SCC 503 - ALGORITMOS E ESTRUTURAS DE DADOS II

Grafos - Conceitos

Prof.: Leonardo Tórtoro Pereira/Maria Cristina

*Material baseado em aulas dos professores: Elaine Parros Machado de Souza, Gustavo Batista, Robson Cordeiro, Moacir Ponti Jr., Maria Cristina Oliveira e Cristina Ciferri.

Por que aprender Grafos?

- ➔ Modelar problemas que envolvam conjuntos de `entidades´ e relacionamentos entre pares dessas entidades
  - ➔ Relacionamentos ⇔ conexões
- ➔ Modelar relações de conectivade e como elas afetam processos em diversos sistemas:
  - ◆ Físicos, biológicos, sociais e de informação

Por que aprender Grafos?

➔ Redes de

- Comunicação (redes sociais, redes de computadores, redes elétricas, ...)
- Organização de dados (redes neurais, redes complexas)
- Dispositivos computacionais (circuitos
- Fluxo de Computação

Por que aprender Grafos?

- ➔ Sistemas de recomendação (Amazon, Netflix, etc.)
- ➔ Otimização de caminhos e rotas (Google Maps, Uber, etc.)
  - ➔ Malha urbana, rodoviária
- ➔ Malha aérea
- ➔ Internet
- ➔ Modelar sintaxe de linguagem natural

Por que aprender Grafos?

- ➔ Estudo de átomos e moléculas
- ➔ Medir prestígio (de páginas, pessoas...)
- ➔ Espalhamento de rumor
- ➔ Amizades entre pessoas
- ➔ Padrões de reprodução de animais
- ➔ Espalhamento de **doenças**
- ➔ Relação entre genes
- ➔ ...

Programa Jupiterweb

- ➔ Conceitos fundamentais e aplicações computacionais de grafos.
- ➔ Estruturas de dados para representação de grafos: lista de arestas, lista de adjacências e matriz de adjacências.
- ➔ Percursos em grafos e aplicações: busca em largura e profundidade.
- ➔ Algoritmos clássicos sobre grafos e aplicações, tais como caminhos mínimos, árvores geradoras mínimas e ordenação topológica.

Definições

Definições

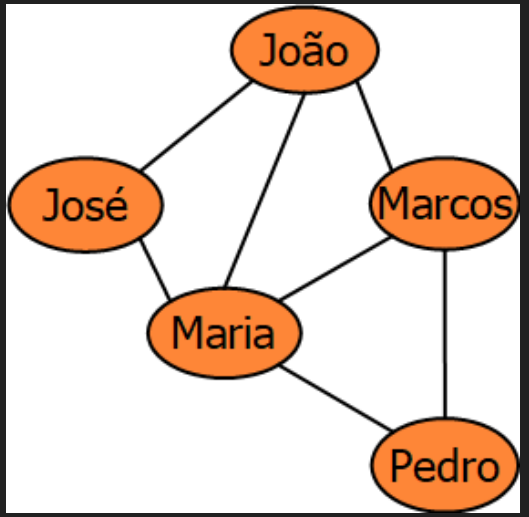
- ➔ Grafos
  - ◆ Estruturas abstratas que modelam entidades e a relação (conexão) entre elas.
- ➔ Teoria dos Grafos
  - ◆ Área de matemática combinatória
  - ◆ Resolução de problemas em computação

Definições

➔ O que é um Grafo?

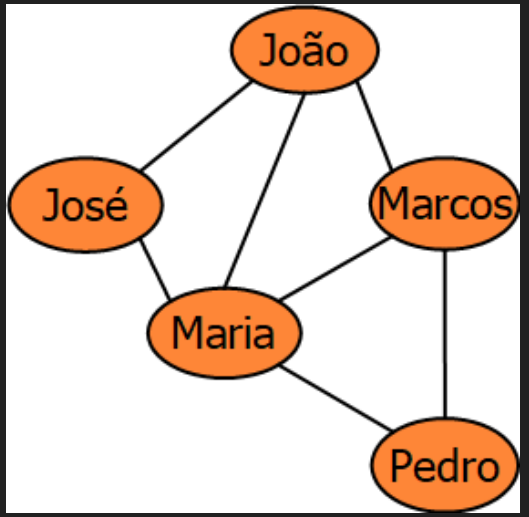
- Uma estrutura (tipo de dado) definida por conjuntos de nós (ou vértices) e as arestas que ligam esses nós
- Grafo **G** definido como um par (V, A)
  - V: Conjunto de nós, ou vértices
  - A: Conjunto de pares de vértices, chamados de arestas, ligações, ou arcos

