AULA 22 – Coloração em grafos

Prof. Daniel Kikuti

Universidade Estadual de Maringá

19 de agosto de 2015

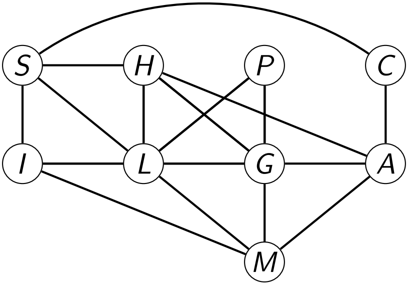
Uma aplicação inicial

Problema de escalonamento de horários

- Você é o responsável por agendar horários de aulas na universidade.
- Seu objetivo é evitar conflitos, isto é, garantir que duas aulas quaisquer com alunos em comum ocorram em horários diferentes.
- Para representar esta informação, você resolveu usar um grafo, onde os vértices representam as disciplinas e uma aresta entre duas disciplinas representa um conflito.

Exemplo[1]

Quantos horários distintos são necessários?



Legenda:

- A Astronomy
- C Chemistry
- G Greek
- H History
- I Italian
- L Latin
- M Music
- P Philosophy
- S Spanish

[1] Copiado de http://web.math.princeton.edu/math_alive/5/Notes2.pdf

Solução?

Coloração

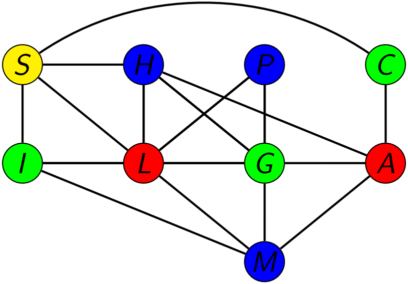
Podemos atribuir uma cor (rótulo) para cada horário (por exemplo, o horário 19:30 – 21:10 pode receber a cor azul), de forma que dois vértices adjacentes não possuam a mesma cor.

Solução?

Coloração

Podemos atribuir uma cor (rótulo) para cada horário (por exemplo, o horário 19:30 – 21:10 pode receber a cor azul), de forma que dois vértices adjacentes não possuam a mesma cor.

Uma coloração possível



Definições

Conjunto independente

Um **conjunto independente** em um grafo $G = (V, E)$ é qualquer subconjunto $V' \subseteq V$, tal que $u, v \in V' \implies (u, v) \notin E$.

Definições

Conjunto independente

Um **conjunto independente** em um grafo $G = (V, E)$ é qualquer subconjunto $V' \subseteq V$, tal que $u, v \in V' \implies (u, v) \notin E$.

Coloração, k-coloração, k-colorível

- Uma **coloração** dos vértices de $G = (V, E)$ é uma função $c : V \to \mathbb{N}$ tal que, dado dois vértices adjacentes $u, v \in V$ quaisquer, associa-os a cores diferentes, isto é, $(u, v) \in E \implies c(u) \neq c(v)$.
- Uma **k-coloração** de um grafo é uma coloração que usa um total de k cores.
- Um grafo que possui uma k-coloração é dito **k-colorível**.

Definições

Conjunto independente

Um **conjunto independente** em um grafo $G = (V, E)$ é qualquer subconjunto $V' \subseteq V$, tal que $u, v \in V' \implies (u, v) \notin E$.

Coloração, k-coloração, k-colorível

- Uma **coloração** dos vértices de $G = (V, E)$ é uma função $c : V \to \mathbb{N}$ tal que, dado dois vértices adjacentes $u, v \in V$ quaisquer, associa-os a cores diferentes, isto é, $(u, v) \in E \implies c(u) \neq c(v)$.
- Uma **k-coloração** de um grafo é uma coloração que usa um total de k cores.
- Um grafo que possui uma k-coloração é dito **k-colorível**.

Partição em conjuntos independentes

A função de coloração c induz uma partição no grafo G em subconjuntos independentes $V_1, V_2, \ldots, V_k$, na qual $V_i \cap V_j = \emptyset$ e $V_1 \cup V_2 \cup \ldots \cup V_k = V$.

Definições

Número cromático

O **número cromático** de um grafo G (representado por $\chi(G)$) é o número mínimo de cores necessário para se colorir o grafo.

Complexidade do problema

Encontrar uma coloração de vértices ótima é um problema NP-difícil[2] (caso geral).

[2] Karp, Richard M. *Reducibility Among Combinatorial Problems.* In R. E. Miller and J. W. Thatcher (editors). Complexity of Computer Computations. New York: Plenum. pp. 85–103, 1972.

