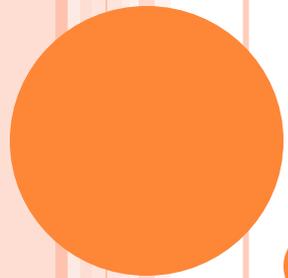


# SCC0216 MODELAGEM COMPUTACIONAL EM GRAFOS

Prof. M. Cristina



**A DISCIPLINA**

# A DISCIPLINA

- Parte essencial dos fundamentos de Computação
- Cadeia de algoritmos e estruturas de dados
  - ICC1 & Lab. ICC1
  - ICC2 & ALG1
  - **Grafos** & Arquivos & POO
  - Alg. Avançados
  - etc.

# A DISCIPLINA

- Programação é o **cerne** da Computação
  - Estruturas de dados são o **cerne** da programação
- Até aqui, diversas estruturas de dados
  - Listas estáticas e dinâmicas, sequenciais e encadeadas, simples ou não
  - Pilhas, filas, deque, *heaps*, etc.
  - Árvores, árvores binárias, balanceadas ou não
  - Etc.
- Agora: as estruturas de dados mais *bonitas e sofisticadas* da Computação
  - Grafos & Arquivos

# GRAFOS

- Intuitivamente, o que são?



# GRAFOS

- E para que servem?



# PORQUE APRENDER GRAFOS?

- Modelar relações e processos em diversos sistemas:
  - Físicos, biológicos, sociais e de informação
- Redes de
  - Comunicação (p.ex. Facebook ou Twitter...)
  - Organização de dados
  - Dispositivos computacionais
  - Fluxo de Computação
- Sistemas de recomendação (Amazon, Netflix,...)
- Otimização de caminhos (Google Maps, Waze,...)

# MODELOS DE GRAFOS

- Modelar sintaxe de linguagem natural
- Estudo de átomos e moléculas
- Medir prestígio/importância
- Espalhamento de rumor
- Amizades entre pessoas
- Padrões de reprodução de animais
- Espalhamento de doenças
- Relação entre genes
- ...

# PROGRAMA DA DISCIPLINA

- Grafos - conceitos fundamentais, grafos dirigidos e não dirigidos, grafos ponderados, caminhos Eulerianos e Hamiltonianos, ciclos, operações básicas sobre grafos, busca em largura e em profundidade, isomorfismo. Representação de grafos via matriz de adjacência e listas de adjacências, operações e análise de algoritmos. Caminhos mínimos, detecção de ciclos, componentes conexos e fortemente conexos, pontes e vértices de articulação, ordenação topológica. Árvores geradoras mínimas, caminhos mínimos, fluxo em rede. Noções de redes complexas e aplicações.

# ESTILO DA AULA

## ○ Slides & lousa

- Considero importante trazer **caderno!**
- Slides, entregas e outros materiais no e-Disciplinas USP

(mas é uma disciplina presencial, i.e., entendo que os alunos estarão nas aulas!)

## ○ As vezes, Exercício/Prática no fim da aula

- Individual ou em **grupos** de 2-3 alunos
- Quem tiver **notebook** pode trazer

