



# Apresentação

## Estrutura de Dados II

Prof. João Dallyson Sousa de Almeida

Núcleo de Computação Aplicada NCA - UFMA

Dep. De Informática - Universidade Federal do Maranhão

# Apresentação

## ▶ Ementa

- ▶ Algoritmos de ordenação e busca.
- ▶ Árvore de busca multidirecional balanceada.
- ▶ Hashing. Noções de organização de arquivos.
- ▶ Noções de grafos: conceitos, coloração, árvores geradoras..
- ▶ Algoritmos em grafos: caminho mínimo, fluxo máximo e outros.

## ▶ Bibliografia: básica

- ▶ CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. Algoritmos: Teoria e Prática. Editora Campus, 2002
- ▶ Algorithms 4th edition by R. Sedgewick and K. Wayne, Addison-Wesley Professional, 2011, ISBN 0-321-57351-X
- ▶ Ziviani, N. Projeto de Algoritmos Com Implementações em Pascal e C, Cengage Learning, 2004.

## ▶ Bibliografia: complementar

- ▶ TENENBAUM, Aaron; LANGSAM, Yedidyah; AUGENSTEIN, Moshe J. Estruturas de dados usando C. São Paulo: Makron Books, 1995. ISBN: 9788534603485
- ▶ ASCENCIO, Ana Fernanda Gomes; ARAUJO, Graziela Santos. Estruturas de Dados: Algoritmos, análise da complexidade e implementações em Java e C/C++. Pearson Prentice Hall, 2010
- ▶ DROZDEK, Adam. Adam Drozdek. Data Structures and Algorithms in Java. 2. Cengage Learning. 2004. 2. Cengage Learning. 2004
- ▶ GOODRICH, Michael T. Estruturas de dados e algoritmos em java. 4 ED. Porto Alegre: Bookman, 2007. 600.
- ▶ SKIENA, Steven S.. The Algorithm Design Manual. 2. Springer-Verlag. 2008

# Apresentação

- ▶ Prof. João Dallyson
  - ▶ [Joao.dallyson@ufma.br](mailto:Joao.dallyson@ufma.br) / [jdallyson@nca.ufma.br](mailto:jdallyson@nca.ufma.br)
  - ▶ Doutor em Engenharia Elétrica – UFMA
    - ▶ Área: Ciência da Computação
  - ▶ Sala 214, Bloco 6, CCET
- ▶ Pesquisador em:
  - ▶ Processamento de Imagens Oftalmológicas, Inteligência Computacional Aplicada e Séries Temporais
- ▶ Lattes:
  - ▶ <http://lattes.cnpq.br/6047330108382641>



# A Disciplina

- ▶ As Estruturas de Dados são os blocos de algoritmos básicos utilizados na organização das informações.
- ▶ Esta disciplina visa apresentar e permitir compreender as principais estruturas de dados avançadas, como implementá-las e usá-las em problemas genéricos
- ▶ Será adotada a linguagem JAVA na disciplina (Os exemplos serão apresentados em pseudocódigo)
- ▶ Pré-requisitos: Conhecimentos das linguagem Java ou C ( manipulação de ponteiros) e recursividade
- ▶ Conhecimento das estruturas de dados básicas (pilha, lista e fila) e de árvores

# Objetivos

---

- ▶ Solução de problemas que utilizem estruturas de dados complexos.
- ▶ Aprendizado de algoritmos e estruturas de dados usados em ordenação e busca.
- ▶ Modelagem e solução de problemas usando grafos.
- ▶ Implementação de algoritmos para solução de problemas clássicos.

