ACH2001-Introdução à Ciência da Computação

Segundo exercício programa.

Fábio Nakano e Norton Trevisan Roman

Dieta

Joaquim é um sujeito muito preocupado com sua boa forma (esférica) e por isso resolveu fazer uma dieta radical para voltar a enxergar os pés...

Disposto a qualquer sacrifício, pesquisou na internet e descobriu que tinha algumas opções, todas elas em quantidades de nutrientes e não em quantidade de alimentos. Ele, que é muito esperto, sabe que alimentos são compostos por nutrientes e em um piscar de olhos arrumou uma tabela que apresenta a quantidade de nutrientes de cada alimento, mas não conseguiu ir muito além disso, então resolveu pedir sua ajuda.

Tarefa

Construir um programa em que dadas as necessidades de nutrientes, os alimentos e sua composição, calcula a quantidade de cada alimento a consumir. No exemplo serão usados 4 alimentos e 4 nutrientes. Os alimentos e os nutrientes têm índices 0 a 3. Uma porção do alimento 0 fornece a quantidade $a_{0,0}$ do nutriente 0, uma porção do alimento 1 fornece a quantidade $a_{0,1}$ do nutriente 0, uma porção do alimento 2 fornece a quantidade $a_{0,2}$ do nutriente 0 e uma porção do alimento 3 fornece a quantidade $a_{0,3}$ do nutriente 0. Dada a quantidade de cada alimento x_0 até x_3, calcula-se a quantidade do nutriente 0 pela fórmula $b_0 = a_{0,0} * x_0 + a_{0,1} * x_1 + a_{0,2} * x_2 + a_{0,3} * x_3$. Reproduzindo essa fórmula para todos os nutrientes, resulta em:

$$
\begin{aligned}
b_0 &= a_{0,0} * x_0 + a_{0,1} * x_1 + a_{0,2} * x_2 + a_{0,3} * x_3 \\
b_1 &= a_{1,0} * x_0 + a_{1,1} * x_1 + a_{1,2} * x_2 + a_{1,3} * x_3 \\
b_2 &= a_{2,0} * x_0 + a_{2,1} * x_1 + a_{2,2} * x_2 + a_{2,3} * x_3 \\
b_3 &= a_{3,0} * x_0 + a_{3,1} * x_1 + a_{3,2} * x_2 + a_{3,3} * x_3
\end{aligned}
$$

Essa fórmula relaciona quantidade de alimento e quantidade de nutrientes. Pode-se usá-la para a partir da quantidade de alimentos calcular a quantidade de nutrientes e também para a partir da quantidade de nutrientes calcular a quantidade de alimentos, que é o que João quer. Para isso, precisamos resolver o sistema de 4 equações e quatro incógnitas (x_0 até

x_3):

$$a_{0,0}*x_0+a_{0,1}*x_1+a_{0,2}*x_2+a_{0,3}*x_3=b_0$$
$$a_{1,0}*x_0+a_{1,1}*x_1+a_{1,2}*x_2+a_{1,3}*x_3=b_1$$
$$a_{2,0}*x_0+a_{2,1}*x_1+a_{2,2}*x_2+a_{2,3}*x_3=b_2$$
$$a_{3,0}*x_0+a_{3,1}*x_1+a_{3,2}*x_2+a_{3,3}*x_3=b_3$$

pela definição de multiplicação entre matrizes (linha vezes coluna, lembra?), é possível escrever esse sistema como:

$$\begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} * \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

ou $A*x=b$ onde A é uma **matriz** n x n, x é um **vetor** n x 1 e b é um **vetor** n x 1.

Método para resolução

Para determinar os valores de x que resolvem o sistema é necessário concatenar A e b em uma única matriz:

$$M=\begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} & b_0 \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} & b_1 \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} & b_2 \\ a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} & b_3 \end{bmatrix}$$

é possível eliminar (zerar) o elemento de índice zero das linhas 1 a n-1, $a_{i,0}$ combinando cada linha com a linha zero.