## ○ Informal!

# AValiação

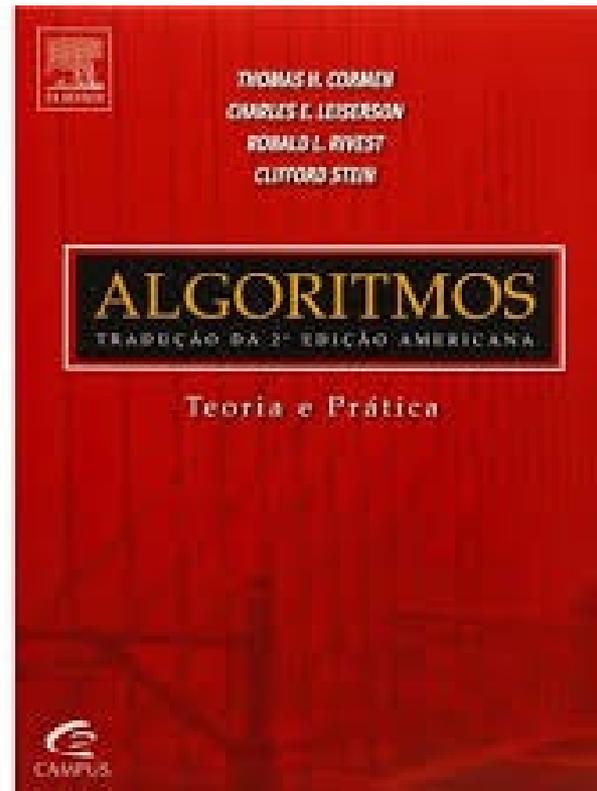
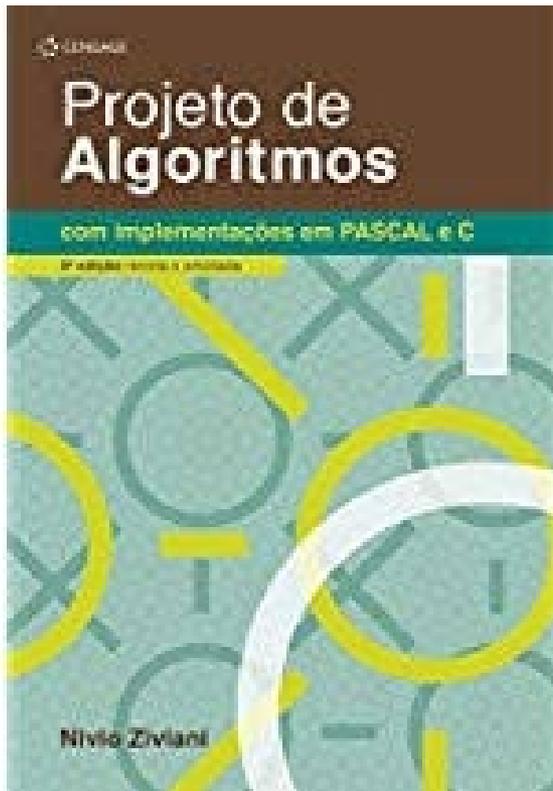
- **Provas**

	<b>Turma A (terça-feira)</b>	<b>Turma B (segunda-feira)</b>
Prova 1	16/05	15/05
Prova 2	11/07	10/07

- **2 trabalho práticos** (individuais, run.codes)
- **2 entregas de atividade em grupo** (2 ou 3 alunos)
- **\*Plágios/cópias não serão tolerados\***
- Frequência mínima de 70%, conforme lista de presença
- Média final = 40% média das provas + 40% média dos trabalhos práticos + 20% média das atividades

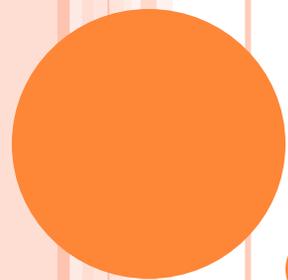
# LIVROS

- Há muitos na nossa biblioteca!



# A DISCIPLINA

- Dúvidas?



# UM POUCO DA 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
  - Resolveu o **problema das 7 pontes** na cidade de Königsberg/Alemanha em 1735



Quadro de Johann Georg Brucker

## O PROBLEMA

- *Na antiga cidade prussiana de Königsberg (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.*

