Coloração de Grafos

Prof. Andrei Braga

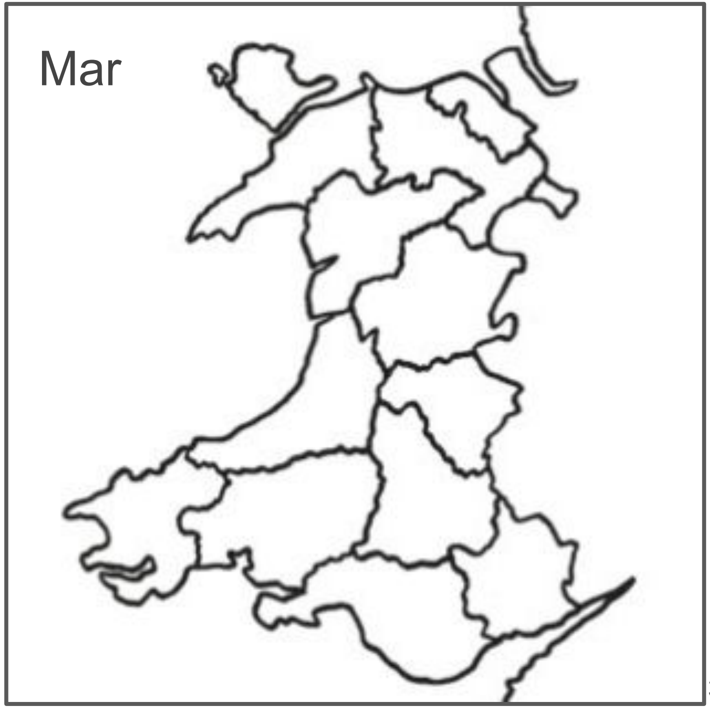
UNIVERSIDADE FEDERAL DA FRONTEIRA SUL

Conteúdo

- Motivação
- Coloração de grafos
- Exercícios
- Referências

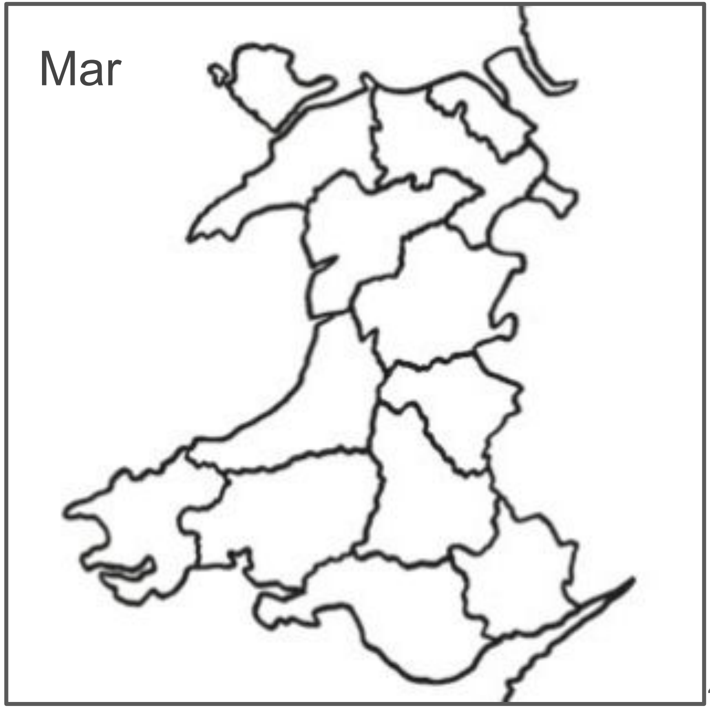
Colorindo um mapa

- Considere que a nossa tarefa é colorir o mapa ao lado
- Um requisito que faz muito sentido para esta tarefa é
  - colorir regiões vizinhas com cores diferentes (para facilitar a visualização das regiões)
- Como podemos fazer isso?



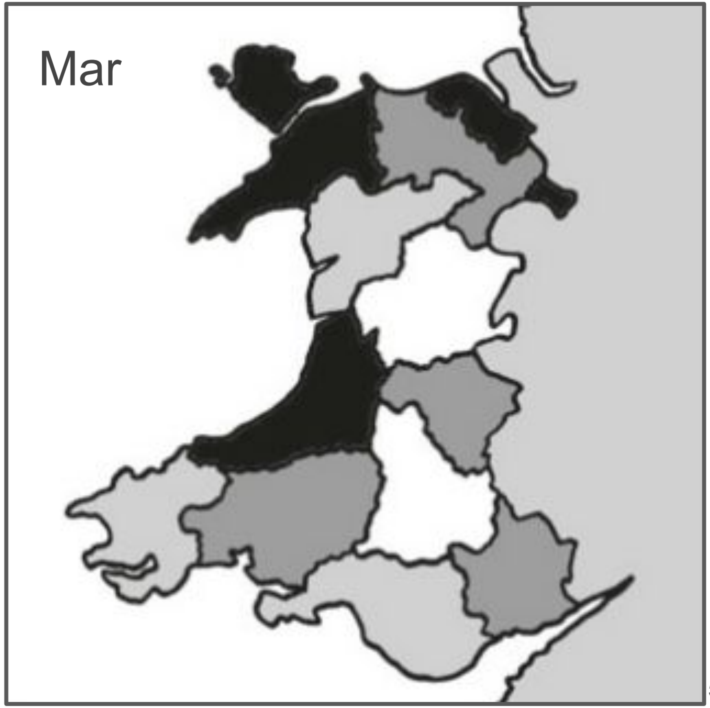
Colorindo um mapa

- Considere que a nossa tarefa é colorir o mapa ao lado
- Um requisito que faz muito sentido para esta tarefa é
  - colorir regiões vizinhas com cores diferentes (para facilitar a visualização das regiões)
- Como podemos fazer isso?
- Outro requisito que também faz sentido para esta tarefa é
  - usar o menor número de cores possível (para minimizar a distração visual do usuário)
- Como podemos fazer isso?



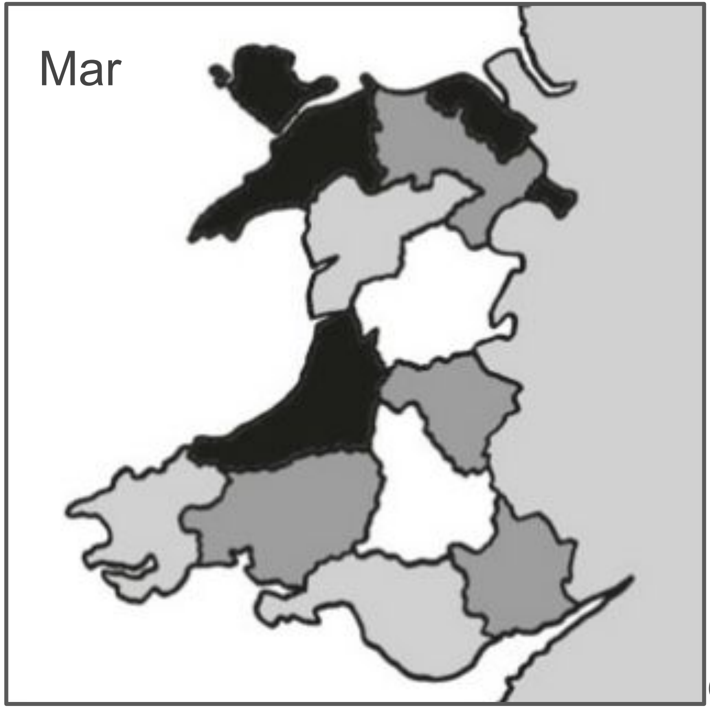
Colorindo um mapa

- Considere que a nossa tarefa é colorir o mapa ao lado
- Um requisito que faz muito sentido para esta tarefa é
  - colorir regiões vizinhas com cores diferentes (para facilitar a visualização das regiões)
- Como podemos fazer isso?
- Outro requisito que também faz sentido para esta tarefa é
  - usar o menor número de cores possível (para minimizar a distração visual do usuário)
- Como podemos fazer isso? Com 4 cores!