# Ementa

---

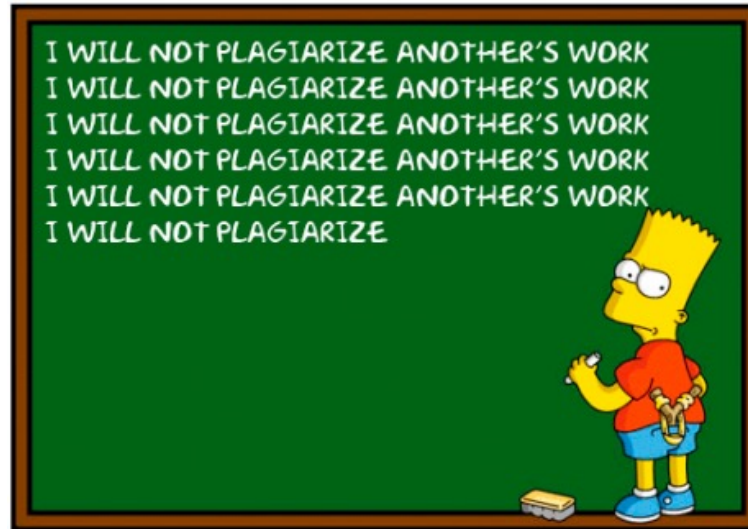
- ▶ Algoritmos de ordenação e busca.
- ▶ Árvore de busca multidirecional balanceada.
- ▶ Hashing. Noções de organização de arquivos.
- ▶ Noções de grafos: conceitos, coloração, árvores geradoras..
- ▶ Algoritmos em grafos: caminho mínimo, fluxo máximo e outros.

# Conteúdo Programático

- ▶ Análise de algoritmos (complexidade)
- ▶ Recorrência
- ▶ Algoritmos de ordenação e busca (Seleção, Inserção, MergeSort, QuickSort, HeapSort, CountingSort, BucketSort e RadixSort)
- ▶ Hashing
- ▶ Árvores e Árvores Binárias de Busca (Revisão)
- ▶ Árvores AVL (Revisão)
- ▶ Árvores Rubro-Negras
- ▶ Árvores B
- ▶ Grafos: Conceitos, BFS, DFS, ordenação topológica, Caminhos mínimos (Bellman-Ford, Dijkstra e Floyd-Warshall) e Fluxo Máximo (Ford-Fulkerson)

# Atividades Práticas

- ▶ Serão critérios de avaliação:
  - ▶ Clareza;
  - ▶ Simplicidade
  - ▶ Generalidade.





# Avaliações

Data	Hora	Descrição
10/04/2024	15:50 às 17:30	Prova 1
15/05/2024	15:50 às 17:30	Prova 2
26/06/2024	15:50 às 17:30	Prova 3
03/07/2024	15:50 às 17:30	Reposição
10/07/2024	15:50 às 17:30	Prova Final

- ▶ Notas: Atividade Prática\*0.4 + Prova Discursiva\*0.5 + Lista de Exercício\*0.1

# Por que estudar algoritmos?

- ▶ Melhorar a eficiência de programas
- ▶ Trabalhar em áreas de TI relacionadas a algoritmos e análise de dados
- ▶ Desenvolver habilidades de resolução de problemas
- ▶ Aprender conceitos fundamentais de ciência da computação

# Por que estudar algoritmos?

## ► Descobrir novas oportunidades:

- Estudo de algoritmos data pelo menos da época Euclides (300 aC).
- Formalizado por Church e Turing em 1930.
- Alguns algoritmos importantes foram descobertos por alunos de graduação em um curso como este!



FONTE: [Sedgewick and Wayne, 2011]

# Por que estudar algoritmos?

- ▶ Para se tornar um programador proficiente.
- ▶ Para resolver problemas que não poderiam ser tratados de outra forma.
- ▶ Por diversão e retorno financeiro.
- ▶ Preparar-se para entrevistas de emprego.



- ▶ Segundo **Brookshear (2013)**, um algoritmo é um conjunto ordenado de passos executáveis, não ambíguos, que define um processo finalizável.
- ▶ Segundo **Dijkstra**, um algoritmo corresponde a uma descrição de um padrão de comportamento, expresso em termos de um conjunto finito de ações.
  - ▶ Executando a operação  $a + b$  percebemos um padrão de comportamento, mesmo que a operação seja realizada para valores diferentes de  $a$  e  $b$ .

# Programas

- ▶ Programar é basicamente estruturar dados e construir algoritmos.
- ▶ Programas são formulações concretas de algoritmos abstratos, baseados em representações e estruturas específicas de dados.
- ▶ Programas representam uma classe especial de algoritmos capazes de serem seguidos por computadores.

