

# Estrutura de dados - Resumo

...

Prof<sup>a</sup> Letícia Pieper Jandt

# Estruturas de dados

Estrutura de dados é um conceito fundamental na ciência da computação e na programação, que se refere à organização e armazenamento de informações de forma eficiente para que os dados possam ser acessados, manipulados e processados de maneira eficaz. As estruturas de dados desempenham um papel crucial no desenvolvimento de algoritmos e na otimização de desempenho de programas de computador.

## **Definição de Estrutura de Dados:**

- Uma estrutura de dados é uma forma de organizar e armazenar dados em um programa de computador.
- Ela define como os dados são representados, organizados e acessados.

## Objetivos das Estruturas de Dados:

- Melhorar a eficiência na manipulação e acesso aos dados.
- Facilitar a organização e a manutenção de informações.
- Permitir a implementação de algoritmos eficientes.

## Principais Estruturas de Dados:

- Listas: uma coleção ordenada de elementos.
- Pilhas: uma coleção de elementos com operações de inserção e remoção restritas a um extremo.
- Filas: uma coleção de elementos com operações de inserção em um extremo e remoção no outro.
- Árvores: uma estrutura hierárquica composta por nós com ramificações.
- Grafos: uma estrutura de dados composta por vértices e arestas, usada para representar conexões complexas.

## Operações com Estruturas de Dados:

- Inserção: adicionar um elemento à estrutura de dados.
- Remoção: excluir um elemento da estrutura de dados.
- Pesquisa: encontrar um elemento específico na estrutura de dados.
- Atualização: modificar um elemento existente na estrutura de dados.
- Ordenação: reorganizar os elementos em uma ordem específica.

## Uso de Estruturas de Dados:

- As estruturas de dados são amplamente utilizadas na programação para resolver uma variedade de problemas, desde algoritmos de busca e ordenação até sistemas de gerenciamento de bancos de dados.

Em resumo, as estruturas de dados desempenham um papel fundamental na organização e manipulação eficiente de informações em programas de computador.

A escolha adequada de uma estrutura de dados pode ter um impacto significativo no desempenho e na eficiência de um programa, tornando-a uma área essencial para programadores e cientistas da computação entenderem e dominarem.

Uma pilha é uma estrutura de dados segue o princípio de "last-in, first-out" (LIFO), o que significa que o último elemento inserido é o primeiro a ser retirado.

### **Definição:**

- Uma pilha é uma coleção linear de elementos na qual a inserção e a remoção de elementos ocorrem apenas em uma extremidade, chamada de "topo".
- Os elementos são organizados em ordem sequencial, com o elemento mais recentemente adicionado no topo da pilha.

### **Operações básicas:**

- Push: Adicionar um elemento no topo da pilha.
- Pop: Remover o elemento do topo da pilha.
- Shift: Obter o elemento no topo da pilha sem removê-lo.
- Size: Obter o número de elementos na pilha.

### **Aplicações comuns:**

- Modelar a recursão: As pilhas são frequentemente usadas para rastrear chamadas de função em programas recursivos.
- Reverter a ordem de elementos: Pode ser usada para inverter a ordem dos elementos em uma sequência.
- Navegação em árvores: Algoritmos de busca em profundidade (DFS) muitas vezes usam pilhas para rastrear nós em árvores.

### **Considerações:**

- O estouro da pilha (stack overflow) ocorre quando a capacidade da pilha é excedida, o que pode causar falhas no programa.

No geral, as pilhas são uma estrutura de dados útil para resolver uma variedade de problemas em ciência da computação e programação, especialmente aqueles que envolvem controle de fluxo e manipulação de dados de maneira organizada e eficiente.

Uma fila é uma estrutura de dados que segue o princípio "first-in, first-out" (FIFO), o que significa que o primeiro elemento inserido é o primeiro a ser removido.

### **Definição:**

- Uma fila é uma coleção linear de elementos em que a inserção ocorre no final, chamado de "final da fila", e a remoção de elementos ocorre no início, chamado de "início da fila".
- Os elementos são organizados em ordem sequencial, com o elemento mais antigo no início da fila.

