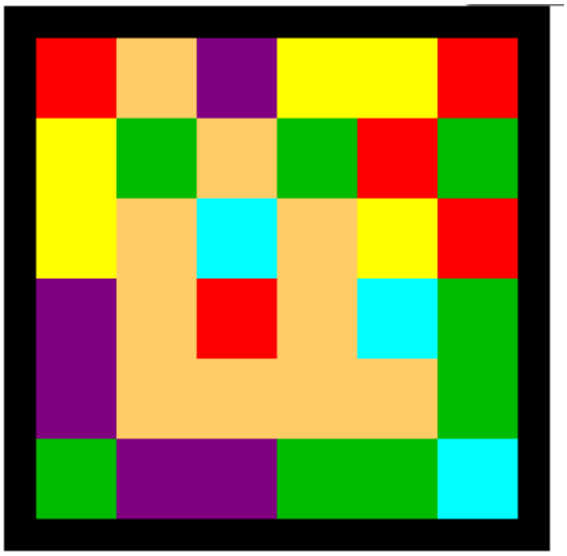
Trabalho 01 - Flood It!

Importante:

- TODOS os membros do grupo devem participar e compreender completamente a implementação.
- Em caso de plágio, fraude ou tentativa de burlar o sistema será aplicado nota 0 na disciplina aos envolvidos.
- Alguns alunos podem ser solicitados para explicar com detalhes a implementação.
- Passar em todos os testes não é garantia de tirar a nota máxima. Sua nota ainda depende do cumprimento das especificações do trabalho, qualidade do código, clareza dos comentários, desempenho do algoritmo, boas práticas de programação e entendimento da matéria demonstrada em possível reunião.
- Esse projeto poderá ser realizado em grupos de até 2 pessoas. Este trabalho deverá ser implementado em linguagem C.
- Você deverá implementar uma **HEURÍSTICA** para resolver o problema proposto.

Flood It! é um jogo em que um tabuleiro n x n é fornecido e cada célula é pintada por uma das 6 cores disponíveis. Na figura abaixo vemos um exemplo de um tabuleiro 6 x 6.



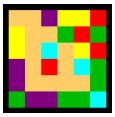
- Você pode mudar a cor da primeira célula no canto superior esquerdo.
- Quando muda a cor de uma célula, todas as células adjacentes (pra cima, pra baixo, pra esquerda e pra direita) que tiverem a mesma cor, também mudam de cor, que por sua vez também mudam a cor das vizinhas seguindo a mesma regra, e assim por diante.
- O objetivo do jogo é fazer com que todas as células tenham a mesma cor, minimizando o número de movimentos (mudanças de cores da primeira célula)

No exemplo acima vamos fazer uma sequência de 8 movimentos:

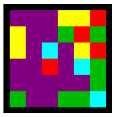
1. Laranja

2. Verde

3. Laranja

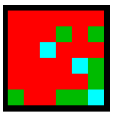


4. Roxo

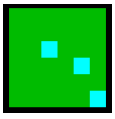


5. Amarelo

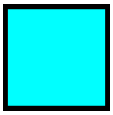
6. Vermelho



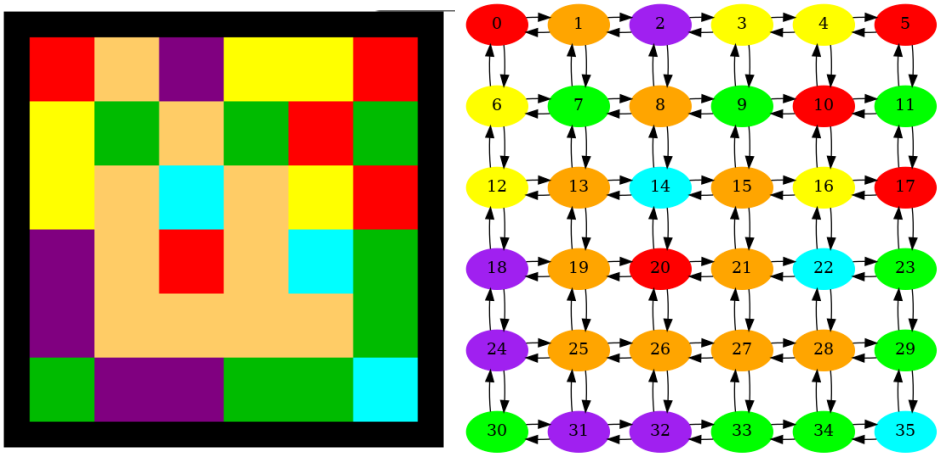
7. Verde



8. Ciano



Podemos entender o tabuleiro de Flood It! como um grafo em que células vizinhas têm arestas que indicam a direção por onde uma cor pode se propagar. No exemplo acima, podemos interpretar o tabuleiro como o seguinte grafo.

