

MAC 115 – Introdução à Ciência da Computação
 INSTITUTO DE FÍSICA - SEGUNDO SEMESTRE DE 2000

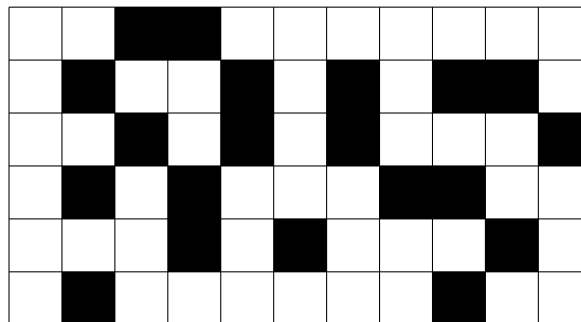
Terceiro Exercício-Programa (EP3)¹

Data de entrega: 5/12/2000

UM MINOTAURO PERDIDO

1 O labirinto e o minotauro perdido

Considere um labirinto como o do desenho abaixo:



Um labirinto desses pode ser representado por uma matriz retangular L , cujo elemento L_{ij} vale 0 ou -1 conforme a casa correspondente do labirinto seja uma passagem livre ou uma parede, respectivamente.

Suponha agora que em alguma posição desse labirinto haja um minotauro perdido, e também uma porta (de entrada/saída). Como bons samaritanos, queremos ajudar o minotauro a encontrar a porta.

Um método geral para resolver esse problema consiste em marcar com o número k ($k = 1, 2, \dots$) as casas livres que estejam a exatamente $k - 1$ passos de distância da porta, pelo caminho mais curto possível. Suponha que, a cada passo, o minotauro possa se deslocar de apenas uma casa na vertical ou na horizontal. Então, rotula-se inicialmente a posição da porta com 1 e para cada $k \geq 2$ examinam-se *todas* as casas livres do labirinto, marcando-se com k aquelas ainda não marcadas e que sejam vizinhas a alguma casa marcada com $k - 1$.

A marcação continua até ser atingido um valor de k (28 no exemplo abaixo) tal que nenhuma casa esteja em condições de ser marcada. Supondo que a $L = [1..6, 1..11]$,² e que a porta esteja na posição $[6, 11]$,³ ao final da marcação teremos a seguinte matriz:

26	27	-1	-1	12	11	10	9	10	11	12
25	-1	0	0	-1	12	-1	8	-1	-1	13
24	25	-1	0	-1	13	-1	7	6	5	-1
23	-1	21	-1	15	14	15	-1	-1	4	3
22	21	20	-1	16	-1	16	17	18	-1	2
23	-1	19	18	17	18	17	18	-1	2	1

¹Terceiro e último; o derradeiro!

²Esta notação deve ser intuitiva o suficiente. . .

³Idem!

Tendo feito esta marcação, um caminho mais curto do minotauro até a porta (se existir) pode ser determinado partindo-se da posição do minotauro e passando a cada etapa para uma casa vizinha cuja numeração seja menor do que a atual.

Por exemplo, se o minotauro estiver na posição $[3, 2]$, este precisará percorrer pelo menos 25 casas para chegar à porta: $[3, 2]$, $[3, 1]$, $[4, 1]$, $[5, 1]$, $[5, 2]$, $[5, 3]$, \dots , $[4, 10]$, $[4, 11]$, $[5, 11]$, $[6, 11]$.

2 Seu programa

Faça um programa em C para resolver o seguinte problema:

Dados dois inteiros positivos m , n , uma matriz $L = [1..m, 1..n]$, representando um labirinto, a posição $[m1, m2]$ de um minotauro e a posição $[p1, p2]$ da porta desse labirinto, determinar (se existir) um caminho mais curto que o minotauro deve percorrer até encontrar a porta. O labirinto é uma matriz de inteiros cujos elementos são 0 ou -1 , conforme a posição correspondente do labirinto seja uma passagem livre ou uma parede, respectivamente.

Se tal caminho não existir, o programa deve imprimir uma mensagem para que uma equipe de salvação vá urgentemente resgatar o minotauro.

O seu programa deve fazer uso de (pelo menos) duas funções, como descrito abaixo.

- Uma função com o protótipo

```
void marque(int L[][MAXCOL], int m, int n, int p1, int p2);
```

que tem como parâmetros um labirinto, isto é, uma matriz de inteiros $L = [1..m, 1..n]$, suas dimensões m e n , e dois inteiros $p1$, $p2$ que indicam a posição $[p1, p2]$ da porta nesse labirinto.

Essa função deve efetuar a marcação da matriz L , como explicado anteriormente. (Após a chamada dessa função, o seu programa deve imprimir essa matriz já marcada.)

- Uma outra função com o protótipo

```
void imprima_caminho(int L[][MAXCOL], int m, int n, int m1, int m2);
```

que tem como parâmetros uma matriz de inteiros $L[1..m, 1..n]$ já marcada, as dimensões m , n dessa matriz, e dois inteiros $m1$, $m2$ indicando a posição $[m1, m2]$ do minotauro nessa matriz.

Esta função deve imprimir, caso exista, o caminho a ser percorrido pelo minotauro até a porta. Aqui, queremos a ordem em que as posições $[i, j]$ da matriz devem ser percorridas.

Adicionalmente, para uma visualização melhor desse caminho no labirinto, essa função deve imprimir também um outro labirinto (matriz tipo *char*), onde apareçam caracteres apropriados indicando as paredes, as posições livres, a posição do minotauro (por exemplo, a letra M), a posição da porta (por exemplo, a letra P), e o caminho que o minotauro deve seguir.

3 O que entregar

Como nos exercícios-programas anteriores, você deve entregar um disquete com o seu programa-fonte, a listagem e a saída do seu programa. Você deve seguir as recomendações gerais que foram dadas desde o primeiro exercício-programa, e que se encontram em <http://www.ime.usp.br/~hitoshi/mac115/ep2/node8.html>

4 Opcional (vale 1.0 ponto adicional)

Para ganhar um ponto extra, o seu programa deve ler os dados de um arquivo, e também imprimir o resultado em um outro arquivo. Veja http://www.ime.usp.br/~yw/fisica2000/mac115/eps/ep3/exemplo_prog3.c para um exemplo de um programa que lê e escreve em arquivos. Note que a “receita” que estamos dando aqui é levemente diferente da que demos no EP2.