

Introdução à Estrutura de Dados

Estrutura de Dados é um conceito fundamental em Ciência da Computação, relacionado com a organização, armazenamento e manipulação de dados. Ela desempenha um papel muito importante no desenvolvimento de algoritmos eficientes e na resolução de problemas complexos. A compreensão e escolha de estruturas de dados adequadas é crucial para que os programadores possam otimizar o desempenho de seus programas, economizando tempo e recursos computacionais.

1. O que é uma Estrutura de Dados?

Uma estrutura de dados é uma forma de organização e armazenamento dos dados com o objetivo de otimizar seu acesso e manipulação. Para isso, é necessário escolher as formas de representação e as operações sobre esses dados capazes de atender as necessidades específicas de uma determinada aplicação.

Cada estrutura de dados possui um conjunto de métodos que pode realizar operações de inserção, exclusão, busca, classificação ou ordenação de elementos de acordo com um conjunto de regras.

Importante: Nenhuma estrutura de dados em específico é capaz de oferecer a melhor alternativa para a resolução de todos os problemas. Por isso, é importante conhecer as diversas estruturas de dados, seus pontos fortes e limitações.

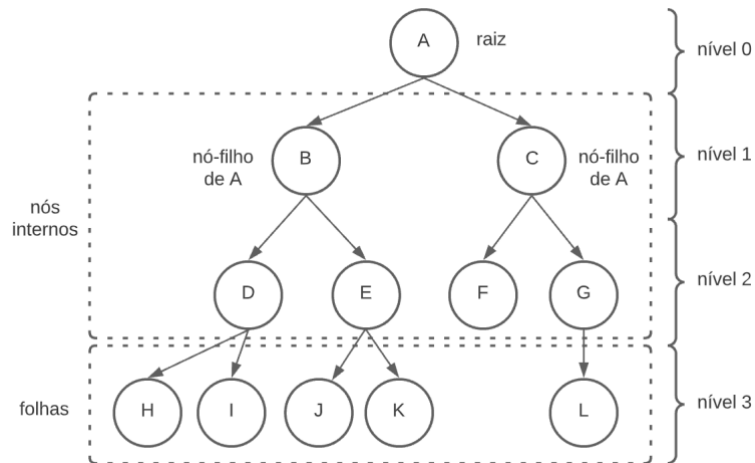
2. Tipos de Estruturas de Dados

A seguir são listados alguns dos principais tipos de estruturas de dados:

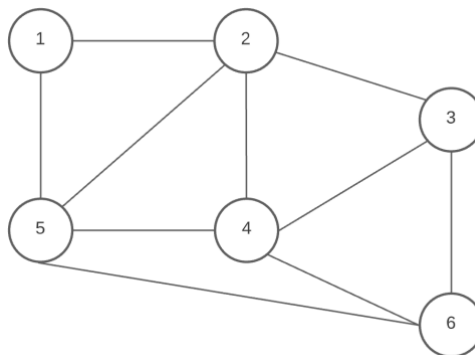
- **Arrays:** Também chamados de vetores ou matrizes, armazenam um conjunto de elementos do mesmo tipo, acessados por índices. São utilizados para armazenar listas ordenadas de elementos, tais como dados numéricos, caracteres, valores de sensores, etc., que podem ser recebidos por meio de uma API (*Application Programming Interface*), banco de dados, arquivos, etc. Os arrays também são utilizados no processamento de imagens para representar uma imagem digital como uma matriz de pixels.
- **Listas Ligadas (*Linked Lists*):** Assim como os arrays, armazenam os elementos sequencialmente. Contudo, os elementos não são armazenados de forma contígua na memória (bloco de memória seguinte), mas por meio de ponteiros, que permitem ao elemento referenciar um endereço de memória qualquer, sem a necessidade de estar no próximo bloco de memória. São elementos encadeados e cada um deles possui um valor e uma referência para o próximo elemento. Listas Ligadas são utilizadas em aplicativos de gerenciamento de memória, editores de texto, gerenciamento do histórico de navegação, álbuns de fotos, *playlists* de músicas ou vídeos.
- **Pilhas (*Stacks*):** Seguem a regra "LIFO" (Last-In-First-Out), na qual o último elemento adicionado é o primeiro a ser removido. Podem ser utilizadas para avaliação de expressões matemáticas, gerenciamento de chamadas de funções em linguagens de programação e sistemas de undo/redo (desfazer/repetir).
- **Filas (*Queues*):** Seguem a regra "FIFO" (First-In-First-Out), na qual o primeiro elemento adicionado é o primeiro a ser removido. Podem ser utilizadas em sistemas de

gerenciamento de tarefas, agendamento de processos em sistemas operacionais, gerenciamento de impressão, gerenciamento de uma fila de banco e/ou restaurante, etc.

- **Árvores (*Trees*):** São estruturas hierárquicas com um nó raiz e nós subsequentes, como árvores binárias, árvores de busca, etc. Elas são aplicadas em algoritmos de tomadas de decisão, nas estruturas de pesquisa de banco de dados e/ou dicionários, na indexação e exibição de arquivos e pastas no explorador de arquivos do sistema operacional, dentre outras situações.



- **Grafos (*Graphs*):** São conjuntos de nós conectados por arestas, que representam relacionamentos entre elementos. O conceito de grafos é utilizado em redes sociais (para modelar conexões entre usuários), sistemas de navegação para determinar as melhores rotas, otimização de redes para planejamento de recursos, etc.



- **Tabelas Hash (*Hash Tables*):** Estruturas que armazenam dados em pares de chave e valor, permitindo acesso rápido a dados usando uma função de *hash*. São utilizadas em bancos de dados, caches de memória, indexação de arquivos e dicionários.

3. Aplicações de Estruturas de Dados

As estruturas de dados estão presentes em diversos tipos de aplicações. Veja alguns exemplos:

- **Bancos de Dados:** Estruturas como árvores B e tabelas *hash* são utilizadas para indexar e organizar dados em bancos de dados.
- **Compiladores:** Pilhas são essenciais para rastrear chamadas de função durante a compilação.
- **Processamento de Imagens:** Matrizes bidimensionais são usadas para representar imagens e aplicar filtros.

- Redes Sociais: Grafos podem representar redes sociais, onde perfis são nós e conexões são arestas.
- Sistemas Operacionais: Filas de processos prontos ou bloqueados são mantidas para gerenciar o escalonamento da CPU.

4. Exercícios

- 4.1. Escolha ao menos duas opções dentre os algoritmos de ordenação listados a seguir e realize as seguintes atividades:
- a) Elabore um breve texto descrevendo o algoritmo, seu propósito de utilização, principais vantagens e desvantagens.
 - b) Crie um exemplo no qual o algoritmo é utilizado.

Algoritmos de ordenação:

- Bubble Sort
- Selection Sort
- Insertion Sort
- Merge Sort

- 4.2. Elabore um programa que realize a ordenação de um array com 5 (cinco) números inteiros. A saída deve contar a ordenação em ordem crescente e decrescente.
- 4.3. Elabore um programa que implemente uma lista de tarefas, possibilitando a inserção de novas tarefas e a impressão da lista.