Otimização Combinatória

O Problema da 3-Coloração de Grafos

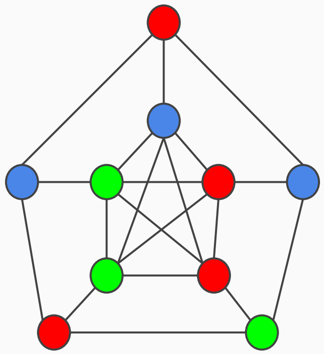
Guilherme Zanardo Borduchi	8937458
Hugo Armando Gualdron Colmenares	8429020
Tiago Moreira Trocoli da Cunha	8531417

Prof.ª Marina Andretta

Introdução ao Problema

Problema das Três Cores

- Seja um grafo G(V,E).
- Objetivo: Colorir os vértices do grafo com somente 3 cores tal que nenhum vértice adjacente tenha cores iguais.



Algoritmos de Aproximação

Solução exata:

- Máximo de cores: 3

Algoritmo de Jonhson

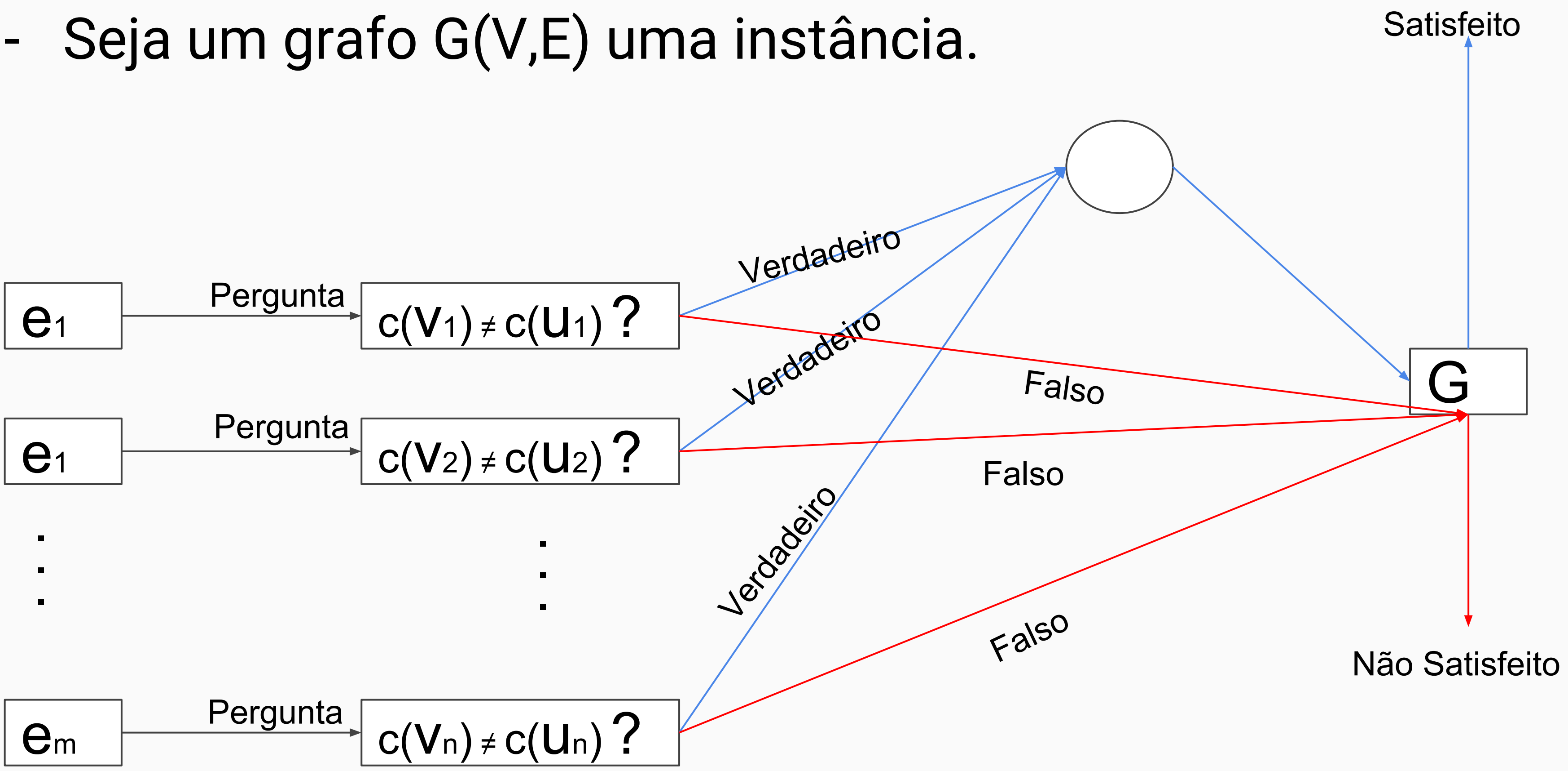
- Máximo de cores: $O(n/\log(n))$

Algoritmo de Wigderson

- Máximo de cores: $O(\sqrt{n})$

Prova NP

- Seja um grafo G(V,E) uma instância.



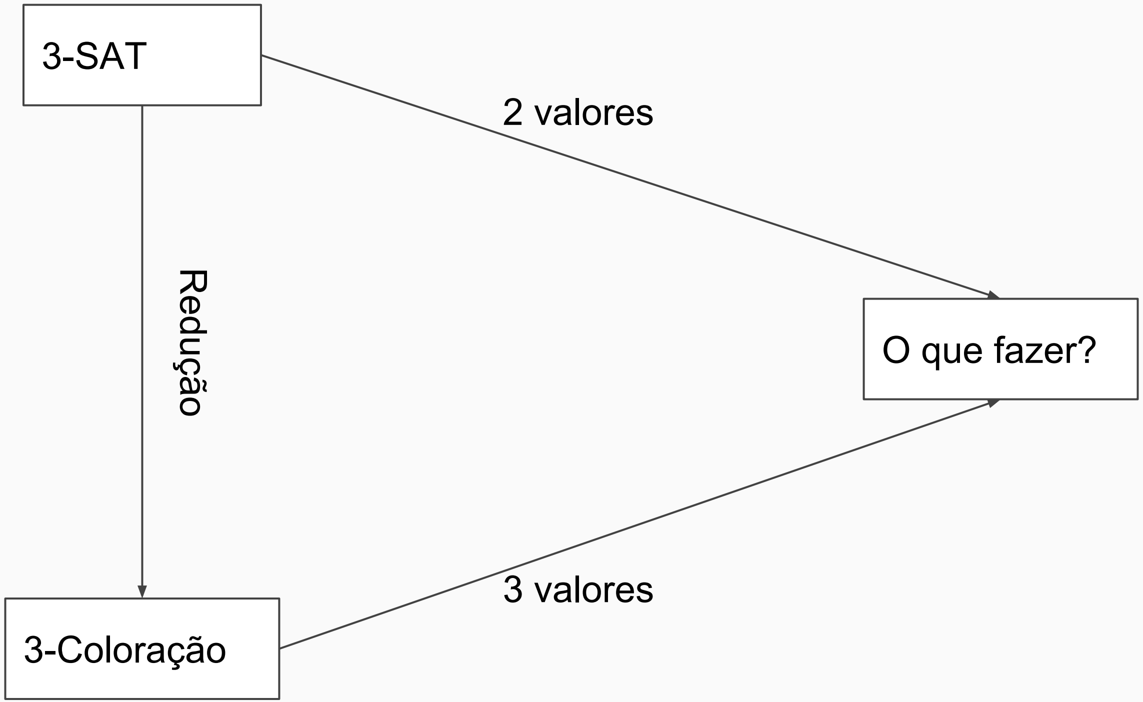
Prova NP-Completo

Redução do 3-SAT para 3-Coloração.

