

Etapa 5 - árvores

Árvores em Estrutura de Dados - Guia Didático

O que são Árvores?

Imagine uma árvore genealógica ou a estrutura de pastas do seu computador. As árvores em programação funcionam de forma similar:

- São estruturas hierárquicas (com diferentes níveis)
- Cada elemento tem um "pai" e pode ter "filhos"
- Formam uma árvore invertida - a raiz fica no topo!

Partes de uma Árvore

Nó (Node)

- O que é: Cada "caixinha" que armazena um dado
- Como funciona: Pode ter conexões para outros nós

Raiz (Root)

- O que é: O nó principal, no topo de tudo
- Característica: Não tem "pai" - é o ancestral de todos

Folhas (Leaves)

- O que é: Os nós que estão nas pontas
- Característica: Não têm "filhos" - são os últimos da linhagem

Galhos (Edges)

- O que é: As "linhas" que conectam os nós
- Função: Mostram as relações entre pais e filhos

Altura

- O que é: O "tamanho" da árvore
- Como medir: Maior distância da raiz até uma folha

Tipos de Árvores Mais Importantes

1. Árvore Binária

...

```
[A]
 /\
[B] [C]
 /\  \
[D] [E] [F]
...
```

- Regra: Cada nó pode ter no máximo 2 filhos
- Filhos: Esquerdo e direito
- Uso: Calculadoras (expressões matemáticas)

2. Árvore Binária de Busca (BST)

...

```
    [50]
   /  \
  [30] [70]
 / \  / \
[20][40][60][80]
```

...

- Regra de Ouro:

- Todos à esquerda são menores
- Todos à direita são maiores
- Vantagem: Busca super rápida!
- Performance: $O(\log n)$ - divide o problema pela metade a cada passo

3. Árvores Balanceadas (AVL e Rubro-Negra)

...

```
    AVL:      Rubro-Negra:
    [B]      [B]
   / \     / \
  [A] [C]  [A] [C]
```

...

- Problema: Árvores podem ficar "desbalanceadas"
- Solução: AVL e Rubro-Negra se auto-ajustam
- Resultado: Sempre mantêm buscas eficientes

4. Heap (Montículo)

...

```
    Heap Máximo:  Heap Mínimo:
    [100]         [10]
   /  \         /  \
  [80] [90]     [20] [30]
 /      \     /
[70]     [40]
```

...

- Heap Máximo: Pai sempre maior que os filhos
- Heap Mínimo: Pai sempre menor que os filhos
- Uso Principal: Filas de prioridade (urgências)

Como Percorrer Árvores?

Pré-Ordem (Pré-Order)

Ordem: Raiz -> Esquerda -> Direita

Como lembrar: Visita o nó ANTES dos filhos

Em-Ordem (In-Order)

Ordem: Esquerda -> Raiz -> Direita

Como lembrar: Visita entre os filhos

Resultado: Valores em ordem crescente!

Pós-Ordem (Post-Order)

Ordem: Esquerda -> Direita -> Raiz

Como lembrar: Visita o nó DEPOIS dos filhos

Vantagens das Árvores

Velocidade

- Busca rápida: $O(\log n)$ em árvores balanceadas
- Comparação: Muito mais rápido que lista ($O(n)$)

Organização Natural

- Hierarquias: Representam estruturas reais
- Flexibilidade: Fácil adicionar/remover elementos

Eficiência

- Divisão e conquista: Problemas divididos em subproblemas
- Ordenação automática: BST mantém dados ordenados

Aplicações no Mundo Real

Sistemas de Arquivos

``

C:/

|-- Documentos/

| |-- Trabalho/

| |-- Pessoal/

|-- Programas/

``

Bancos de Dados

- Índices: Encontram dados rapidamente
- Exemplo: Encontrar um cliente entre milhões em segundos

Compiladores

- Árvores sintáticas: Analisam código fonte
- Exemplo: Verificar se seu programa tem erros de sintaxe

Internet e Redes

- Roteamento: Encontrar melhor caminho entre computadores
- DNS: Encontrar endereços de sites

Inteligência Artificial

- Árvores de decisão: Tomadas de decisão automatizadas
- Jogos: IA de xadrez avalia milhões de jogadas

Aplicações do Dia a Dia

- Autocompletar: Busca preditiva em celulares
- Organizadores: Gerenciamento de tarefas e projetos

Como Estudar Árvores?

Iniciante

1. Entenda Árvores Binárias simples
2. Aprenda BST - a mais útil no dia a dia
3. Pratique os três tipos de travessia

Intermediário

1. Domine árvores balanceadas (AVL)
2. Aprenda Heaps e filas de prioridade
3. Estude algoritmos de balanceamento

Avançado

1. Explore árvores B e B+ (bancos de dados)
2. Aprenda sobre árvores de segmento
3. Estude árvores para problemas específicos

Dica Final

Pense nas árvores como "organizadores inteligentes" de dados. Elas tornam a busca e organização de informações muito mais eficiente, especialmente quando temos grandes quantidades de dados para gerenciar.

A chave para dominar árvores é praticar a construção e travessia de diferentes tipos, sempre prestando atenção em como os dados são organizados em cada estrutura.

Pergunta 1: Propriedade Fundamental

Qual é a propriedade fundamental de uma Árvore Binária de Busca (BST)?

- a) É sempre completa
- b) Todos os níveis estão completamente preenchidos
- c) Para qualquer nó, todos os valores na subárvore esquerda são menores e na direita são maiores
- d) Não pode ter mais de dois filhos por nó

Resposta: c) Para qualquer nó, todos os valores na subárvore esquerda são menores e na direita são maiores

Pergunta 2: Navegação em Árvores

Qual tipo de travessia em uma árvore binária visita os nós na ordem: esquerda, raiz, direita?

- a) Pré-ordem

- b) Em-ordem
- c) Pós-ordem
- d) Por nível

Resposta: b) Em-ordem