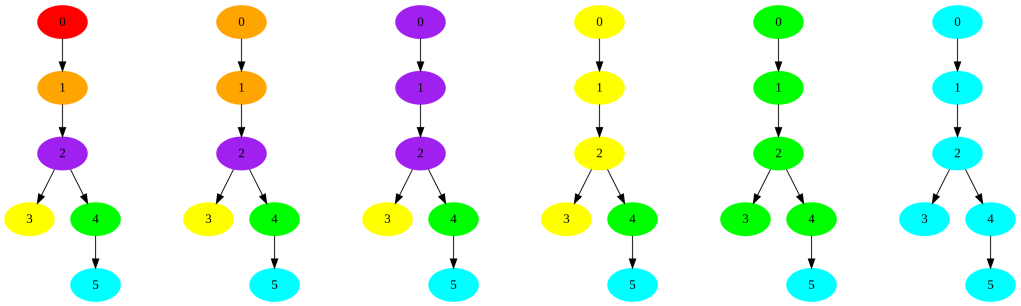


Dessa forma também podemos generalizar o jogo do Flood It! para um problema mais geral, o Problema do Flood It!

Problema do Floot It!

Dado um Grafo direcionado D = (V, A), com um vértice fonte s ∈ V, e cada vértice v ∈ V com uma cor inicial c(v) ∈ {0, 1, 2, 3, 4, 5}. Pintar a fonte de j significa trocar a cor da fonte para j. Sempre que um vértice u troca para a cor j, todo vértices vizinho w, ou seja, que (u, w) ∈ A, que tinha a mesma cor de u também troca a cor para j. Encontrar uma sequência mínima de cores L que será pintado o vértice fonte, de forma que todos os vértices tenham a mesma cor.

Considere o seguinte grafo na primeira imagem e a sequência de movimentos {Laranja, Roxo, Amarelo, Verde, Ciano}. O tamanho dessa lista é 5, ou seja, 5 pinturas são necessárias para resolver o problema, note que esse número é ótimo.



O seu trabalho será encontrar uma solução **heurística** para o problema, que tenha boa qualidade, sem necessariamente ser ótima. Você deverá implementar a função:

void resolvedor(char* entrada_filename, char* solucao_filename);

Que recebe o nome de uma instância e o nome de um arquivo onde você deverá escrever a sua resposta. A instância é dada da seguinte forma, dois inteiros representando o número de vértices n e o número de arcos m. Depois uma linha com n valores entre 0 e 5 representando as cores iniciais de cada vértice (as cores obviamente terão repetições, no exemplo cada vértice ficou com uma cor diferente). E depois m linhas, cada uma com dois valores representando a origem e o destino de cada arco. Abaixo está a instância que representa o grafo da figura acima. O vértice 0 (zero) sempre será o fonte.

```
6 5
0 1 2 3 4 5
0 1
1 2
2 4
2 3
4 5
```

Seu programa deverá escrever no arquivo *solucao_filename* uma linha contendo o número de movimentos, e outra linha contendo a lista com os movimentos. No caso da lista apresentada acima a saída seria:

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

Note que como é uma heurística, não necessariamente você precisa encontrar a solução ótima. Uma outra solução para esse problema seria:

```
23
1 4 3 1 5 1 4 0 3 1 2 1 2 1 5 4 0 0 4 4 5 2 3
```

Essa lista tem 23 movimentos, e apesar de também ser viável é significativamente pior que a de 5 movimentos.

- Você poderá, se quiser, usar parte do código que é fornecido com o trabalho.
- Você poderá, se quiser, buscar literatura e códigos na internet. Entretanto, caso deseje se basear nessa fonte, você deverá compreendê-la completamente e realizar a sua própria implementação, além de explicá-la com detalhes em apresentação.
- Você pode usar bibliotecas conhecidas e confiáveis elaboradas, com a condição de que você explique com antecedência como é feita a instalação no Ubuntu 20.04.2 LTS.
- Se você não tiver certeza se alguma coisa é permitida ou não no trabalho, não hesite em perguntar ao professor!