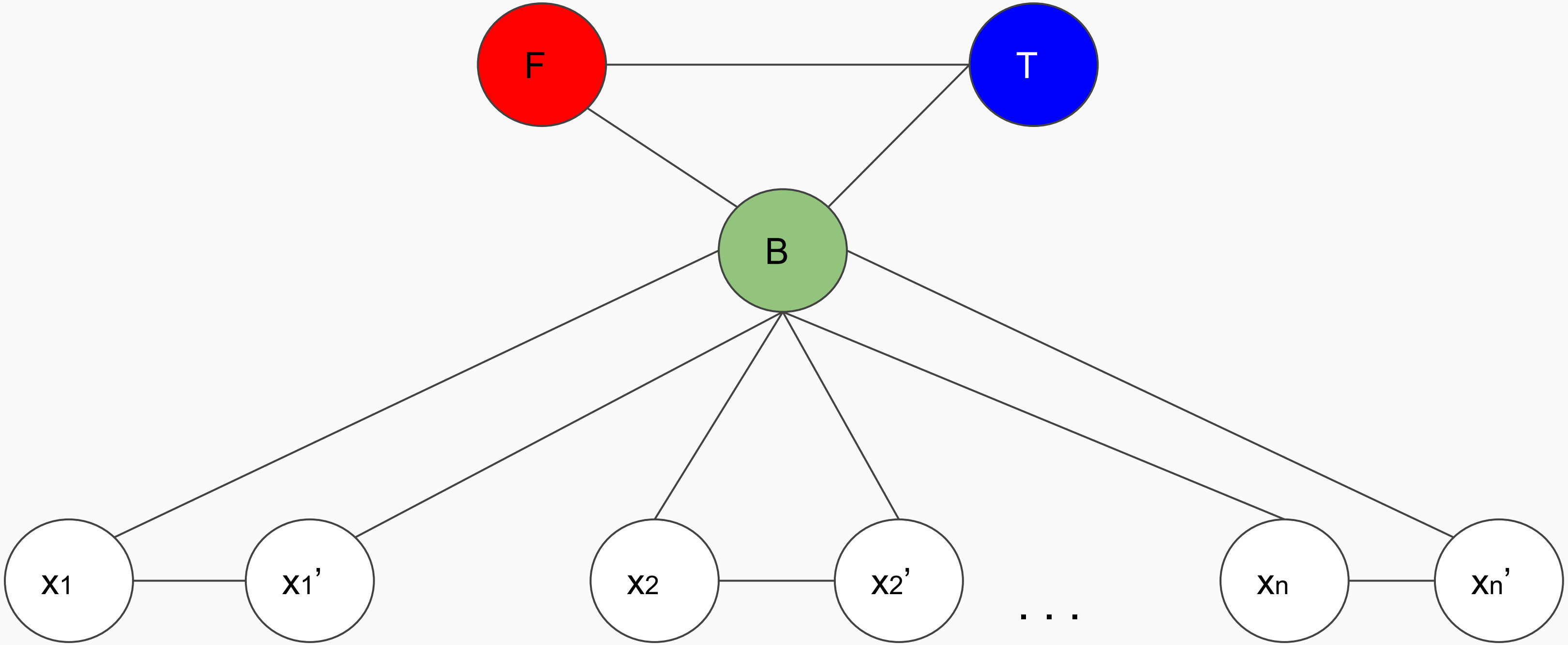
- Seja φ uma instância do 3-SAT, com variáveis $x_1, x_2 \ldots x_n$ e cláusulas $C_1, C_2 \ldots C_m$.

