

# Estruturas de Dados 2

Prof. Silvia Brandão

2024.1



# Plano de Ensino – ED2



Ementa



Bibliografia



Sistema de Avaliação

# Plano de Ensino – ED2

## ▶ EMENTA

### ○ ESTRUTURAS DE DADOS 1

- Estudo dos conceitos das estruturas estáticas e dinâmicas e suas aplicações para o armazenamento de dados. Manipulação de estruturas dinâmicas lineares e não lineares.

### ○ ESTRUTURAS DE DADOS 2

- Apresentação dos métodos de busca e de ordenação de dados em memória principal e secundária, trabalhando com as organizações física e lógica de arquivos. Proposição de situações e problemas, utilizando-se algoritmos que implementem os vários métodos de busca e ordenação.

# Plano de Ensino – ED2

## ► BIBLIOGRAFIA BÁSICA

- ASCENCIO, Ana Fernanda Gomes; ARAUJO, Graziela Santos de. **Estruturas de dados: algoritmos, análise da complexidade e implementações em Java e C/C++**. São Paulo: Pearson Education do Brasil, 2010. 432 p. Disponível em: [https://uniube.bv3.digitalpages.com.br/users/publications/9788576058816/pages/\\_1](https://uniube.bv3.digitalpages.com.br/users/publications/9788576058816/pages/_1) Acesso em: 20 mar. 2017.
- CORMEN, Thomas H. **Algoritmos: teoria e prática**. Rio de Janeiro: Elsevier, 2012. 926 p.
- ZIVIANI, Nivio. **Projeto de algoritmos: com implementações em Pascal e C**. São Paulo: PIONEIRA, 1993. 267 p.
- PREISS, Bruno R. **Estrutura de dados e algoritmos: padrões de projetos orientados a objetos com Java**. Rio de Janeiro: Campus, 2001. 566p.

# Plano de Ensino – ED2

## ► BIBLIOGRAFIA COMPLEMENTAR

- AZEREDO, Paulo Alberto de. **Métodos de classificação de dados e análise de suas complexidades**. Rio de Janeiro: Campus, 1996. 132 p.
- BOAVENTURA NETTO, Paulo Oswaldo. **Grafos: teoria, modelos, algoritmos**. 2. ed. São Paulo (SP): E. Blucher, 2001. 304p.
- DROZDEK, A. **Estrutura de dados e algoritmos em C++**. São Paulo (SP): Thomson, 2002, 579 p.
- PUGA, Sandra; RISSETTI, Gerson. **Lógica de programação de computadores e estruturas de dados com aplicações em Java**. 3. ed. São Paulo: Pearson Prentice Hall, 2009. 263 p. Disponível em: <https://uniube.bv3.digitalpages.com.br/users/publications/9788543019147/pages/-15>. Acesso em: 10 abr. 2017.
- KNUTH, Donald E. **The art of computer programming**. 3. ed. Boston, Massachusetts, EUA: ADDISON WESLEY, 1997–1998.

# Plano de Ensino – ED2

## ▶ RECURSOS

- Simulador *Data Structure Visualizations* – disponível em <http://www.cs.usfca.edu/~galles/visualization/Algorithms.html>
- Linguagem de Programação (C++ ou Java)

## ▶ SISTEMA DE AVALIAÇÃO

- Momento N1:
  - 1ª avaliação: 03/04, 20pts (Projeto: ordenação e análise – Bubble, Selection e Insertion);
  - Atividade de 5pts: aplicação na prática; Uniube+: 5ptos
- Momento N2:
  - 2ª avaliação: 22/05, 20pts (Projeto: ordenação e análise – Quick e Merge);
  - Atividade de 5pts: aplicação na prática; Uniube+: 5ptos
- Momento N3:
  - 3ª avaliação: 25/06, 20pts (Projeto: pesquisa e análise – Quick e Merge);
  - Atividade de 5pts: aplicação na prática; Uniube+: 5ptos; Avaliação Institucional: 18/06, 10pts

# Revisão

## Estruturas de Dados 1

# Estruturas de Dados

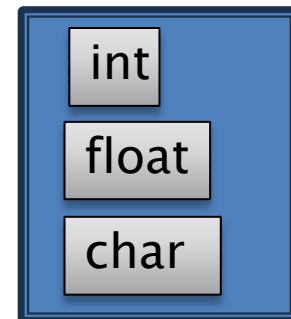
- Estruturas de dados (*data structures*) servem para organizar os dados de um problema de modo que eles possam ser processados mais eficientemente.
- **Eficiência tem a ver com *escalabilidade*:**
  - Como o tempo de processamento cresce quando a quantidade de dados (ou seja, o tamanho) do problema aumenta?
  - O tempo cresce de maneira moderada?
  - Cresce de maneira explosiva?
- **Exemplo:**
  - Qual a diferença entre encontrar uma determinada palavra em uma lista de 10 palavras e encontrar a palavra em uma lista de 10000 palavras?
  - O tempo de processamento é apenas dez vezes maior?
  - É mil vezes maior?
  - Ou é um milhão de vezes maior?



