. O objetivo é observar se dois ou mais pontos estão conectados e o padrão desta conexão. Por exemplo, as redes sociais podem representar indivíduos e suas relações através de um diagrama deste tipo (Figura 1). Neste caso, indivíduos, representados por pontos, estão ligados a fim de estabelecer uma relação de amizade, representada por uma linha. Ao analisar este diagrama, por exemplo, podemos observar padrões de relacionamento em um grupo, bem como identificar oportunidades para disseminação de informações, considerando os caminhos existentes entre indivíduos que podem ou não estar ligados diretamente.

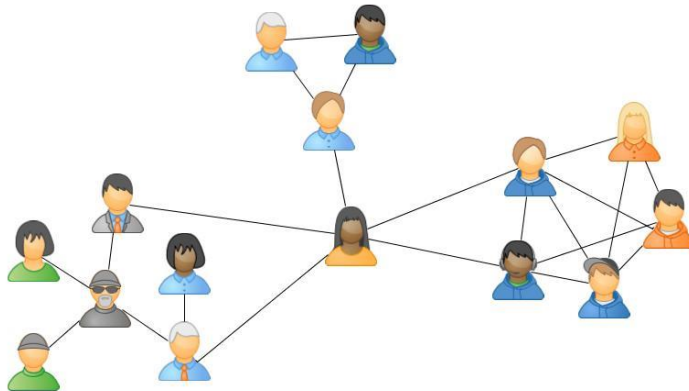


Figura 1. Ilustração gráfica de uma rede social.

Grafos são utilizados na prática para modelar diversos conceitos presentes em nossas vidas. A web, por exemplo, é um grafo virtual gigantesco onde cada página é um vértice e as arestas são links que levam de uma página para outra. Aplicações com GPS fazem uso de grafos para representar pontos de interesse e calcular rotas de um ponto a outro ponto.

Hoje em dia, praticamente toda e qualquer rede de informações e conhecimento humano é representada através de grafos. Algoritmos sobre grafos são aplicados a fim de encontrar soluções para os diferentes problemas em diferentes representações. Neste sentido, o estudo da Teoria dos Grafos nos ajuda a conhecer melhor esta poderosa ferramenta, incluindo modelos, propriedades e algoritmos diversos que podem ser aplicados para resolver problemas práticos, não só em Ciência da Computação. O estudo de grafos vai muito além de conceitos básicos de programação deste conceito (alvo das disciplinas de programação e estrutura de dados), mas se remete, principalmente, a representação e solução de problemas. Este é o objetivo central desta disciplina.

Problemas Clássicos

As 7 pontes de *Konigsberg* ligam as ilhas A e B e as margens C e D (Figura 2). **Existe uma trilha que passe por cada uma das pontes apenas uma vez?**

Este problema pode ser representado através de um grafo onde os vértices representam as porções de terra e as arestas representam as pontes.

Como vamos aprender em aulas posteriores, não existe esta trilha.

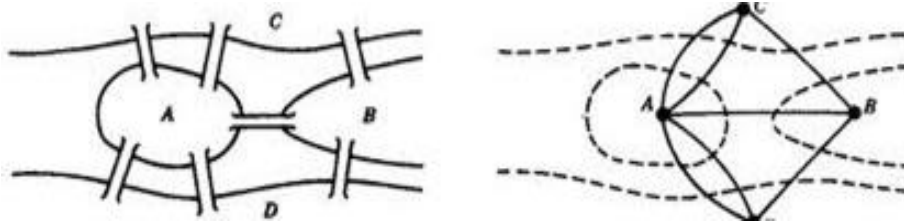


Figura 2: Pontes de Konigsberg.

Um outro problema clássico que pode ser representado através de grafos é o problema das 4 cores. **Quantas cores são necessárias para colorir um mapa de forma que regiões vizinhas possuam cores diferentes?**

É sempre possível colorir qualquer figura planar com apenas 4 cores (Figura 3). Podemos representar problemas semelhantes através da coloração de vértices onde vértices ligados diretamente não podem receber a mesma cor.

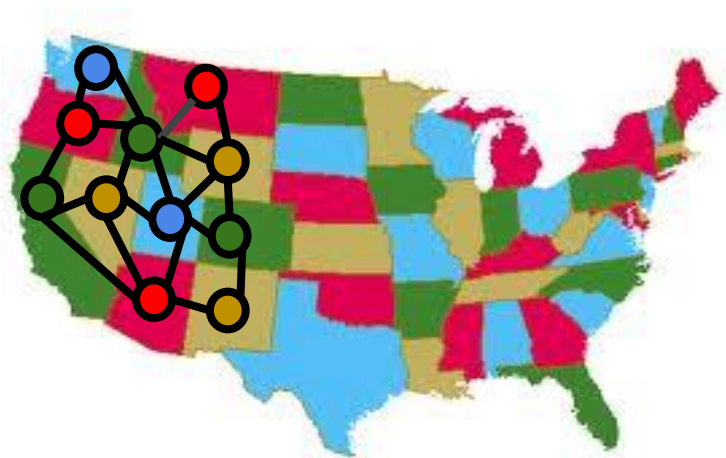


Figura 3: Coloração de mapas planares.