# 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?*

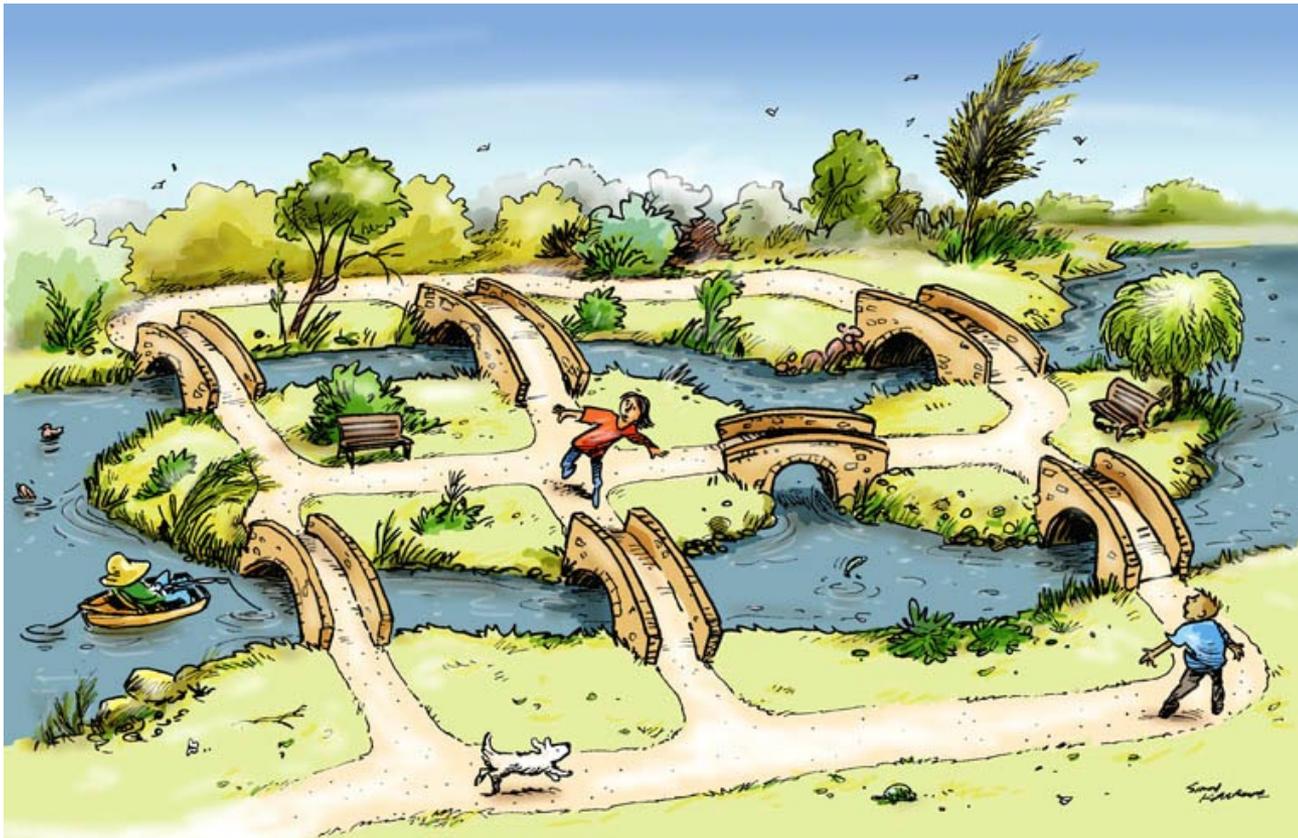
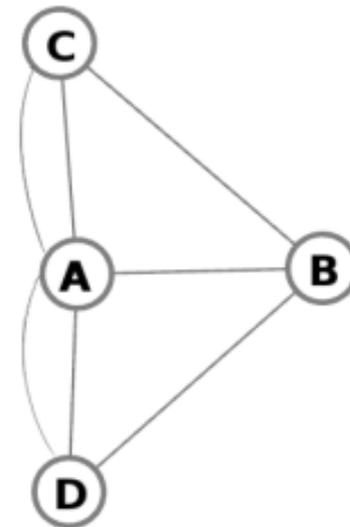


Imagem de Simon Kneebone

# A REPRESENTAÇÃO

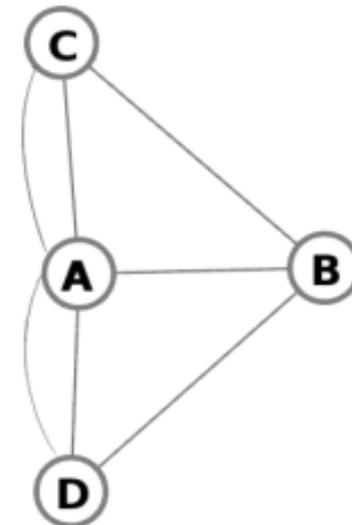
- Representação esquemática do problema com “vértices” e “arestas”
  - Nasce o que é considerado oficialmente por muitos como o **primeiro grafo**