Prova NP-Completo

Parte 1



Prova NP-Completo

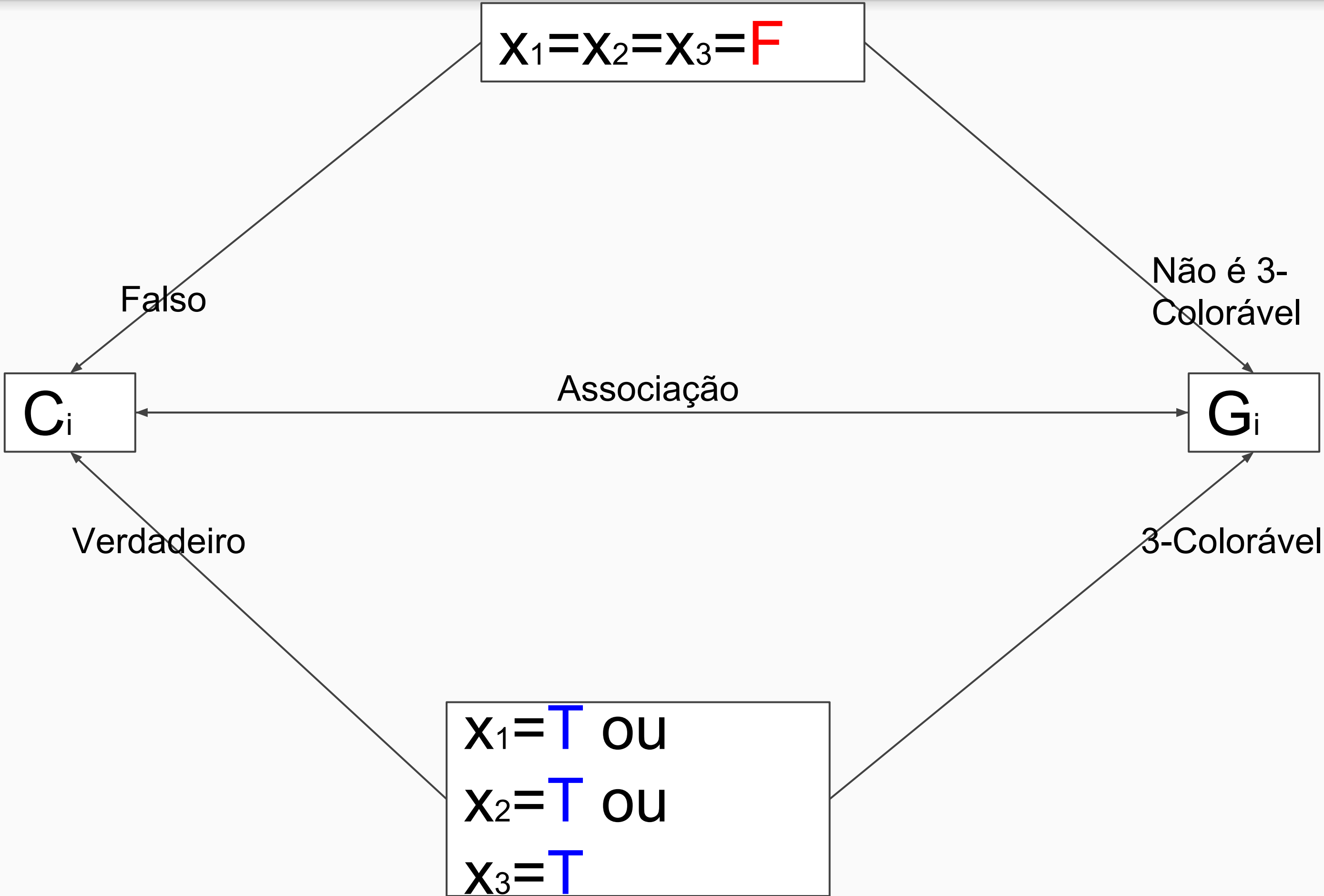


Prova NP-Completo

- Como os vértices das variáveis não podem ser iguais a cor BASE, então só podem ser coloridas com TRUE e FALSE.