### **Operações básicas:**

- Enqueue: Adicionar um elemento no final da fila.
- Dequeue: Remover o elemento do início da fila.
- IsEmpty: Verificar se a fila está vazia.
- Size: Obter o número de elementos na fila.

### **Aplicações comuns:**

- Processamento de tarefas em ordem de chegada: Filas são usadas para garantir que as tarefas sejam processadas na ordem em que chegaram, como em sistemas de impressão ou gerenciamento de solicitações de rede.
- Controle de recursos compartilhados: Filas são usadas para gerenciar recursos compartilhados, como processadores de computadores.

### **Implementação:**

- Filas podem ser implementadas usando arranjos (arrays) ou listas encadeadas, dependendo das necessidades específicas.
- A escolha da implementação pode afetar a eficiência das operações, especialmente em cenários que requerem inserção e remoção frequentes.

### **Considerações:**

- É importante garantir que a fila seja usada de maneira justa, respeitando a ordem de chegada.
- **A fila de prioridade é uma variação da fila que atribuir prioridades aos elementos e pode ser usada para lidar com tarefas com prioridades diferentes.**

Uma fila circular, também conhecida como fila circular (ou circular queue, em inglês), é uma variação da estrutura de dados fila em que o último elemento da fila está conectado ao primeiro, formando um ciclo. Isso permite que os elementos sejam armazenados de forma eficiente em um espaço de memória limitado, reutilizando os locais que foram desocupados quando elementos foram removidos da fila.

**Estrutura circular:** A fila circular mantém um buffer circular no qual os elementos são armazenados. Quando o último elemento da fila é alcançado, a próxima inserção ocorre no início do buffer, aproveitando os espaços liberados.

Operações básicas: As operações fundamentais de uma fila circular são semelhantes às de uma fila tradicional:

- Enqueue: Adicionar um elemento no final da fila circular.
- Dequeue: Remover o elemento do início da fila circular.
- IsEmpty: Verificar se a fila circular está vazia.
- IsFull: Verificar se a fila circular está cheia.

**Implementação eficiente:** A fila circular é especialmente útil em cenários em que a memória é limitada, e a alocação e desalocação de memória dinâmica podem ser custosas. Ela evita o desperdício de espaço e permite reutilizar áreas de memória.

**Utilizações comuns:** As filas circulares são frequentemente usadas em situações em que os dados são processados em um ciclo contínuo, como sistemas de gerenciamento de tarefas com recursos limitados, sistemas embarcados e buffers de comunicação em sistemas de tempo real.

**Desafios:** Gerenciar adequadamente a fila circular requer atenção para evitar erros, como condições de estouro (quando a fila está cheia) e condições de subfluxo (quando a fila está vazia). As operações de enfileiramento e desenfileiramento devem ser cuidadosamente implementadas para manter a integridade da fila circular.

Em resumo, uma fila circular é uma estrutura de dados eficiente para armazenar elementos em um ciclo contínuo, oferecendo um equilíbrio entre utilização eficiente de memória e acesso rápido aos elementos. Ela é amplamente utilizada em situações em que os dados precisam ser processados de forma cíclica e com recursos limitados.



Uma lista é uma estrutura de dados que armazena uma coleção ordenada de elementos. Existem várias variações de listas em ciência da computação, cada uma com características e operações específicas.

Lista Simplesmente Encadeada:

- Uma lista simplesmente encadeada consiste em nós (ou elementos) ligados sequencialmente, onde cada nó contém um valor e uma referência ao próximo nó.
- Operações principais incluem inserção no início, no final e no meio, além de remoção e busca.

Lista Duplamente Encadeada:

- Similar à lista simplesmente encadeada, mas cada nó possui referências tanto para o próximo nó quanto para o nó anterior.
- Isso permite percorrer a lista em ambas as direções.
- Operações incluem inserção e remoção em ambas as extremidades e busca bidirecional.

Lista Circular:

- Uma lista na qual o último nó se liga ao primeiro, criando um ciclo.
- Pode ser implementada como uma lista simplesmente encadeada ou duplamente encadeada.
- Útil para casos em que a ordem cíclica dos elementos é importante.