# A SOLUÇÃO

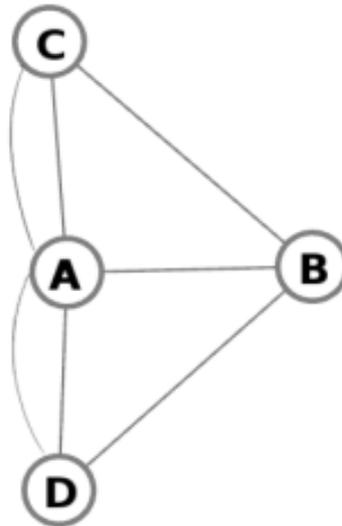
- Considerando vértices de partida, de chegada e intermediários
  - Vértices intermediários devem ter número par de arestas, pois é preciso chegar e sair do vértice sem passar pela mesma aresta
  - Vértices de partida e de chegada
    - Não faz diferença

Todos os vértices têm número ímpar de arestas ☹



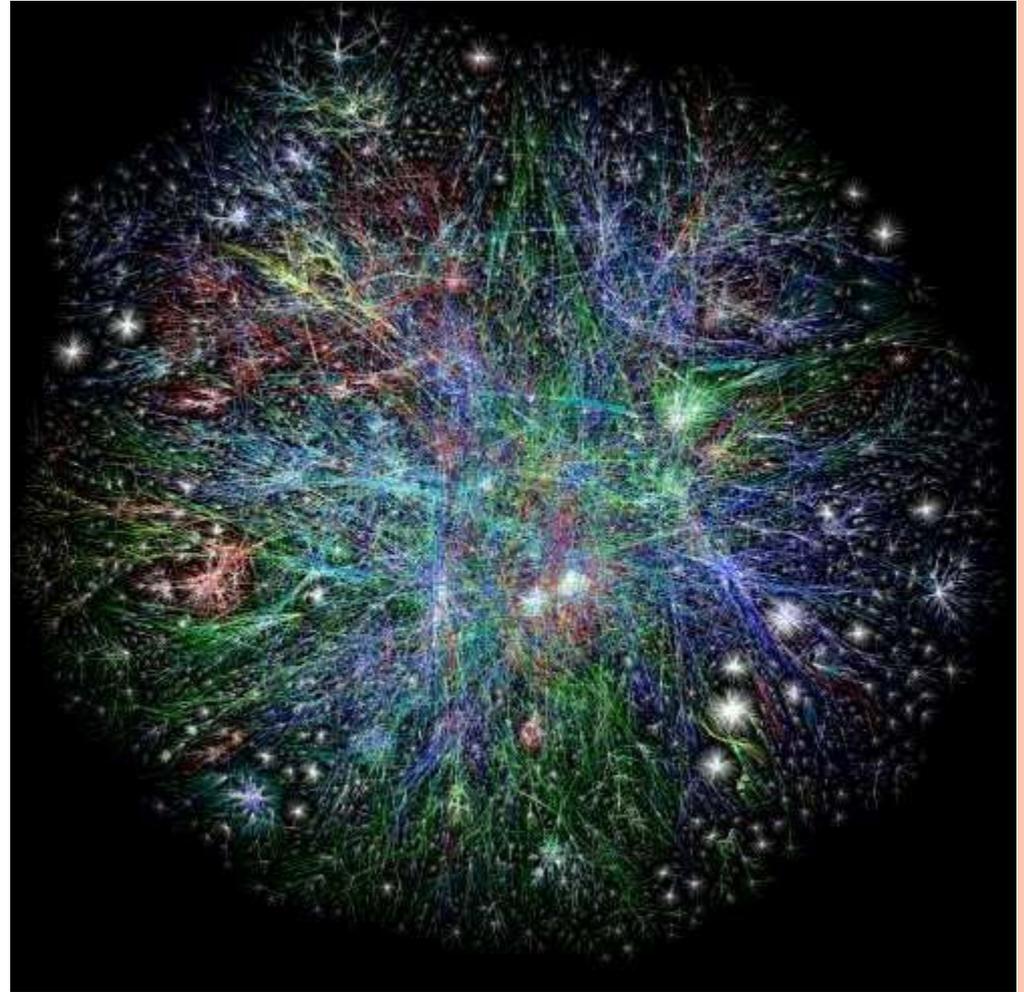
## QUESTÃO

- Esse tipo de estrutura nos é estranha? Ou já vimos coisas parecidas em outras disciplinas do curso?