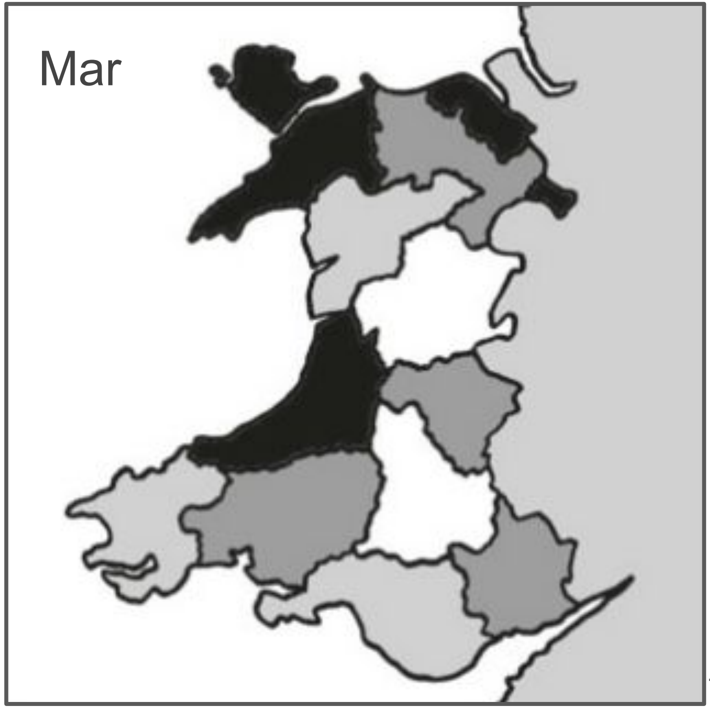
Colorindo um mapa

- Acredita-se que uma tarefa deste tipo foi considerada pela primeira vez por Francis Guthrie, no século 19, enquanto ele coloria um mapa de regiões da Inglaterra
- Ele conseguiu colorir o mapa usando 4 cores
- Além disso, ele considerou a seguinte questão:
  É possível colorir qualquer mapa com no máximo 4 cores (com regiões vizinhas sendo coloridas com cores diferentes)?



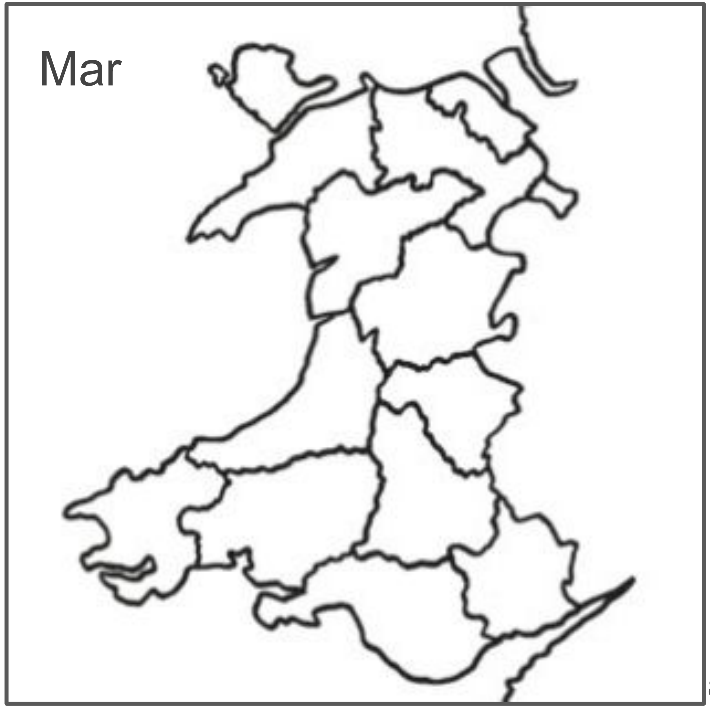
Colorindo um mapa

- Acredita-se que uma tarefa deste tipo foi considerada pela primeira vez por Francis Guthrie, no século 19, enquanto ele coloria um mapa de regiões da Inglaterra
- Ele conseguiu colorir o mapa usando 4 cores
- Além disso, ele considerou a seguinte questão:
  É possível colorir qualquer mapa com no máximo 4 cores (com regiões vizinhas sendo coloridas com cores diferentes)?
  Sim! Teorema das Quatro Cores



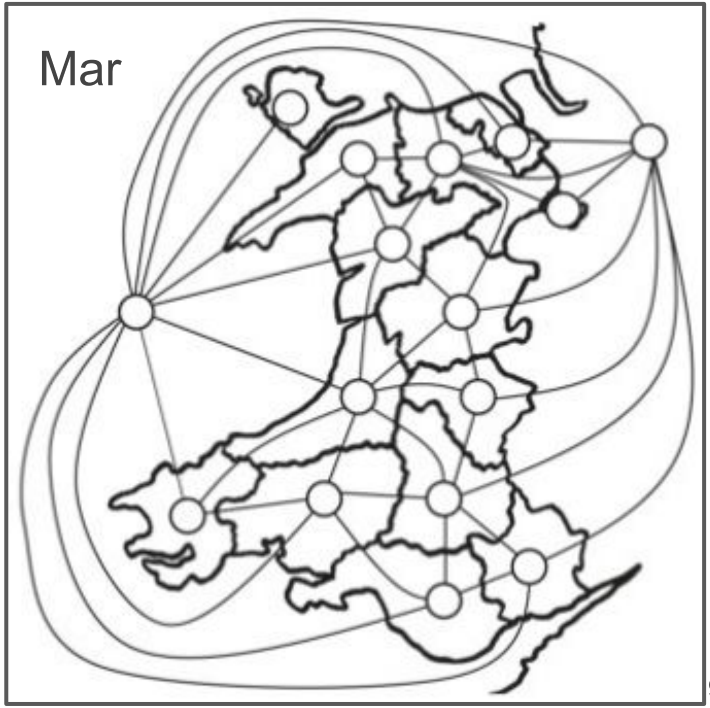
Colorindo um mapa

- Como podemos usar um grafo na realização da tarefa citada?
- Podemos representar o mapa como um grafo tal que
  - os vértices do grafo correspondem às regiões do mapa e
  - existe uma aresta entre os vértices v_i e v_j se e somente se as regiões *i* e *j* são vizinhas



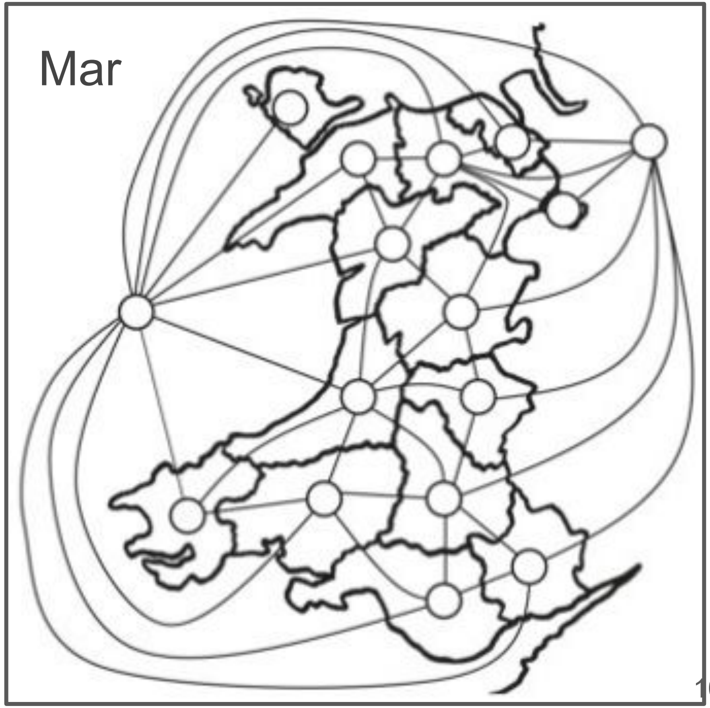
Colorindo um mapa

- Como podemos usar um grafo na realização da tarefa citada?
- Podemos representar o mapa como um grafo tal que
  - os vértices do grafo correspondem às regiões do mapa e
  - existe uma aresta entre os vértices v_i e v_j se e somente se as regiões *i* e *j* são vizinhas