Cada variação de lista tem suas próprias vantagens e desvantagens, e a escolha da estrutura de dados dependerá dos requisitos específicos de um problema. As listas são usadas para armazenar e manipular coleções de dados de maneira organizada, tornando-as uma parte fundamental da programação e da ciência da computação.

Uma árvore é uma estrutura de dados hierárquica amplamente utilizada na ciência da computação para organizar e representar dados de forma hierárquica. Ela é composta por nós (ou vértices) interconectados, onde cada nó contém um valor e pode ter zero ou mais nós filhos, formando uma estrutura em forma de árvore.

Nó Raiz:

- O nó superior na hierarquia da árvore é chamado de nó raiz. Ele é o ponto de partida para a navegação na árvore.

Nós, Filhos e Pais:

- Cada nó da árvore pode ter zero ou mais nós filhos. Os nós filhos são aqueles que se ramificam a partir de um nó pai.
- O nó pai é o nó que possui um ou mais nós filhos.

Folhas:

- Os nós que não têm filhos são chamados de folhas. São os nós terminais da árvore.

Nível:

- O nível de um nó indica a sua posição na hierarquia da árvore. O nó raiz está no nível 0, seus filhos diretos estão no nível 1, e assim por diante.

Altura:

- A altura da árvore é a quantidade máxima de níveis em qualquer caminho da raiz às folhas. É um indicador da profundidade da árvore.

## Árvore Binária:

- Uma árvore binária é um tipo especial de árvore em que cada nó tem no máximo dois nós filhos: um à esquerda e um à direita.

## Árvore de Busca Binária (BST):

- Uma árvore de busca binária é uma árvore binária especial na qual os valores dos nós são organizados de tal forma que, para qualquer nó, os valores dos nós à esquerda são menores e os valores dos nós à direita são maiores.

## Operações Comuns:

- Algumas operações comuns em árvores incluem a inserção de elementos, a remoção de elementos, a busca por um valor específico, e travessias (inorder, preorder, postorder) para percorrer a árvore.

## Utilizações Comuns:

- Árvores são usadas em inúmeras aplicações, como a implementação de estruturas de dados (árvores binárias de busca, árvores AVL, árvores B), representação de estruturas hierárquicas (árvores de diretórios), e análise sintática de expressões (árvores de análise sintática).

- 

Árvores são uma parte fundamental da ciência da computação e são amplamente aplicadas para resolver uma variedade de problemas de organização e manipulação de dados em sistemas de software. A escolha da estrutura de árvore adequada depende dos requisitos específicos de cada aplicação.

Um grafo é uma estrutura de dados que consiste em um conjunto de vértices (ou nós) interconectados por arestas (ou arestas), que representam relacionamentos entre os vértices.

Vértices e Arestas:

- Os vértices são os pontos individuais no grafo e representam entidades, objetos ou pontos de dados.
- As arestas são conexões entre vértices e representam relacionamentos ou conexões entre os vértices.

Grafo Direcionado e Não Direcionado:

- Um grafo direcionado (ou digrafo) é um grafo em que as arestas têm direção, indicando uma relação de um vértice de origem para um vértice de destino.
- Um grafo não direcionado é um grafo em que as arestas não têm direção, representando uma relação bidirecional entre os vértices.

Grafo Ponderado:

- Um grafo ponderado atribui valores numéricos (pesos) às arestas para representar alguma medida, como distância, custo, tempo, etc.

Aplicações:

- Grafos são usados em diversas aplicações, como representação de redes sociais, sistemas de recomendação, mapas de estradas, análise de redes, problemas de transporte, grafos de dependência em programação e muito mais.

Algoritmos:

- Diversos algoritmos são utilizados em grafos, incluindo busca em profundidade (DFS), busca em largura (BFS), algoritmos de caminho mínimo (como Dijkstra e Bellman-Ford).

Estruturas de Dados:

- Grafos podem ser implementados usando diversas estruturas de dados, incluindo listas de adjacência e matrizes de adjacência, cada uma com suas próprias vantagens e desvantagens.

Grafos são uma estrutura de dados poderosa e versátil utilizada para modelar e resolver problemas complexos que envolvem relações entre entidades. Eles desempenham um papel fundamental em muitas aplicações computacionais e são fundamentais para resolver uma variedade de problemas em ciência da computação e engenharia.