Definições



- ➔ Exemplo de Grafo
  - ◆ Rede social de amizade
    - Cada vértice representa uma pessoa
    - Cada aresta representa uma relação de amizade entre pessoas

Definições



➔ Exemplo de Grafo

G= (V, A)

V = {José, João, Marcos, Pedro, Maria}

A = {(João,José), (João,Maria), (João,Marcos), (José,Maria),(Maria,Marcos), (Maria,Pedro), Marcos,Pedro)}

|V| = 5 (a `ordem´ do grafo G), |A| = 7

Terminologia

Terminologia

- **Vértices adjacentes**: dois vértices u e v são ditos adjacentes (ou vizinhos) se são os extremos de uma mesma aresta a = (u,v)

- **Arestas adjacentes**: duas arestas são ditas adjacentes (ou vizinhas) se possuem um mesmo vértice extremo

Definições

➔ Grafo Vazio: $G=(\varnothing, \varnothing)$

➔ Grafo Trivial

$V = \{v_1\}$

$A = \varnothing$

$|V| = 1$ e $|A| = 0$

História: Leonhard Paul Euler (1707-1783)

➔ Considerado o "pai" da teoria dos grafos

- Matemático e físico
- Viveu na Rússia e na Alemanha
- Ficou parcialmente cego aos 28 anos, e totalmente cego nas 2 últimas décadas de vida
- Motivação: resolver o problema das 7 pontes na cidade de Konigsberg/Alemanha em 1735

Quadro de Johann Georg Brucker

História: o problema

- Na antiga cidade prussiana de Konigsberg (atual Kaliningrado, na Rússia) havia 7 pontes que conectavam 2 ilhas.

- Hoje somente três pontes daquela época ainda existem (duas da época de Euler, e uma que foi reconstruída). Duas foram destruídas durante a segunda guerra, outras duas foram destruídas para formar uma única via moderna.

História: o problema

➔ A questão que intrigava os moradores da cidade era essa: é possível fazer um caminho por todas as pontes passando uma e somente uma vez em cada uma das delas?

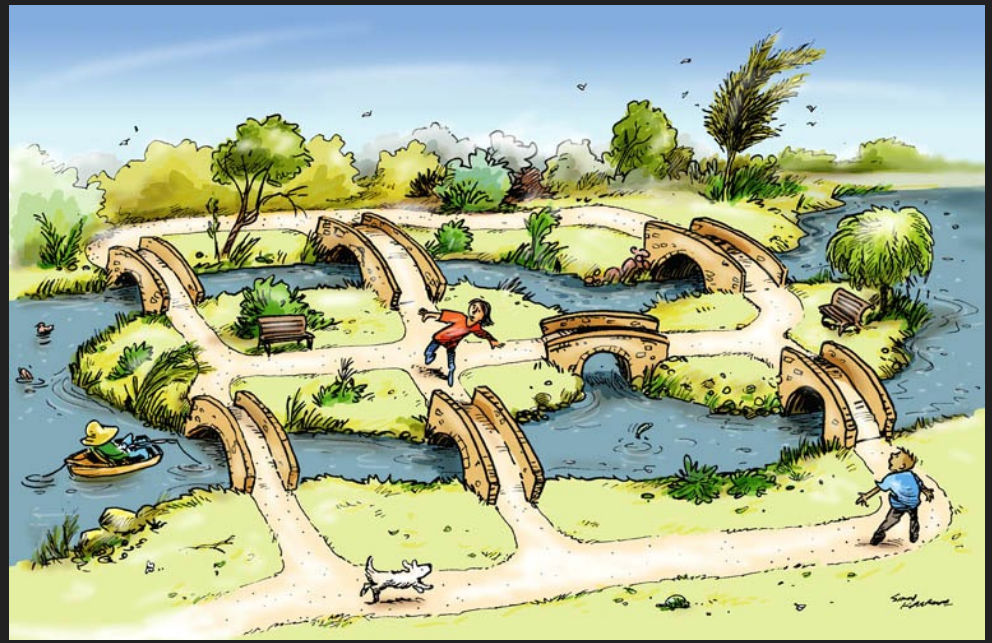
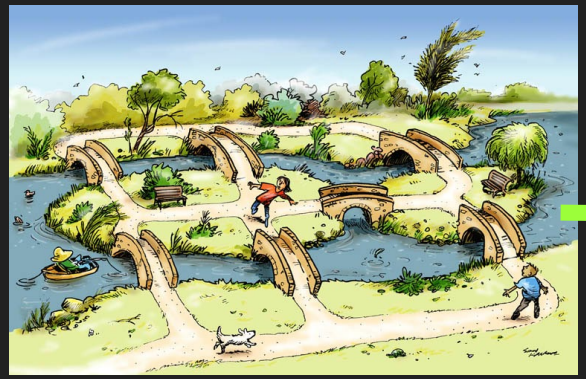