# DESDE ENTÃO...

- Internet



<https://pt.wikipedia.org/wiki/Internet>

# DESDE ENTÃO...

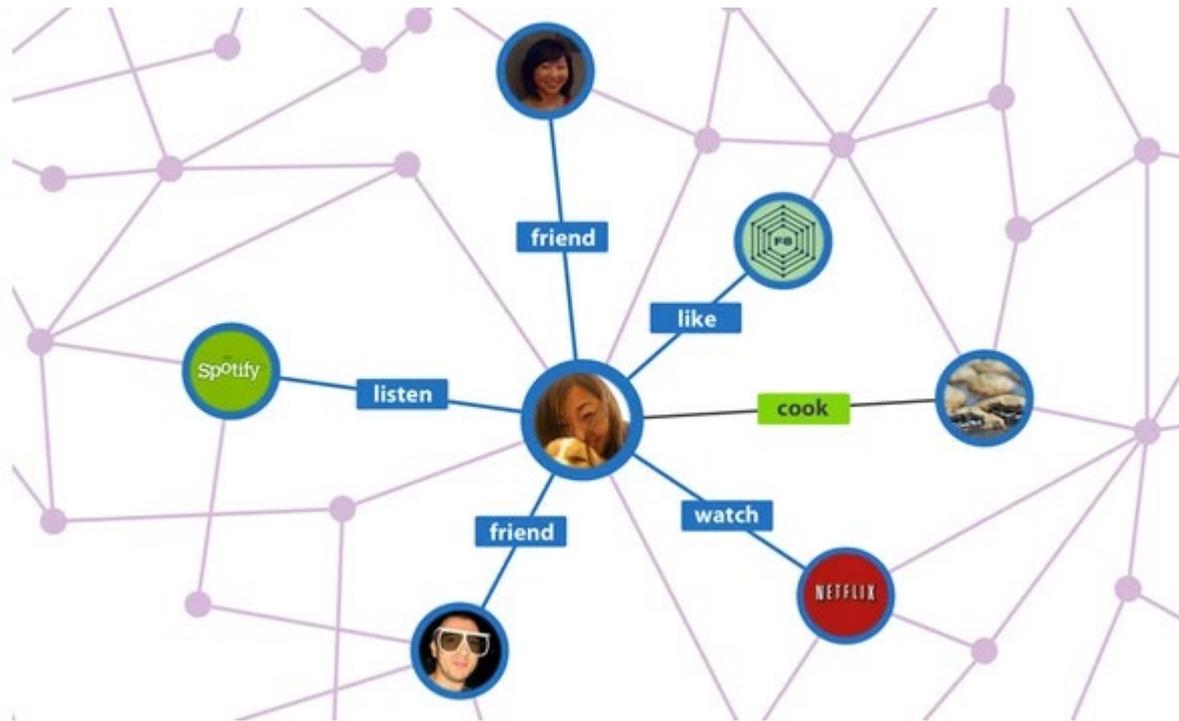
- Buscador na web



<https://pt.wikipedia.org/wiki/PageRank>