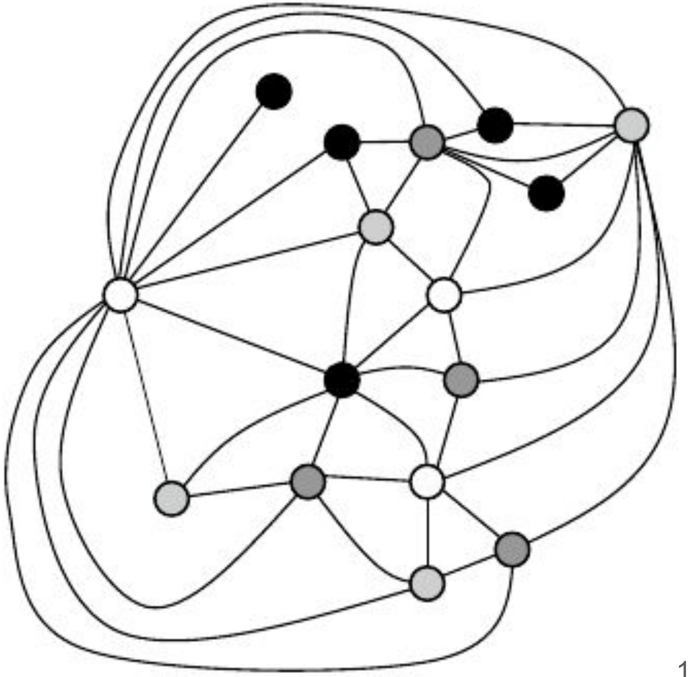
Colorindo um mapa

- Como podemos usar um grafo na realização da tarefa citada?
- Vamos colorir os vértices deste grafo de modo que
  - vértices vizinhos tenham cores diferentes e
  - o número de cores utilizado seja o menor possível



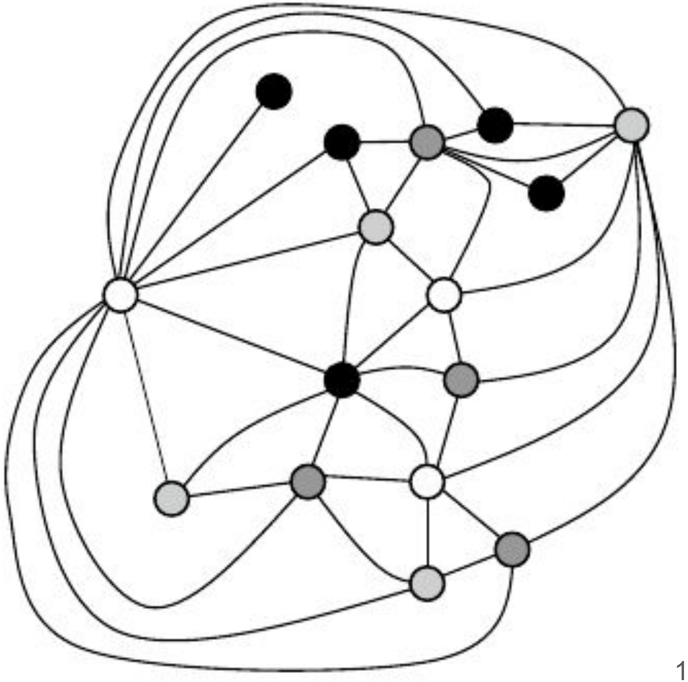
Colorindo um mapa

- Como podemos usar um grafo na realização da tarefa citada?
- Vamos colorir os vértices deste grafo de modo que
  - vértices vizinhos tenham cores diferentes e
  - o número de cores utilizado seja o menor possível



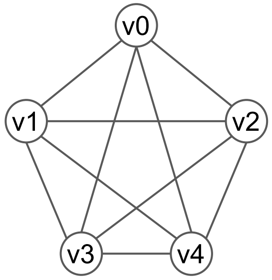
Colorindo um mapa

- Comentamos que é possível colorir qualquer mapa com no máximo 4 cores (com regiões vizinhas tendo cores diferentes)
- Então, é possível colorir os vértices de qualquer grafo com no máximo 4 cores (com vértices vizinhos tendo cores diferentes)? **Não!**
- Isto é verdade **apenas** para grafos **planares**, que são grafos que podem ser desenhados em um plano sem cruzamento de arestas (grafos que representam mapas são deste tipo)



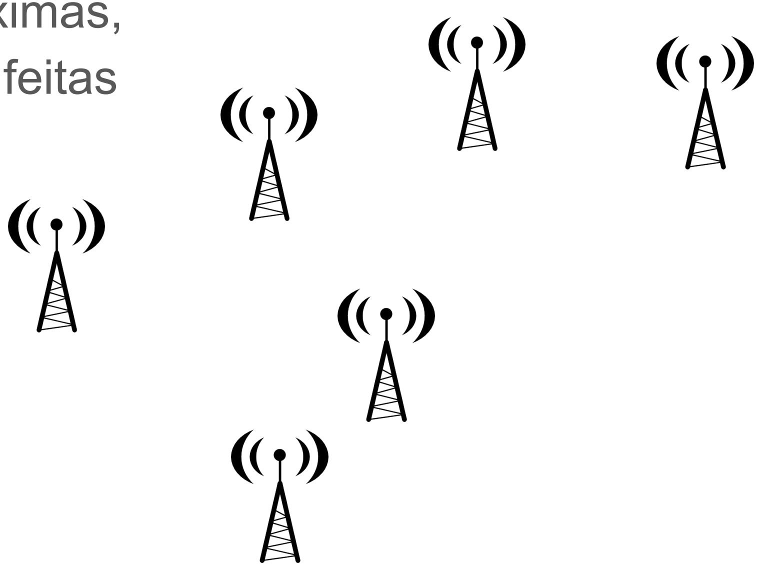
Colorindo um mapa

- Comentamos que é possível colorir qualquer mapa com no máximo 4 cores (com regiões vizinhas tendo cores diferentes)
- Então, é possível colorir os vértices de qualquer grafo com no máximo 4 cores (com vértices vizinhos tendo cores diferentes)?
  Não!
- Isto não é verdade para o grafo ao lado



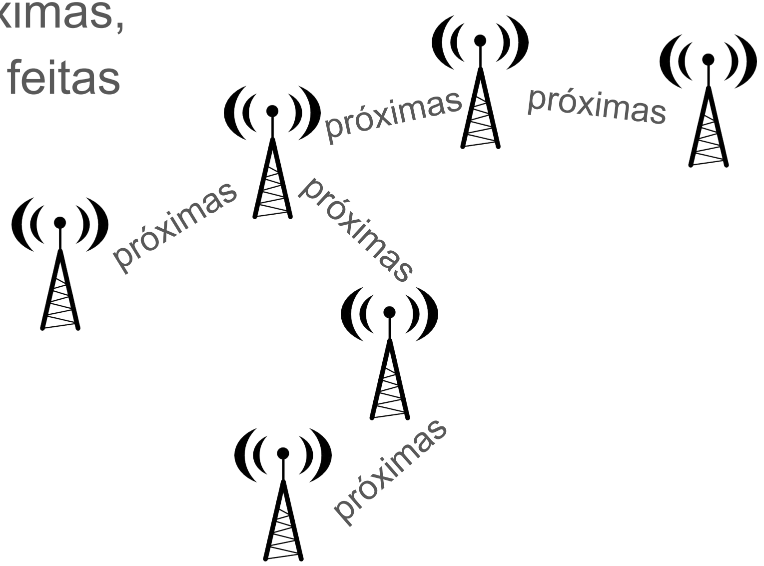
Atribuição de frequências para transmissão sem fio

- Considere agora o seguinte problema
- Temos um conjunto de antenas de onde são feitas transmissões sem fio
- Se duas antenas estão suficientemente próximas, então as suas transmissões não podem ser feitas usando a mesma frequência; caso contrário, haverá uma interferência prejudicial às transmissões