Limites do número cromático

1. $1 \leq \chi(G) \leq |V|$.
2. Para um grafo completo K_n, $\chi(K_n) = n$.
3. Se G contém um **clique** de tamanho k, então $\chi(G) \geq k$.
4. Grafos bipartidos (incluindo florestas e árvores) são 2-coloríveis.
5. Todo grafo planar pode ser colorido com 4 cores (Appel e Haken, 1976).
6. Uma coloração gulosa mostra que todo grafo pode ser colorido com uma cor a mais que o grau máximo de um vértice, $\chi(G) \leq \Delta(G) + 1$.

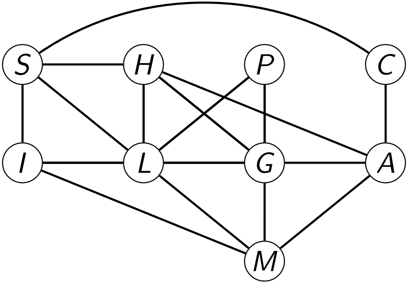
Algoritmo sequencial

Entrada: Um grafo G e uma lista de vértices (ordem) $v_1, v_2, \ldots, v_n$.
Saída: Uma coloração de vértices $c : V_G \to \mathbb{N}$.

```
Para i = 1 até n faça
    Seja c(v_i) = o menor número de cor não usado nos
                  vizinhos de menor índice de v_i
Devolva a coloração de vértices c.
```

Exemplo

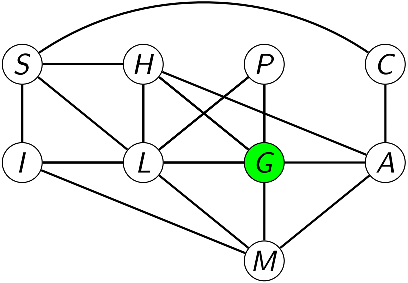
Execução do algoritmo sequencial considerando a ordem $G, L, H, P, M, A, I, S, C$.



Cores:

Exemplo

Execução do algoritmo sequencial considerando a ordem $G, L, H, P, M, A, I, S, C$.

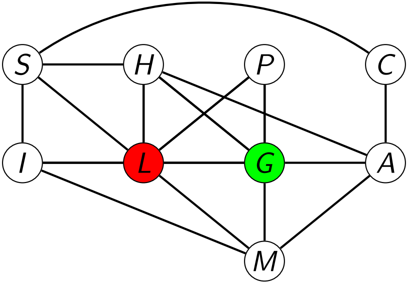


Cores:

cor 1

Exemplo

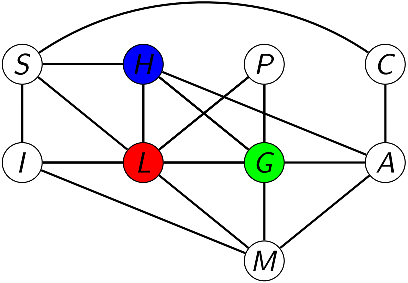
Execução do algoritmo sequencial considerando a ordem $G, L, H, P, M, A, I, S, C$.



Cores:
cor 1
cor 2

Exemplo

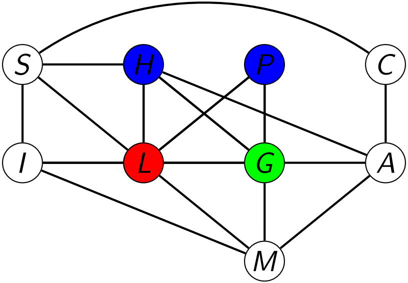
Execução do algoritmo sequencial considerando a ordem $G, L, H, P, M, A, I, S, C$.



Cores:
cor 1
cor 2
cor 3

Exemplo

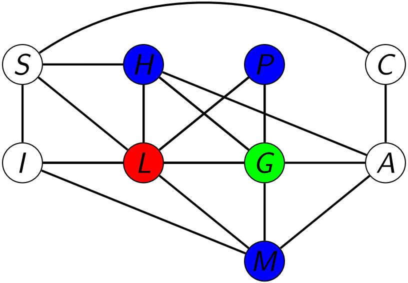
Execução do algoritmo sequencial considerando a ordem $G, L, H, P, M, A, I, S, C$.



Cores:
cor 1
cor 2
cor 3

Exemplo

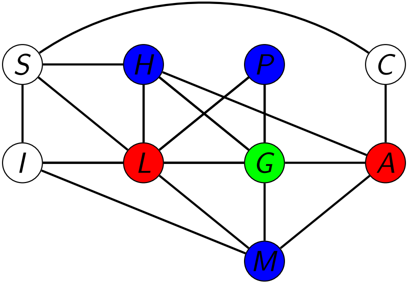
Execução do algoritmo sequencial considerando a ordem $G, L, H, P, M, A, I, S, C$.