# Ex: Problema de Ordenação

## ▶ Entrada:

- ▶ Uma sequência de  $n$  números:  $\langle a_1, a_2, \dots, a_n \rangle$

## ▶ Saída:

- ▶ Uma permutação dos números de entrada:  $\langle a'_1, a'_2, \dots, a'_n \rangle$ , tal que  $a'_1 \leq a'_2 \leq \dots \leq a'_n$  (ordenação crescente)

## ▶ Algoritmo:

- ▶ Sequência de comandos que leva uma instância de entrada a uma saída correta.

- ▶ Instância de um problema:
  - ▶ É a entrada, que satisfaz a quaisquer restrições impostas pelo problema, necessária para se calcular uma solução do problema
- ▶ Algoritmo correto:
  - ▶ É quando, para qualquer instância do problema, o algoritmo termina com a saída correta.
  - ▶ Resolve o problema computacional

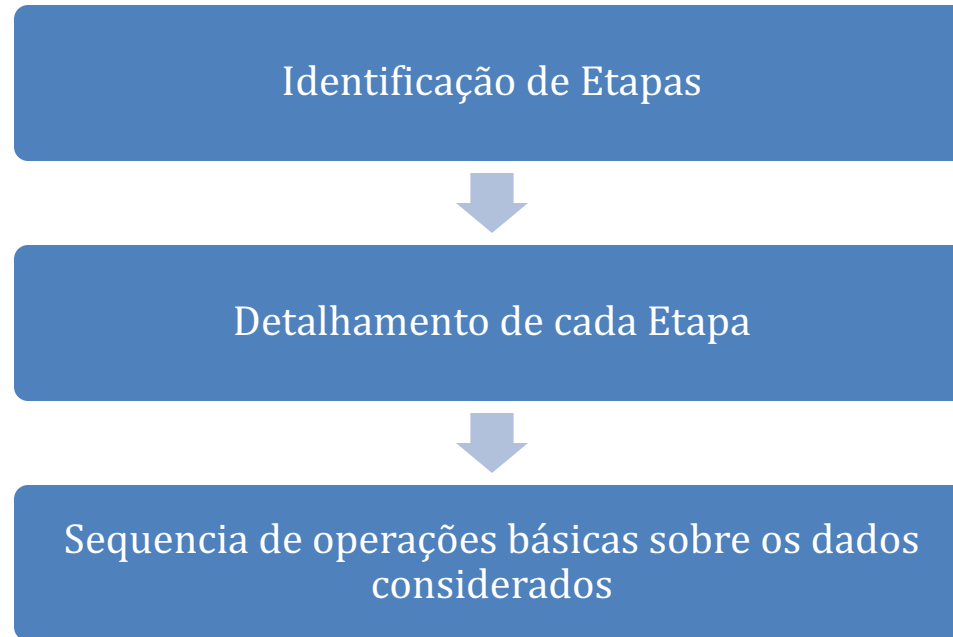


# Formas de Descrição de Algoritmos

---

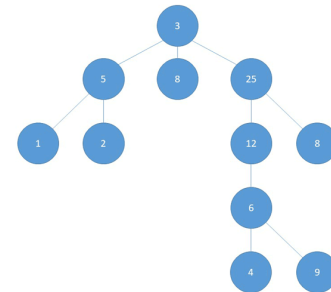
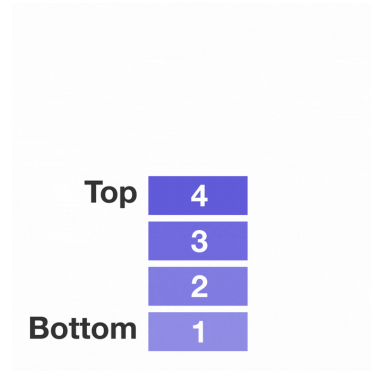
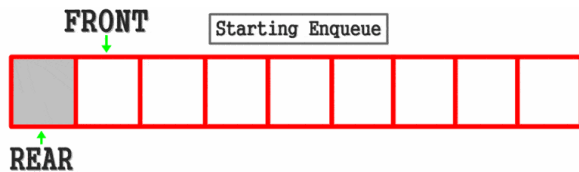
- ▶ Pseudocódigo
- ▶ Linguagem de Programação
- ▶ Fluxograma
- ▶ Linguagem Natural (ambiguidade!)

