MAC 115 – Introdução à Computação para Ciências Exatas e Tecnologia

Instituto de Física – Segundo Semestre de 2001 - Bacharelado Noturno

Terceiro Exercício-Programa (EP3) Prazo de entrega: 29 de novembro de 2001

Alagações na USSP

Devido a um forte temporal que caiu em São Paulo, várias regiões da cidade universitária da USSP (Universidade de Sábios Samaritanos Pragmáticos) ficaram alagadas. Representamos o *campus* da USSP por um reticulado, como o da figura abaixo, onde 0 representa uma posição **seca** e -1 representa uma posição **alagada**.

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

Dizemos que duas posições deste reticulado são **adjacentes** se possuem uma aresta em comum. Uma **região R** é definida como um conjunto de posições tal que é possível a partir de uma posição de **R**, atingir qualquer outra posição de **R** através de posições adjacentes. Se todas as posições de uma região contêm -1 então a região é dita **alagada**. Uma **região alagada maximal** é uma região alagada que não está contida em nenhuma região alagada a não ser ela mesma.

Exercício-programa

Faça um programa em C que resolva o seguinte problema: dado um reticulado do tipo acima (uma matriz), marcar todas as regiões alagadas maximais. Especificações a serem seguidas:

1. Dados: dois inteiros m e n e $U_{m \times n}$ uma matriz inteira representando um reticulado com entradas 0 e -1. (Na figura acima temos $m = n = 10$.)
2. A partir da matriz U construir uma nova matriz M (de mesma dimensão) marcando cada **região alagada maximal** com um número inteiro, da seguinte forma: Se existirem k regiões alagadas maximais então cada posição de uma mesma região recebe um número entre 1 a k, sendo que todas as posições de uma mesma região devem receber um mesmo número. Por exemplo, o reticulado acima tem 8 regiões alagadas maximais. Neste caso, um exemplo de saída seria:

1	0	0	0	0	2	2	0	3	3
1	1	1	0	0	0	2	0	3	0
0	1	0	0	4	0	0	0	0	0
0	0	0	4	4	4	4	4	0	0
0	0	0	0	4	4	0	0	5	5
0	0	6	0	0	0	0	5	5	5
0	0	6	6	6	0	0	0	5	5
7	0	6	6	6	6	0	0	0	5
7	0	0	0	0	0	8	8	0	0
7	0	0	0	0	0	8	0	0	0

3. Imprimir a matriz dada U e a matriz marcada M.

4. Use obrigatoriamente seguintes funções:
   - função **void CriaLista**
     - parâmetros:
       m, n: números inteiros
       MATRIX: matriz inteira $m \times n$
       LISTA: matriz inteira $(m * n) \times 3$
     - descrição:
       A partir da matriz MATRIX essa função deve construir a matriz LISTA. Esta deverá indicar quais são as posições alagadas de MATRIX. A informação de que uma posição (x, y) de MATRIX está alagada deve ficar armazenada numa linha da matriz LISTA: a primeira coluna armazena x e a segunda coluna armazena y (a terceira coluna não é alterada por essa função). Se a matriz MATRIX tiver t posições alagadas, então as t primeiras linhas (juntamente com as correspondentes duas primeiras colunas) da matriz LISTA armazenarão essas informações. As linhas restantes deverão conter -3 por convenção.
   - função **int EhVizinho**
     - parâmetros de entrada:
       x, y, z, w: números inteiros
     - descrição:
       se a posição (x, y) for adjacente à posição (z, w), então a função deve devolver o valor 1; caso contrário, devolver o valor 0.
   - função **void MarcaComK**
     - parâmetros:
       k, lin, m, n: números inteiros
       MAT: matriz inteira $m \times n$
       LISTA: matriz inteira $(m * n) \times 3$
     - descrição:
       Esta função recebe como entrada a matriz MAT e a devolve marcada. Para isso, ela deve usar as duas funções anteriores. O parâmetro *lin* indica qual é a linha da matriz LISTA que está sendo usada como ponto de partida para atribuição do valor k. Ou seja, a partir da posição (x, y) que está na linha *lin* da matriz LISTA, deve-se percorrer a LISTA e identificar posições que são adjacentes a (x, y), e armazenar tanto em MAT(x, y) como na 3a. coluna de LISTA o valor k. Note que as posições que estão na LISTA e que receberam o valor k na 3a. coluna devem agora ser testadas para verificar se têm vizinhos adjacentes que também devem receber o valor k. Ou seja, o processo de marcar o valor k deve continuar enquanto houver alguma posição na LISTA que recebeu o valor k e ainda não teve os seus vizinhos testados. Nessa função você deve usar as duas funções anteriores.
       Sugestão: à medida que posições na LISTA vão sendo testadas e vão sendo marcadas (com algum inteiro k), para indicar que estas já foram marcadas, em vez de removê-las da LISTA, você pode armazenar nessas posições um valor negativo, como por exemplo, -3 ou -4 para indicar isso.

5. **Observações**
   - Para imprimir a matriz, basta imprimir os números (não é necessário imprimir as linhas horizontais e verticais).
   - Teste o seu programa obrigatoriamente com a matriz dada no exemplo ($m = n = 10$).
   - Você pode utilizar mais outras funções, além das que foram mencionadas acima.

6. **Bônus (opcional)**

   Se, adicionalmente, o seu programa imprimir uma matriz do tipo *char* correspondente à matriz M (marcada), indicando as regiões secas e as diferentes regiões alagadas maximais com caracteres bem adequados e bonitos, você receberá uns pontos adicionais na nota deste EP3. Use sua imaginação (não precisa necessariamente numerar as regiões alagadas)! Também sugerimos que você faça leitura de um arquivo e impressão em um arquivo.

Boa sorte!