



Coloração

Gabriel de Paula Meira - 222050003

A coloração de grafos é um problema que envolve atribuir cores a vértices de um grafo de modo que vértices adjacentes não tenham a mesma cor, usando o menor número possível de cores.

No jogo de xadrez, a rainha é uma peça cuja movimentação se dá livremente nas linhas colunas e diagonais do tabuleiro. Nesse problema, todas as rainhas deve estar associadas a uma classe de cor. O objetivo é criar uma solução em que uma rainha só possa estar no caminho de outra se suas cores forem distintas. Duas rainhas da mesma cor não podem estar no caminho uma da outra.



Algoritmo

O problema não possui um algoritmo determinístico conhecido que resolve o problema em tempo polinomial, portanto, uma estratégia é utilizar de heurísticas capazes de encontrar uma solução aproximada.

A heurística utilizada é a de coloração por classe, na qual é criada uma classe nova toda vez que não é possível adicionar mais vértices à todas as classes existentes.



Rodando o programa

[> Ver main.cpp](#) < [> Ver graph.hpp](#) < [> Ver gml.hpp](#) < [> Ver coloring.hpp](#) <

Para compilar o programa basta possuir o compilador G++ para o código-fonte que está na linguagem C++.

Utilize o comando:

```
g++ main.cpp -o main
```

Em seguida execute com um arquivo de teste, informando o arquivo `.gml` com as incompatibilidades do tabuleiro:

```
./main tests/board.gml
```

Testes de Execução

[> Ver pasta ./tests <](#)

A entrada do programa consiste na leitura de um arquivo `.gm1` com estrutura similar à exibida abaixo.

O `id` define o número da casa no tabuleiro da esquerda para a direita.

A `label` não tem impacto no funcionamento do algoritmo.

```
graph [
  node [
    id 0
    label "-"
  ]
  node [
    id 1
    label "-"
  ]
  edge [
    source 0
    target 1
  ]
]
```

Saída

As κ primeiras linhas contém o identificador das κ cores usadas na solução e uma sequência com as posições das rainhas que satisfazem a condição do problema. O restante das linhas representa o tabuleiro completo com cada casa contendo o número identificador da cor.

```
0 { 0 10 20 25 35 }
1 { 1 11 16 26 36 }
2 { 2 8 19 30 41 52 63 }
3 { 3 9 22 28 32 47 53 58 }
4 { 4 14 17 27 37 40 50 }
5 { 5 15 18 24 38 44 49 }
6 { 6 12 23 29 42 59 }
7 { 7 13 33 43 54 }
8 { 21 31 34 46 51 57 }
9 { 39 45 48 62 }
10 { 55 56 }
11 { 60 }
12 { 61 }
```

```
[ 0][ 1][ 2][ 3][ 4][ 5][ 6][ 7]
[ 2][ 3][ 0][ 1][ 6][ 7][ 4][ 5]
[ 1][ 4][ 5][ 2][ 0][ 8][ 3][ 6]
[ 5][ 0][ 1][ 4][ 3][ 6][ 2][ 8]
[ 3][ 7][ 8][ 0][ 1][ 4][ 5][ 9]
[ 4][ 2][ 6][ 7][ 5][ 9][ 8][ 3]
[ 9][ 5][ 4][ 8][ 2][ 3][ 7][10]
[10][ 8][ 3][ 6][11][12][ 9][ 2]
```

O exemplo acima é gerado a partir do arquivo de teste > [tests/board.gml](#) <