

# Seminário I - Estruturas de Dados Baseadas em Árvores

## Heaps (Capítulo 6)

### Definição e Tipos

- **Heap:** Árvore binária completa com propriedade específica:
  - **Max-heap:** Chave do pai  $\geq$  chave dos filhos.
  - **Min-heap:** Chave do pai  $\leq$  chave dos filhos.
- **Representação:**
  - Armazenado em **array** (índice 0 ou 1).
  - Atributo `heap_size` indica último nó válido.

### Operações Principais

Operação	Descrição	Complexidade
MAX-HEAPIFY	Corrige subárvore violando max-heap.	$O(\log n)$
BUILD-MAX-HEAP	Converte array arbitrário em max-heap.	$O(n)$
HEAPSORT	Ordenação usando heap.	$O(n \log n)$
INSERT / EXTRACT-MAX	Operações para filas de prioridade.	$O(\log n)$

---

## Outras Estruturas Baseadas em Árvores

### 1. Árvores Enraizadas (Capítulo 10)

- **Representação:**
  - Listas ligadas ( `left-child` , `right-sibling` ).
- **Aplicação:** Hierarquias genéricas.

### 2. Árvores de Busca Binária (BSTs - Capítulo 12)

- **Propriedade:**
  - Filho esquerdo  $\leq$  nó  $\leq$  filho direito.
- **Operações** (tempo  $O(h)$  ):
  - Busca, mínimo/máximo, inserção, exclusão.

- **Representação:**
  - Ponteiros para `pai`, `esquerdo`, `direito`.

### 3. Árvores Vermelho-Pretas (Capítulo 13)

- **Balanceamento:**
  - Altura garantida  $O(\log n)$ .
- **Operações:**
  - Inserção/exclusão com rotações e ajustes de cor.
- **Complexidade:**  $O(\log n)$  no pior caso.

### 4. Aumentação de Estruturas (Capítulo 17)

- **Exemplo:**
  - Árvores vermelho-pretas com tamanho da subárvore.
- **Operações Adicionais:**
  - Estatísticas de ordem (`SELECT` em  $O(\log n)$ ).

### 5. B-Trees (Capítulo 18)

- **Propósito:**
  - Otimizado para armazenamento secundário (discos).
- **Propriedades:**
  - Fator de ramificação  $t$  (altura  $O(\log_t n)$ ).
- **Operações:**
  - Busca/inserção/exclusão com divisão/fusão de nós.

### 6. Estruturas de Conjunto Disjunto (Capítulo 19)

- **Representação:**
  - Florestas de árvores.
- **Otimizações:**
  - União por *rank* + compressão de caminho.
- **Complexidade Amortizada:**
  - Quase constante ( $\alpha(n)$  inversa de Ackermann).

### 7. Árvores em Algoritmos de Grafos (Parte VI)

Tipo	Algoritmo Associado	Aplicação
Árvores BFS/DFS	Busca em largura/profundidade	Exploração de grafos.
MST (Kruskal/Prim)	Árvore de extensão mínima	Redes de conexão ótimas.

Tipo	Algoritmo Associado	Aplicação
Florestas de Fluxo	Algoritmo de Ford-Fulkerson	Caminhos aumentantes.

---

## Análise

- **Abordagem do Livro:**
  - Descrição clara de operações via **pseudocódigo**.
  - Foco em:
    - Manipulação de ponteiros/arrays.
    - Manutenção de invariantes (ex: propriedade do heap).
- **Análise de Complexidade:**
  - Relacionada à **altura da árvore** ( $h$ ).
  - Exemplos:
    - BSTs desbalanceadas:  $h = O(n)$ .
    - Red-Black/B-Trees:  $h = O(\log n)$ .