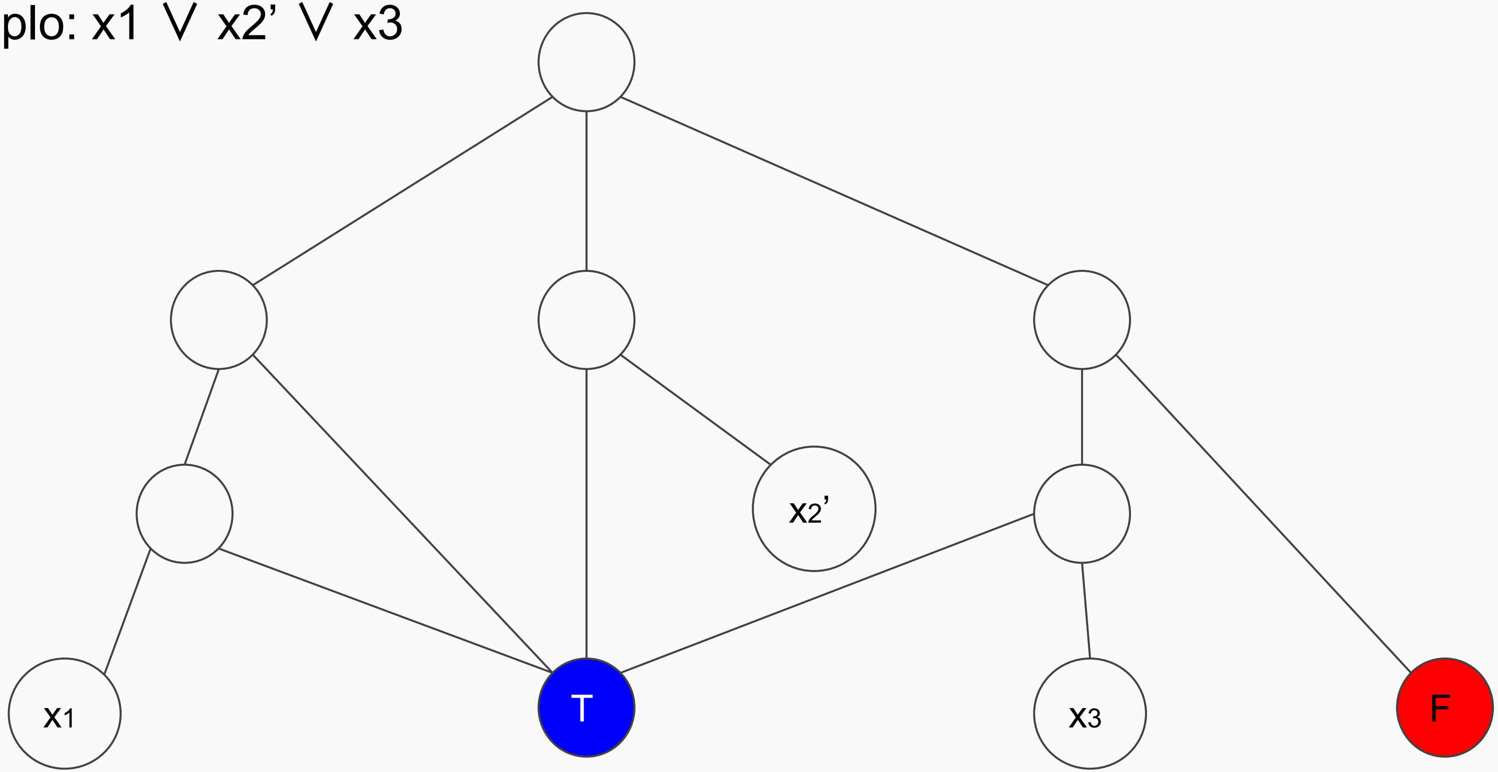
Prova NP-Completo

Parte 2

