

3ª Prova de Algoritmos e Estruturas de Dados I

25/06/2010

Perguntas comuns e suas respostas:

- P: Tenho uma dúvida na questão tal.
R: A compreensão do enunciado faz parte da prova.
- P: Se eu consultar algum material próprio ou de algum colega, o que acontecerá comigo?
R: A prova é individual e sem consulta. Qualquer tentativa de fraude acarretará abertura de processo administrativo na UFPR.
- P: Posso fazer a prova a lápis?
R: prova é um documento, portanto deve ser feita a caneta.
- P: O que será corrigido?
R: A lógica, a criatividade, a sintaxe, o uso correto dos comandos, a correta declaração dos tipos, os nomes das variáveis, a indentação, uso equilibrado de comentários no código e, evidentemente, a clareza.

Introdução

Uma matriz é chamada de *esparsa* quando possui uma grande quantidade de elementos que valem zero. Por exemplo, a matriz de ordem 5×4 seguinte é esparsa, pois contém apenas 4 elementos não nulos.

	1	2	3	4
1	0	17	0	0
2	0	0	0	0
3	13	0	-12	0
4	0	0	25	0
5	0	0	0	0

Obviamente, a representação computacional padrão para matrizes é ineficiente em termos de memória, pois gasta-se um espaço inútil para se representar muitos elementos nulos.

Nesta prova, vamos usar uma representação alternativa que vai permitir uma boa economia de memória.

A idéia é representar apenas os elementos não nulos. Para isto usaremos três vetores, dois deles (L e C) para guardar as coordenadas dos elementos não nulos e o terceiro (D) para guardar os valores dos elementos daquelas coordenadas. Também usaremos três variáveis para representar o número de linhas e colunas da matriz completa e o número de elementos não nulos da matriz.

Considere as seguintes definições de tipos:

CONST

```
MAX = 6;          (* um valor bem menor que 5 x 4, dimensao da matriz *)
```

TYPE

```
vetor_coordenadas = array [1..MAX] of integer; (* coordenadas *)  
vetor_elementos   = array [1..MAX] of real;   (* dados *)
```

VAR

```
L, C: vetor_coordenadas; (* L: linhas, C: colunas *)
```

```

D: vetor_elementos;      (* D: dados *)
N_lin, N_col: integer;   (* para armazenar as dimensoes da matriz *)
N_elementos: integer     (* numero de elementos nao nulos *)

```

Definição 1 Um elemento $M[i,j]$ da matriz completa pode ser obtido da representação compactada:

- se existe um k tal que $L[k] = i$ e $C[k] = j$, então $M[i,j] = D[k]$;
- caso contrário, $M[i,j] = 0$.

A matriz do exemplo anterior pode então ser assim representada:

```
N_elementos:= 4; N_lin:= 5; N_col:= 4;
```

	1	2	3	4	5	6
L	1	3	3	4		
C	2	1	3	3		
D	17	13	- 12	25		

Questões (25 pontos cada)

1. Fazer um procedimento que leia da entrada padrão:

- dois inteiros, representando as dimensões da matriz (linha, coluna);
- trincas de elementos l, c, d , onde l e c são inteiros e d é real, representando respectivamente a linha, a coluna e o valor de um elemento não nulo da matriz. A leitura termina quando for lido uma trinca $0, 0, 0$. Para cada trinca, devem ser criados os três vetores que representam a matriz conforme descrito acima. Veja o exemplo de entrada de dados, abaixo.

Exemplo para a entrada de dados:

```

5 4
1 2 17
3 1 13
3 3 -12
4 3 25
0 0 0

```

2. Fazer uma função que, dada uma coordenada (l, c) , respectivamente para uma linha e coluna, retorne o valor de elemento $M[l,c]$, conforme a definição ??.
3. Fazer um procedimento que, dadas duas matrizes no formato compactado descrito acima, obtenha uma terceira matriz compactada que é a soma das duas primeiras.
4. Fazer um procedimento que, dada uma matriz no formato compactado, imprima na tela uma matriz no formato padrão, contendo os zeros.