

sqtpm

[jhdapr]

[voltar](#)**Trabalho:** 05-VC

Data de abertura: 08/09/2021 11:11:00

Data limite para envio: 17/09/2021 16:00:00 (aberto)

Número máximo de envios: 31

Arquivos a enviar: entre 1 e 3

Enviar:Linguagem:

Arquivos:

Nenhum arquivo selecionado

Matrizes esparsas em vetor de coordenadas

Uma matriz é esparsa se tem proporcionalmente muitos zeros. "Muitos" não é uma quantidade definida claramente, mas em geral se uma matriz com n linhas e m colunas tem um número de entradas não-zero da ordem de $n+m$ então ela é considerada esparsa.

Quando uma matriz esparsa é colocada na memória, uma parte significativa do espaço é usada para armazenar zeros, que em muitos casos é simplesmente ausência de informação.

Estruturas de dados para matrizes esparsas não armazenam os zeros. Elas sacrificam o tempo de acesso constante a qualquer elemento oferecido pelas matrizes (pelo menos em C) para usar menos memória. Algumas estruturas de dados são melhores para percorrer os elementos de uma mesma linha ou coluna da matriz. Algumas são boas para recuperação, mas ruins para atualização da matriz.

Uma dessas representações é o vetor de coordenadas: cada elemento não-zero em i,j é colocado em um vetor em que cada posição armazena uma tripla $(i,j,\text{elemento})$. Normalmente o vetor é mantido ordenado por i e, para o mesmo i , por j .

Por exemplo, a matriz

```
0 0 0 0
7 0 0 0
0 0 9 0
0 0 0 0
0 8 0 5
```

pode ser representada em um vetor de coordenadas da forma:

1,0,7	2,2,9	4,1,8	4,3,5
-------	-------	-------	-------

Neste trabalho você deve implementar um vetor de coordenadas com operações de recuperação e atualização dos elementos de uma matriz esparsa. Seu programa deve usar um vetor de structs.

A matriz começa com um conjunto de k ($k \leq n+m$) elementos não zero. Depois é atualizada e consultada por comandos que são dados na entrada.

Entrada

A entrada para o programa começa com uma linha que especifica as dimensões n e m de uma matriz de números inteiros. Vamos chamá-la de M . O vetor de coordenadas deve armazenar até $n+m$ elementos não-zero de M .

Depois vem uma linha com o número k de linhas que serão as próximas na entrada e têm elementos em M que são diferentes de zero. k é menor ou igual a $n+m$.

Em seguida vêm k triplas com a forma $[i,j] x$, indicando que $M[i,j] = x$. Essas triplas estão ordenadas por i e por j .

As próximas linhas têm uma das seguintes formas:

- a $[i,j] x$

Ao ler uma linha dessa forma, se $M[i,j]$ já estiver no vetor de coordenadas então o valor desse elemento deve ser atualizado tornando-se igual a x , mas se x for igual a zero então $M[i,j]$ deve ser removido do vetor de coordenadas. Se $M[i,j]$ não estiver no vetor de coordenadas e x for diferente de zero, esse elemento deve ser inserido.

Se o vetor de coordenadas já estiver cheio (i.e. com $n+m$ elementos) e uma inserção for necessária então ela não deve ser feita e uma mensagem deve ser impressa com a forma ilustrada abaixo.

- r $[i,j]$

Linhas dessa forma indicam que $M[i,j]$ deve ser recuperado e impresso com o formato ilustrado no exemplo a seguir.

- p

Ao ler uma linha dessa forma seu programa deve imprimir o conteúdo do vetor de coordenadas na forma de uma lista de triplas ordenadas por linha e coluna, com um espaço depois de cada elemento, em uma linha, como exemplificado abaixo.

sqtpm
[jhdapr]

[voltar](#)

Se o vetor de coordenadas estiver vazio, o programa deve imprimir apenas o rótulo "VC: " e um fim-de-linha.

- t

Ao ler uma linha dessa forma seu programa deve terminar.

Os índices na entrada sempre estarão dentro dos limites, i.e., $0 \leq i < n$ e $0 \leq j < m$.

Saída

A saída do seu programa deve conter a saída dos comandos a, r e p.

Exemplo

Entrada	
7	7
9	
[0,4]	3
[0,5]	7
[1,0]	1
[2,2]	2
[2,3]	6
[2,4]	5
[4,1]	4
[4,4]	6
[4,5]	8
p	
a	[4,6] 9
a	[0,2] 9
a	[2,5] 9
a	[4,4] 9
a	[4,4] 0
a	[0,0] 9
a	[1,1] 9
a	[6,6] 9
p	
a	[5,5] 9
p	
r	[2,2]
r	[5,5]
t	

Saída	
VC: (0,4,3) (0,5,7) (1,0,1) (2,2,2) (2,3,6) (2,4,5) (4,1,4) (4,4,6) (4,5,8)	
VC: (0,0,9) (0,2,9) (0,4,3) (0,5,7) (1,0,1) (1,1,9) (2,2,2) (2,3,6) (2,4,5) (2,5,9) (4,1,4) (4,5,8) (4,6,9) (6,0 vetor de coordenadas esta' cheio.	
VC: (0,0,9) (0,2,9) (0,4,3) (0,5,7) (1,0,1) (1,1,9) (2,2,2) (2,3,6) (2,4,5) (2,5,9) (4,1,4) (4,5,8) (4,6,9) (6,0 vetor de coordenadas esta' cheio.	
M[2][2] == 2	
M[5][5] == 0	

Observações

- Comece fazendo um programa que apenas lê a entrada e imprime os valores. Depois acrescente as demais funcionalidades.
- Se for usar scanf para a entrada, pode ser necessário ajustar espaços na string de scanf para que os colchetes sejam casados no momento certo.
- Observe que os elementos da matriz estão sendo fornecidos em ordem, primeiro por linhas e depois por colunas. Manter o seu vetor de coordenadas ordenados provavelmente vai facilitar a impressão da matriz.

Requisitos adicionais

- Não deve haver variáveis globais. Uma variável é global se for definida fora de uma função. Variáveis definidas dentro da função main não são globais. A nota será zero se houver variáveis globais.

Sobre organização do código e comentários

- Faça um programa organizado, bem endentado e que seja fácil de ler.
- Adicione comentários que vão ser úteis para entender o programa se você for relê-lo daqui a alguns anos: comentar cada linha vai ser redundante; documentar blocos de código e a estratégia usada na solução vai ser muito útil.