



**INSTITUTO
FEDERAL**

Sudeste de
Minas Gerais

Campus
Manhuaçu

Estruturas de Dados I

Listas

Prof. Leonardo C. R. Soares - leonardo.soares@ifsudestemg.edu.br

Instituto Federal do Sudeste de Minas Gerais

21 de novembro de 2024





Estrutura de dados

Definição

- Estruturas de dados são um modo particular de armazenamento e organização de conjuntos de dados;





Estrutura de dados

Definição

- ▶ Estruturas de dados são um modo particular de armazenamento e organização de conjuntos de dados;
- ▶ Estes conjuntos de dados manipulados por algoritmos podem crescer, diminuir, serem alterados e também pesquisados – por isso são chamados de conjuntos dinâmicos;





Estrutura de dados

Definição

- ▶ Estruturas de dados são um modo particular de armazenamento e organização de conjuntos de dados;
- ▶ Estes conjuntos de dados manipulados por algoritmos podem crescer, diminuir, serem alterados e também pesquisados – por isso são chamados de conjuntos dinâmicos;
- ▶ Estes conjuntos podem ser representados por estruturas de dados simples, que por sua vez podem compor estruturas avançadas;





Estrutura de dados

Definição

- ▶ Estruturas de dados são um modo particular de armazenamento e organização de conjuntos de dados;
- ▶ Estes conjuntos de dados manipulados por algoritmos podem crescer, diminuir, serem alterados e também pesquisados – por isso são chamados de conjuntos dinâmicos;
- ▶ Estes conjuntos podem ser representados por estruturas de dados simples, que por sua vez podem compor estruturas avançadas;
- ▶ A representação computacional escolhida deve ser capaz de suportar as operações em conjuntos dinâmicos.





Operações frequentes

- ▶ $\text{Pesquisar}(S, k)$: Dado um conjunto S , pesquisar a existência do elemento k ;
- ▶ $\text{Inserir}(S, x)$: Insere o elemento x no conjunto S ;
- ▶ $\text{Remover}(S, x)$: Remove o elemento x do conjunto S ;
- ▶ $\text{Mínimo}(S)$: Pesquisa o conjunto S e retorna a menor chave encontrada;
- ▶ $\text{Máximo}(S)$: Pesquisa o conjunto S e retorna a maior chave encontrada;
- ▶ $\text{Sucessor}(S, x)$: Pesquisa o conjunto ordenado S e retorna o elemento após x , ou, caso x seja o último elemento, retorna NULL ;
- ▶ $\text{Antecessor}(S, x)$: Pesquisa o conjunto ordenado S e retorna o elemento anterior a x , ou, caso x seja o primeiro elemento, retorna um valor nulo.





Listas

Descrição

Listas são uma forma simples de interligar conjuntos, uma sequência linear de zero ou mais elementos x_0, x_1, \dots, x_{n-1} , na qual x_i é de um determinado tipo e n representa o tamanho da lista.

Sua principal propriedade estrutural envolve as posições relativas dos elementos em uma dimensão:

- ▶ Assumindo $n > 0$, x_0 é o primeiro item da lista e x_{n-1} é o último item da lista;
- ▶ x_i precede x_{i+1} para $i = 0, \dots, n-2$;
- ▶ x_i sucede x_{i-1} para $i = 1, \dots, n-1$.





Listas

Operações

O conjunto de operações a ser definido depende exclusivamente da aplicação. Entretanto, um conjunto de operações normalmente necessário à maioria das aplicações são:

- ▶ Criar uma lista linear vazia;
- ▶ Inserir um novo item imediatamente após o i -ésimo item;
- ▶ Remover o i -ésimo item;
- ▶ Localizar o i -ésimo item para examinar e/ou alterar o conteúdo de seus componentes;
- ▶ Combinar duas ou mais listas lineares em uma lista única;
- ▶ Dividir uma lista linear em duas ou mais listas;





Listas

Operações

- ▶ Fazer uma cópia da lista linear;
- ▶ Ordenar os elementos da lista em ordem ascendente ou descendente, de acordo com alguns de seus componentes;
- ▶ Pesquisar a ocorrência de um item com um valor particular em algum componente.





Listas

Várias estruturas de dados podem ser usadas para representar listas lineares, cada uma com vantagens e desvantagens particulares.

As duas representações mais utilizadas são:

- ▶ Implementação por arranjos;
- ▶ Implementação por ponteiros (ou referência).





Listas

Implementação por arranjos

- ▶ Os elementos são armazenados em um arranjo de tamanho suficiente para armazenar a lista;
- ▶ A lista pode ser percorrida em qualquer direção;
- ▶ O campo último aponta para a posição seguinte à do último elemento da lista;
- ▶ O i -ésimo elemento da lista está armazenado na $(i-1)$ -ésima posição do arranjo;
- ▶ O tamanho do arranjo limita o tamanho máximo permitido para a lista.





Listas

Vantagens da implementação por arranjos

- ▶ Economia de memória (os apontadores são implícitos nesta estrutura);
- ▶ Acesso a qualquer elemento da lista é feito em tempo constante ($O(1)$).

Desvantagens da implementação por arranjos

- ▶ Custo para inserir elementos da lista pode ser linear ($O(n)$ - todos os elementos podem precisar ser movidos);
- ▶ Custo para remover elementos da lista pode ser $O(n)$;
- ▶ Quando não existe previsão sobre o crescimento da lista, o arranjo que define a lista deve ser alocado de forma dinâmica;
- ▶ O arranjo pode precisar ser realocado completamente.





Listas

Exemplos de implementações por arranjos

Baixe o exemplo aqui.





Listas

Exemplos de implementações por arranjos

Baixe o exemplo aqui.

Olhe atentamente o método de remoção. O que acontece se todas as posições do vetor estiverem ocupadas e tentarmos remover um elemento?





Listas

Exercício de fixação (não entra na lista de exercícios avaliativa)

1. Desenvolva uma lista de Filmes (codigo, nome e genero). O método de inserção deverá adicionar itens na lista mantendo-se a ordenação pelo código (considere que a lista estará ordenada inicialmente ou vazia).
2. Escreva um método que permita excluir um determinado filme da lista (pelo código).
3. Escreva um método que permita aumentar o tamanho da lista.
4. Escreva um método que permita localizar um determinado filme pelo nome.
5. Escreva um menu de controle que permita ao usuário acessar as opções do programa desenvolvido (pode-se utilizar GUI).







Referências

- ▶ CARVALHO, Marco Antonio Moreira de. **Projeto e análise de algoritmos**. 01 mar. 2018, 15 jun. 2018. Notas de Aula. PPGCC. UFOP
- ▶ GOODRICH, Michael T.; TAMASSIA, Roberto. **Estruturas de Dados & Algoritmos em Java**. Bookman Editora, 2013.
- ▶ ZIVIANI, Nivio. **Projeto de Algoritmos com implementações em Java e C++**, 2007.

