

LABORATÓRIO DE PROGRAMAÇÃO AVANÇADA
NONO TRABALHO PRÁTICO
-- BUSCA EM PROFUNDIDADE EM GRAFOS --

Este trabalho é uma continuação exercício 8 (**Problema do Labirinto**) e, quem fez o trabalho, não terá qualquer dificuldade para fazer esse outro trabalho. Portanto, vamos continuar resolvendo o problema do labirinto (exercício 7 e 8) só que agora usando a **BUSCA EM PROFUNDIDADE** em grafos. **Da mesma forma que o trabalho 8**, vamos continuar lendo uma matriz de números 0's e 1's, onde o número 0 indica caminho aberto e o número 1 indica caminho fechado.

As informações do labirinto devem ser passadas por um arquivo de entrada (`argv[1]`). A primeira linha deve conter as dimensões da matriz da seguinte forma:

```
<linhas>\t<colunas>\n
```

Em seguida deve ser lido a matriz que representa o labirinto. No arquivo deve estar uma linha por vez, da seguinte forma:

```
<valor 1,1>\t<valor 1,2>\t ... \t<valor 1,N>\n
<valor 2,1>\t<valor 2,2>\t ... \t<valor 2,N>\n
. . .
<valor M,1>\t<valor M,2>\t ... \t<valor M,N>\n
```

Após ler a matriz de entrada, você deverá transformar essa matriz em um grafo não-direcionado, isto é, se o nó A está ligado ao nó B, então necessariamente o nó B está ligado ao nó A. A saída deve ser no formato “lista de adjacência”, da seguinte forma:

```
<Nó 0>:\t<nó N01>\t<nó N02>\t ... \t<nó N0x>
<Nó 1>:\t<nó N11>\t<nó N12>\t ... \t<nó N1y>
...
<Nó k>:\t<nó Nk1>\t<nó Nk2>\t ... \t<nó Nkz>
```

EXEMPLO

Aqui é só um exemplo e você deverá testar com outros valores.

Supondo que o labirinto seja uma matriz de dimensões 7x6 (7 linhas e 6 colunas), cujos valores sejam:

0	1	1	1	1	1
1	0	0	0	1	1
1	0	0	0	0	1
1	0	1	0	0	1
1	1	1	0	0	1
1	0	0	0	0	1
1	1	1	1	1	0

Nesse caso, a entrada deve ser:

7	6				
0	1	1	1	1	1
1	0	0	0	1	1
1	0	0	0	0	1
1	0	1	0	0	1
1	1	1	0	0	1
1	0	0	0	0	1
1	1	1	1	1	0

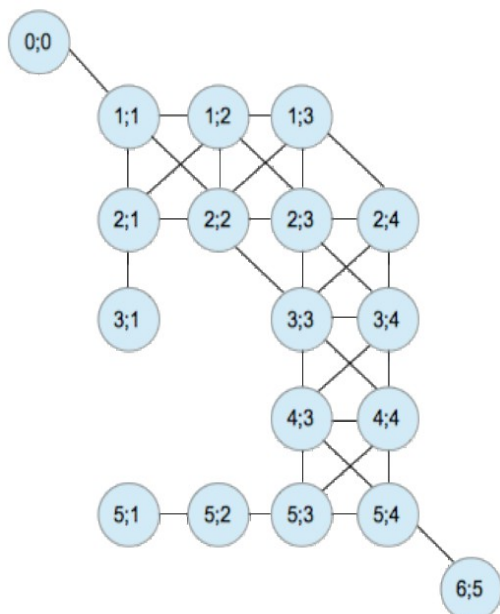
Você deverá gerar uma estrutura de dados do tipo **lista de adjacência** para armazenar o grafo que representa o labirinto de entrada. **Note que o grafo é uma representação intermediária e que, nesse trabalho, não será a saída.** Para a matriz de entrada, o seguinte grafo deve ser gerado.

```

0_0: 1_1
1_1: 0_0 1_2 2_1 2_2
1_2: 1_1 1_3 2_1 2_2 2_3
1_3: 1_2 2_2 2_3 2_4
2_1: 1_1 1_2 2_2
2_2: 1_1 1_2 1_3 2_1 2_3 3_3
2_3: 1_2 1_3 2_2 2_4 3_3 3_4
2_4: 1_3 2_3 3_3 3_4
3_1: 2_1
3_3: 2_2 2_3 2_4 3_4 4_3 4_4
3_4: 2_3 2_4 3_3 4_3 4_4
4_3: 3_3 3_4 4_4 5_3 5_4
4_4: 3_3 3_4 4_3 5_3 5_4
5_1: 5_2
5_2: 5_1 5_3
5_3: 4_3 4_4 5_2 5_4
5_4: 4_3 4_4 5_3 6_5
6_5: 5_4

```

Cuja representação gráfica do grafo acima é (não precisa gerar tal representação):



De posse do grafo, executar a **busca em profundidade** em grafos para, a partir do nó **0_0** chegar ao nó **M_N**. A saída deve ser um dos possíveis caminhos do rato no seguinte formato:

```
L(x1, y1) \n
L(x2, y2) \n
...
L(xm, yn) \n
```

Onde necessariamente $x_1 = y_1 = 0$; e $x_m = M$ e $y_n = N$. No caso do exemplo anterior, o nó final seria o nó **6_5**.

Este trabalho deve ser entregue no dia **27/06/2014**. Entretanto, vou liberar a entrega até a data máxima de **30/06/2014** (segunda) até meia-noite.

IMPORTANTE! Após esta data, o trabalho não será mais aceito.

Envie também alguns testes que você tenha feito, tanto as entradas quanto as saídas. Para gerar a saída em arquivos, você pode redirecionar a saída. Por exemplo:

```
lpav_tp9 entrada.txt > saida.txt
```

O trabalho deve ser possível ser compilado pelo **GCC** e executado no **Linux**.

Envie para o professor (xbarretox@gmail.com) e o monitor (marrco.santos@gmail.com).

Coloque no assunto: [**LPAV-TP09**].