# Desenvolvimento



# Estruturas de Dados

- ▶ Estruturas de dados e algoritmos estão intimamente ligados:
  - ▶ não se pode estudar estruturas de dados sem considerar os algoritmos associados a elas;
  - ▶ assim como a escolha dos algoritmos em geral depende da representação e da estrutura dos dados.



# Escolha da representação dos dados

- ▶ A escolha da representação dos dados é determinada, entre outras, pelas operações a serem realizadas sobre os dados.
- ▶ Considere a operação de adição
  - ▶ Para pequenos números, uma boa representação é por meio de barras verticais (caso em que a operação de adição é bastante simples)
  - ▶ Já a representação por dígitos decimais requer regras relativamente complicadas, as quais devem ser memorizadas.
  - ▶ Entretanto, quando consideramos a adição de grandes números é mais fácil a representação por dígitos decimais (devido ao princípio baseado no peso relativo da posição de cada dígito).

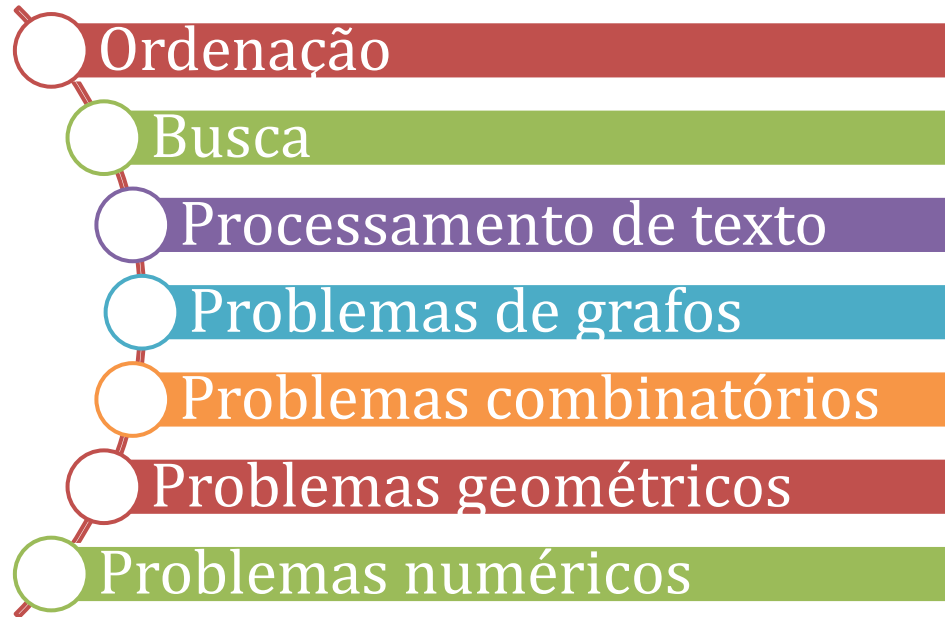
# Tipo Abstrato de Dados (TAD)

- ▶ TAD:
  - ▶ Descreve um domínio de valores e o conjunto de operações que são especificados independente da implementação
  - ▶ **Exemplo:** o conjunto dos inteiros acompanhado das operações de adição, subtração e multiplicação.
- ▶ TAD's são utilizados extensivamente como base para o projeto de algoritmos.
- ▶ A representação por trás do tipo abstrato de dados é realizada mediante uma estrutura de dados.

# Estrutura de Dados

- ▶ Tipos de Dados
  - ▶ Define o conjunto de valores que uma variável pode assumir
  - ▶ int, char, float, etc.
- ▶ Tipos Abstratos de Dados (TAD)
  - ▶ Filas, Pilhas, Listas, etc.
  - ▶ Inclui operações para manipulação dos dados
- ▶ Estruturas de dados:
  - ▶ Relacionamento lógico entre tipos de dados
  - ▶ Método particular de se implementar um TAD

# Tipos de problemas importantes



# Problemas de ordenação

- ▶ O problema de ordenação é reorganizar os itens de uma dada lista em uma determinada ordem.