# DESDE ENTÃO...

- Redes sociais e aplicativos relacionados



# DESDE ENTÃO...

- Aeroportos & vôos



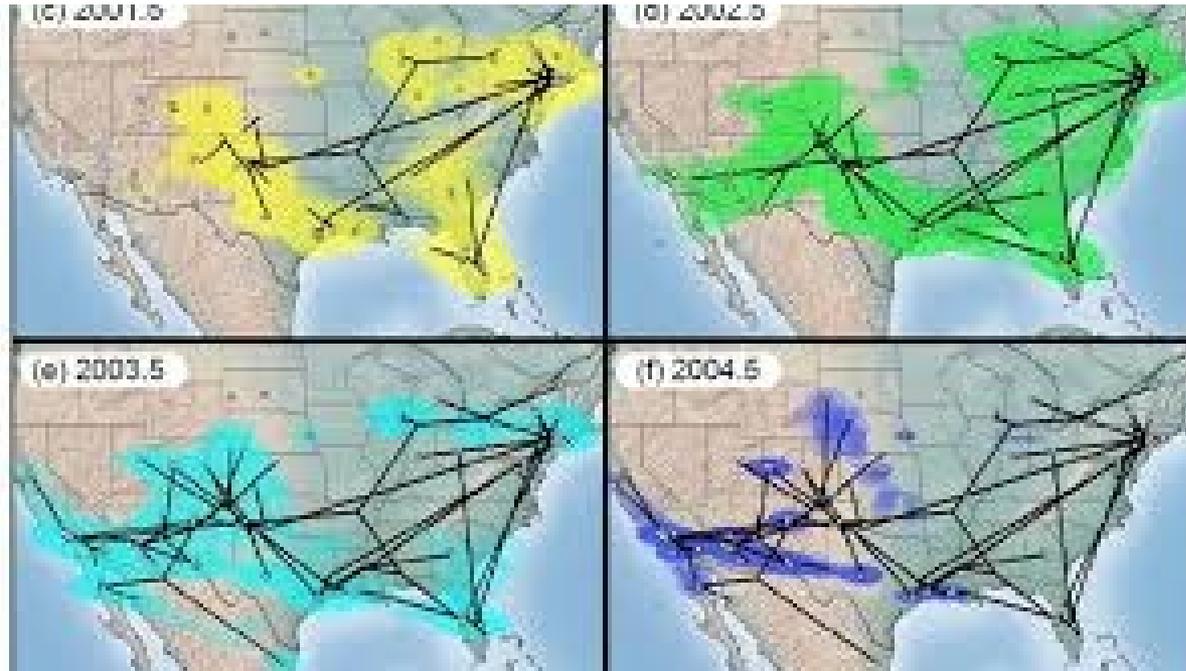
# DESDE ENTÃO...

- GPS



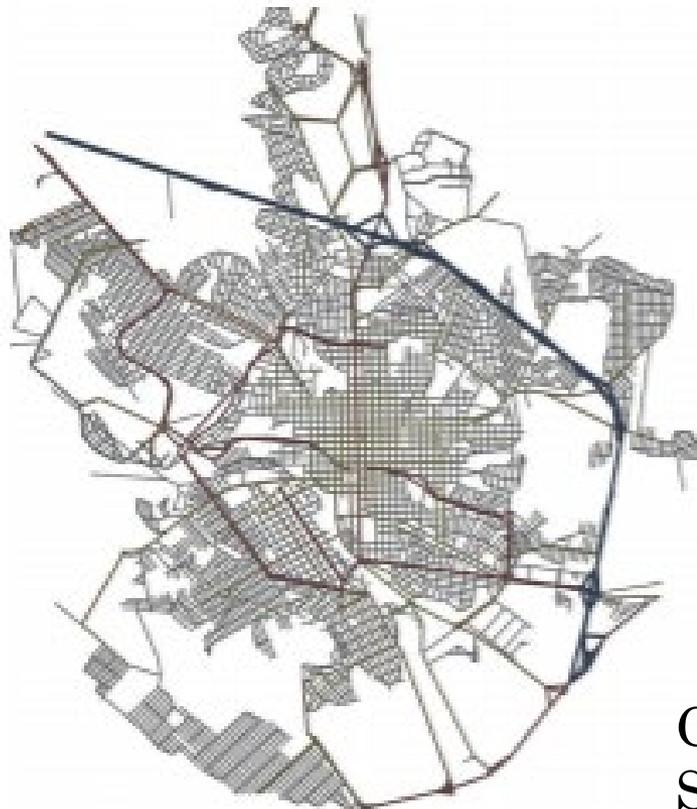
# DESDE ENTÃO...

