

# Aulas 09-12: Árvores e Grafos

## Aula 09: Árvore Binária de Busca

**Objetivos:** Compreender conceitos de árvores binárias, implementar BST, dominar operações básicas.

### Conceitos Fundamentais

- **Árvore:** Estrutura hierárquica com nós conectados
- **Árvore Binária:** Cada nó tem no máximo 2 filhos
- **BST:** Propriedade de ordenação (esquerda < raiz < direita)

### Implementação Básica

```
#include <stdio.h>
#include <stdlib.h>

typedef struct No {
    int data;
    struct No *esquerda, *direita;
} No;
```

# Aula 10: Percurso em Árvores Binárias

**Objetivos:** Dominar algoritmos de travessia, implementar versões recursivas e iterativas.

## Tipos de Percurso

### 1. Em Ordem (In-order)

```
void inorder(No *raiz) {  
    if (raiz != NULL) {  
        inorder(raiz->esquerda);  
        printf("%d ", raiz->data);  
        inorder(raiz->direita);  
    }  
}
```

### 2. Pré-ordem (Pre-order)

```
void preorder(No *raiz) {  
    if (raiz != NULL) {
```

# Aula 11: Balanceamento de Árvore e Árvore AVL

**Objetivos:** Compreender problemas de desbalanceamento, implementar árvores AVL.

## Conceitos de Balanceamento

- **Fator de Balanceamento:** altura(direita) - altura(esquerda)
- **Árvore AVL:**  $|\text{fator de balanceamento}| \leq 1$  para todos os nós

## Rotações

### Rotação Simples à Direita

```
No* rotacao_direita(No *y) {  
    No *x = y->esquerda;  
    No *T2 = x->direita;  
  
    // Rotação  
    x->direita = y;  
    y->esquerda = T2;
```

# Aula 12: APS (Grafos) - Conceitos e Representação

**Objetivos:** Compreender conceitos básicos de grafos, implementar representações.

## Definições Básicas

- **Grafo:**  $G = (V, E)$  onde  $V$  são vértices e  $E$  são arestas
- **Grafo Direcionado:** Arestas têm direção
- **Grafo Não-Direcionado:** Arestas bidirecionais
- **Peso:** Valor associado à aresta

## Representação por Matriz de Adjacência

```
#define MAX_VERTICES 100

typedef struct {
    int matriz[MAX_VERTICES][MAX_VERTICES];
    int num_vertices;
} GrafoMatriz;

void inicializar_matriz(GrafoMatriz *g, int vertices) {
```