

MAC 122 - Princípios de Desenvolvimento de Algoritmos**Segundo semestre de 2007**Primeiro Exercício-Programa – Entrega: **23 de agosto****Identificando regiões em imagens binárias**

O objetivo deste primeiro exercício-programa é o uso e manipulação de listas ligadas. Há várias aplicações interessantes do problema que tratamos neste EP tais como: aplicações militares (reconhecimento de instalações ou objetos em imagens de satélite), aplicações na agroindústria (previsão de safras) ou na medicina (identificação de regiões cancerígenas em imagens de tomografia).

Neste primeiro EP vocês implementarão um algoritmo de identificação de regiões conexas numa imagem binária (preto e branco). É dada uma imagem através de uma matriz de “pixels”, onde cada pixel pode assumir o valor 0 (branco) ou 1 (preto). Desejamos identificar os objetos conexas que compõem esta imagem.

Exemplo: Considere a matriz dada abaixo.

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

A matriz contém 5 objetos conexas (estamos considerando que as posições vizinhas de uma posição da matriz são: a posição de cima, de baixo, da esquerda e da direita). Na figura abaixo os pixels

de um mesmo objeto recebem o mesmo rótulo, e cada objeto tem um rótulo único:

```
0 0 1 1 0 0 2 0 2 0
0 1 1 0 0 0 2 0 2 0
0 0 1 1 0 0 2 0 2 0
0 0 0 0 0 0 2 2 2 0
0 0 0 0 0 0 0 0 0 0
0 0 3 0 0 0 4 0 4 0
0 0 3 0 0 4 4 4 4 4
0 0 3 0 0 0 0 0 0 0
3 3 3 3 3 0 0 0 0 0
0 0 3 0 0 0 5 5 5 0
0 0 3 0 0 0 5 5 5 0
0 0 3 0 0 0 5 5 5 0
```

A tarefa deste exercício-programa será resolver o problema acima. Ou seja, dada a matriz da imagem, seu programa deverá rotular os “pixels” da matriz de forma a identificar os objetos conexos nela contidos. Seu programa deverá ser bastante eficiente, evitando percorrer toda a matriz de entrada várias vezes. Uma estratégia que **deverá** ser implementada é a que descrevemos a seguir. A matriz é varrida de cima para baixo e da esquerda para a direita. Se um “pixel” tem valor 1, verificamos seu vizinho de cima e o da esquerda. Se ambos são 0, um novo objeto começa neste pixel. Se um deles é 1, este pixel pertence ao mesmo objeto do anterior. Se ambos são 1 e iguais, o pixel também pertence ao objeto anterior. Finalmente, se ambos os vizinhos têm rótulos diferentes, a matriz deverá ser corrigida para que ambos tenham o mesmo rótulo (ou seja, um dos rótulos vai desaparecer e todos os pixels daquele rótulo vão passar a ter o outro). A fim de realizar eficientemente esta última operação cada objeto deverá manter uma lista ligada com os “pixels” associados com seu rótulo.

Entrada

São dadas várias instâncias. Cada instância é composta pelo número m de linhas e de n colunas da imagem e, a seguir, por m linhas compostas cada uma por n 0's ou 1's. A entrada termina com uma matriz 0×0 .

Exemplo:

```
4 3
0 1 0
1 1 1
0 0 0
1 0 1
4 5
1 0 1 0 1
1 0 1 0 1
1 0 1 0 1
1 1 1 1 1
0 0
```

Saída

Para cada instância seu programa deverá imprimir uma mensagem identificando a instância e a matriz modificada, com um rótulo para cada objeto conexo da imagem. Observe o exemplo acima correspondente à entrada dada na seção anterior.

Instância 1

```
0 1 0
1 1 1
0 0 0
2 0 3
```

Instância 2

```
1 0 1 0 1
1 0 1 0 1
1 0 1 0 1
1 1 1 1 1
```

Observações

- Qual o número máximo de vezes que um mesmo “pixel” é analisado utilizando esta estratégia? Você é capaz de fazer uma estratégia mais eficiente?
- Capriche em todos os aspectos do seu trabalho: documentação, identificação, eficiência, clareza, etc. Isso será levado em conta na sua avaliação.
- Fique a vontade para implementar adicionalmente outras estratégias para resolver o problema.
- O EP deve ser feito individualmente. Se você discutir parte de seu EP com seus colegas façam suas implementações separadamente para garantir que não se configurará cola nos EPs.