Problemas Comuns na Engenharia de Software

Grafos são utilizados para modelar programas a fim de tornar possível realizar diferentes análises sobre os mesmos tal como a cobertura de testes. Neste caso, programas são usualmente representados através de Grafos de Fluxo de Controle. Na Figura 4, o trecho de código é representado por um grafo onde os vértices representam as linhas de código que são ligadas através de arcos os quais representam o direcionamento do fluxo de execução.

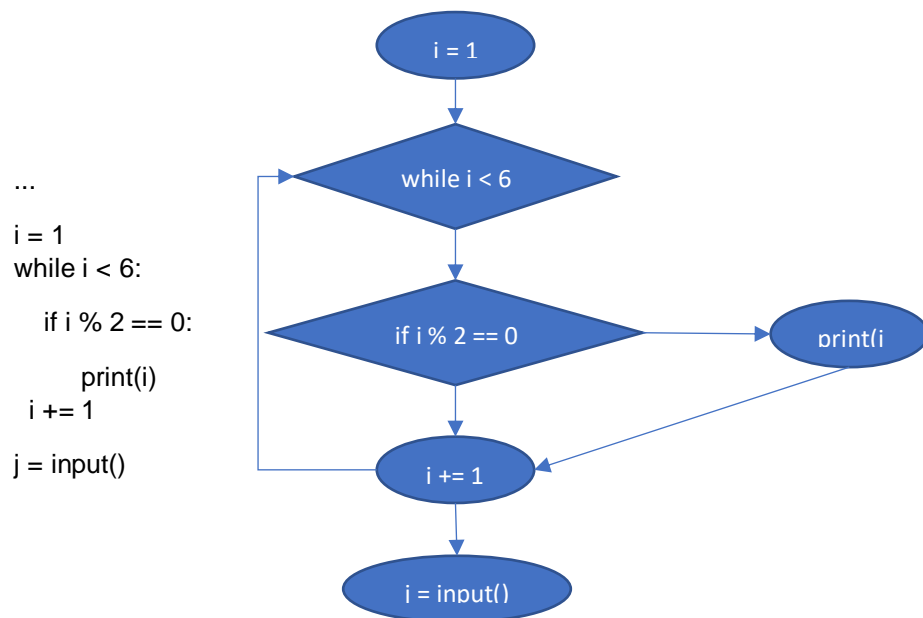


Figura 4: Grafo de fluxo de controle.

Como uma meta de avaliação da qualidade de uma suite de testes, espera-se que casos de teste percorram de forma sistemática os diferentes caminhos e elementos do grafo a fim de garantir que as linhas de código foram efetivamente testadas.

A partir da análise do grafo de fluxo de controle, podemos investigar os seguintes problemas:

- **Existencial:** Existe um conjunto finito de casos de teste que executa todos os comandos de um programa pelo menos uma vez?
- **Construção:** Como construir este conjunto?
- **Enumeração:** Quantos conjuntos existem? Podem ser listados? São finitos?
- **Otimização:** Qual o menor conjunto?

Outro problema importante que pode ser tratado através do uso de grafos é o problema da modificação de um software. ***Que classes devem ser testadas após a realização de uma modificação?***

Para tal precisamos identificar que partes do software são afetadas pela modificação. Isto pode ser feito observando a relação de dependência entre as classes. Uma classe *A* é dependente de uma classe *B* se *A* usa *B*. Isto pode ocorrer, por exemplo, quando *A* possui um objeto da classe *B* como parâmetro em um ou mais métodos ou *A* possui uma variável local da classe *B*.

Através da análise do **Grafo de Dependências** podemos investigar os custos e o impacto de modificações em um software. Como exemplo, considere o grafo de dependência de classes apresentado na Figura 5, onde os arcos representam a relação de dependência entre duas classes. Neste exemplo, observamos que todas as classes são dependentes da classe **Card**. Em particular, a classe **Address** é dependente de **Card** através de **Ship**. Assim, modificações em **Card** devem ser validadas através do teste de todas as classes. O mesmo não acontece com a classe **View**: modificações nesta classe requerem teste apenas nela. Observe também que **Card**, **Ship** e **Address** possuem uma relação de *dependência circular*. Portanto, alterações nestas classes podem levar a ciclos repetitivos de correções.