# Tipos de Dados

- Caracteriza o conjunto de valores a que uma constante pertence, ou que podem ser assumidos por uma variável ou expressão, ou que podem ser gerados por uma função.
- Tipos simples de dados:
  - ▶ os tipos básicos `int`, `boolean`, `char` e `float` de Java.
  - ▶ Ex: uma variável do tipo *boolean* pode assumir o valor verdadeiro ou o valor falso, e nenhum outro valor.
- Os tipos estruturados de dados em geral definem uma coleção de valores simples, ou um agregado de valores de tipos diferentes.

Trata-se de um relacionamento  
lógico entre tipos de dados.



# Tipo Abstrato de Dados, TAD (ou ADT – *Abstract Data Type*)

1. Arrays

2. Strings

3. Pilhas

4. Filas

5. Listas

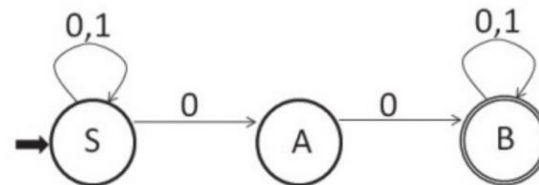
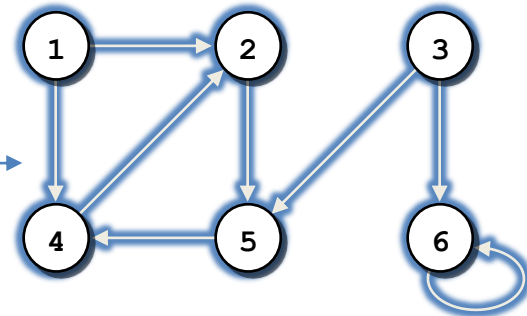
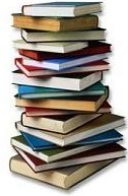
6. Árvores

7. Grafos

8. Autômatos

9. Tabelas Hash

10. ...



# Implementação de TADs

- ❖ Como as estruturas de dados funcionam por baixo dos panos?
- ❖ Qual é o tempo de execução e performance dessas estruturas?
- ❖ Qual a **solução mais viável**?

## ➤ Armazenamento sequencial em Vetores

0	1	2	3	4	5	6	7	8	9	...
João	José	Carla	Danilo	Pedro	null	null	null	null	null	null

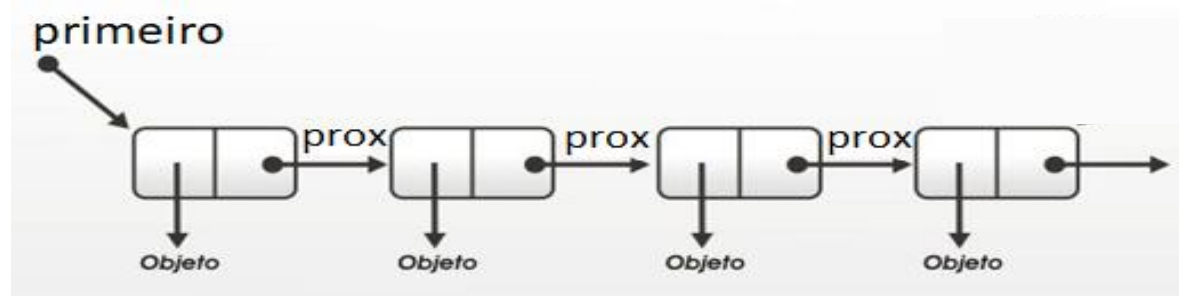
- Inserção e remoção de dados em vetor – **nada performático**