Cores:
cor 1
cor 2
cor 3

Exemplo

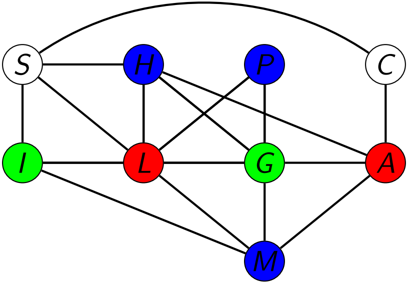
Execução do algoritmo sequencial considerando a ordem $G, L, H, P, M, A, I, S, C$.



Cores:
cor 1
cor 2
cor 3

Exemplo

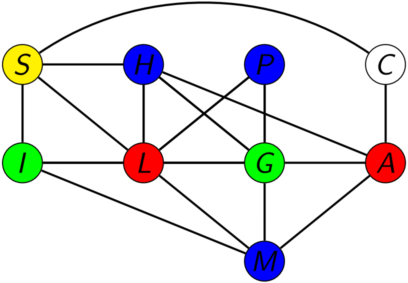
Execução do algoritmo sequencial considerando a ordem $G, L, H, P, M, A, I, S, C$.



Cores:
cor 1
cor 2
cor 3

Exemplo

Execução do algoritmo sequencial considerando a ordem $G, L, H, P, M, A, I, S, C$.



Cores:

cor 1

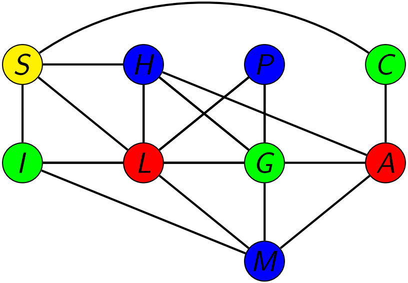
cor 2

cor 3

cor 4

Exemplo

Execução do algoritmo sequencial considerando a ordem $G, L, H, P, M, A, I, S, C$.



Cores:
cor 1
cor 2
cor 3
cor 4

Análise do algoritmo sequencial

- O algoritmo produz uma coloração própria porque evita conflitos toda vez que vai colorir um vértice.
- O tempo de execução é $O(V + E)$.
- Quantas cores serão usadas? Depende da ordem escolhida para colorir os vértices.
- Produz uma coloração ótima se for dada uma ordenação ótima. O problema é que achar esta ordenação ótima é NP-Difícil também.
- Uma propriedade interessante é que, uma vez colorido o grafo, é possível gerar a ordem dos vértices que gera esta coloração (simplesmente listando os vértices de acordo com sua cor).
- É um algoritmo eficiente, mas não eficaz.

Análise do algoritmo sequencial

- O algoritmo produz uma coloração própria porque evita conflitos toda vez que vai colorir um vértice.
- O tempo de execução é $O(V + E)$.
- Quantas cores serão usadas? Depende da ordem escolhida para colorir os vértices.
- Produz uma coloração ótima se for dada uma ordenação ótima. O problema é que achar esta ordenação ótima é NP-Difícil também.
- Uma propriedade interessante é que, uma vez colorido o grafo, é possível gerar a ordem dos vértices que gera esta coloração (simplesmente listando os vértices de acordo com sua cor).
- É um algoritmo eficiente, mas não eficaz.

Exercício

Mostre que o Algoritmo Sequencial nem sempre produz uma coloração que usa o número cromático de cores.

Algoritmo heurístico Maior Grau Primeiro

Entrada: um grafo G com n vértices.
Saída: Uma coloração de vértices $c : V_G \to \mathbb{N}$.

```
1 Enquanto existir vértices não coloridos em G faça
2     Entre os vértices sem cor de maior grau,
      escolha o vértice v com o maior grau
      de coloração;
3     Atribua a menor cor k possível para o
      vértice v :  c(v) = k;
4 Devolva a coloração de vértices c.
```

Grau de coloração

É o número de cores diferentes usadas para os vértices coloridos adjacentes de v.

Outras aplicações

Coloração de vértices

- Alocação de faixas de frequência (rádio ou TV).
- Colorir mapas.
- Separação de produtos explosivos.
- Otimização em compiladores (alocação de registradores).

Outros problemas de coloração

- Coloração de arestas.
- Coloração de faces.