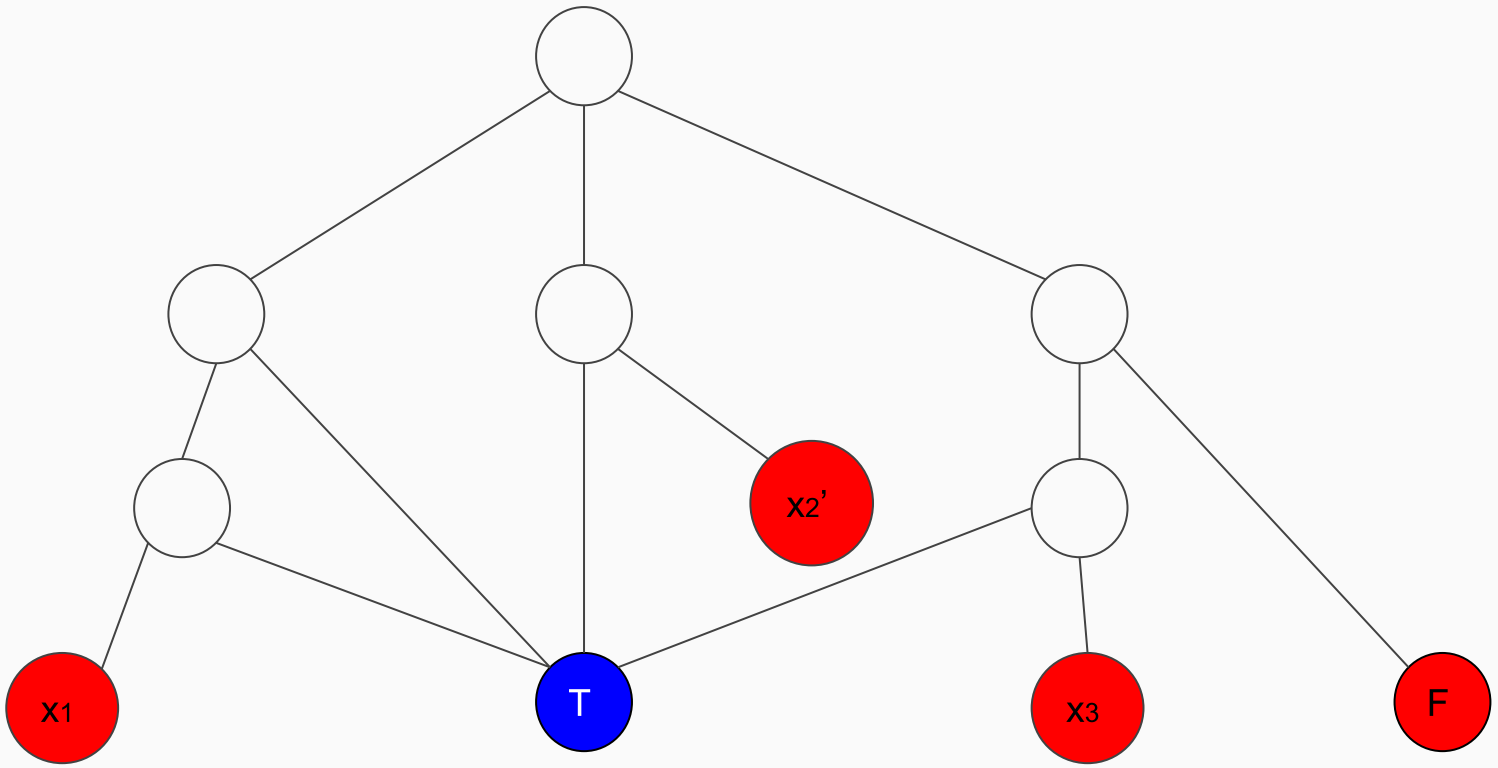


Prova NP-Completo

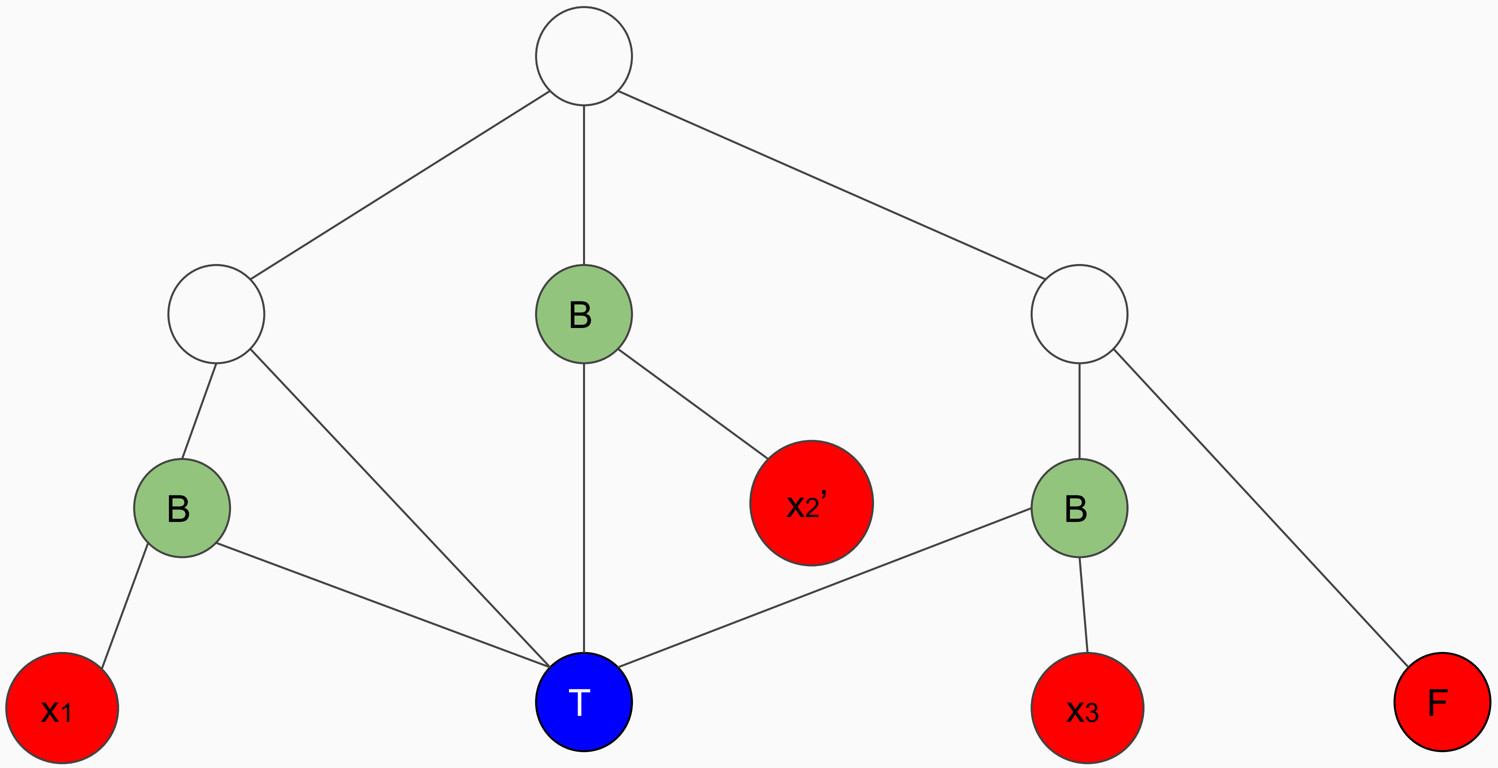