## ➤ Listas ligadas ou Listas dinâmicas

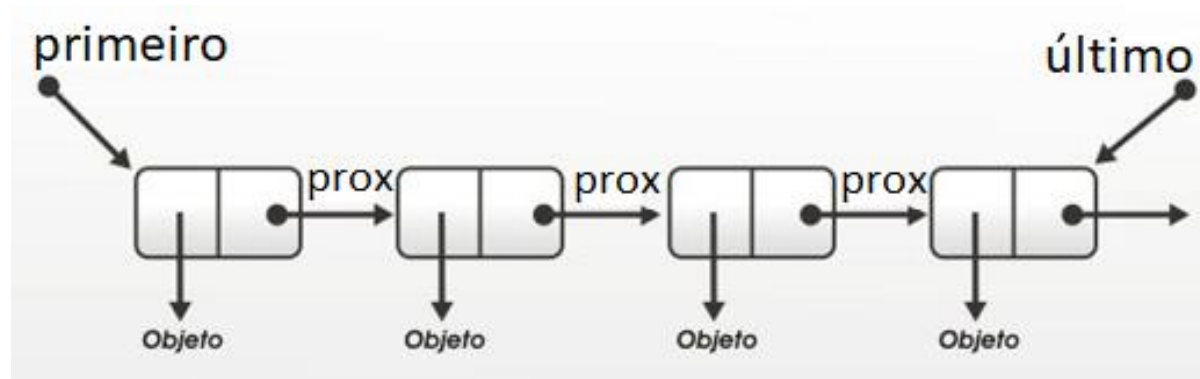


# Lista Linear Simplesmente Encadeada

- Com o endereço do primeiro elemento da lista



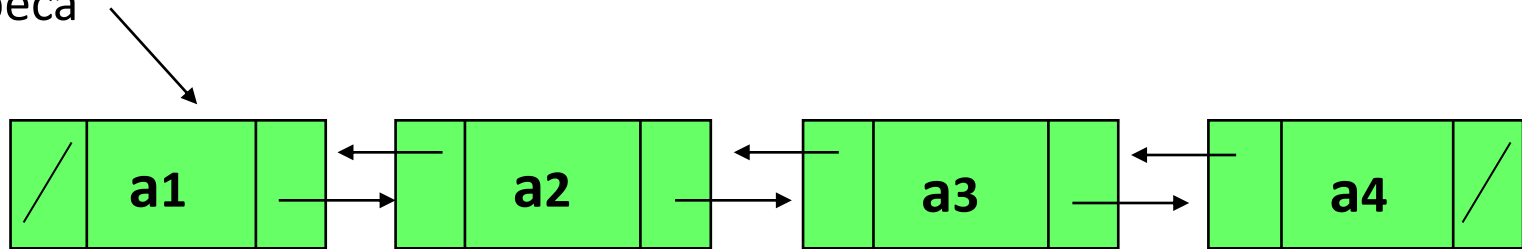
- Com os endereços do primeiro e do último elementos da lista