Para isso, para cada linha calcula-se $p_i=\frac{a_{i,0}}{a_{0,0}}$ e recalcula-se cada elemento da i-ésima linha usando a fórmula ,

$a_{i,j}=a_{i,j}-p_i*a_{0,j}$ o que resulta em

$$M=\begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} & b_0 \\ 0 & a_{1,1} & a_{1,2} & a_{1,3} & b_1 \\ 0 & a_{2,1} & a_{2,2} & a_{2,3} & b_2 \\ 0 & a_{3,1} & a_{3,2} & a_{3,3} & b_3 \end{bmatrix}$$

(note que os elementos da i-ésima linha tem seus valores modificados!!)

usando a mesma ideia é possível eliminar o elemento de índice 1 das linhas 2 a n-1, $a_{i,1}$, combinando a linha 1 com as demais: calcula-se $p_i=\frac{a_{i,1}}{a_{1,1}}$ e $a_{i,j}=a_{i,j}-p_i*a_{1,j}$, o que resulta na matriz

$$M=\begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} & b_0 \\ 0 & a_{1,1} & a_{1,2} & a_{1,3} & b_1 \\ 0 & 0 & a_{2,2} & a_{2,3} & b_2 \\ 0 & 0 & a_{3,2} & a_{3,3} & b_3 \end{bmatrix}$$

(I) Generalizando, pode-se combinar a linha zero com as seguintes, eliminando a coluna zero, depois combinar a linha 1 com as seguintes, eliminando a coluna 1, a linha k com as demais, eliminando a coluna k. Este ẽ o mẽtodo de eliminação de Gauss. Para isto calcula-se $p_{i,k}=\frac{a_{i,k}}{a_{k,k}}$ e $a_{i,j}=a_{i,j}-p_{i,k}*a_{k,j}$. Ao final do processo no exemplo:

$$M=\begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} & b_0 \\ 0 & a_{1,1} & a_{1,2} & a_{1,3} & b_1 \\ 0 & 0 & a_{2,2} & a_{2,3} & b_2 \\ 0 & 0 & 0 & a_{3,3} & b_3 \end{bmatrix}$$

então calcula-se pela linha 3 o valor $x_3=\frac{b_3}{a_{3,3}}$, substituindo o valor de x_3 na linha 2 e resolvendo para x_2 tem-se $x_2=\frac{b_2-a_{2,3}*x3}{a_{2,2}}$ substituindo $x_3 e x_2$ na linha 1 resulta em $x_1=\frac{b_1-a_{1,3}*x3-a_{1,2}*x2}{a_{1,1}}$ e substituindo $x_3, x_2 e x_1$ na linha 0 resulta em $x_0=\frac{b_0-a_{0,3}*x3-a_{0,2}*x2-a_{0,1}*x_0}{a_{0,0}}$. **(II) A generalização** resulta na fórmula $x_i=\frac{b_i-\sum_{j=i+1}^{n} a_{i,j}*x_j}{a_{i,i}}$.

Exemplo numérico passo a passo

Joaquim escolheu coentro, abóbora, biscoito recheado e catalonha – cada um colocado em uma coluna contendo o valor energético (kcal), carboidratos (g), lipídios (g) e proteínas (g). A última coluna contém a quantidade de nutrientes que devem ser supridos.

$$M=\begin{bmatrix} 309.0 & 48.0 & 472.0 & 24.0 & 830.239 \\ 20.9 & 1.4 & 6.4 & 1.9 & 34.0 \\ 10.4 & 0.7 & 19.6 & 0.3 & 30.0 \\ 48.0 & 10.8 & 70.5 & 4.8 & 130.0 \end{bmatrix}$$

Aplicando I passo a passo:
1-) combina a linha (k=0), com a linha (i=1) para eliminar elemento da coluna (j=0). Note que os demais elementos da linha são recalculados.

$$p=\frac{20.9}{309.0}=0.06763754045307444$$

$$M=\begin{bmatrix} 309.0 & 48.0 & 472.0 & 24.0 & 830.239 \\ -3.552713678800501\text{E-}15 & -1.846601941747573 & -25.524919093851132 & 0.27669902912621347 & -22.155323948220065 \\ 10.4 & 0.7 & 19.6 & 0.3 & 30.0 \\ 48.0 & 10.8 & 70.5 & 4.8 & 130.0 \end{bmatrix}$$

(note que o elemento não foi anulado - há um valor residual de aprox. -3e-15 - resultado de erros de arredondamento e que neste exercício será desprezado).