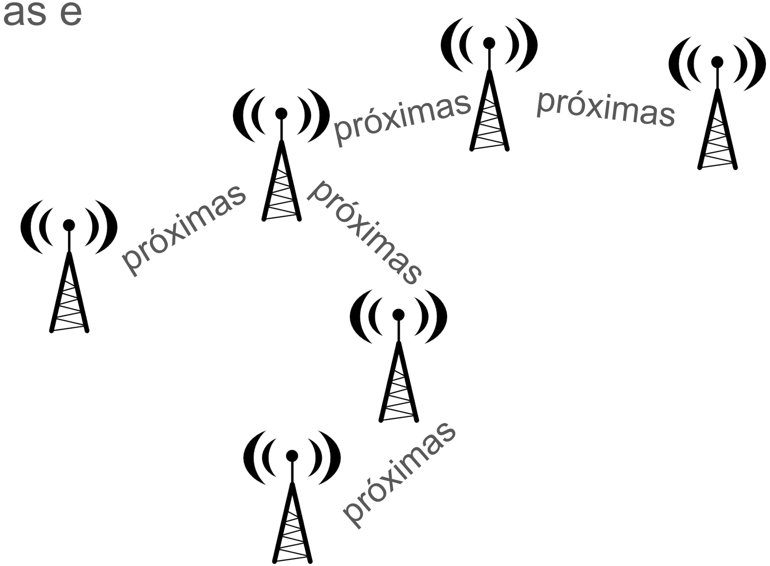
Atribuição de frequências para transmissão sem fio

- Considere agora o seguinte problema
- Temos um conjunto de antenas de onde são feitas transmissões sem fio
- Se duas antenas estão suficientemente próximas, então as suas transmissões não podem ser feitas usando a mesma frequência; caso contrário, haverá uma interferência prejudicial às transmissões
- Para diminuir custos, desejamos utilizar o menor número possível de frequências



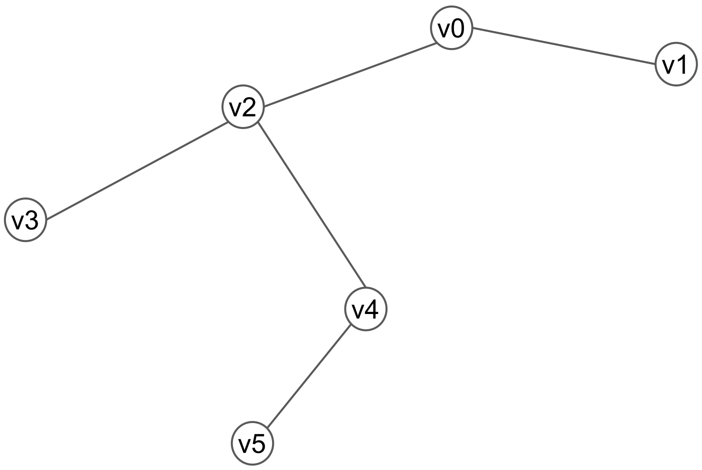
Atribuição de frequências para transmissão sem fio

- Como podemos usar um grafo na resolução deste problema?
- Podemos construir um grafo tal que
  - os vértices do grafo correspondem às antenas e
  - existe uma aresta entre os vértices v_i e v_j se e somente se as antenas *i* e *j* estão suficientemente próximas



Atribuição de frequências para transmissão sem fio

- Como podemos usar um grafo na resolução deste problema?
- Podemos construir um grafo tal que
  - os vértices do grafo correspondem às antenas e
  - existe uma aresta entre os vértices v_i e v_j se e somente se as antenas *i* e *j* estão suficientemente próximas



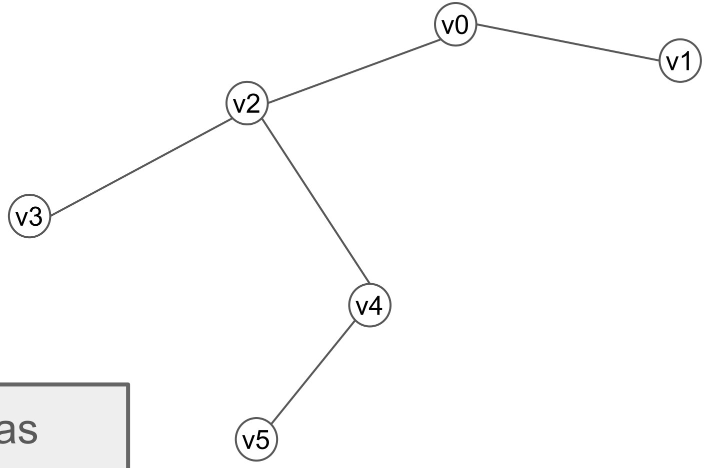
Atribuição de frequências para transmissão sem fio

- Como podemos usar um grafo na resolução deste problema?
- Vamos colorir os vértices deste grafo de modo que
  - vértices vizinhos tenham cores diferentes e
  - o número de cores utilizado seja o menor possível

antenas suficientemente próximas

transmitam usando frequências diferentes

o número de frequências utilizadas seja o menor possível



Sudoku

- Outro problema que podemos considerar é o seguinte
- O Sudoku é um jogo onde temos um quebra-cabeça parcialmente preenchido como o mostrado ao lado
- Devemos completar este quebra-cabeça de modo que
    - cada linha contenha exatamente um dos números 1, 2, …, 9;
    - cada coluna contenha exatamente um dos números 1, 2, …, 9 e
    - cada *quadrado 3x3 destacado* contenha exatamente um dos números 1, 2, …, 9

	2	4			7			
6								
		3	6	8		4	1	5
4	3	1			5			
5							3	2
7	9						6	
2		9	7	1		8		
	4			9	3			
3	1				4	7	5	

Sudoku

- Outro problema que podemos considerar é o seguinte
- O Sudoku é um jogo onde temos um quebra-cabeça parcialmente preenchido como o mostrado ao lado
- Devemos completar este quebra-cabeça de modo que
  - cada linha contenha exatamente um dos números 1, 2, …, 9;
  - cada coluna contenha exatamente um dos números 1, 2, …, 9 e
  - cada *quadrado 3x3 destacado* contenha exatamente um dos números 1, 2, …, 9

1	2	4	9	5	7	3	8	6
6	8	5	3	4	1	2	9	7
7	9	3	6	8	2	4	1	5
4	3	1	2	6	5	9	7	8
5	6	8	4	7	9	1	3	2
7	9	2	1	3	8	5	6	4
2	5	9	7	1	6	8	4	3
8	4	7	5	9	3	6	2	1
3	1	6	8	2	4	7	5	9

Sudoku

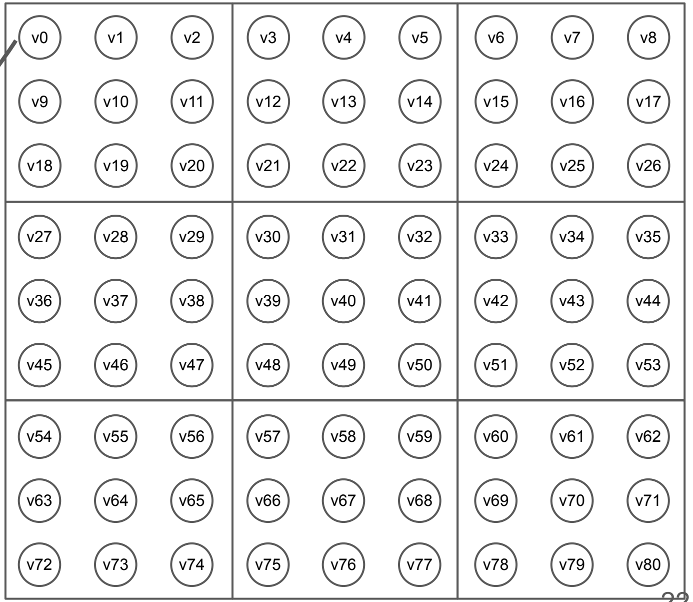
- Como podemos usar um grafo na resolução deste problema?
- Podemos construir um grafo tal que
  - os vértices do grafo correspondem aos quadrados do quebra-cabeça e
  - existe uma aresta entre os vértices v_i e v_j se e somente se os quadrados *i* e *j* estão na mesma linha, na mesma coluna ou no mesmo quadrado 3x3