# Lista Linear Duplamente Encadeada

- Com o endereço do primeiro elemento da lista

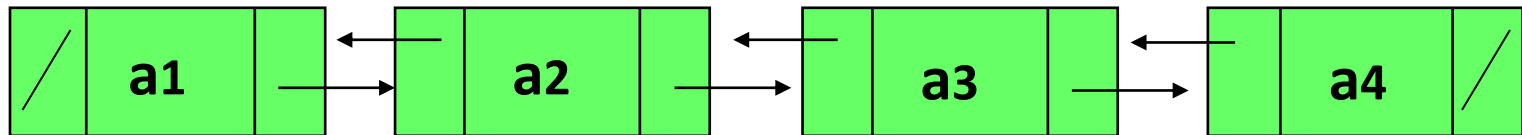
cabeca



- Com os endereços do primeiro e do último elementos da lista

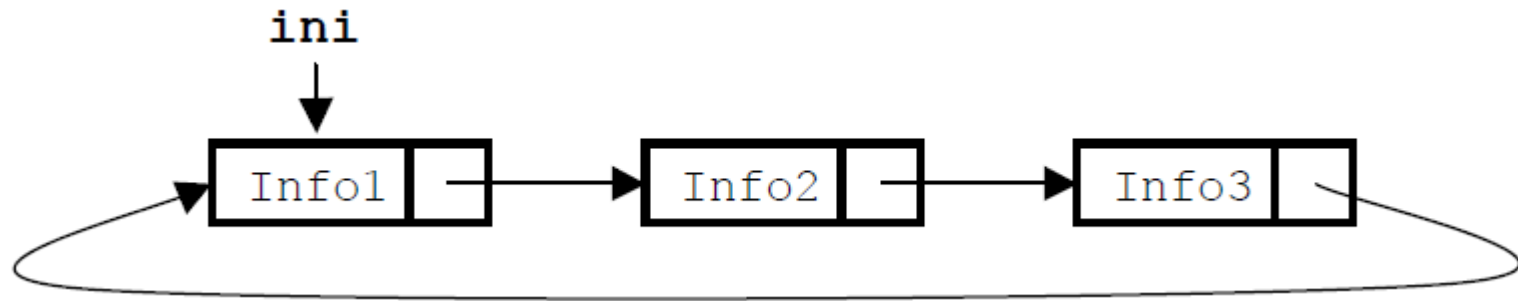
primeiro

ultimo

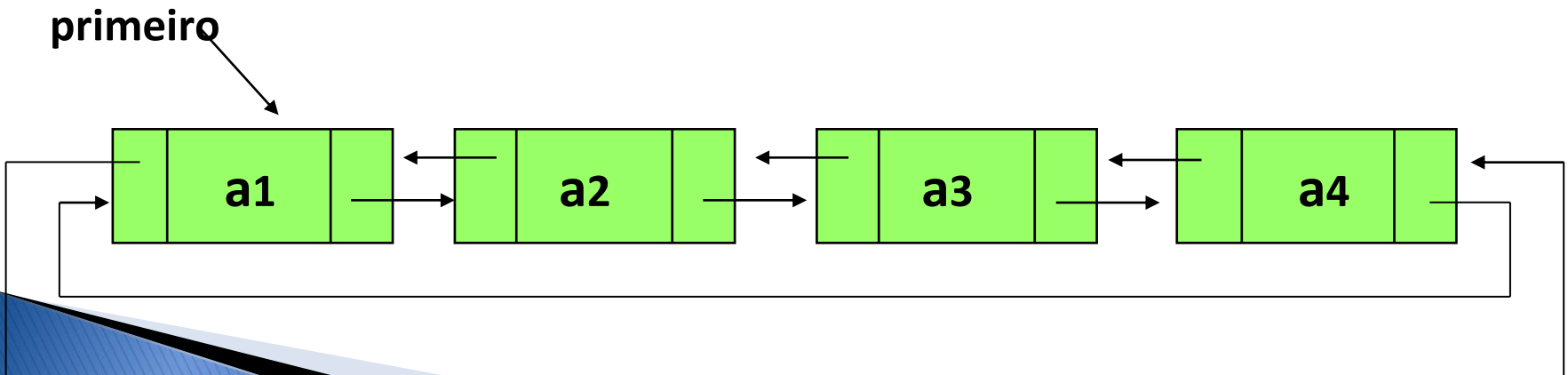


# Lista Circular

- Lista simplesmente encadeada circular



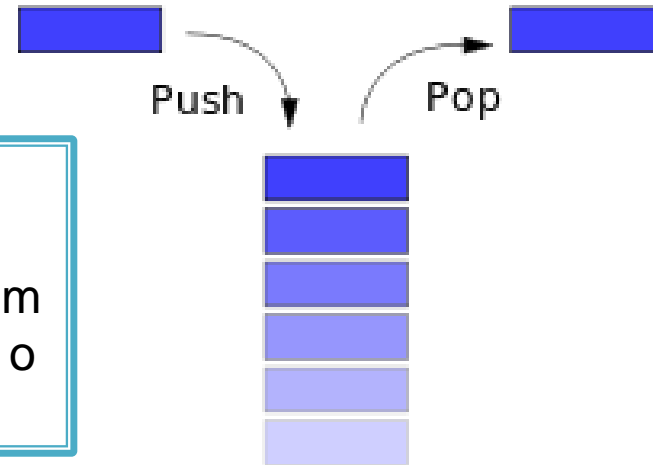
- Lista duplamente encadeada circular



# Pilha

## Estratégia LIFO (*Last in, first out*)

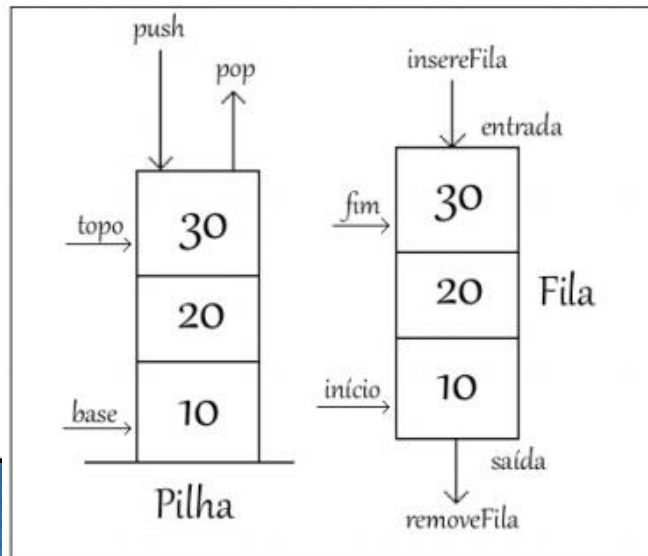
Os elementos da pilha são retirados na ordem inversa à ordem em que foram introduzidos: o último a entrar é o primeiro a sair.



# Fila

## Estratégia FIFO (*Fist in, first out*)

A estrutura de fila é análoga ao conceito que temos de filas em geral. O primeiro a chegar é sempre o primeiro a sair, e a entrada de novos elementos sempre se dá no fim da fila.

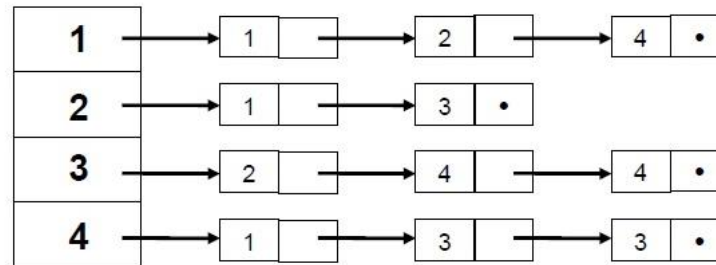
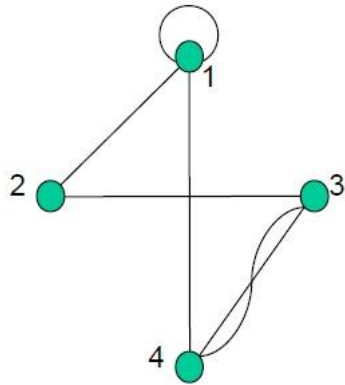