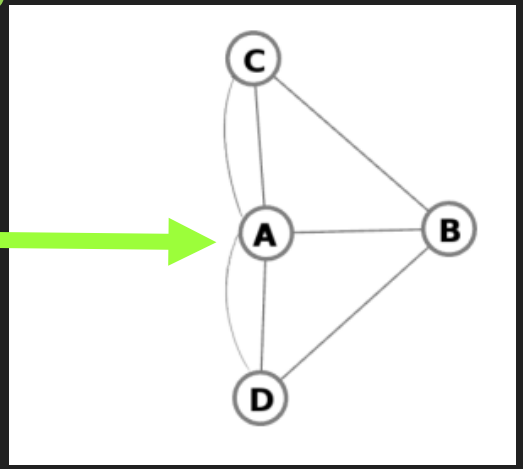
Imagem de Simon Kneebone

História: a representação

➔ Representação esquemática do problema com “vértices” e “arestas”

- Nasce o que é considerado oficialmente por muitos como o primeiro grafo





$G = (V, \mathcal{A})$
$V = \{A, B, C, D\}$
$\mathcal{A} =$
$\{(A,C),(A,C),(A,D),(A,D),$
$(C,B),(A,B),(B,D)\}$

Definições

Um grafo ponderado (valorado/com pesos) G(V,A) consiste de um conjunto finito, não vazio, de vértices (V) conectados por um conjunto de arestas com pesos (A)

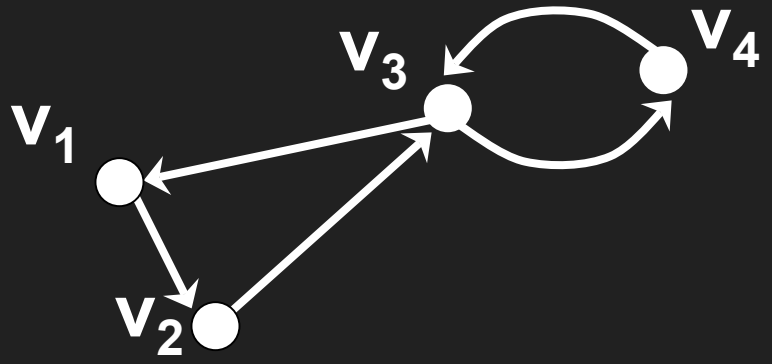
Os elementos de A são triplas distintas da forma (v,w,valor), em que v e w são vértices pertencentes a V e valor é um número inteiro/real

Definições

Um grafo orientado (dirigido/dígrafo) G(V,A) consiste de um conjunto finito, não vazio, de vértices (V) conectados por um conjunto de arestas dadas por pares ordenados de vertices distintos (A)