2-) combina a linha (k=0), com a linha (i=2) para eliminar elemento da coluna (j=0)

$$p=\frac{10.4}{309.0}=0.03365695792880259$$

$$M=\begin{bmatrix} 309.0 & 48.0 & 472.0 & 24.0 & 830.239 \\ -3.552713678800501E\text{-}15 & -1.846601941747573 & -25.524919093851132 & 0.27669902912621347 & -22.155323948220065 \\ 0.0 & -0.9155339805825244 & 3.713915857605178 & -0.5077669902912623 & 2.0566809061488627 \\ 48.0 & 10.8 & 70.5 & 4.8 & 130.0 \end{bmatrix}$$

3-) combina a linha (k=0), com a linha (i=3) para eliminar elemento da coluna (j=0)

$$p=\frac{48.0}{309.0}=0.1553398058252427$$

$$M=\begin{bmatrix} 309.0 & 48.0 & 472.0 & 24.0 & 830.239 \\ -3.552713678800501E\text{-}15 & -1.846601941747573 & -25.524919093851132 & 0.27669902912621347 & -22.155323948220065 \\ 0.0 & -0.9155339805825244 & 3.713915857605178 & -0.5077669902912623 & 2.0566809061488627 \\ 7.105427357601002E\text{-}15 & 3.343689320388351 & -2.820388349514559 & 1.071844660194175 & 1.0308349514563133 \end{bmatrix}$$

4-) combina a linha (k=1), com a linha (i=2) para eliminar o elemento da coluna (j=1)

$$p=\frac{-0.9155339805825244}{-1.846601941747573}=0.4957939011566772$$

$$M=\begin{bmatrix} 309.0 & 48.0 & 472.0 & 24.0 & 830.239 \\ -3.552713678800501E\text{-}15 & -1.846601941747573 & -25.524919093851132 & 0.27669902912621347 & -22.155323948220065 \\ 0.0 & 0.0 & 16.36901507185419 & -0.6449526813880126 & 13.041155397826845 \\ 7.105427357601002E\text{-}15 & 3.343689320388351 & -2.820388349514559 & 1.071844660194175 & 1.0308349514563133 \end{bmatrix}$$

5-) combina a linha (k=1), com a linha (i=3) para eliminar o elemento da coluna (j=1)

$$p=\frac{3.343689320388351}{-1.846601941747573}=-1.8107255520504737$$

$$M=\begin{bmatrix} 309.0 & 48.0 & 472.0 & 24.0 & 830.239 \\ -3.552713678800501\text{E-}15 & -1.846601941747573 & -25.524919093851132 & 0.27669902912621347 & -22.155323948220065 \\ 0.0 & 0.0 & 16.36901507185419 & -0.6449526813880126 & 13.041155397826845 \\ 7.105427357601002\text{E-}15 & 0.0 & -49.039011566771826 & 1.5728706624605677 & -39.08637623554154 \end{bmatrix}$$

6-) combina a linha (k=2), com a linha (i=3) para eliminar elemento da coluna (j=2)

$$p=\frac{-49.039011566771826}{16.36901507185419}=-2.995843754282582$$

$$M=\begin{bmatrix} 309.0 & 48.0 & 472.0 & 24.0 & 830.239 \\ -3.552713678800501\text{E-}15 & -1.846601941747573 & -25.524919093851132 & 0.27669902912621347 & -22.155323948220065 \\ 0.0 & 0.0 & 16.36901507185419 & -0.6449526813880126 & 13.041155397826845 \\ 7.105427357601002\text{E-}15 & 0.0 & 0.0 & -0.359306799883514 & -0.017112288333407832 \end{bmatrix}$$

calcula a solução usando (II):

$$x=\begin{bmatrix} 1.313175535454139 \\ 0.9666198872282575 \\ 0.7985741203099078 \\ 0.04762584047659431 \end{bmatrix}$$

Testes mais simples podem ser feitos, por exemplo usando A como uma matriz diagonal, logo M ficaria algo como:

$$M=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 2 \\ 0 & 2 & 0 & 0 & 0 & 6 \\ 0 & 0 & 3 & 0 & 0 & 12 \\ 0 & 0 & 0 & 4 & 0 & 20 \\ 0 & 0 & 0 & 0 & 5 & 30 \end{bmatrix}$$ cuja solução é $$x=\begin{bmatrix} 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}$$