Exemplo: x1 V x2’ V x3



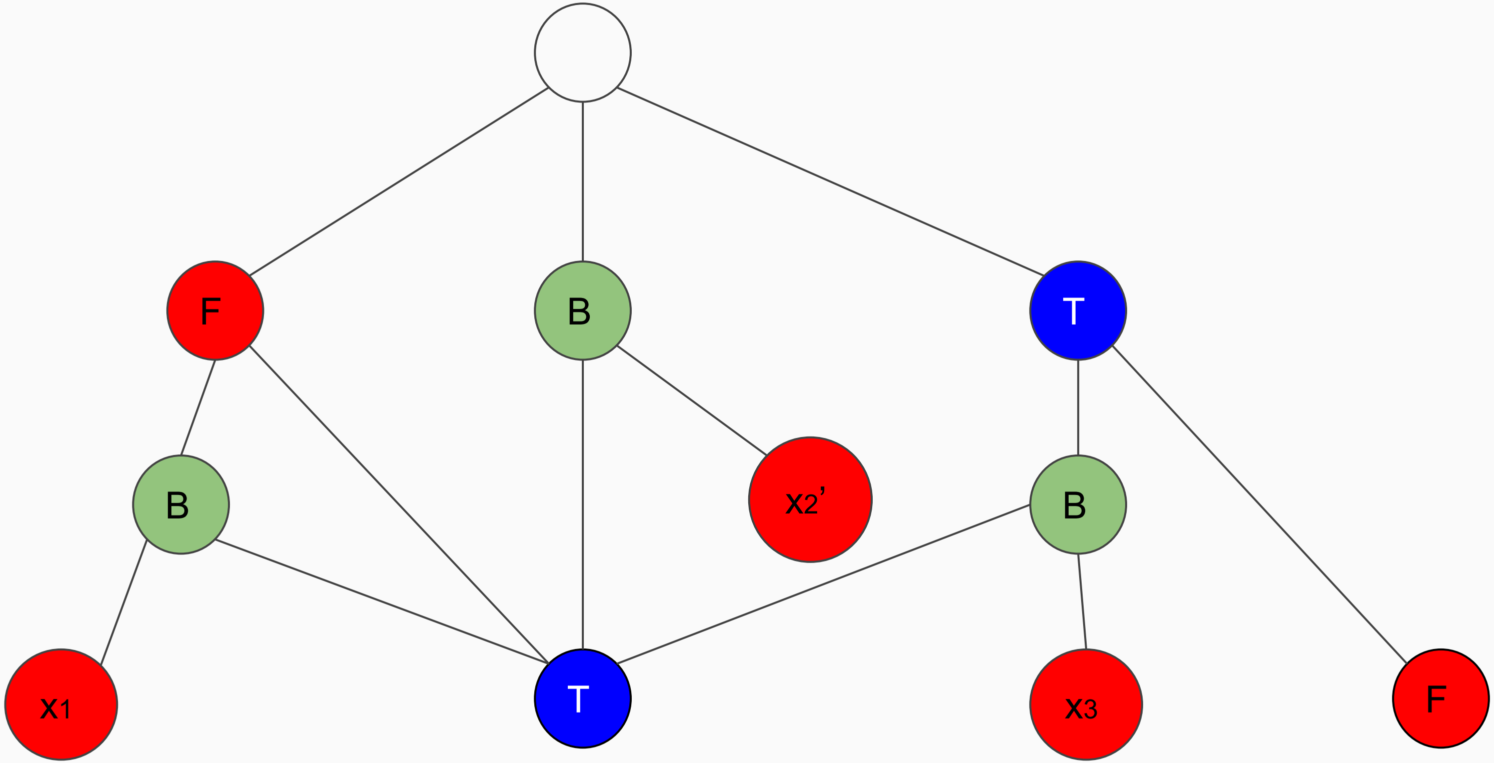
Prova NP-Completo



