

alpha

<ed/tech>

Python
Estrutura de Dados

<Módulo 07 />

Estrutura de Dados

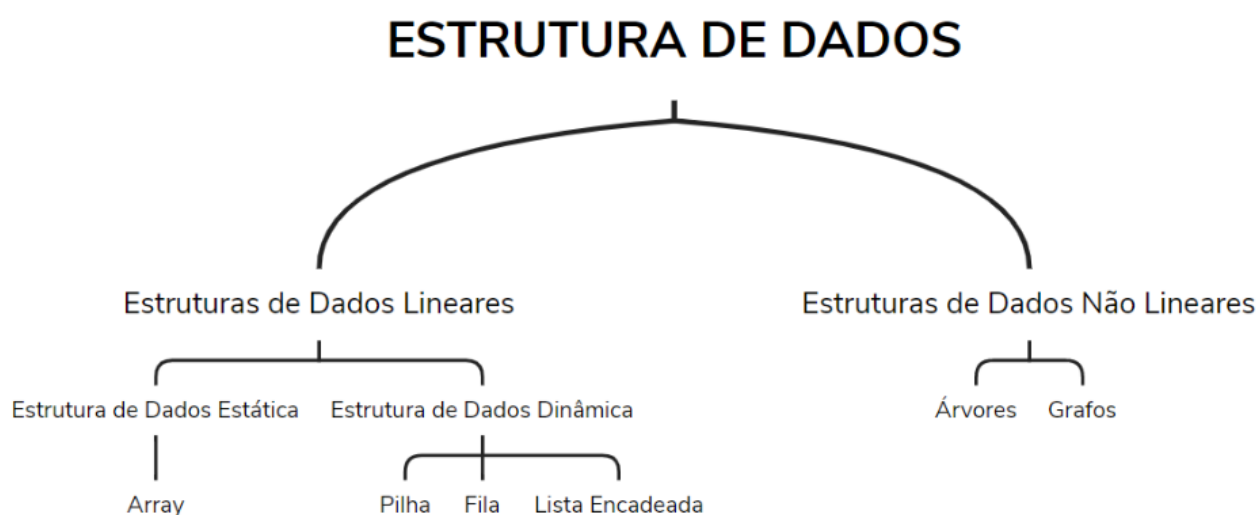
Introdução

Estruturas de dados são maneiras de organizar, processar, retornar e armazenar dados de forma que possam ser acessados e trabalhados de maneira eficiente. Elas são fundamentais para a criação de algoritmos eficientes e para o gerenciamento de grandes volumes de dados.

Importância

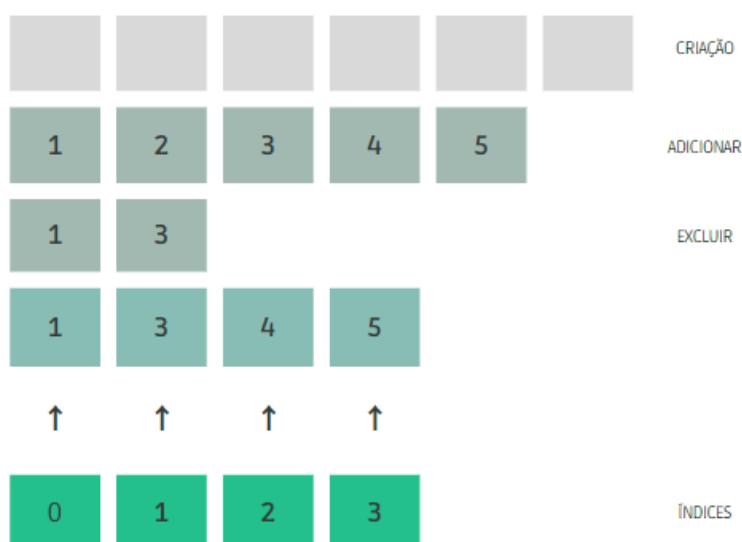
- Facilitam a organização e o gerenciamento de dados.
- Tornam os programas mais eficientes e rápidos.
- Permitem a manipulação de dados complexos e grandes volumes de informação.