Unsorted Array



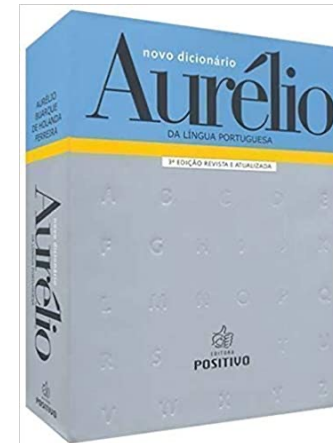


# Problemas de ordenação

- ▶ O problema de ordenação é reorganizar os itens de uma dada lista em uma determinada ordem.
  - ▶ A natureza dos itens deve permitir ordená-los
  - ▶ Exemplos

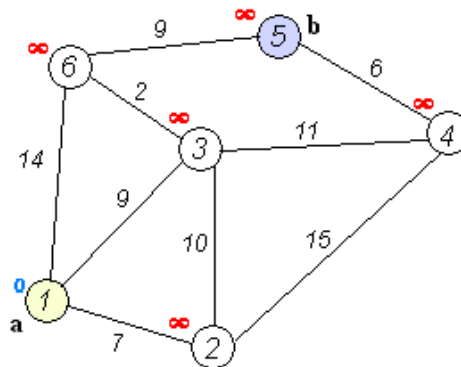
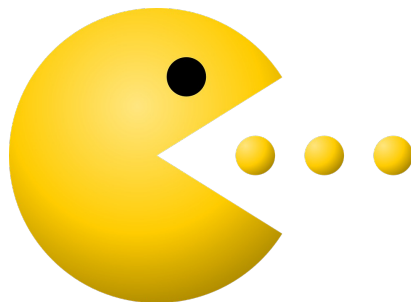


FRANCA		FEAC	
MODALIDADE: Básico		SÉRIE	
TÉCNICO: Doriane Henrique de Araújo Gomes		ASSISTENTE TÉCNICO: Ricardo Junior De Sousa Santos	
LOCAL: Ciepq - Fortaleza em 1		DIÁRIO: de 1993 a 2008	
DIAS SEQUENCIAIS		DIÁRIO	
Nº	ATIVIDADE	1	2
1	JARTHUR OLIVEIRA BILAL	1	2
2	JARTHUR OLIVEIRA	3	4
3	JARTHUR OLIVEIRA	5	6
4	JARTHUR OLIVEIRA	7	8
5	JARTHUR OLIVEIRA	9	10
6	JARTHUR OLIVEIRA	11	12
7	JARTHUR OLIVEIRA	13	14
8	JARTHUR OLIVEIRA	15	16
9	JARTHUR OLIVEIRA	17	18
10	JARTHUR OLIVEIRA	19	20
11	JARTHUR OLIVEIRA	21	22
12	JARTHUR OLIVEIRA	23	24
13	JARTHUR OLIVEIRA	25	26
14	JARTHUR OLIVEIRA	27	28
15	JARTHUR OLIVEIRA	29	30
16	JARTHUR OLIVEIRA	31	32
17	JARTHUR OLIVEIRA	33	34
18	JARTHUR OLIVEIRA	35	36
19	JARTHUR OLIVEIRA	37	38
20	JARTHUR OLIVEIRA	39	40
21	JARTHUR OLIVEIRA	41	42
22	JARTHUR OLIVEIRA	43	44
23	JARTHUR OLIVEIRA	45	46
24	JARTHUR OLIVEIRA	47	48
25	JARTHUR OLIVEIRA	49	50
26	JARTHUR OLIVEIRA	51	52
27	JARTHUR OLIVEIRA	53	54
28	JARTHUR OLIVEIRA	55	56
29	JARTHUR OLIVEIRA	57	58
30	JARTHUR OLIVEIRA	59	60
31	JARTHUR OLIVEIRA	61	62
32	JARTHUR OLIVEIRA	63	64
33	JARTHUR OLIVEIRA	65	66
34	JARTHUR OLIVEIRA	67	68
35	JARTHUR OLIVEIRA	69	70
36	JARTHUR OLIVEIRA	71	72
37	JARTHUR OLIVEIRA	73	74
38	JARTHUR OLIVEIRA	75	76
39	JARTHUR OLIVEIRA	77	78
40	JARTHUR OLIVEIRA	79	80
41	JARTHUR OLIVEIRA	81	82
42	JARTHUR OLIVEIRA	83	84
43	JARTHUR OLIVEIRA	85	86
44	JARTHUR OLIVEIRA	87	88
45	JARTHUR OLIVEIRA	89	90
46	JARTHUR OLIVEIRA	91	92
47	JARTHUR OLIVEIRA	93	94
48	JARTHUR OLIVEIRA	95	96
49	JARTHUR OLIVEIRA	97	98
50	JARTHUR OLIVEIRA	99	100