- Epidemias



## DESDE ENTÃO...

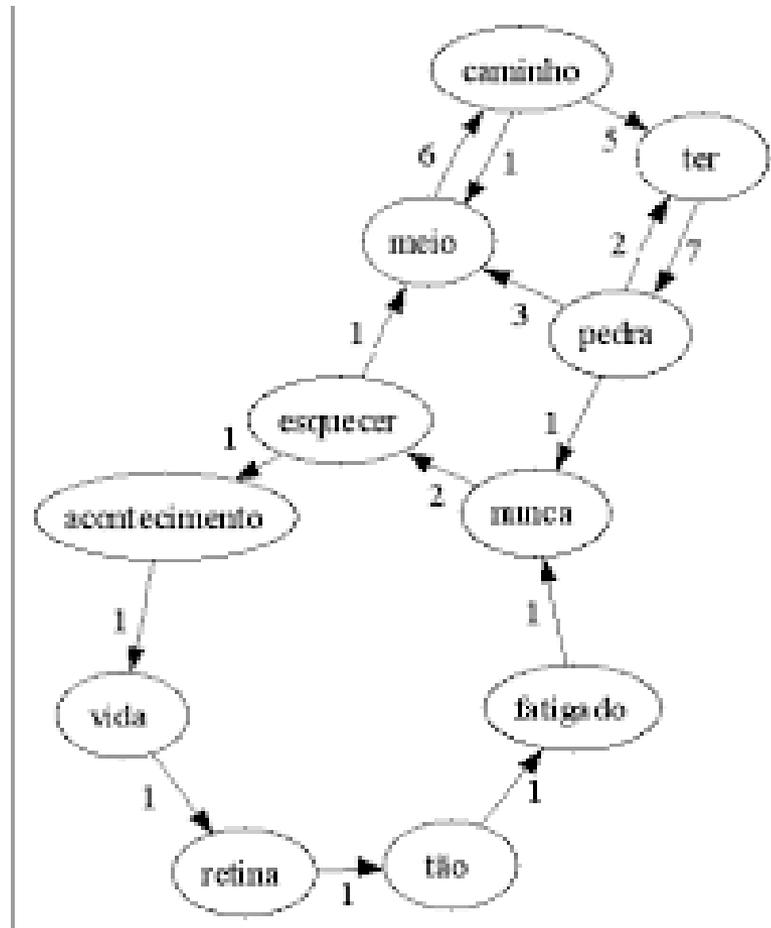
- Topologia urbana: densidade populacional, dinâmica do tráfego, planejamento



Cidade de  
São Carlos

# DESDE ENTÃO...

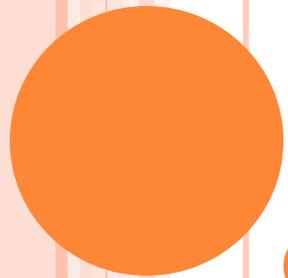
- Linguagem



# DESDE ENTÃO...

- Incontáveis outras aplicações
  - Redes de computadores e roteamento de dados
  - Redes Neurais
  - Planejamento de movimento de robôs autônomos
  - Modelagem de cenários de jogos
  - Circuitos
  - Genealogia
  - Ecologia
  - Etc.
- Grafos são uma estrutura onipresente em computação





