

ESTRUTURA DE DADOS: VETORES

Prof. Josildo Silva

AGENDA

- ⊙ Motivação
- ⊙ Introdução à estrutura de dados
- ⊙ Estrutura de dados: matrizes
- ⊙ Vetores: definição e operações
- ⊙ Exemplos

MOTIVAÇÃO

Suponha que haja necessidade de se manipular quatro números, por exemplo, para mostrá-los ordenados:

Em Portugal, a entrada de dados seria:

```
escreva("Informe n1: ") leia(n1)
```

```
escreva("Informe n2: ") leia(n2)
```

```
escreva("Informe n3: ") leia(n3)
```

```
escreva("Informe n4: ") leia(n4)
```

MOTIVAÇÃO

Suponha que haja necessidade de se manipular quatro números, por exemplo, para mostrá-los ordenados:

Em Portugal, a entrada de dados seria:

```
escreva("Informe n1: ") leia(n1)
```

```
escreva("Informe n2: ") leia(n2)
```

```
escreva("Informe n3: ") leia(n3)
```

```
escreva("Informe n4: ") leia(n4)
```

E se fossem cem, duzentos, mil números ?

MOTIVAÇÃO

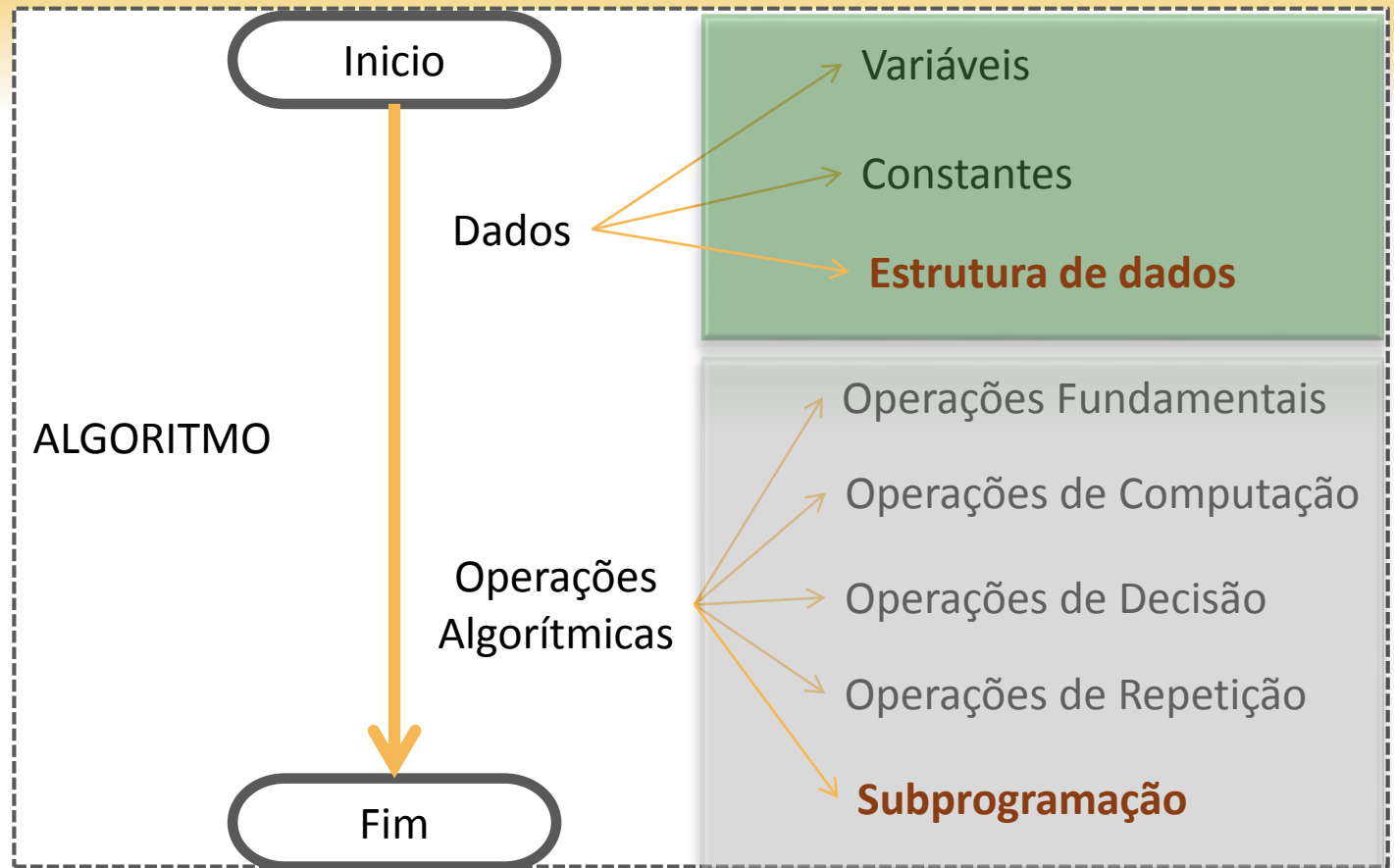
Suponha que haja necessidade de se manipular quatro números, por exemplo, para mostrá-los ordenados:

Estruturas de Dados

Uma estrutura de dados é uma coleção (de dados) que pode ser caracterizada por sua organização e pelas operações que a definem.

E se fossem cem, duzentos, mil números ?

ALGORITMO COMPUTACIONAL



Conjunto ordenado de passos automatizáveis atuando sobre dados que definem um processo finito.

TIPOS DE DADOS

As linguagens de programação disponibilizam mecanismos de manipulação de dados de duas maneiras:

Dados simples	Dados estruturados	
	Estáticos	Dinâmicos
✓ Inteiro	✓ Matrizes	✓ Listas
✓ Real	✓ Registros	✓ Pilhas
✓ Caractere	✓ Arquivos	✓ Filas
✓ Lógico	✓ Cadeias	✓ Árvores
		✓ Grafos

ESTRUTURAS DE DADOS ESTÁTICA/DINÂMICA

- ⊙ Quanto à criação ou definição, os dados estruturados podem ser classificados de maneiras: **estáticas** e **dinâmicas**.
- ⊙ As estruturas de dados estáticas são aquelas em que o tamanho ocupado na memória é definido antes que o programa seja executado e não pode ser modificado durante a execução do programa.

ESTRUTURA DE DADOS MATRIZES

- ⊙ Uma matriz é uma estrutura de dados homogênea, ou seja, é um conjunto de dados do mesmo tipo, representado pelo mesmo nome.
- ⊙ Conjunto de variáveis, cada uma podendo representar um valor, como se fossem variáveis simples, mas todas compartilham o mesmo nome.

ESTRUTURA DE DADOS

MATRIZES

- ⊙ Uma matriz é um conjunto finito e ordenado de elementos homogêneos.
- ⊙ A propriedade ordenado significa que o elemento primeiro, segundo, terceiro ... enésimo de uma matriz pode ser identificado.
- ⊙ Os elementos de uma matriz guardam relação de contiguidade e vizinhança na memória.

REPRESENTAÇÃO DE MATRIZ

A:

1	2	3	4	5	6

1 x 6

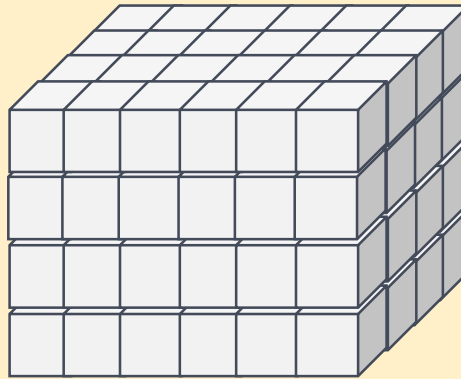
B:

	1	2	3	4	5	6
1						
2						
3						
4						

4 x 6

REPRESENTAÇÃO DE MATRIZ

C:



4 x 6 x 4

ESTRUTURA DE DADOS

MATRIZES

Uma matriz é definida pelo o tipo de dado que armazena, um nome e quantidade de elementos:

```
tipo_dado NOME_MATRIZ[n1][n2][n3]
```

Exemplos:

```
int idades[20]
```

```
char nome[60]
```

```
char nomes[40][60]
```

```
float notas_disciplina[40][6]
```

```
float notadisciplina_curso[40][6][2]
```

MATRIZ UNIDIMENSIONAL

- ◎ Conjunto de variáveis, cada uma podendo representar um valor, como se fossem variáveis simples, mas todas compartilham o mesmo nome, DISTRIBUIDAS EM UMA SÓ LINHA OU COLUNA.