# Problemas de ordenação

- ▶ Por que queremos uma lista ordenada?
  - ▶ Facilitar a busca.
  - ▶ A ordenação é utilizada em etapas intermediárias de outros algoritmos.
  - ▶ Ex: Algoritmos gulosos necessitam de entradas ordenadas.



# Busca

- ▶ O problema da busca consiste em encontrar uma chave em um determinado conjunto de dados.
- ▶ Existem diversos algoritmos de busca:
  - ▶ **Busca sequencial**, busca binária, busca eficiente

Linear Search



## ► Busca sequencial X Busca Binária

Binary search

steps: 0




Sequential search

steps: 0



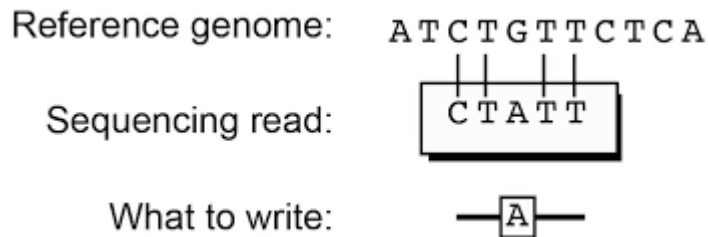
# Busca

- ▶ Não existe um algoritmo único que se adapte eficientemente a todos os problemas de busca.
    - ▶ Alguns algoritmos são mais eficientes que outros. Entretanto, podem consumir mais memória.
  - ▶ Desafio em aplicações reais.
    - ▶ Texto ? Áudio ? Vídeo ?
- 
- A word cloud graphic in the bottom right corner, featuring various terms related to data science and technology. The most prominent words are 'big data', 'search', 'software', 'analysis', 'development', 'future', 'based', 'san', 'bytes', 'volume', 'at', 'process', 'technology', 'management', 'approaches', 'national', 'well', 'decision', 'related', 'per', 'flow', 'year', 'times', 'business', 'related', 'factor', 'years', 'integration', 'government', 'definition', 'terabytes', 'map', 'reduced', 'networks', 'making', 'initiative', 'currently', 'process', 'data', 'volume', 'at', 'bytes', 'san', 'based', 'search', 'analysis', 'development', 'future', 'year', 'flow', 'per', 'related', 'decision', 'factor', 'years', 'integration', 'government', 'definition', 'terabytes', 'map', 'reduced', 'networks', 'making', 'initiative', 'currently'.