Os elementos de A são duplas distintas da forma (v,w), indicando que existe uma relação orientada de v para w

Grafo orientado - exemplo



$V(G) = \{v_1, v_2, v_3, v_4\}$

$A(G) = \{(v_1, v_2);\ (v_3, v_1);\ (v_2, v_3);\ (v_3, v_4);\ (v_4, v_3)\}$

Representação

Representação

- ➔ Para implementar um Tipo Abstrato de Dado (TAD) Grafo geralmente usamos uma dessas 2 soluções
  - ◆ Listas de Adjacência
  - ◆ Matriz de Adjacência
- ➔ A escolha entre elas depende das características do grafo e dos algoritmos a serem utilizados na resolução do problema
  - ◆ A escolha da ED tem impacto no desempenho dos algoritmos

Representação

- É possível descrever os algoritmos e operações em termos do TAD, independente da implementação específica do TAD
- É possível criar implementações de algoritmos/operações que independem de como o TAD é implementado

Operações Básicas

Operações Básicas

- ➔ Inicializa(G)
  - ◆ Cria um grafo G vazio
- ➔ InsereVertice(G, v)
  - ◆ Insere um vértice v isolado no grafo G
- ➔ RemoveVertice(G,v)
  - ◆ Remove do grafo G o vértice v e suas arestas incidentes

Operações Básicas

- ➔ InsereAresta(G, v, u, P)
  - ◆ Insere a aresta (v,u) no grafo G com peso P
- ➔ ExisteAresta(G, v, u)
  - ◆ Verifica se existe a aresta (v,u) no grafo G
- ➔ RetiraAresta(G, v, u, P)
  - ◆ Retira a aresta (v, u) do grafo G e retorna seu peso
- ➔ LiberaGrafo(G)
  - ◆ Libera o espaço ocupado pelo grafo G

Operações Básicas

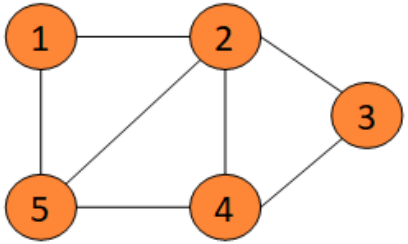
- ExisteAdj(G, v)
  - Retorna verdadeiro se existe algum vértice adjacente a v
- PrimeiroAdj(G, v)
  - Retorna o endereço do primeiro vértice adjacente a v.
- PróximoAdj(G, v, p)
  - Retorna o endereço do próximo vértice adjacente a v a partir de p

Matriz de Adjacências

Matriz de Adjacências

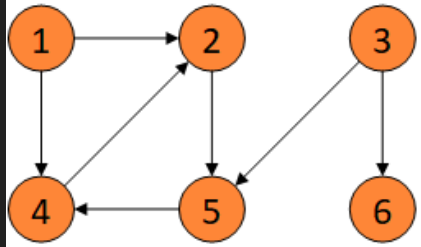
- Dado G = (V, A) um grafo com n vértices
  - n = |V| e n >= 1
- A matriz de adjacências de G é uma matriz M
  - n x n
  - M[v,u] = 1 se e somente se existe aresta do vértice v para o vértice u.
  - Grafos ponderados: M[v,u] contém o peso da aresta

grafo não direcionado



	1	2	3	4	5
1		1			1
2	1		1	1	1
3		1		1	
4		1	1		1
5	1	1		1	

grafo direcionado



	1	2	3	4	5	6
1		1		1		
2					1	
3					1	1
4		1				
5				1		
6						

Matriz de Adjacências

Possível implementação...

Referências

- WIRTH,N. Algorithms and Data Structures, Englewood Cliffs, Prentice-Hall, 1986.
- CORMEN, H.T.; LEISERSON, C.E.; RIVEST, R.L. Introduction to Algorithms, MIT Press, McGraw-Hill, 1999.
- ZIVIANI, N. Projeto de Algoritmos, Thomson, 2a. Edição, 2004.
- SZWARCFITER,J.L. Grafos e Algoritmos Computacionais. Editora Campus, 1983.