Representação de uma Fila e uma Pilha

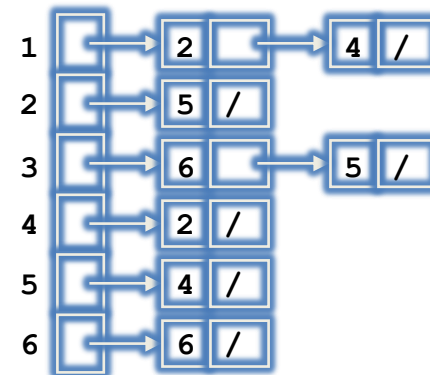
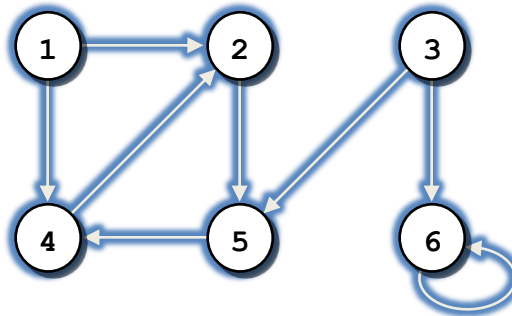
# Grafo:

## Lista de adjacência ou dicionário

### Primeiro caso: Grafo Não-Orientado



### Segundo caso: Grafo Orientado ou Dígrafo

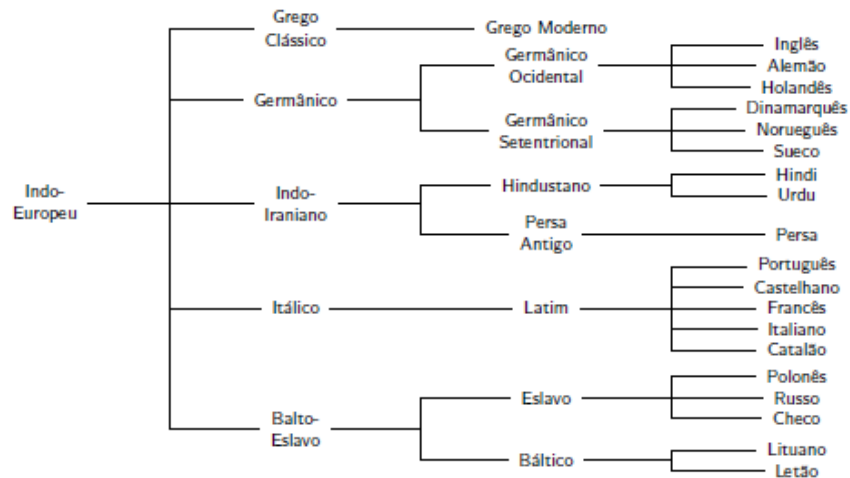


Obs.: No lugar do vetor {1,2,3,4,5,6} pode-se utilizar uma outra lista encadeada.



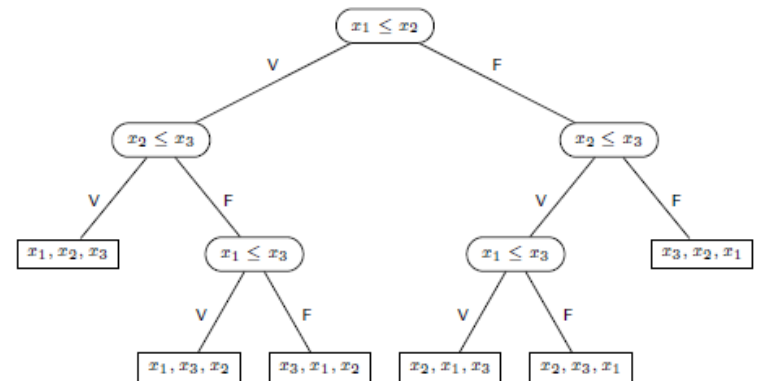
# Árvores

## descendentes



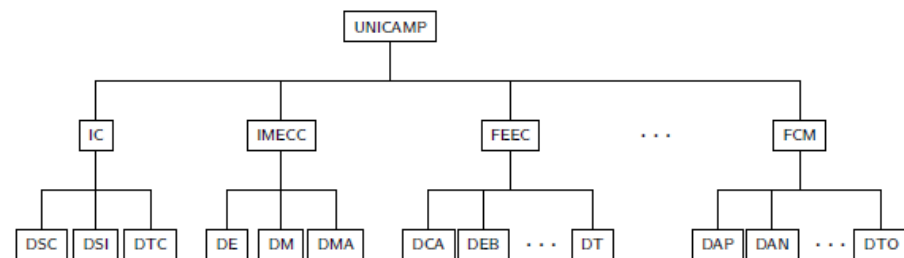
- ▶ A árvore é incompleta e não necessariamente exata.
- ▶ Cada elemento pode ter um número variável de sucessores: *árvore geral*.