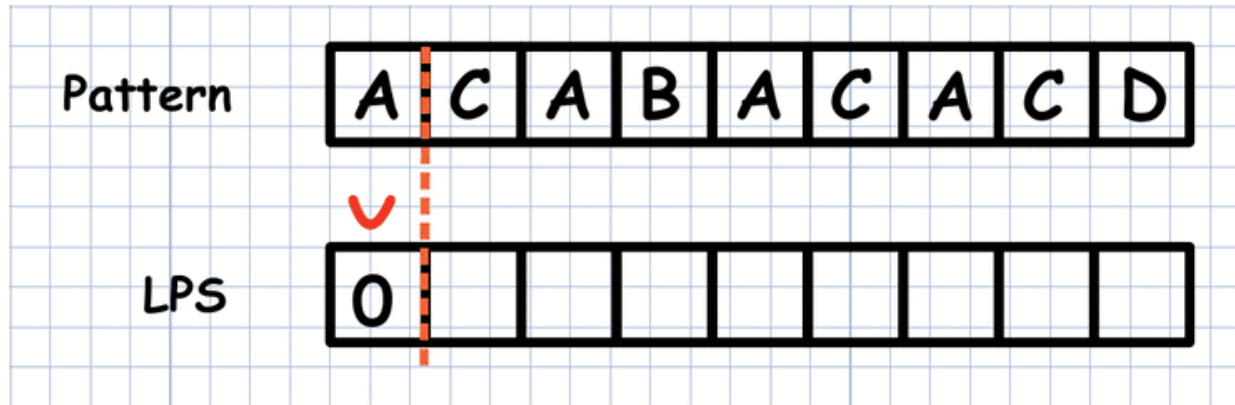
# Processamento de Texto

- ▶ Manipulação de dados não numéricos (String)
- ▶ String é uma sequência de caracteres que incluem letras, números, caracteres especiais, cadeias de bits e sequência de genes (podem ser representados por sequência de 4 caracteres).



# Processamento de Texto

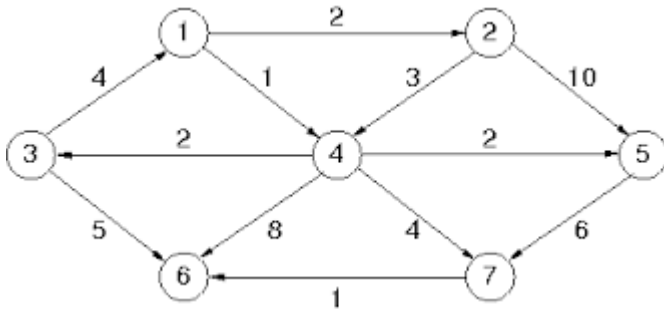
- ▶ Algoritmos de processamento de texto tem sido importante para as linguagens de programação em problemas de compilação.
- ▶ Exemplo de problema: String Matching. Busca de uma dada palavra em um texto.



# Grafos

## ► Grafo:

- Definição Informal: Coleção de pontos chamados vértices que são conectados por segmentos chamados arestas.





# Grafos

- ▶ Grafos podem ser utilizados para modelar diversas aplicações práticas nas diversas áreas (transporte, comunicação, social, redes, econômica, jogos, etc..)
- ▶ Exemplos:
  - ▶ Como alcançar todos os pontos em uma rede?
  - ▶ Qual a melhor rota entre duas cidades?

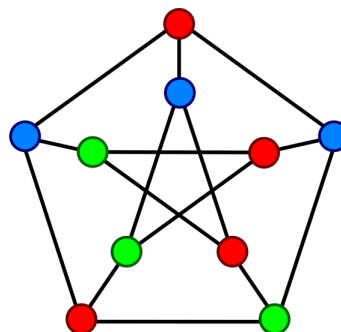
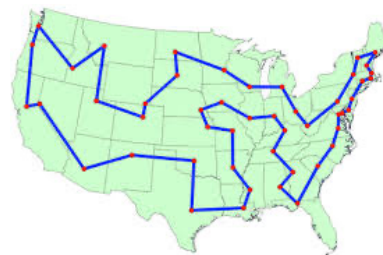
# Grafos

## ▶ Problemas difíceis:

- ▶ Caixeiro viajante

- ▶ Coloração de grafos:

- ▶ procura atribuir o menor número de cores aos vértices de um gráfico de forma que não haja dois vértices adjacentes da mesma cor. Aplicação em problemas de escalonamento.

