Algoritmos e Estruturas de Dados 1

Trabalho Prático
2º Semestre de 2019 – DC-UFSCar
Data sugerida para a entrega: 26/Out

1 Introdução

Neste trabalho prático é solicitada a entrega de um programa que solucione o problema apresentado na próxima seção. O arquivo entregue deve seguir os itens abaixo:

- O trabalho deverá ser feito individualmente e plágio não será tolerado;
- O cabeçalho de cada função pedida deve seguir o modelo indicado na sua descrição;
- Você pode adicionar mais bibliotecas caso necessite;
- Outras funções auxiliares podem ser criadas para facilitar o desenvolvimento, desde que as funções pedidas no enunciado estejam presentes e se comportem de acordo com o que foi pedido;
- O trabalho deve ser entregue no run codes (https://run.codes) em um arquivo contendo código em linguagem C e com um cabeçalho com as informações do estudante (nome, curso, RA);
- Cada estudante deve se cadastrar no run codes (https://run.codes) informando Nome Completo, escolhendo "UFSCar - Universidade Federal de São Carlos" no campo Universidade e colocando seu RA no campo Núm. Matrícula. Depois de cadastrado, basta logar no run codes e se matricular na disciplina "1001502 - Algoritmos e Estruturas de Dados 1" usando o Código de Matrícula WAE8.

2 O problema

"*Quero encontrar a ilha desconhecida, quero saber quem sou eu quando nela estiver*". José Saramago, *O Conto da Ilha Desconhecida.*

Em sua jornada por auto-conhecimento, você está em busca de uma ilha. Seus problemas acabaram (ou quase)! Você recebeu e-mail de um príncipe de um reino distante oferecendo-lhe uma ilha de seu arquipélago. Por sorte, o e-mail não foi filtrado como *spam*. O príncipe *Steck Über Flux* lhe enviou um mapa de seu arquipélago para que você fizesse sua escolha. No entanto, você ainda não está livre de sua ambição e deseja escolher a ilha certa.

Nesse trabalho prático, deve ser desenvolvida uma solução para encontrar a maior ilha em um arquipélago. O mapa da região, que pode conter uma ou mais ilhas, é representado por uma matriz binária. Cada entrada/célula da matriz a_{ij} é definida como:

$$a_{ij} = \begin{cases} 0, & \text{se a célula representa uma área marítma,} \\ 1, & \text{se a célula representa uma área terrestre.} \end{cases}$$

Duas células são consideradas adjacentes (partes da mesma ilha) se são vizinhas horizontal ou verticalmente na matriz. Ou seja, cada célula possui, no máximo, quatro vizinhos (Figura 1).

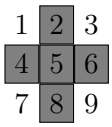

Figura 1: Representação gráfica dos vizinhos da célula 5. Nesse caso, os vizinhos são as células 2, 4, 6 e 8

Com isso, pode-se apresentar o problema da seguinte forma.

> Dada uma matriz binária bidimensional M. Encontre o número de elementos do maior conjunto de células adjacentes com entradas iguais a 1.

3 Tarefa

Como tarefa deste trabalho prático, você deve implementar a função *search_max_area* que recebe a matriz m e seu respectivo número de linhas e colunas para que o programa funcione corretamente. A função possui o seguindo cabeçalho.

```
1  int search_max_area(int **m, int n_rows, int n_cols)
```

Também é fornecida a função *main*.

```c
int main(void){
    int n_rows = 0, n_cols = 0;
    scanf("%d %d\n", &n_rows, &n_cols);
    int **m = create_matrix(n_rows, n_cols);
    read_input(m, &n_rows, &n_cols);
    int area = search_max_area(m, n_rows, n_cols);
    printf("%d", area);
    /*Funcao para desalocar a matriz.*/
    destroy_matrix(m, n_rows, n_cols);
    return 0;
}
```

Além do código já apresentado, também são fornecidas implementações da função que aloca a matriz m (*create_matrix*) e da função que realiza a leitura da entrada de dados (*read_input*).

```c
int** create_matrix(int n_rows, int n_cols){
    int** m = NULL;
    m = malloc(n_rows*(sizeof(int*)));
    for (int i = 0; i < n_rows; ++i)
        m[i] = malloc(n_cols*sizeof(int));
    return m;
}


void read_input(int** m, int *n_rows, int *n_cols){
    char line[MAX_COLS];
    for (int i = 0; i < *n_rows; ++i) {
        fgets(line, MAX_COLS, stdin);
        for (int j = 0; j < *n_cols; ++j) {
      if(line[j] == '0')
          m[i][j] = WATER;
      else if(line[j] == '1')
          m[i][j] = LAND;
      else{
          printf("Wrong char in buffer\n");
          printf("%s\n", line);
          exit(EXIT_FAILURE);
            }
        }
    }
}
```

É importante seguir o modelo dado acima para possibilitar a correção. Você pode adicionar o que for necessário, como bibliotecas e alterar a ordem de funções, desde que o que existe nesse modelo continue existindo no trabalho enviado.

4 Casos de teste

Os dados de entrada são as dimensões da matriz (número de linhas n e colunas m) e suas entradas a_{ij}. A primeira linha da entrada é composta por dois inteiros representando n e m, respectivamente. Então, seguem-se n linhas, cada uma com m caracteres (0 ou 1). Um exemplo é dado a seguir, considerando uma matriz 10 x 10.

```
10 10
0000000000
0011000000
0011000000
0011000000
0010000000
0000001100
0000011100
0000000000
0000000000
0000000000
```

Como saída esperada, o programa desenvolvido deve fornecer o tamanho (número de células) da maior ilha. Ou seja, deve imprimir um número inteiro positivo. Para o exemplo apresentado, a saída esperada é:

```
7
```

Outro exemplo é dado a seguir:

```
10 10
0000000000
0011000000
0011000000
0011111000
0010001000
0000001100
0000011100
0000000000
0000000000
0000000000
```

Cuja saída esperada é:

```
16
```