

Algoritmos e Estrutura de Dados na Prática

Algoritmos e Estrutura de Dados na Prática

Autor: Sérgio Santos

Sumário

1. Introdução
2. O que são Algoritmos?
3. Variáveis, Tipos e Operadores
4. Condicionais e Laços de Repetição
5. Funções e Modularização
6. Vetores e Matrizes
7. Estruturas de Dados Lineares: Listas, Pilhas e Filas
8. Árvores e Grafos (Introdução)
9. Algoritmos de Ordenação e Busca
10. Análise de Complexidade (Big-O)
11. Projeto Final: Agenda Inteligente com Listas Ligadas
12. Recursos Extras e Próximos Passos

Algoritmos e Estrutura de Dados na Prática

Capítulo 1 - Introdução

A lógica de programação e o domínio de estruturas de dados são as bases para resolver problemas computacionais de forma eficaz. Este eBook irá guiá-lo passo a passo pelos fundamentos dos algoritmos, tipos de estruturas de dados e sua implementação prática em Python.

Capítulo 2 - O que são Algoritmos?

Algoritmos são sequências finitas de passos bem definidos que resolvem um problema. Exemplo: uma receita de bolo é um algoritmo do mundo real. Em programação, algoritmos são usados para realizar tarefas como ordenação, busca e cálculos.

Capítulo 3 - Variáveis, Tipos e Operadores

Variáveis armazenam valores. Tipos comuns em Python: int, float, str, bool. Operadores: +, -, *, /, %, ==, !=, >, <, and, or.

Capítulo 4 - Condicionais e Laços de Repetição

Condicionais: if, elif, else. Permitem decisões.

Laços: while e for. Permitem repetições. Exemplo:

```
for i in range(5):  
    print(i)
```

Capítulo 5 - Funções e Modularização

Funções agrupam comandos reutilizáveis:

```
def saudacao(nome):  
    print(f"Olá, {nome}!")
```

Modularização é dividir o código em módulos reutilizáveis.

Algoritmos e Estrutura de Dados na Prática

Capítulo 6 - Vetores e Matrizes

Vetores (listas) armazenam dados sequenciais. Matrizes são listas de listas:

```
matriz = [[1,2],[3,4]]  
print(matriz[0][1])
```

Capítulo 7 - Estruturas de Dados Lineares: Listas, Pilhas e Filas

Listas são coleções mutáveis. Pilhas (LIFO) e Filas (FIFO) controlam ordem de inserção e remoção.

Exemplo:

```
from collections import deque  
fila = deque()  
fila.append('cliente')  
fila.popleft()
```

Capítulo 8 - Árvores e Grafos (Introdução)

Árvores têm estrutura hierárquica (raiz, nós, folhas). Grafos são conjuntos de nós conectados por arestas. Representam redes, mapas, conexões.

Capítulo 9 - Algoritmos de Ordenação e Busca

Ordenação: bubble sort, insertion sort, quick sort. Busca: linear e binária. Exemplo de busca binária:

```
def busca_binaria(lista, valor): ...
```

Capítulo 10 - Análise de Complexidade (Big-O)

Big-O mede a eficiência dos algoritmos:

$O(1)$ constante, $O(n)$ linear, $O(n^2)$ quadrático, etc. Ajuda a escolher a melhor solução para grandes entradas.

Algoritmos e Estrutura de Dados na Prática

Capítulo 11 - Projeto Final: Agenda Inteligente com Listas Ligadas

Lista ligada é uma estrutura dinâmica com nós encadeados:

class No:

```
def __init__(self, dado):  
    self.dado = dado  
    self.proximo = None
```

Permite criar uma agenda onde contatos podem ser inseridos, buscados e removidos.

Capítulo 12 - Recursos Extras e Próximos Passos

Estude mais com livros (Cormen, Goodrich), cursos online (Coursera, edX), pratique em sites (HackerRank, LeetCode) e continue evoluindo com projetos próprios.