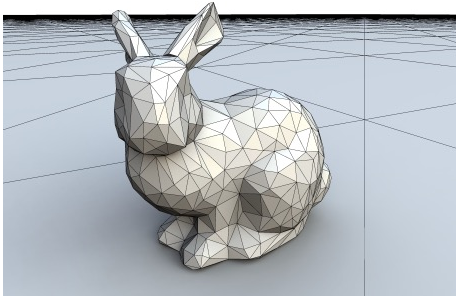


# Problemas combinatórios

- ▶ O problema do caixeiro viajante e da coloração de grafo são exemplos de problemas combinatórios.
- ▶ Problemas que buscam um objeto combinatório: permutação, uma combinação ou subconjunto que satisfazem certa condição.
- ▶ São problemas difíceis por crescerem muito rápido.
- ▶ Não possuem algoritmos conhecidos para resolverem de forma eficiente (tempo aceitável)

# Problemas geométricos

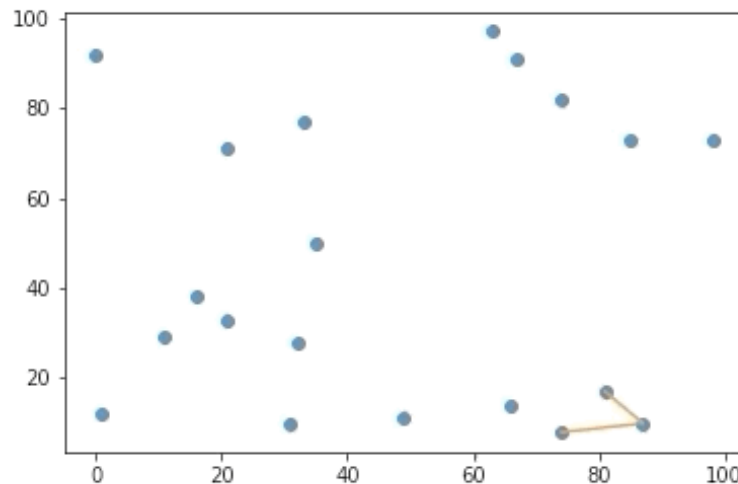
- ▶ Os algoritmos geométricos lidam com objetos geométricos, como pontos, linhas e polígonos.
- ▶ Aplicações atuais:
  - ▶ Computação gráfica, robótica, tomografia etc...



# Problemas geométricos

## ▶ Ex de classes de problemas geométricos:

- ▶ Problema do par mais próximo.
- ▶ Problema do fecho convexo. →
- ▶ Problema de triangulação
- ▶ Triângulos de Delaunay



# Problemas numéricos

- ▶ Problemas que envolvem objetos matemáticos de natureza contínua:
  - ▶ Solução de equações, sistemas de equações, integrais, equações diferenciais, matrizes esparsas, Transformada de Fourier, avaliação de funções, etc...
- ▶ A maioria dos problemas são resolvidos apenas por aproximação.

# Referências

## Básica

- ▶ CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. Algoritmos: Teoria e Prática. Editora Campus, 2002
- ▶ Algorithms 4th edition by R. Sedgewick and K. Wayne, Addison-Wesley Professional, 2011, ISBN 0-321-57351-X
- ▶ Ziviani, N. Projeto de Algoritmos Com Implementações em Pascal e C, Cengage Learning, 2004.

## Complementar

- ▶ TENENBAUM, Aaron; LANGSAM, Yedidyah; AUGENSTEIN, Moshe J. *Estruturas de dados usando C. São Paulo: Makron Books, 1995. ISBN: 9788534603485*
- ▶ ASCENCIO, Ana Fernanda Gomes; ARAUJO, Graziela Santos. Estruturas de Dados: Algoritmos, análise da complexidade e implementações em Java e C/C++. Pearson Prentice Hall, 2010
- ▶ DROZDEK, Adam. Adam Drozdek. Data Structures and Algorithms in Java. 2. Cengage Learning. 2004. 2. Cengage Learning. 2004
- ▶ GOODRICH, Michael T. Estruturas de dados e algoritmos em java. 4 ED. Porto Alegre: Bookman, 2007. 600.
- ▶ SKIENA, Steven S.. The Algorithm Design Manual. 2. Springer-Verlag. 2008

# Referências

- ▶ Livros gratuitos:
  - ▶ <https://algs4.cs.princeton.edu/home/>
  - ▶ <https://goalkicker.com/AlgorithmsBook/>
  - ▶ <http://dept.cs.williams.edu/~bailey/JavaStructures/Book.html>
  - ▶ <https://people.cs.vt.edu/~shaffer/Book/>



# Perguntas....