## árvore de decisão



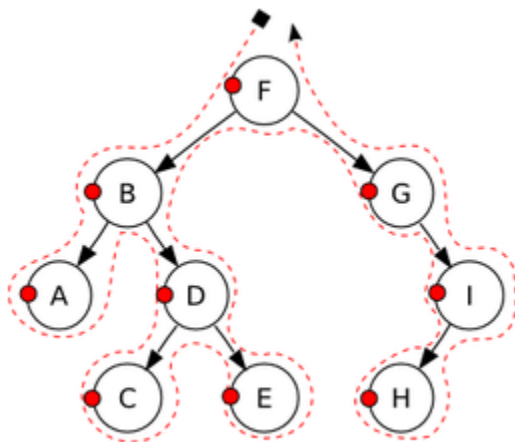
- ▶ A árvore representa as decisões tomadas por um algoritmo de ordenação usando operações de comparação; V: verdadeiro, F: falso.
- ▶ Devido à natureza das comparações, a árvore é binária.

## organograma

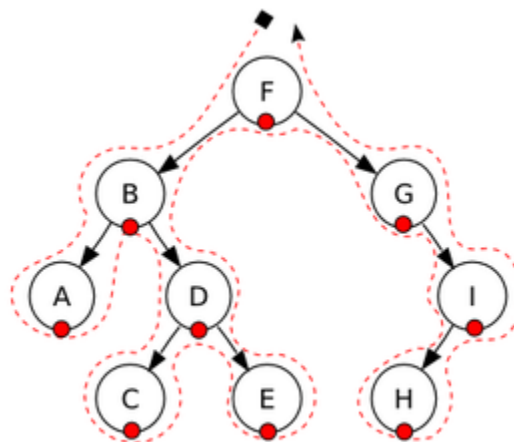


Obs.: A UNICAMP tem 21 unidades acadêmicas. Algumas unidades têm mais de 10 departamentos.

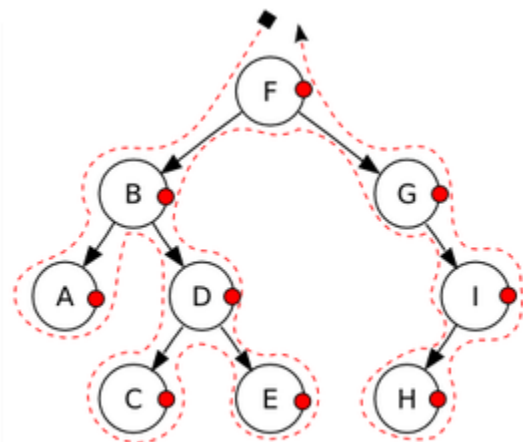
# Árvores Binárias



**Pré-ordem:** F, B, A, D, C, E, G, I, H



**Ordem simétrica:** A, B, C, D, E, F, G, H, I



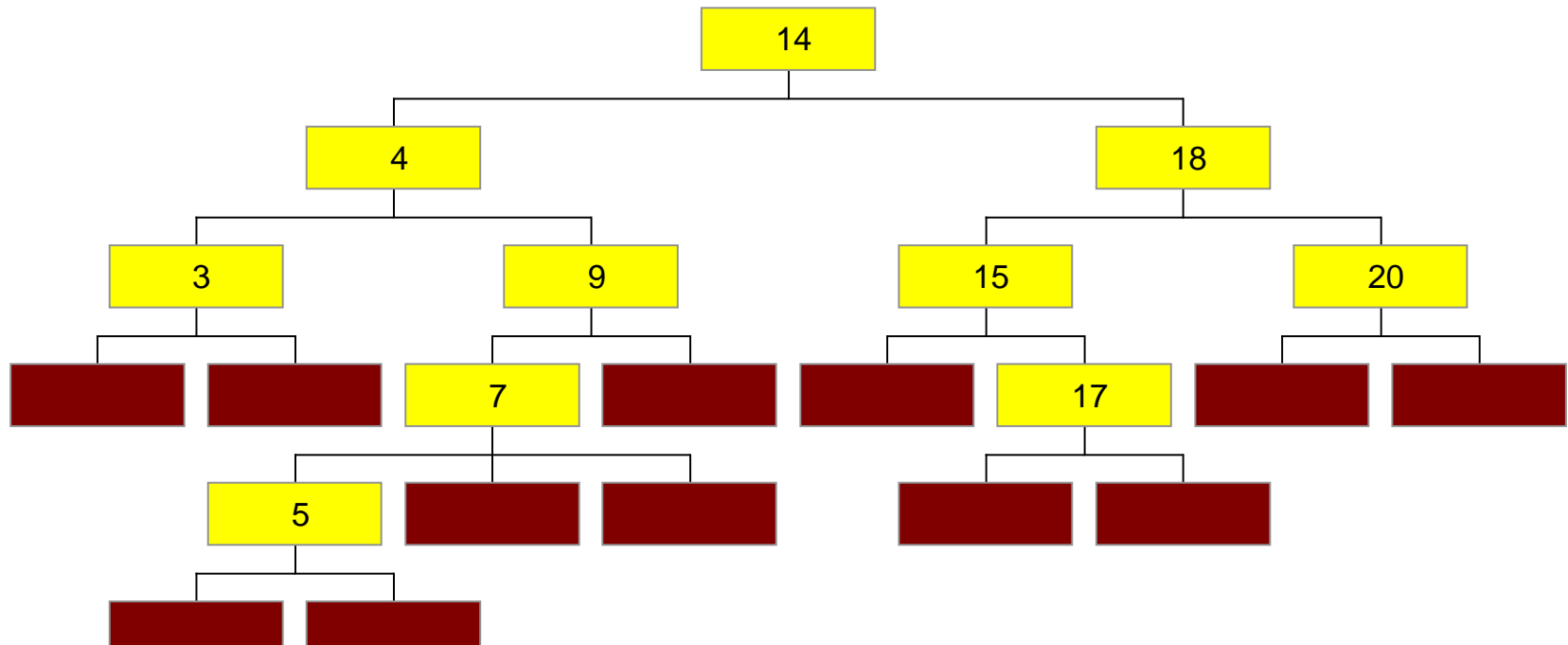
**Pós-ordem:** A, C, E, D, B, H, I, G, F