Tipos de Estruturas de Dados



Os dados organizados de forma sequencial são chamados de Estruturas de Dados Lineares, sendo:

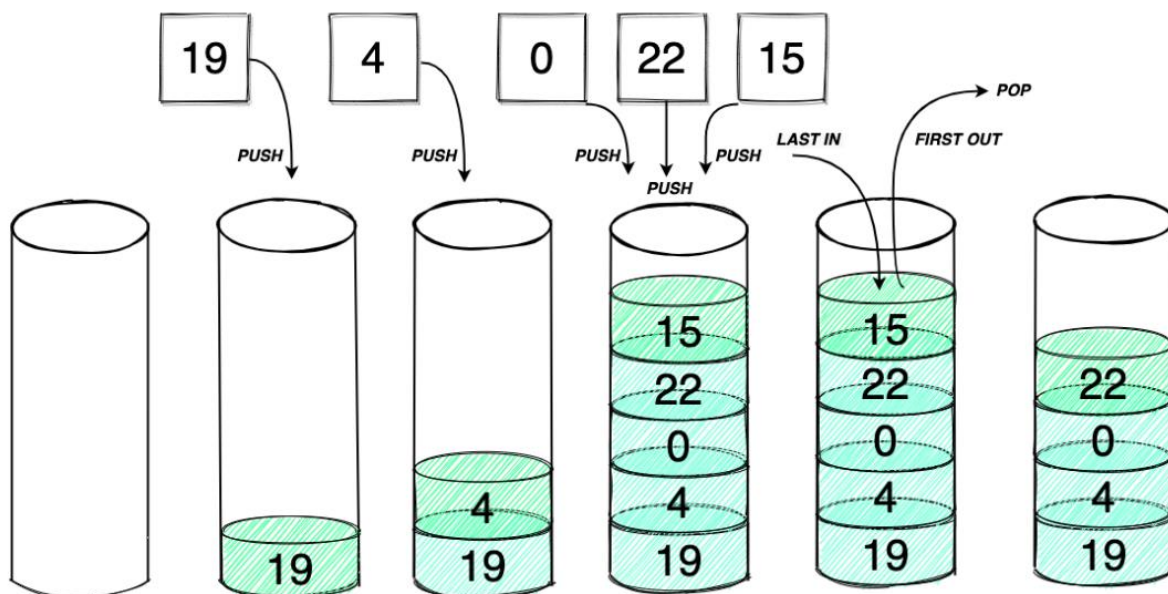
- **Arrays:** Coleção de elementos do mesmo tipo, acessíveis por índices.



- **Lista Encadeada:** Sequência de elementos, onde cada um aponta para o próximo.



- **Pilhas:** Estrutura LIFO (*Last In, First Out*), como uma pilha de livros.

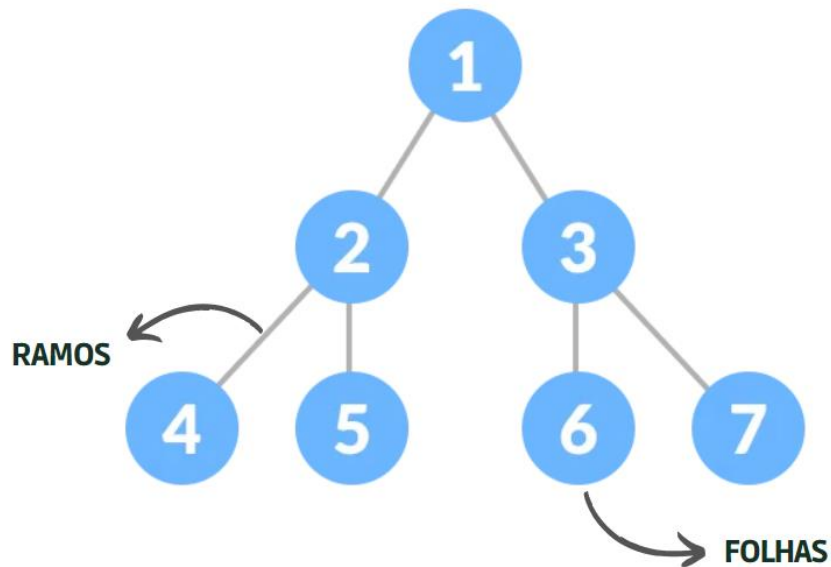


- **Filas:** Estrutura FIFO (*First In, First Out*), como uma fila de supermercado.