Um teste um pouco mais complexo usa uma matriz já triangular, por exemplo:

$$M=\begin{bmatrix} 1 & -1 & 1 & 0 & 0 & 3 \\ 0 & 2 & 0 & -1 & 1 & 7 \\ 0 & 0 & 3 & 0 & 2 & 24 \\ 0 & 0 & 0 & 4 & 1 & 26 \\ 0 & 0 & 0 & 0 & 5 & 30 \end{bmatrix}$$ que tem a mesma solução acima.

Arquivos

O arquivo Matriz.java contém o atributo M, que conterá a matriz, o método de leitura de arquivos le(String nomeArquivo) e o método de impressão na tela imprime(). Suas funcionalidades estão descritas no próprio arquivo. Sua tarefa é preencher o método resolve(), que deve resolver o sistema linear usando o método descrito e retornar o array contendo a solução.

É permitido criar novos métodos dentro da classe Matriz, lembrando que o sistema tem que ser resolvido com uma única chamada a resolve().

Os arquivos teste*.m contém matrizes para teste de seu programa. Os dois primeiros números desse arquivo devem ser inteiros e representam o número de linhas e o número de colunas da matriz M. Depois devem seguir os valores dos elementos de M, em ordem de linha.

Recomenda-se que você teste seu programa exaustivamente.

Avaliação

As soluções serão avaliadas pela clareza e formatação do código (edentação, comentários, etc) e sua capacidade de resolver testes diferentes dos apresentados nos exemplos e que não serão disponibilizados a priori.

Se a solução causar erro ou exceção em determinado teste, receberá zero nesse teste.

Receberão zero as soluções que:

- não compilarem;
- estiverem fora da especificação (inclusive formato de entrega);
- tiverem **modificado** métodos pré-existentes;
- forem copiadas (detecção por sistema automático).

É permitido criar novos métodos dentro da classe Matriz, lembrando que o sistema tem que ser resolvido com uma única chamada a resolve().

Entrega

Apenas o arquivo Matriz.java, contendo a sua solução, deve ser entregue, **compactado** em formato zip ou rar. O nome do arquivo compactado deve ser seu número USP, sem prefixos ou sufixos. Por exemplo se seu número é 1234567, então seu arquivo deve ser 1234567.zip.

A entrega deve ser feita através do CoL até 27/05/2012. Não serão aceitas entregas após este prazo.

Dicas

Para evitar variações na edentação em função da relação entre espaços e TABs, recomenda-se que toda edentação seja feita com TABs e que se for possível no editor de texto escolhido, estes sejam ajustados para exibir recuo equivalente a 4 espaços.

Outros testes

Farinha de centeio integral, coentro, couve-manteiga refogada e macarrão instantâneo.

$$M=\begin{bmatrix} 336.0 & 309.0 & 90.0 & 436.0 & 830.239 \\ 12.5 & 20.9 & 1.7 & 8.8 & 34.0 \\ 1.8 & 10.4 & 6.6 & 17.2 & 30.0 \\ 73.3 & 48.0 & 8.7 & 62.4 & 130.0 \end{bmatrix}$$

1-)

$$p=\frac{12.5}{336.0}=0.03720238095238095$$

$$M=\begin{bmatrix} 336.0 & 309.0 & 90.0 & 436.0 & 830.239 \\ 0.0 & 9.404464285714285 & -1.6482142857142856 & -7.420238095238094 & 3.113132440476189 \\ 1.8 & 10.4 & 6.6 & 17.2 & 30.0 \\ 73.3 & 48.0 & 8.7 & 62.4 & 130.0 \end{bmatrix}$$

2-)

$$p=\frac{1.8}{336.0}=0.005357142857142857$$

$$M=\begin{bmatrix} 336.0 & 309.0 & 90.0 & 436.0 & 830.239 \\ 0.0 & 9.404464285714285 & -1.6482142857142856 & -7.420238095238094 & 3.113132440476189 \\ 0.0 & 8.744642857142857 & 6.117857142857143 & 14.864285714285714 & 25.55229107142857 \\ 73.3 & 48.0 & 8.7 & 62.4 & 130.0 \end{bmatrix}$$