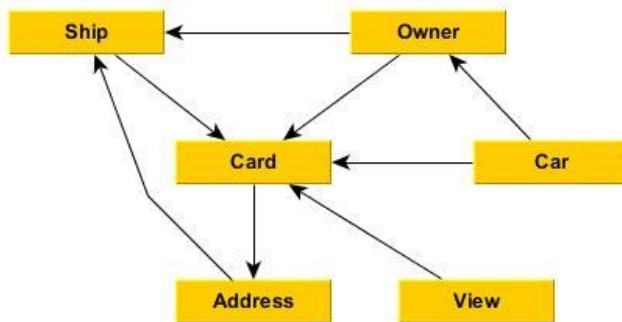


Figura 5: Grafo de Dependência de Classes

Grafos também podem ser utilizados para representar situações onde queremos identificar grupos de elementos fortemente relacionados. Por exemplo, podemos usar um grafo de dependência de classes para particionar o esforço de evolução de um software entre diferentes equipes, minimizando a necessidade de interação entre estas equipes. Cada equipe recebe um grupo de classes que possuem forte dependência entre si, mas pouca dependência direta com as demais como podemos observar na Figura 6.

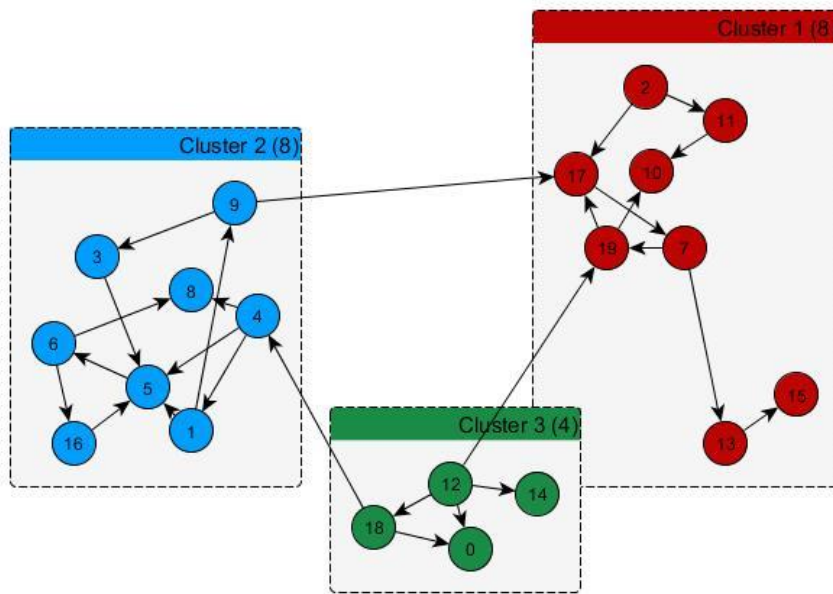


Figura 6: Grupos de classes relacionadas por dependência.

Histórico

Não é possível determinar exatamente quando estudos e usos de grafos tiveram início uma vez que se trata de um conceito intuitivo que pode ter sido usado desde épocas remotas da civilização. Abaixo listamos alguns pontos e fatos marcantes na evolução de conceitos e formalização do que conhecemos hoje como Teoria dos Grafos.

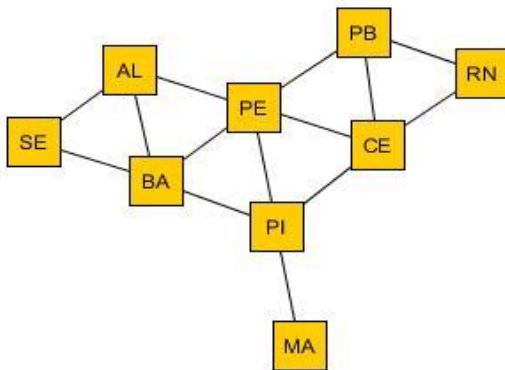
- Königsberg (Euler 1735)
- Grafo Hamiltoniano (Hamilton 1805-65)
- Conceito de Árvore (Kirchhoff 1824-87)
- Grafos Planares (Mobius/Kuratowski/Dudeney 1840/1913)
- Conjectura das 4-cores (1852)
- Busca por algoritmos eficientes
- Larga aplicabilidade: Matemática, Computação, Comunicação e Otimização Combinatória
- Soluções para algoritmos intratáveis

