

## Lab 0: Representação de Vetores e Matrizes

Prof. Waldemar Celes  
Departamento de Informática, PUC-Rio

9 de março de 2022

**A correção dos trabalhos será feita automaticamente por programas de teste.**  
Sigam as especificações nos seus detalhes!

O objetivo deste laboratório é criar funções para representação e manipulação de vetores e matrizes dinâmicas. A matriz será representada por um vetor de ponteiros, onde cada elemento aponta para o vetor linha.

Pede-se:

1. Crie um módulo com funções para vetores. O arquivo `vetor.h` deve ter a interface do módulo (protótipos das funções), e o arquivo `vetor.c` deve ter a implementação das funções do módulo.

- (a) Implemente uma função que crie dinamicamente um vetor de dimensão  $n$ , onde  $n$  representa o número de elementos do vetor:

```
double* vet_cria (int n);
```

- (b) Implemente uma função que libere a memória de um vetor previamente criado.

```
void vet_libera (double* v);
```

- (c) Implemente uma função que calcule e retorne o valor do produto escalar entre dois vetores de dimensão  $n$ .

$$v \cdot w = \sum_{i=0}^{n-1} v_i w_i$$

```
double vet_escalar (int n, double* v, double* w);
```

- (d) Implemente uma função que calcule a norma-2 de um vetor de dimensão  $n$ . Sabe-se que a norma-2 de um vetor é dado por:

$$\|v\|_2 = \sqrt{\sum_{i=0}^{n-1} v_i^2}$$

```
double vet_norma2 (int n, double* v);
```

- (e) Implemente uma função que calcule a multiplicação de um vetor  $v$  por um valor escalar  $s$ , preenchendo o vetor  $w$  com o resultado (já alocado):

$$w_i = s v_i$$

```
void vet_mults (int n, double* v, double s, double* w);
```

- (f) Implemente uma função que exiba os elementos de um vetor na tela. Sugere-se que os elementos do vetor sejam exibidos na mesma linha, separados por brancos.

```
void vet_imprime (int n, double* v);
```

2. Crie um módulo com funções para matrizes, implementando `matriz.h` e `matriz.c`.

- (a) Implemente uma função que crie dinamicamente uma matriz de dimensão  $m \times n$ , onde  $m$  representa o número de linhas e  $n$  representa o número de colunas:

```
double** mat_cria (int m, int n);
```

- (b) Implemente uma função que libere a memória de uma matriz previamente criada. A função recebe o número de linhas  $m$  da matriz:

```
void mat_libera (int m, double** A);
```

- (c) Implemente uma função que preencha a transposta de uma dada matriz. A função recebe as dimensões  $m \times n$  da matriz original, a matriz original  $A$  e a matriz transposta a ser preenchida  $T$  (com dimensão  $n \times m$ , já alocada):

$$T_{ji} = A_{ij}$$

```
void mat_transposta (int m, int n, double** A, double** T);
```

- (d) Implemente uma função que receba uma matriz e um vetor, e preencha um outro vetor, já alocado, com o resultado da multiplicação da matriz pelo vetor. A função recebe a dimensão  $m \times n$  da matriz e assume que o primeiro vetor  $v$  tem dimensão  $n$  e o vetor resultado  $w$  tem dimensão  $m$ :

$$w_i = \sum_{j=0}^{n-1} A_{ij} v_j$$

```
void mat_multv (int m, int n, double** A, double* v, double* w);
```

- (e) Implemente uma função que calcule a multiplicação entre duas matrizes:  $C = AB$ . A função recebe as dimensões  $m$ ,  $n$  e  $q$ , e as matrizes  $A_{m \times n}$ ,  $B_{n \times q}$  e  $C_{m \times q}$ , preenchendo  $C$ , já alocada:

$$C_{ik} = \sum_{j=0}^{n-1} A_{ij} B_{jk}$$

```
void mat_multm (int m, int n, int q, double** A, double** B, double** C);
```

- (f) Implemente uma função que exiba os elementos de uma matriz na tela. Sugere-se que cada linha da matriz seja exibida em uma mesma linha na saída, com os valores separados por brancos.

```
void mat_imprime (int m, int n, double** A);
```

Escreva um outro módulo “main.c” para testar sua implementação.

**Entrega:** O código fonte deste trabalho (isto é, os arquivos “vetor.h”, “vetor.c”, “matriz.h”, “matriz.c” e “main.c”) devem ser enviados via página da disciplina no EAD. Este laboratório não entra na avaliação da disciplina. Para ser corrigido, o prazo final para envio é **domingo, dia 13 de março**.