

Trabalho 1

Implementar um programa em C que recebe como entrada uma instância do jogo Floodit e que gera como saída uma sequência de cores que pinta o tabuleiro todo com uma única cor.

Especificação:

=====

O seu programa deve receber uma instância do jogo a partir da entrada padrão do sistema.

Uma instância é dada por um arquivo texto no seguinte formato:

- a primeira linha do arquivo tem 3 números inteiros n , m e k , separados por espaço, onde n é o número de linhas do tabuleiro, m é o número de colunas do tabuleiro e k é o número de cores presentes no tabuleiro;
- o restante do arquivo contém n linhas com m números inteiros cada, separados por espaços, sendo que cada um destes valores pertence ao intervalo $[1..k]$;
- o exemplo abaixo é uma instância do jogo com 4 linhas, 5 colunas e 3 cores:

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

A saída do seu programa deve ser pela saída padrão do sistema e deve conter duas linhas de texto:

- a primeira linha deve conter um número inteiro que indica o tamanho da sequência de números da segunda linha;
- a segunda linha deve conter uma sequência de números inteiros pertencentes ao intervalo $[1..k]$, separados por espaço e terminada por uma quebra de linha.

Esta sequência representa as cores que devem ser usadas para pintar o tabuleiro com uma única cor. A operação de pintura muda a cor da região que contém a célula do canto superior esquerdo do tabuleiro.

Por exemplo, considere a seguinte sequência de cores, para o tabuleiro dado como exemplo:

```
5
3 2 1 3 2
```

Ao aplicar a operação de pintura, usando esta sequência de cores, as seguintes transformações ocorrem no tabuleiro:

Pintar com a cor 3:

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

Pintar com a cor 2:

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

Pintar com a cor 1:

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

Pintar com a cor 3:

```
3 3 3 3 2
3 3 3 3 3
3 3 3 3 2
3 3 3 3 3
```

Pintar com a cor 2:

```
2 2 2 2 2
2 2 2 2 2
2 2 2 2 2
2 2 2 2 2
```

Note que ao final o tabuleiro tem uma única cor. Portanto, esta sequência resolve a instância dada como exemplo.

Equipe:

=====

O trabalho pode ser feito em equipes de até dois alunos.

Avaliação:

=====

O seu trabalho será avaliado considerando:

- Estratégia utilizada;
- Qualidade da implementação e código fonte;
- Desempenho na competição com as outras implementações;
- Seu trabalho deve resolver uma instância de 100x100 com 20 cores em um tempo máximo de 120 segundos em um espaço de no máximo 8GB de memória;
- Defesa do código fonte se necessário.

Entrega:

=====

O trabalho deve ser entregue por email para:

- fabiano@inf.ufpr.br
- com o assunto/subject: trabalho1
- até 23:59 do dia 03/04/2022
- o corpo do email deve conter o nome completo e GRR dos membros da equipe
- o anexo do email deve conter um arquivo no formato .tar.gz com:
 - código fonte
 - makefile
 - leiametext
 - este último deve conter a descrição detalhada da estratégia implementada e outras informações que considerar relevantes para a avaliação do trabalho

Material de apoio:

=====

Os arquivos e programas abaixo podem ajudar:

- `exemplo_mapa_30_30_10.txt`: exemplo de instância do problema;
- `exemplo_soluc o_30_30_10.txt`: exemplo de uma solu  o poss vel para a inst ncia exemplo;
- `geramapa`: programa para gerar inst ncias aleat rias do problema;
- `verifica`: programa para verificar se um determinado mapa   resolvido por uma determinada sequ ncia de cores, retornando 0 se sim e 1 caso contr rio, exemplo de uso:
\$ cat `exemplo_mapa_30_30_10.txt` `exemplo_soluc o_30_30_10.txt` | `./verifica`
para pegar o resultado no bash:
if ["\$?" -eq 0]
then
echo ok
else
echo fail
fi
- `anima`: programa para gerar uma anima  o da solu  o de um mapa no terminal
exemplo de uso:
\$ cat `exemplo_mapa_30_30_10.txt` `exemplo_soluc o_30_30_10.txt` | `./anima`
- `floodit_h1`: resolvidor exemplo que usa um algoritmo guloso que escolhe a cor que mais aparece na borda da regi o do canto superior esquerdo;
- `floodit_h2`: resolvidor exemplo que usa um algoritmo guloso que escolhe a cor que mais aparece nas regi es adjacentes   regi o do canto superior esquerdo;
- `floodit_h4`: resolvidor exemplo que usa uma estrat gia baseada em dist ncias.