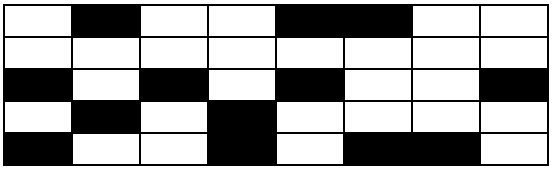
MAC110 – Introdução à Computação
Primeiro Semestre de 2018 – BMAC - IMEUSP
Exercício Programa 3 – Prof. Marcilio - Entregar até 18/06/2018

Considere o labirinto representado pela figura abaixo:



Pode ser representado por uma matriz onde elementos-parede tem valor -1 e elementos-espaço tem valor 0. Neste caso, a matriz tem 5 linhas por 8 colunas.

0	-1	0	0	-1	-1	0	0
0	0	0	0	0	0	0	0
-1	0	-1	0	-1	0	0	-1
0	-1	0	-1	0	0	0	0
-1	0	0	-1	0	-1	-1	0

Suponha agora que exista uma porta de saída (um dos elementos-espaço das bordas) e desejamos saber qual o menor caminho, se existir, de certo elemento até a porta de saída.
Por exemplo, (vamos numerar as linhas de 1 a 5 e as colunas de 1 a 8):
Supondo que a porta seja o elemento [5,8], será que existe caminho de [2,2] até a porta? E de [4,3] ?

Vamos supor que os movimentos podem ser realizados na horizontal e vertical.
Para resolver esse problema, podemos rotular os elementos a partir da porta da seguinte forma:
Rotula-se inicialmente a porta com 1. Em seguida, rotula-se seus vizinhos possíveis (que ainda não tenham sido rotulados) com 2. Em seguida, rotula-se com 3 os vizinhos possíveis dos elementos rotulados com 2 (que ainda não tenham sido marcados). E assim por diante, até que haja mais nada a rotular.

Dessa forma, o caminho mais curto de um elemento até a porta (se existir) pode ser determinado, partindo-se desse elemento e passando a cada vez, para um elemento vizinho cuja numeração seja menor do que a atual.

Veja como fica a numeração no exemplo de labirinto dado:

12	-1	10	9	-1	-1	6	7
11	10	9	8	7	6	5	6
-1	11	-1	9	-1	5	4	-1
0	-1	0	-1	5	4	3	2
-1	0	0	-1	6	-1	-1	1

Da posição [2,2] existe caminho. É necessário percorrer 10 passos para se chegar à porta. Da posição [4,3] não há caminho.

Faça um programa que:

a) Leia uma matriz labirinto com 0s e -1s . Como a matriz é grande (n linhas e m colunas = n * m elementos), em vez de digitá-la, deve ser lida de um arquivo de texto **`<arquivo>.txt`**. A função **`LeiaMatriz`** abaixo é um exemplo de como fazer essa leitura. Digite a matriz usando o próprio editor do Python e salve no próprio diretório onde está o seu programa com extensão **`txt`**.
b) Para cada uma das portas (portas são os elementos com 0 que estão nas bordas da matriz:
   - Efetue a numeração dos elementos como especificado
   - Imprime a matriz
   - Imprime as posições com caminho máximo até essa porta

- Imprime as posições sem caminho até essa porta

(lembre-se que os índices das linhas e colunas de uma matriz (listas com listas) começam em 0).

Faça pelo menos as seguintes funções:

```
numera_vizinhos (lab, n, m, k):
# Varre a matriz lab[0..n-1][0..m-1], procurando todos os elementos
# [i][j] tais que de lab[i][j] == k.
# Rotula com k+1 os vizinhos destes elementos (horizontal e vertical)
# que ainda não tenham sido rotulados.
```

A função devolve True se encontrou algum elemento igual a k, ou -1 caso contrário e nesse caso o programa principal deve parar de chamá-la, pois todos os possíveis.

```
def imprime_mais_distante(lab, n, m, dist):
# Imprime as posições com distância igual a dist. Usada para
# imprimir as posições de caminho máximo e sem caminho (dist = 0).
```

Ler a matriz

```
def LeiaMatriz(Mat, NomeArq):
    # retorna True se conseguiu ler o arquivo e False caso contrário

    # abrir o arquivo para leitura
    try:
        arq = open(NomeArq, "r")
    except:
        print("erro no open")
        return False

    # ler cada uma das linhas do arquivo
    i = 0
    for linha in arq:
        # se der alguma exception retorna False
        try:
            v = linha.split()# separa os elementos da string
            Mat.append([])   # adiciona uma nova linha a matriz
            # transforma os strings numéricos em números inteiros
            # e adiciona cada int à matriz
            for j in range(len(v)):
                Mat[i].append(int(v[j]))
            i = i + 1
        except:
            # algum erro no trecho acima
            print("erro no split, int ou nos appends")
            return False
    # consistência dos valores da matriz
    # verificar se todas as linhas tem o mesmo número de elementos
    # . . .
    # verificar se todos os elementos são 0s e -1s
    # . . .
    # fechar arquivo
    arq.close()
    return True

# exemplo de chamada da LeiaMatriz
mat = []
print(LeiaMatriz(mat, "meulab.txt"))
print(mat)
```

Exemplo de saída do programa:

```
Entre com o nome do arquivo:meulabirinto.txt
Matriz labiринto com 5 linhas por 8 colunas
   0  -1   0   0  -1  -1   0   0
   0   0   0   0   0   0   0   0
  -1   0  -1   0  -1   0   0  -1
   0  -1   0  -1   0   0   0   0
  -1   0   0  -1   0  -1  -1   0

Porta [0, 0]:
Caminhos possíveis e comprimento:
   1  -1   5   6  -1  -1   9  10
   2   3   4   5   6   7   8   9
  -1   4  -1   6  -1   8   9  -1
   0  -1   0  -1  10   9  10  11
  -1   0   0  -1  11  -1  -1  12

Posições com caminho de máximo comprimento:
[4, 7]

Posições sem caminho:
[3, 0]
[3, 2]
[4, 1]
[4, 2]

Porta [1, 0]:
Caminhos possíveis e comprimento:
...
...
```