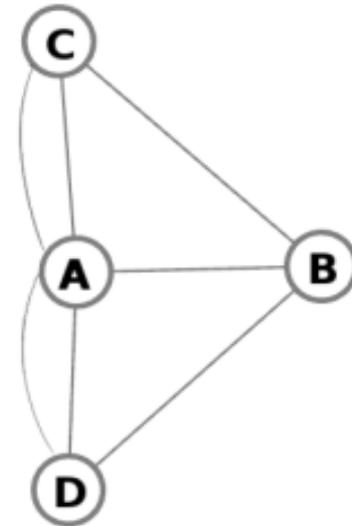
# ALGUMAS DEFINIÇÕES

# DEFINIÇÃO BÁSICA

- *Um grafo é uma estrutura de dados constituída de um conjunto de vértices  $V$  e um conjunto de arestas  $A$  conectando pares de vértices.*
  - $G = (V, A)$ 
    - Nomenclatura usual
      - Vértice, nó (*node, vertex*)
      - Aresta, ligação, arco (*edge, link, arc*)
- *Grau do vértice*: número de arestas incidentes ao vértice

# EXEMPLO

- Possível grafo das 7 pontes de Königsberg
  - $V = \{A, B, C, D\}$
  - $A = \{(A,B), (A,C), (C,A), (A,D), (D,A), (B,C), (B,D)\}$
  - $G1 = (\{A, B, C, D\}, \{(A,B), (A,C), (C, A), (A,D), (D, A), (B,C), (B,D)\})$
  - Grau do vértice B é 3



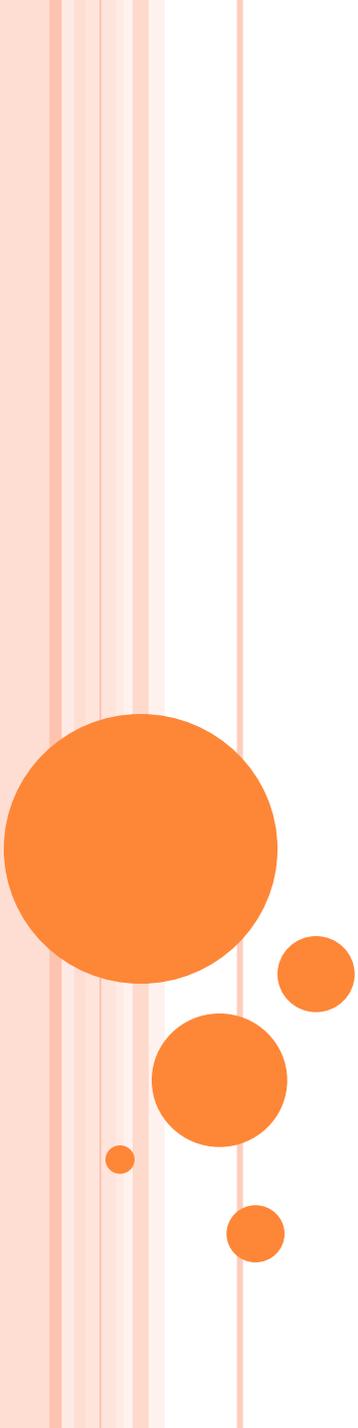
# HÁ MUITO MAIS ALÉM DO BÁSICO!

- Grafos podem ser direcionados (ou não!)
  - Para quê?
- Grafos podem ter ciclos
  - Sério?
- Arestas podem ter peso
  - Para quê?
- Grau pode ser de entrada ou de saída
  - Que diferença faz?
- Etc.

Vamos com calma! Temos o semestre inteiro pela frente... 😊

# CARACTERÍSTICAS DOS GRAFOS

- Como logo veremos
  - **Muita liberdade e flexibilidade**
    - Grande poder de representação
  - Necessidade ainda maior de **controle**



# PRÁTICA EM GRUPOS

*NOTEBOOKS PERMITIDOS*

# EXERCÍCIO

- Desenhe um grafo do ICMC



## EXERCÍCIO 2

- Pensando em um TAD Grafo, declare em C a estrutura de dados de um grafo

## EXERCÍCIO 3

- Desenhe o esquema da sua estrutura de dados instanciada com o seu grafo do ICMC

## EXERCÍCIO 4 (PARA CASA)

- Considerando o TAD Grafo e usando a sua estrutura de dados:
  - Escreva uma função em C que retorna o número de vértices de um grafo de entrada (dado pelo usuário)
  - Escreva uma função em C que retorna o vértice com mais conexões de um grafo de entrada