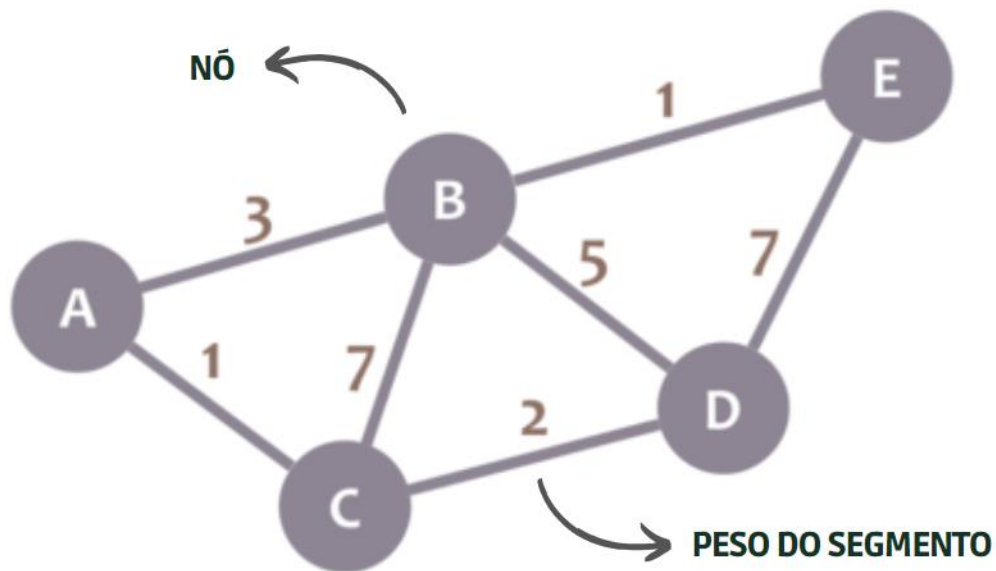


Já os dados que não possuem uma organização sequencial, são denominados Estruturas de Dados Não Lineares, e idealmente são utilizados para representar relações hierárquicas ou em rede.

- **Árvores:** Estrutura hierárquica, com um elemento raiz e sub-árvores.



- **Grafos:** Conjunto de nós conectados por arestas, representando relações entre elementos.



Comparação entre Diferentes Tipos

Quando trabalhamos com estruturas de dados, existem alguns que são mais comumente utilizados, sendo: Arrays, Listas Encadeadas, Pilhas e Filas. Mas qual a principal diferença entre estes tipos?

- **Arrays x Listas Encadeadas:** Arrays oferecem acesso rápido a elementos, enquanto listas ligadas permitem inserções e exclusões eficientes.
- **Pilhas x Filas:** Pilhas são ideais para situações que exigem acesso ao último elemento inserido, enquanto filas são adequadas para processos que atendem elementos na ordem de chegada.

Aplicações Práticas de Diferentes Estruturas de Dados

Quando trabalhamos com estruturas de dados, existem alguns que são mais comumente utilizados, sendo: Arrays, Listas Encadeadas, Pilhas e Filas. Mas qual a principal diferença entre estes tipos?

- **Arrays x Listas Encadeadas:** Arrays oferecem acesso rápido a elementos, enquanto listas ligadas permitem inserções e exclusões eficientes.
- **Pilhas x Filas:** Pilhas são ideais para situações que exigem acesso ao último elemento inserido, enquanto filas são adequadas para processos que atendem elementos na ordem de chegada.



Backtracking é uma técnica de programação para **resolver problemas recursivamente**, tentando construir uma solução de forma incremental, item a item. Quando se percebe que a solução atual não pode ser completada para satisfazer todas as condições do problema, o algoritmo descarta a solução parcial e volta (ou "*backtracks*") para uma etapa anterior, tentando uma abordagem diferente. Esse método é frequentemente utilizado em situações onde a busca por soluções exige a exploração de todos os cenários possíveis.

- **Filas:** Aplicadas em gerenciamento de tarefas, operações de buffer, etc.

Introdução à Complexidade de Tempo e Espaço

Ao utilizar estruturas de dados, temos que pensar em três variáveis importantes para nosso código, sendo:

- **Complexidade de Tempo:** Refere-se ao tempo de execução de um algoritmo, que pode variar dependendo do tamanho da entrada e da estrutura de dados usada.
- **Complexidade de Espaço:** Relacionada à quantidade de memória usada pelo algoritmo durante sua execução.
- **Relevância:** A escolha da estrutura de dados correta pode significativamente melhorar a eficiência do programa em termos de tempo e espaço.