Percursos

# Árvores Binárias

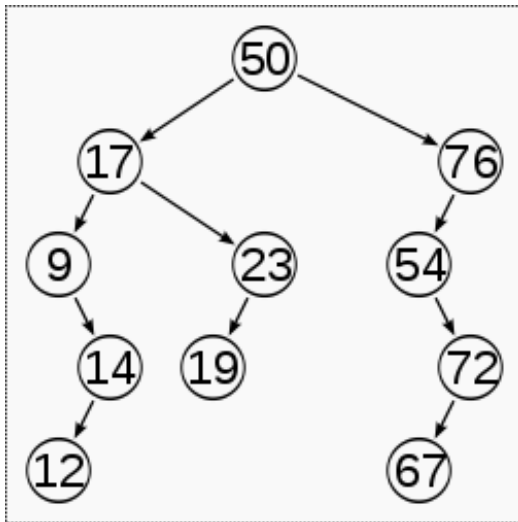
## Aplicações – ordenação de dados

14, 18, 4, 9, 7, 15, 3, 5, 17, 4, 20, 9, 5

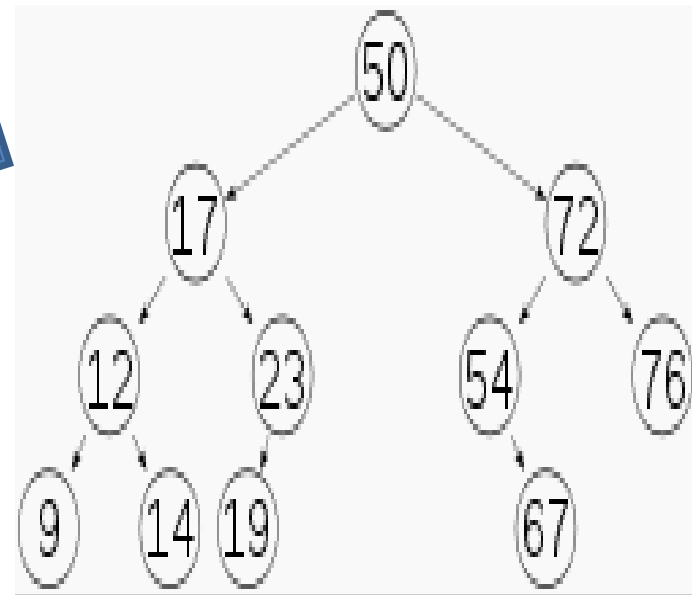


# Árvore AVL

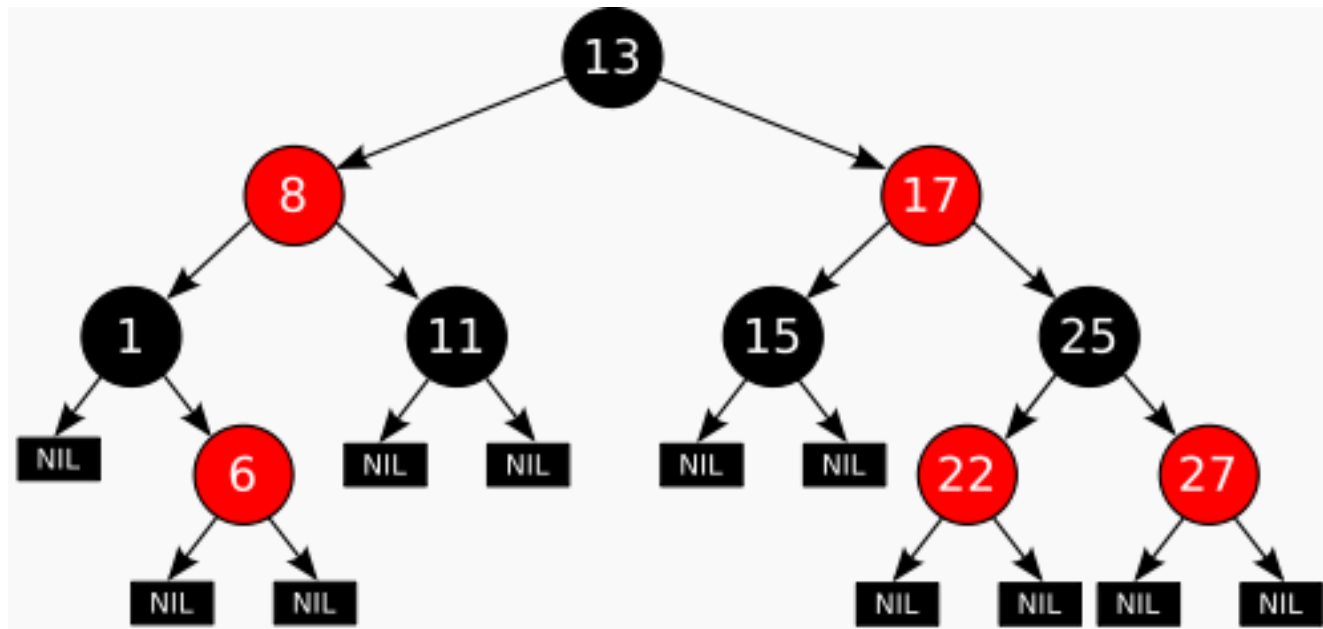
## Árvore não AVL



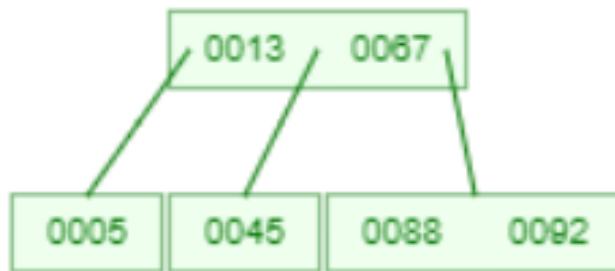
**A mesma árvore após balanceamento por altura, agora AVL:**



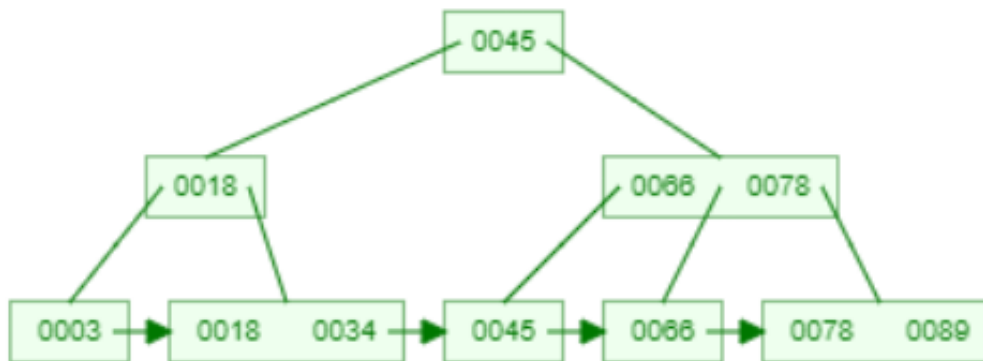
# Árvore Rubro-negra



# Árvores B e $B^+$



Exemplo de B-Tree



Exemplo de B+-Tree

# Prática

- ▶ Faça uso do simulador *Data Structure Visualizations* – disponível em <http://www.cs.usfca.edu/~galles/visualization/Algorithms.html> para a visualização das estruturas apresentadas.
- ▶ Construa a B-tree com os seguintes valores inseridos nesta ordem:  
1 12 8 2 25 5 14 28 17 7 52 16 48 68 3 26 29 53 55 4  
5.

Escolha você um grau para a árvore B e informe-o antes da sua construção.

## Conceitos básicos:

### ❖ Ordenação

- ❖ Ato de colocar um conjunto de dados em uma determinada ordem predefinida

### ❖ Fora de ordem: 5, 2, 1, 3, 4

### ❖ Ordenado: 1, 2, 3, 4, 5 OU 5, 4, 3, 2, 1

### ❖ Algoritmo de ordenação

- ❖ Coloca um conjunto de elementos em uma certa ordem

# Ordenação

## Estruturas de Dados 2

- Definir o problema da ordenação
- Diferenciar os métodos de classificação interna e externa
- Analisar a complexidade dos algoritmos de ordenação