	2	4			7			
6								
		3	6	8		4	1	5
4	3	1			5			
5							3	2
7	9						6	
2		9	7	1		8		
	4			9	3			
3	1				4	7	5	

Sudoku

- Como podemos usar um grafo na resolução deste problema?
- Podemos construir um grafo tal que
  - os vértices do grafo correspondem aos quadrados do quebra-cabeça e
  - existe uma aresta entre os vértices v_i e v_j se e somente se os quadrados *i* e *j* estão na mesma linha, na mesma coluna ou no mesmo quadrado 3x3

v_0 é vizinho de v_1, v_2, … v_8,
de v_9, v_{18}, v_{27}, v_{36}, v_{45}, v_{54}, v_{63}, v_{72},
e de v_{10}, v_{11}, v_{19}, v_{20}



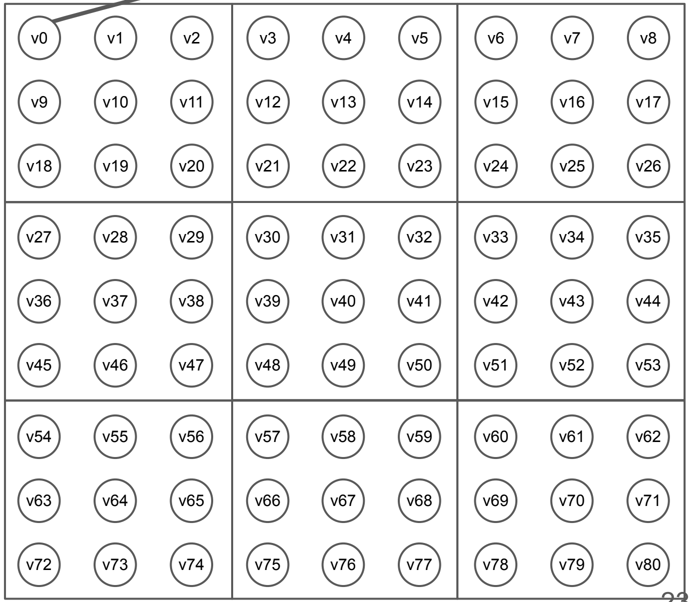
Sudoku

v_0 é vizinho de v_1, v_2, … v_8,
de v_9, v_{18}, v_{27}, v_{36}, v_{45}, v_{54}, v_{63}, v_{72},
e de v_{10}, v_{11}, v_{19}, v_{20}

- Como podemos usar um grafo na resolução deste problema?
- Vamos colorir os vértices deste grafo de modo que
  - vértices vizinhos tenham cores diferentes e
  - o número de cores utilizado seja 9

quadrados que estejam na mesma linha, na mesma coluna ou no mesmo quadrado 3x3

contenham números diferentes



Outros problemas modeláveis por coloração de vértices

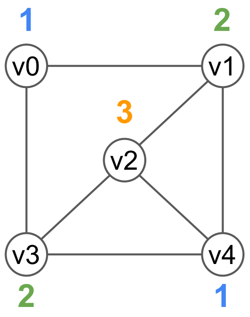
- Alocação de variáveis a registradores
- Agendamento de eventos
- Alocação de motoristas a solicitações de corrida

k-coloração

- Dado um grafo G, uma ***k*-coloração** de G é uma função $c: V(G) \rightarrow \{1, 2, \dots, k\}$; ou seja, é a atribuição de um valor do conjunto $\{1, 2, \dots, k\}$ a cada vértice de G
- Dizemos que $c(v)$ é a **cor** de v

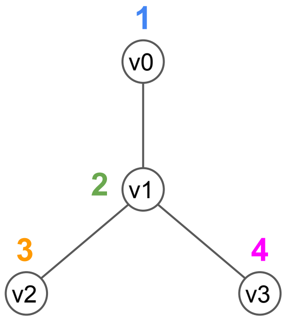
- Exemplo:

G



3-coloração

G

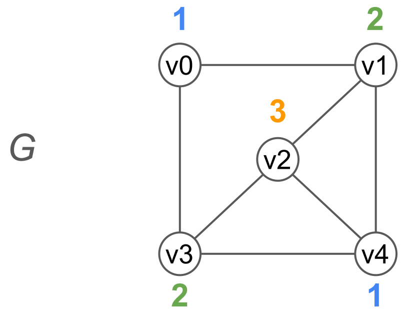


4-coloração

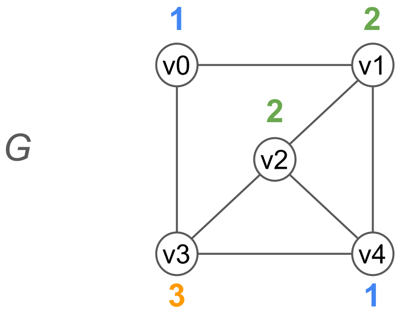
k-coloração própria

- Dado um grafo G, uma k-coloração c de G é **própria** se, para todo par de vértices vizinhos v_i, v_j de G, $c(v_i) \neq c(v_j)$; ou seja, as cores a atribuídas a v_i e v_j são diferentes

- Exemplo:



3-coloração própria



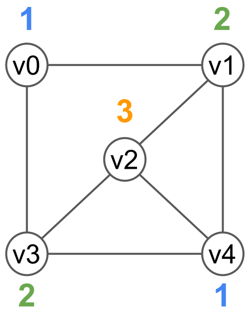
~~3-coloração própria~~

Número cromático

- O **número cromático** de um grafo *G* é o menor inteiro positivo *k* tal que *G* possui uma *k*-coloração própria
- Denotamos por χ(*G*) o número cromático de *G*

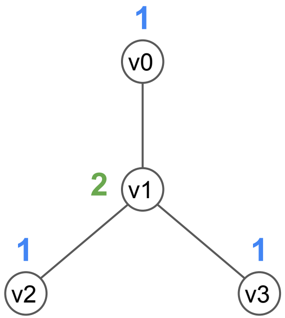
- Exemplo:

G



$\boxed{\chi(G) = 3}$

G



$\boxed{\chi(G) = 2}$

Problema da coloração mínima

- **Problema:** Dado um grafo *G*, encontre uma *k*-coloração própria de *G* tal que *k* seja mínimo
- Ao resolver este problema, determinamos $\chi(G)$

Colorindo com 2 cores

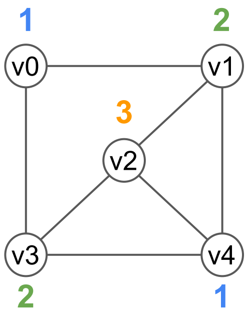
- Se um grafo G tem pelo menos uma aresta, então $\chi(G) \geq 2$
- Para quais grafos o número cromático é exatamente 2?

k-coloração própria e conjuntos independentes

- Note que, em uma *k*-coloração própria de um grafo, vértices que têm a mesma cor não são vizinhos
- Por consequência, uma *k*-coloração própria de um grafo *G* **particiona** $V(G)$ em conjuntos de vértices $V_1, V_2, \ldots, V_k$ tal que $V_1, V_2, \ldots, V_k$ são **conjuntos independentes**

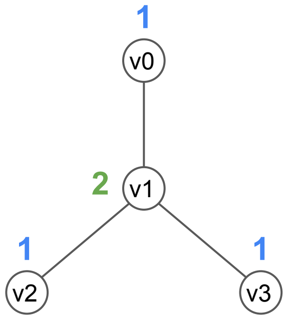
- Exemplo:

G



V_1 = { v0, v4 } é um conjunto independente
V_2 = { v1, v3 } é um conjunto independente
V_3 = { v2 } é um conjunto independente