3-)

$$p=\frac{73.3}{336.0}=0.2181547619047619$$

$$M=\begin{bmatrix} 336.0 & 309.0 & 90.0 & 436.0 & 830.239 \\ 0.0 & 9.404464285714285 & -1.6482142857142856 & -7.420238095238094 & 3.113132440476189 \\ 0.0 & 8.744642857142857 & 6.117857142857143 & 14.864285714285714 & 25.55229107142857 \\ 0.0 & -19.409821428571433 & -10.933928571428574 & -32.71547619047619 & -51.12059136904762 \end{bmatrix}$$

4-)

$$p=\frac{8.744642857142857}{9.404464285714285}=0.9298395518845534$$

$$M=\begin{bmatrix} 336.0 & 309.0 & 90.0 & 436.0 & 830.239 \\ 0.0 & 9.404464285714285 & -1.6482142857142856 & -7.420238095238094 & 3.113132440476189 \\ 0.0 & 0.0 & 7.650431975695433 & 21.763916579638597 & 22.657577398018923 \\ 0.0 & -19.409821428571433 & -10.933928571428574 & -32.71547619047619 & -51.12059136904762 \end{bmatrix}$$

4 5

5-)

$$p=\frac{-19.409821428571433}{9.404464285714285}=-2.063894427038831$$

$$M=\begin{bmatrix} 336.0 & 309.0 & 90.0 & 436.0 & 830.239 \\ 0.0 & 9.404464285714285 & -1.6482142857142856 & -7.420238095238094 & 3.113132440476189 \\ 0.0 & 0.0 & 7.650431975695433 & 21.763916579638597 & 22.657577398018923 \\ 0.0 & 0.0 & -14.335668850280076 & -48.03006424253932 & -44.695414674515014 \end{bmatrix}$$ 4 5

6-)

$$p = \frac{-14.335668850280076}{7.650431975695433} = -1.8738378297882907$$

4 5

$$M = \begin{bmatrix} 336.0 & 309.0 & 90.0 & 436.0 & 830.239 \\ 0.0 & 9.404464285714285 & -1.6482142857142856 & -7.420238095238094 & 3.113132440476189 \\ 0.0 & 0.0 & 7.650431975695433 & 21.763916579638597 & 22.657577398018923 \\ 0.0 & 0.0 & 0.0 & -7.248014031255927 & -2.2387890147510063 \end{bmatrix}$$

$$x = \begin{bmatrix} 0.6479504151632062 \\ 0.9397857186688291 \\ 2.082898261387085 \\ 0.3088830961276536 \end{bmatrix}$$

Soluções dos arquivos de teste

Solucao para teste1.m

$$x = \begin{bmatrix} 0.25 \end{bmatrix}$$

Solucao para teste2.m

$$x = \begin{bmatrix} 4 \\ 1 \end{bmatrix}$$

Solucao para teste3.m

$$x=\begin{bmatrix}-2572.8734429962724\\50.569643513414185\\68.90848952590959\\73.75853152727464\\77.47151782432928\\82.9836323196129\\76.76536987452093\\70.29469864020581\\64.70639879129405\\57.73575891216465\end{bmatrix}$$

Solucao para teste4.m

$$x=\begin{bmatrix}1.313175535454139\\0.9666198872282575\\0.7985741203099078\\0.04762584047659431\end{bmatrix}$$

Solucao para teste5.m

$$x=\begin{bmatrix}0.6479504151632062\\0.9397857186688291\\2.082898261387085\\0.3088830961276536\end{bmatrix}$$

Solucao para teste6.m

$$x=\begin{bmatrix}2.0\\3.0\\4.0\\5.0\\6.0\end{bmatrix}$$

Solucao para teste7.m

$$x = \begin{bmatrix} -9.129516728023845 \\ 16.666814340883946 \\ 22.30995187711009 \\ -1.5143697460364103 \\ -244.73719735350298 \\ -940.8653540317752 \\ 147.2491666212461 \\ -122.19346648764036 \\ 798.298213300251 \\ 1.3258286361400176 \end{bmatrix}$$