

Trabalho 1

1 Flood Fill

O preenchimento de inundação (*Flood Fill*), também chamado de preenchimento de semente (*Seed Fill*), é um algoritmo que determina a área conectada a um determinado nó em um vetor multidimensional (matriz). É usado na ferramenta de preenchimento “balde” dos programas de pintura para preencher áreas conectadas de cores semelhantes com uma cor diferente e em jogos como Go e Minesweeper para determinar quais peças são limpas. O algoritmo de preenchimento usa três parâmetros: um nó inicial, uma cor de destino e uma cor de substituição. O algoritmo procura todos os nós na matriz que estão conectados ao nó inicial por um caminho da cor de destino e os altera para a cor de substituição [1].

2 Descrição das Atividades

2.1 Programa

Você deverá escrever um programa em linguagem C que aplique o algoritmo *Flood Fill* para encontrar todas as posições conectadas, de mesmo valor, em uma matriz binária (0s e 1s). O seu programa deve:

- Ler, do teclado, o nome de um arquivo que deverá ser aberto para leitura;
- Ler as informações do arquivo, estruturado conforme descrito posteriormente;
- Aplicar o algoritmo *Flood Fill* na matriz lida;
- Retornar, em tela, o número de posições do mesmo valor conectadas, a partir da posição inicial, seguido pelas coordenadas de todas estas posições.

2.2 Contagem de operações

Em arquivo separado, você deve enviar a contagem de operações do seu algoritmo. Como a contagem é dependente da estrutura da matriz, você deverá realizá-la apenas para as informações abaixo:

```
11 11
1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1
```

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

3 Estruturação dos arquivos

Os arquivos estarão estruturados da seguinte maneira

- 1ª Linha: Dois inteiros, X e Y , representando o número de linhas e de colunas da matriz, respectivamente;
- X linhas seguintes: Matriz com $X * Y$ valores, separados em X linhas, com Y valores em cada linha, separados por um espaço. Os valores serão apenas 0 ou 1;
- Última linha: Dois inteiros, M e N , representando as coordenadas da posição inicial do algoritmo (M = linha, N = coluna):

$$- 0 \leq M < X$$

$$- 0 \leq N < Y$$

Exemplo:

```
8 10
0 0 0 0 0 0 0 0 0 0
0 0 1 1 0 0 0 0 0 0
0 0 0 0 1 1 0 0 0 0
0 0 0 0 1 1 0 0 0 0
0 0 1 1 1 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 1 1 0 0 0 0 0 0
1 2
```

4 Instruções adicionais

- Outros arquivos, de mesma estruturação dos enviados, poderão ser utilizados para avaliação de seu código;
- A contagem de operações deve ser feita para a sua implementação do algoritmo;

USP-ICMC-EngComp
ED-II
Trabalho 1 (continuação)

- Nas primeiras linhas do código ou, se preferir, em arquivo separado, descreva a sua abordagem para implementação do algoritmo (estratégia utilizada etc);
- A utilização de estruturas de dados (*structs*, filas, pilhas) facilita a implementação do algoritmo;
- É ideal que o algoritmo principal seja programado com **recursão**;
- Comentários e organização do código auxiliam na correção e entendimento de sua lógica. Será esperado o mínimo de organização para uma boa leitura do código.

5 Exemplo entrada e saída

Entrada	Saída
8 10	Total:10
0 0 0 0 0 0 0 0 0 0	1 2
0 0 1 1 0 0 0 0 0 0	1 3
0 0 0 0 1 1 0 0 0 0	2 4
0 0 0 0 1 1 0 0 0 0	2 5
0 0 1 1 1 0 0 0 0 0	3 4
0 0 0 0 1 0 0 0 0 0	3 5
0 0 0 0 0 0 0 0 0 0	4 2
0 0 1 1 0 0 0 0 0 0	4 3
1 2	4 4
	5 4

6 Prazo

O trabalho deverá ser implementado em grupo e entregue (fonte e relatório) através do escaninho de um membro do grupo na plataforma ae4.tidia-ae.usp.br até às 23h59 do dia 21 de abril de 2020. Esse prazo é **improrrogável**. O grupo deve se inscrever até o dia 15/4 pelo escaninho do Tidia-Ae. Além disso, haverá uma apresentação em forma de mini-seminário de 15 minutos nas aulas dos dias 22 e 23/4/2020. Todos os elementos do grupo deverão estar presentes.

Referências

[1] Wikipedia, https://en.wikipedia.org/wiki/Flood_fill.