G



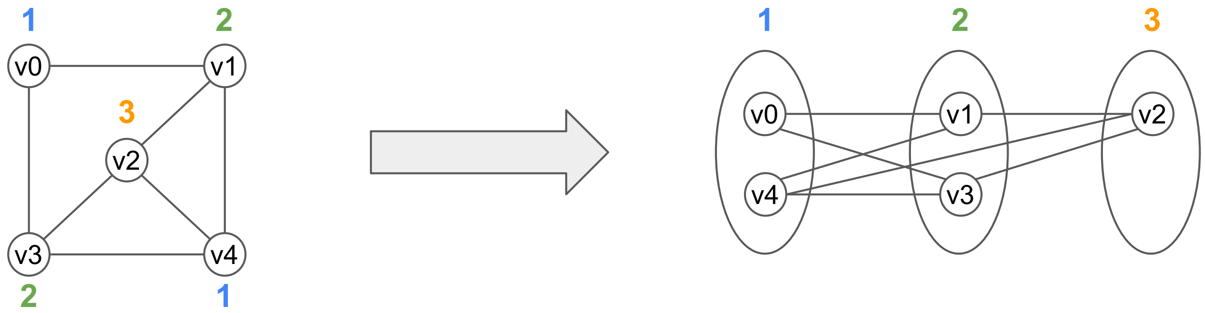
V_1 = { v0, v2, v3 } é um conjunto independente
V_2 = { v1 } é um conjunto independente

k-coloração própria e conjuntos independentes

- Note que, em uma *k*-coloração própria de um grafo, vértices que têm a mesma cor não são vizinhos
- Por consequência, uma *k*-coloração própria de um grafo *G* **particiona** $V(G)$ em conjuntos de vértices $V_1, V_2, \ldots, V_k$ tal que $V_1, V_2, \ldots, V_k$ são **conjuntos independentes**

- Exemplo:

G



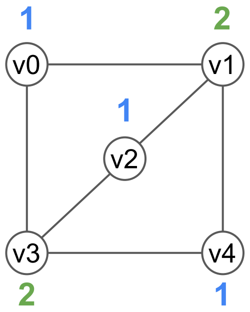
V_1 = { v0, v4 } é um conjunto independente
V_2 = { v1, v3 } é um conjunto independente
V_3 = { v2 } é um conjunto independente

Grafo bipartido

- Um grafo G é **bipartido** se $V(G)$ pode ser particionado em dois conjuntos de vértices V_1 e V_2 tal que V_1 e V_2 são conjuntos independentes

- Exemplo:

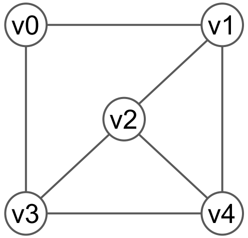
G



V_1 = { v0, v2, v4 } é um conjunto independente
V_2 = { v1, v3 } é um conjunto independente

Grafo bipartido

G



Não é possível particionar $V(G)$ em dois conjuntos independentes V_1 e V_2

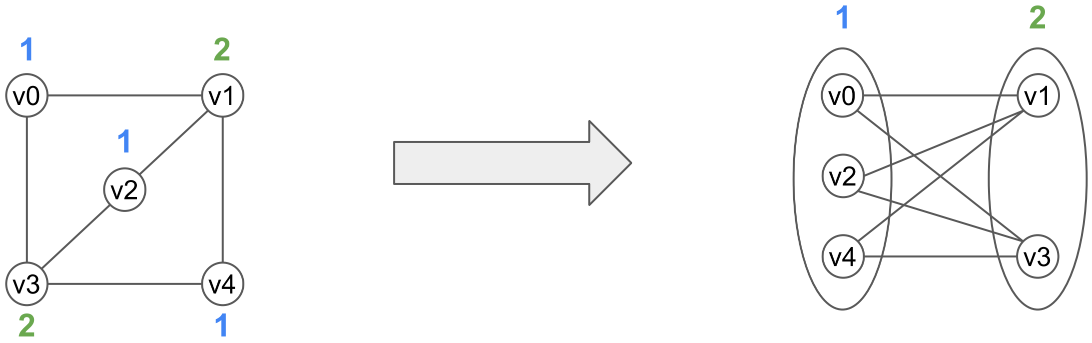
~~Grafo bipartido~~

Grafo bipartido

- Um grafo *G* é **bipartido** se *V*(*G*) pode ser particionado em dois conjuntos de vértices V_1 e V_2 tal que V_1 e V_2 são conjuntos independentes

- Exemplo:

G



V_1 = { v0, v2, v4 } é um conjunto independente
V_2 = { v1, v3 } é um conjunto independente

Grafo bipartido

Colorindo com 2 cores

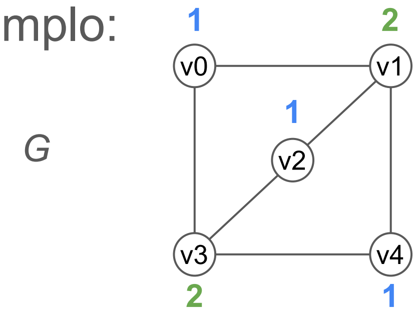
- Se um grafo G é bipartido, então G possui uma 2-coloração própria
  - $V(G)$ pode ser particionado em V_1 e V_2 tal que V_1 e V_2 são conjuntos independentes
  - Atribuímos a cor 1 para os vértices de V_1 e atribuímos a cor 2 para os vértices de V_2
- Se G possui uma 2-coloração própria, então G é bipartido
  - Vimos que uma 2-coloração própria de G particiona $V(G)$ em conjuntos de vértices V_1 e V_2 tal que V_1 e V_2 são conjuntos independentes
  - Então, G é bipartido

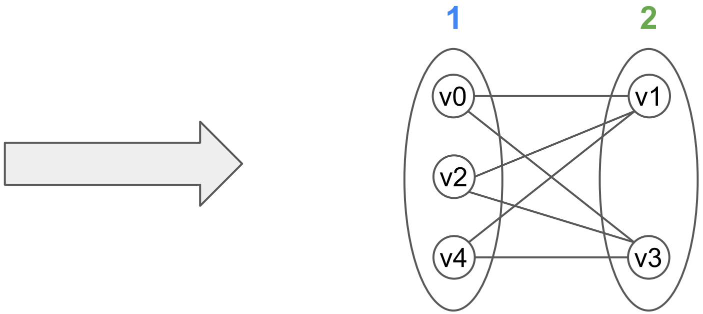
- Dado um grafo G que tem pelo menos uma aresta,
  $\chi(G) = 2$ se e somente se G é bipartido

Distâncias em um grafo bipartido

- **Teorema:** Considere um grafo bipartido G tal que $V(G)$ pode ser particionado em dois conjuntos independentes V_1 e V_2.
  Dado um vértice $v \in V_1$,
  - para todo vértice $w \in V_1$, $d(v, w)$ é par;
  - para todo vértice $w \in V_2$, $d(v, w)$ é ímpar.

- Exemplo:





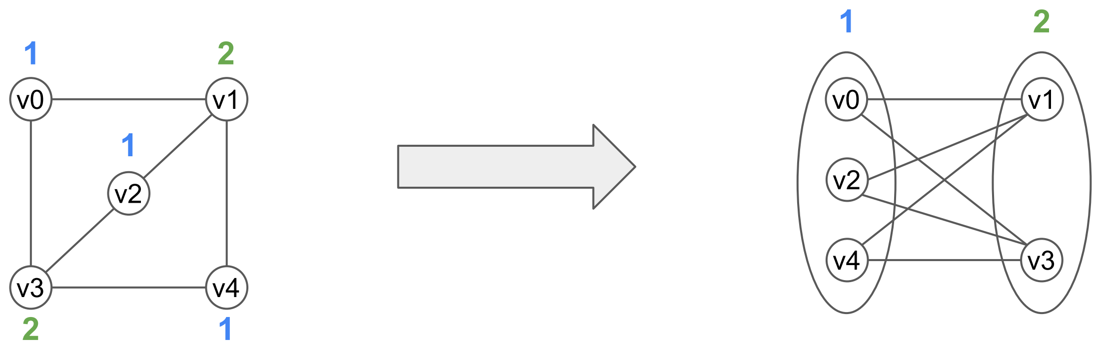
Vértices em V_1: d(v0, v0) = , d(v0, v2) = , d(v0, v4) =
Vértices em V_2: d(v0, v1) = , d(v0, v3) =

Distâncias em um grafo bipartido

- **Teorema:** Considere um grafo bipartido G tal que $V(G)$ pode ser particionado em dois conjuntos independentes V_1 e V_2.
  Dado um vértice $v \in V_1$,
  - para todo vértice $w \in V_1$, $d(v, w)$ é par;
  - para todo vértice $w \in V_2$, $d(v, w)$ é ímpar.