OU SEJA... UMA DIMENSÃO

MATRIZ UNIDIMENSIONAL

Matriz Coluna

0		1	
1		2	
2		3	
3		4	
4		5	
5		6	
6		7	
7		8	
8		9	
9		10	

Matriz Linha

1	2	3	4	5	6	7	8	9	10
0	1	2	3	4	5	6	7	8	9

VETORES

- ⦿ Vetor é um tipo especial de matriz:
UNIDIMENSIONAL.

**ESTRUTURA DE DADOS
UNIDIMENSIONAL
=
VETOR em Computação**

VETORES

Um vetor é uma variável composta homogênea unidimensional, formada por uma sequência de variáveis, todas do mesmo tipo de dados, com o mesmo identificador (mesmo nome) e alocadas sequencialmente na memória.

OPERAÇÕES COM VETORES

As operações que podem ser realizadas com vetores durante o processo de resolução de um problema são:

- ✓ Atribuição,
- ✓ Leitura/escrita,
- ✓ Varredura (acesso sequencial),
- ✓ Busca,
- ✓ Atualização (inserção, exclusão),
- ✓ Ordenação.

OPERAÇÕES COM VETORES

✓ Atribuição

```
float notas[20]
```

A atribuição de valores a um elemento do vetor será realizada com a instrução de atribuição:

```
notas[0] = 7.5;  
notas[5] = 8.5;
```

Atribuir valores a todos os elementos de um vetor, deve-se recorrer às estruturas de repetição:

```
for (i=0; i<20; i++) {  
    notas[i] = notas[i]+0.5  
}
```

OPERAÇÕES COM VETORES

✓ Atribuição

```
float notas[10]
```

Atribuir valores a todos os elementos de um vetor,
deve-se recorrer às estruturas de repetição:

0	1	2	3	4	5	6	7	8	9
7	6	9	8	6	7	8	8	9	6

```
for (i=0;i<10;i++) {  
    notas[i] = notas[i]+0.5  
}
```

0	1	2	3	4	5	6	7	8	9
7.5	6.5	9.5	8.5	6.5	7.5	8.5	8.5	9.5	6.5

OPERAÇÕES COM VETORES

✓ Leitura/escrita

```
float notas[20]
```

A leitura/escrita de valores a um elemento do vetor será realizada por:

```
scanf("%f", &notas[1])  
printf(notas[1])
```

Leitura/escrita a todos os elementos de um vetor, deve-se recorrer às estruturas de repetição:

```
for(i=0; i<20; i++)  
{  
    scanf("%f", &notas[i])  
    printf(notas[i])  
}
```

OPERAÇÕES COM VETORES

✓ Varredura

```
float notas[20]
```

- A operação de efetuar uma ação geral sobre todos os elementos de um vetor.
- A varredura é realizada com objetivo de alguma outra operação: preenchimento do vetor, exibição, busca, atualização.

```
s = 0
for (i=0; i<20; i++)
{
    s = s + notas[i]
}
```

OPERAÇÕES COM VETORES

Exemplo 1

Dado uma turma de vinte alunos, desenvolver um programa usando vetores que permita informar as notas da turma. Após preenchimento deve-se acrescentar todas notas de 0,5 ponto e exibir as notas na ordem inversa (do último valor informado para o primeiro).

OPERAÇÕES COM VETORES

Exemplo 2

Dados salários de trinta funcionários, desenvolver um programa usando vetores que receba os salários e informe o maior e o menor salário.

OPERAÇÕES COM VETORES

Exemplo 3

Desenvolver um programa que receba uma frase (com limite de 1000 caracteres) informe a quantidade de vogais da frase detalhando cada uma delas.

REFERÊNCIAS

1. AGUILAR, L.J. Fundamentos de Programação: Algoritmos, estruturas de dados e objetos. McGraw-Hill, São Paulo, 3.ed, 2008..
2. ASCENCIO, Ana F.G. e CAMPOS, Edilene A. V. Fundamentos da Programação de Computadores – Algoritmos, Pascal e C/C++ e Java. São Paulo: Prentice Hall.

TIPO CHAR EM C

j	o	s	i	l	d	o		p	e	r	e	i	r	a		x	x	x	