Exercícios: Exemplos de Modelagem de Problemas Usando Grafos

Rotas de Transporte

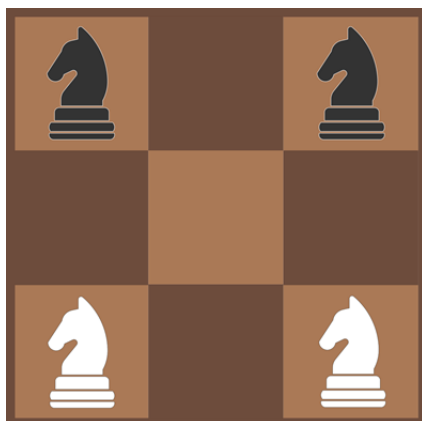
Os governos estaduais cobram impostos para a entrada de mercadorias vindas de outros estados. Para minimizar os custos de transportes, é importante escolher rotas de transporte de um estado a outro que atravesse o menor número possível de estados. Como podemos resolver este problema usando grafos? Construímos um grafo onde os estados são vértices e as arestas representam a existência de fronteira entre 2 estados.

Para encontrar a melhor rota, calculamos o menor caminho no grafo entre dois vértices.



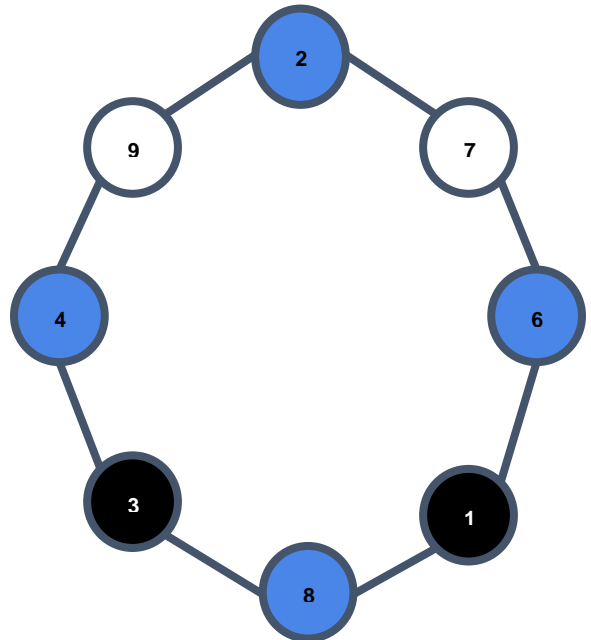
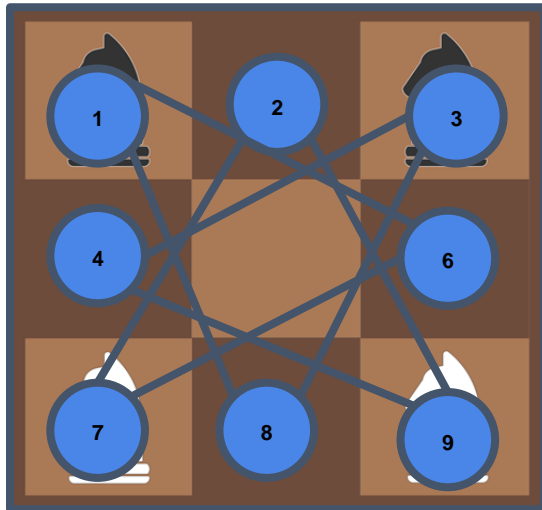
Guarani's Puzzle

Existem 4 cavalos em um tabuleiro de xadrez 3x3: dois brancos nos cantos inferiores e dois pretos nos cantos superiores. Qual é o número mínimo de movimentos necessários para que os cavalos brancos ocupem as posições dos cavalos pretos e vice-versa?



Criamos um grafo onde cada célula do tabuleiro é representada por um vértice e adicionamos uma aresta entre dois vértices se o cavalo pode se mover entre as células correspondentes. Representando graficamente este mesmo grafo de outra forma, observamos que é na verdade um círculo. Na

representação abaixo, os vértices de cor preta representam as células onde os cavalos pretos se encontram, enquanto os vértices com de cor branca representam as células onde os cavalos brancos se encontram.



Através deste grafo, descobrimos que mover os cavalos brancos para as células dos cavalos pretos e vice-versa consiste em deslocarmos os mesmos no grafo seguindo a ordem circular, um de cada vez. Ou seja, determinar o número mínimo de movimentos no tabuleiro resume-se a contar o número mínimo de deslocamentos dos cavalos no círculo, considerando que apenas um cavalo se move por vez.

Referências

- https://en.wikipedia.org/wiki/Graph_theory
- **Introdução a Algoritmos:**
[Aula no Youtube:](http://www.youtube.com/watch?v=3hv5_hWPleo) http://www.youtube.com/watch?v=3hv5_hWPleo