- Exemplo:

G



Vértices em V_1: d(v0, v0) = 0, d(v0, v2) = 2, d(v0, v4) = 2
Vértices em V_2: d(v0, v1) = 1, d(v0, v3) = 1

Testando se é possível colorir com 2 cores

Testa2Cores(G)

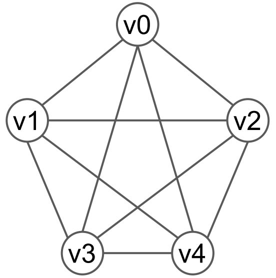
1. Enquanto existe um vértice não visitado v de G faça:
2. Execute o algoritmo de busca em largura começando por v testando o seguinte:
   Para um vértice w, se $d(v,w)$ é par e w tem um vizinho u tal que $d(v,u)$ é par ou se $d(v,w)$ é ímpar e w tem um vizinho u tal que $d(v,u)$ é ímpar, então o grafo não pode ser colorido com 2 cores

Limitantes inferiores para χ(*G*)

- **Teorema:** Seja *k* o tamanho máximo de uma clique de um grafo *G*. Então, $\chi(G) \geq k$.

- Exemplo:

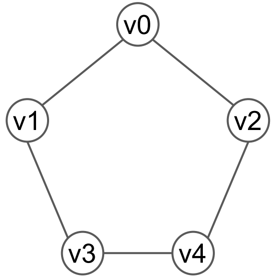
G



O tamanho máximo de uma clique de *G* é 5

$\boxed{\chi(G) = 5}$

G



O tamanho máximo de uma clique de *G* é 2

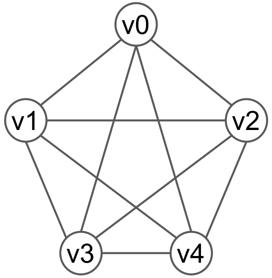
$\boxed{\chi(G) = 3}$

Limitantes inferiores para χ(*G*)

- **Teorema:** Seja *k* o tamanho máximo de um conjunto independente de um grafo *G*. Então, $\chi(G) \geq |V(G)| / k$.

- Exemplo:

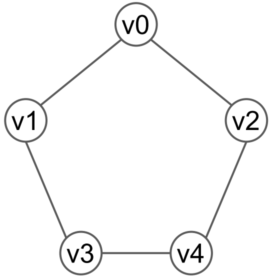
G



O tamanho máximo de um conjunto independente de *G* é 1 → $|V(G)| / 1 = 5$

$\boxed{\chi(G) = 5}$

G



O tamanho máximo de um conjunto independente de *G* é 2 → $|V(G)| / 2 = 2{,}5$

$\boxed{\chi(G) = 3}$

Construindo uma coloração

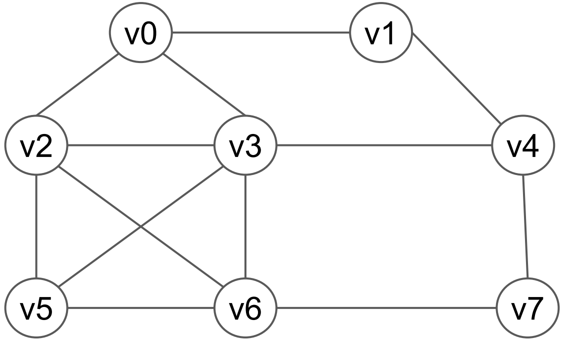
- Até hoje, não se conhece um algoritmo *eficiente* para resolver o problema da coloração mínima
- No entanto, existem algoritmos eficientes para simplesmente construir uma coloração própria de um grafo (a coloração construída não é garantidamente mínima)
- Um algoritmo simples para isso utiliza uma abordagem *gulosa*

Construindo uma coloração

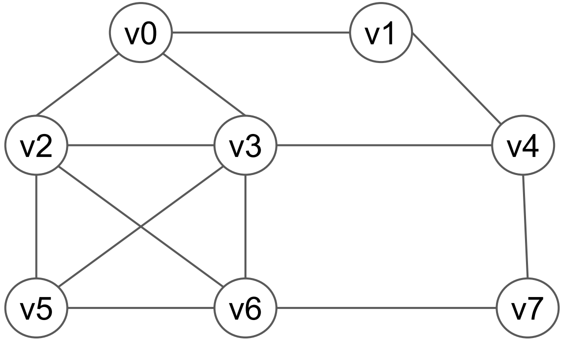
ConstroiColoracao(G)

1. Para cada vértice v de G com os vértices de G considerados em alguma ordem arbitrária:
2. Atribua a v a cor que tenha o menor índice e que ainda não tenha sido atribuída a nenhum dos seus vizinhos que já foram coloridos

Construindo uma coloração - Algoritmo guloso



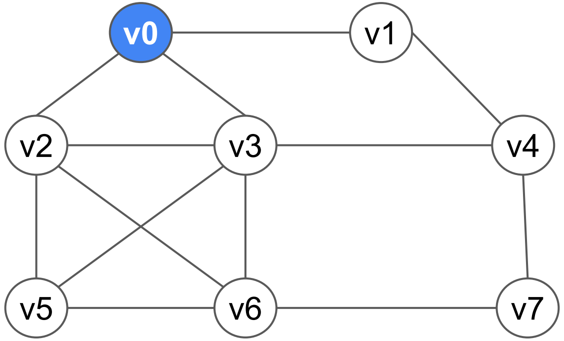
Construindo uma coloração - Algoritmo guloso



cor:

0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0

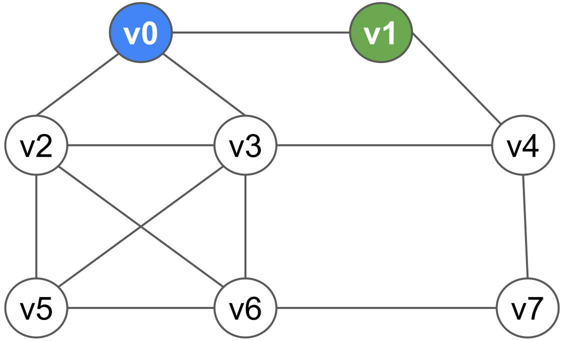
Construindo uma coloração - Algoritmo guloso



cor:

0	1	2	3	4	5	6	7
1	0	0	0	0	0	0	0

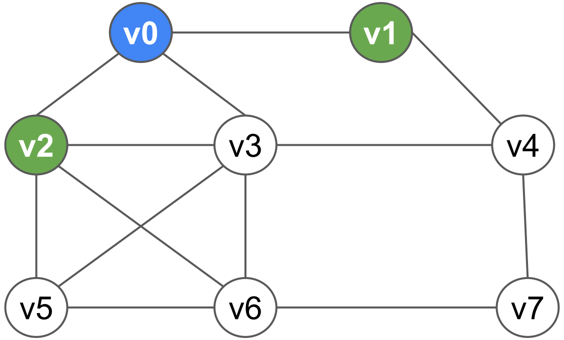
Construindo uma coloração - Algoritmo guloso



cor:

0	1	2	3	4	5	6	7
1	2	0	0	0	0	0	0

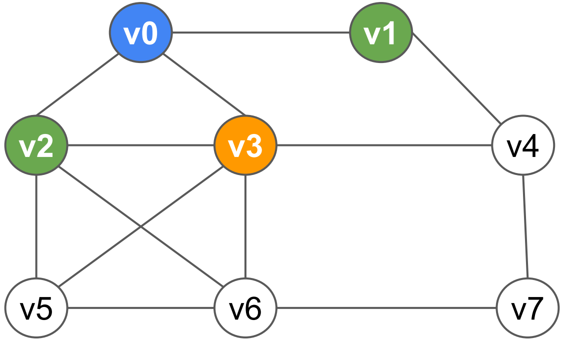
Construindo uma coloração - Algoritmo guloso



cor:

1	2	2	0	0	0	0	0
0	1	2	3	4	5	6	7

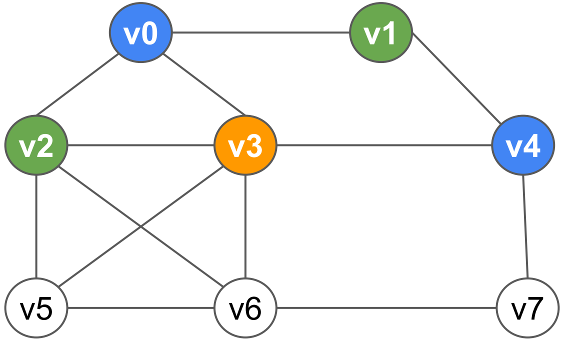
Construindo uma coloração - Algoritmo guloso



cor:

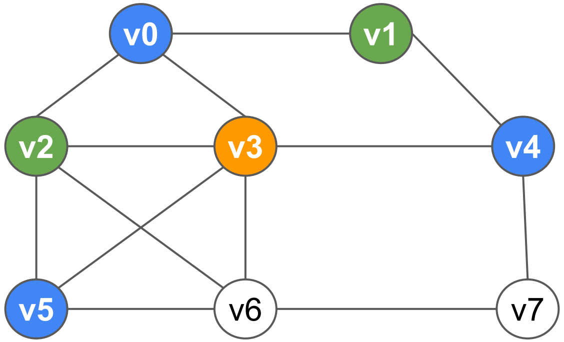
0	1	2	3	4	5	6	7
1	2	2	3	0	0	0	0

Construindo uma coloração - Algoritmo guloso



cor:	1	2	2	3	1	0	0	0
	0	1	2	3	4	5	6	7

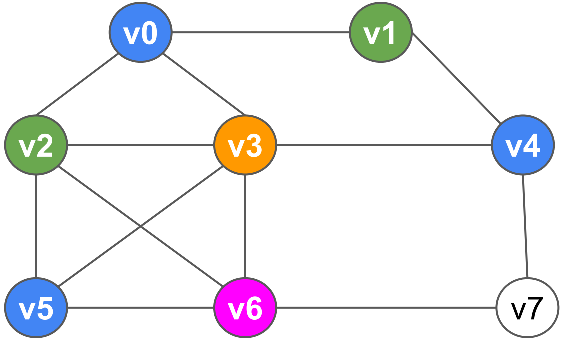
Construindo uma coloração - Algoritmo guloso



cor:

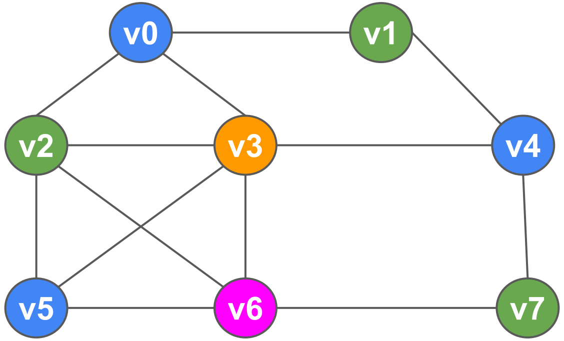
1	2	2	3	1	1	0	0
0	1	2	3	4	5	6	7

Construindo uma coloração - Algoritmo guloso



cor:	1	2	2	3	1	1	4	0
	0	1	2	3	4	5	6	7

Construindo uma coloração - Algoritmo guloso



cor:

0	1	2	3	4	5	6	7
1	2	2	3	1	1	4	2

Limitante superior para $\chi(G)$

- **Teorema:** Dado um grafo G, $\chi(G) \leq \Delta(G) + 1$. ($\Delta(G)$ é o grau máximo de G.)

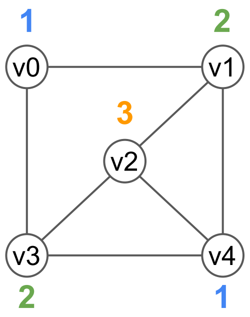
- **Prova:**
  - Vamos considerar o algoritmo guloso visto anteriormente
  - Quando o algoritmo vai atribuir uma cor a um vértice v, podem existir no máximo $\Delta(G)$ vizinhos de v já coloridos
  - Neste caso, o algoritmo vai atribuir uma cor diferente a v, totalizando $\Delta(G) + 1$ cores atribuídas
  - Então, o número máximo de cores atribuídas pelo algoritmo é $\Delta(G) + 1$
  - Logo, G possui uma $(\Delta(G) + 1)$-coloração própria e, portanto, $\chi(G) \leq \Delta(G) + 1$ □

Limitante superior para χ(*G*)

- **Teorema:** Se *G* é um grafo conexo que não é um grafo completo nem composto apenas por um ciclo ímpar, então $\chi(G) \leq \Delta(G)$. ($\Delta(G)$ é o grau máximo de *G*.)

- Exemplo:

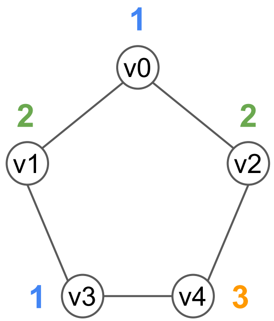
G



$\Delta(G) = 3$

$\boxed{\chi(G) = 3}$

G



$\Delta(G) + 1 = 3$

$\boxed{\chi(G) = 3}$

Algoritmo guloso melhorado

- Podemos melhorar a abordagem gulosa vista anteriormente
- Em vez de considerar os vértices em uma ordem arbitrária, podemos considerar os vértices em ordem decrescente dos seus graus (mais precisamente, em ordem não-crescente)

- A partir deste algoritmo, podemos provar um outro limitante superior para o número cromático de um grafo

Algoritmo guloso melhorado

ConstroiColoracaoMelhorado(G)

1. Para cada vértice v de G com os vértices de G considerados em **ordem não-crescente dos seus graus**:
2. Atribua a v a cor que tenha o menor índice e que ainda não tenha sido atribuída a nenhum dos seus vizinhos que já foram coloridos

Algoritmos exatos

- Como comentado anteriormente, não se conhece um algoritmo *eficiente* para resolver o problema da coloração mínima
- Um algoritmo *muito ineficiente* consiste em considerar todas as possibilidades de atribuição de cores aos vértices do grafo e verificar qual delas é uma *k*-coloração própria tal que *k* é o menor possível

Algoritmo exato 1 (muito ineficiente)

AlgoritmoExato1(G)

1. Para $k = 1, \ldots, |V(G)|$:
2. Para cada possível atribuição c de k cores aos vértices de G:
3. Se c é uma k-coloração própria de G:
4. Retorne c

Algoritmos exatos

- Outro algoritmo *muito ineficiente* para resolver o problema da coloração mínima pode ser definido com base no algoritmo ConstroiColoracao (veja este slide)
- A ideia consiste em executar o algoritmo ConstroiColoracao para todas as possíveis ordens dos vértices do grafo
- Note que cada uma destas ordens corresponde a uma permutação dos vértices do grafo

Algoritmo exato 2 (muito ineficiente)

AlgoritmoExato2(G)

1. Para cada possível ordem *ord* dos vértices de G:
2.   Execute ConstroiColoracao(G) (veja este slide) com os vértices de G sendo considerados de acordo com *ord*
3. Retorne c tal que c é uma k-coloração própria de G construída nos Passos 1 e 2 com k sendo o menor possível

Referências

- Esta apresentação é baseada nos seguintes materiais:
    1. Livro

Lewis, R. Guide to Graph Colouring. 2nd. ed. Springer, 2021.