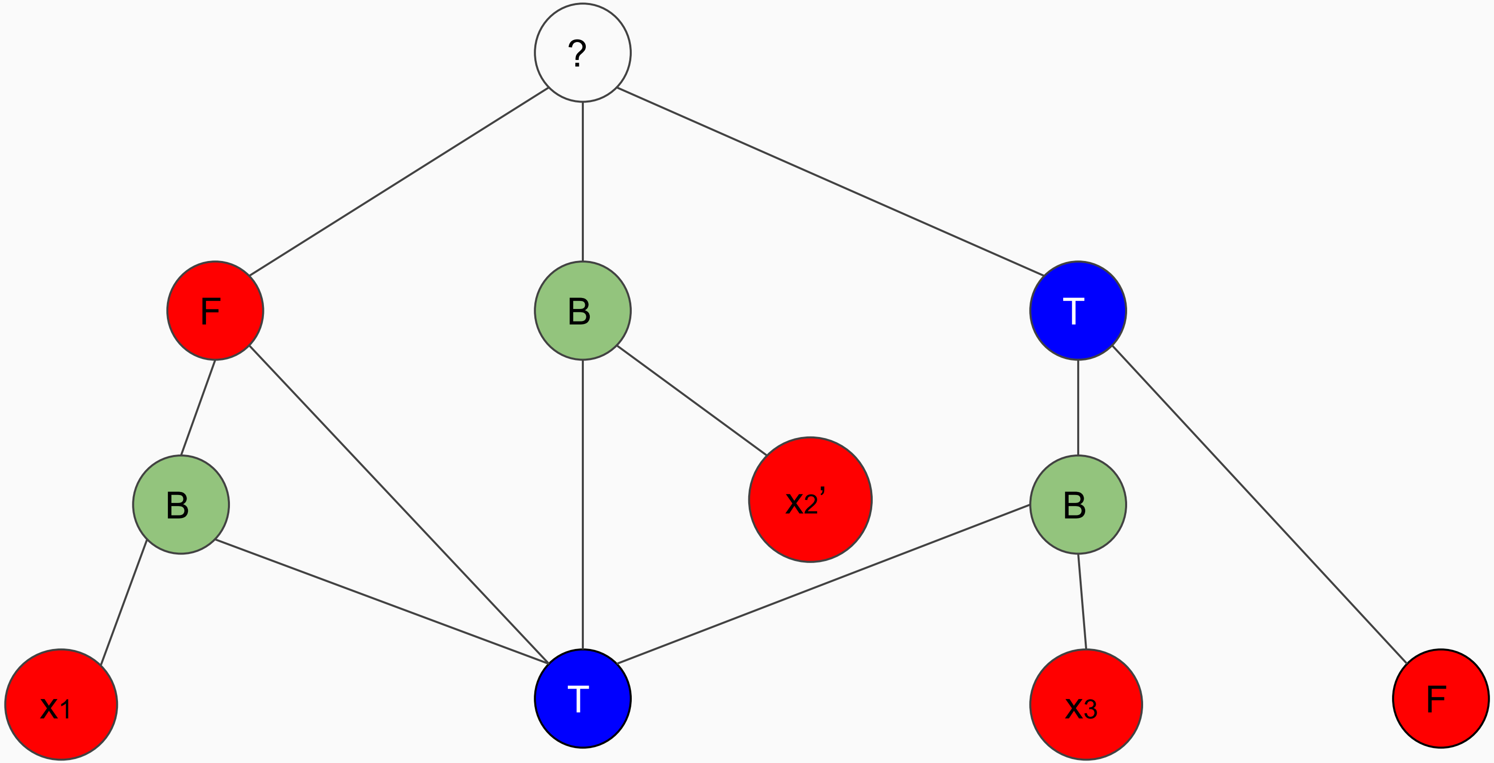
Prova NP-Completo



Prova NP-Completo

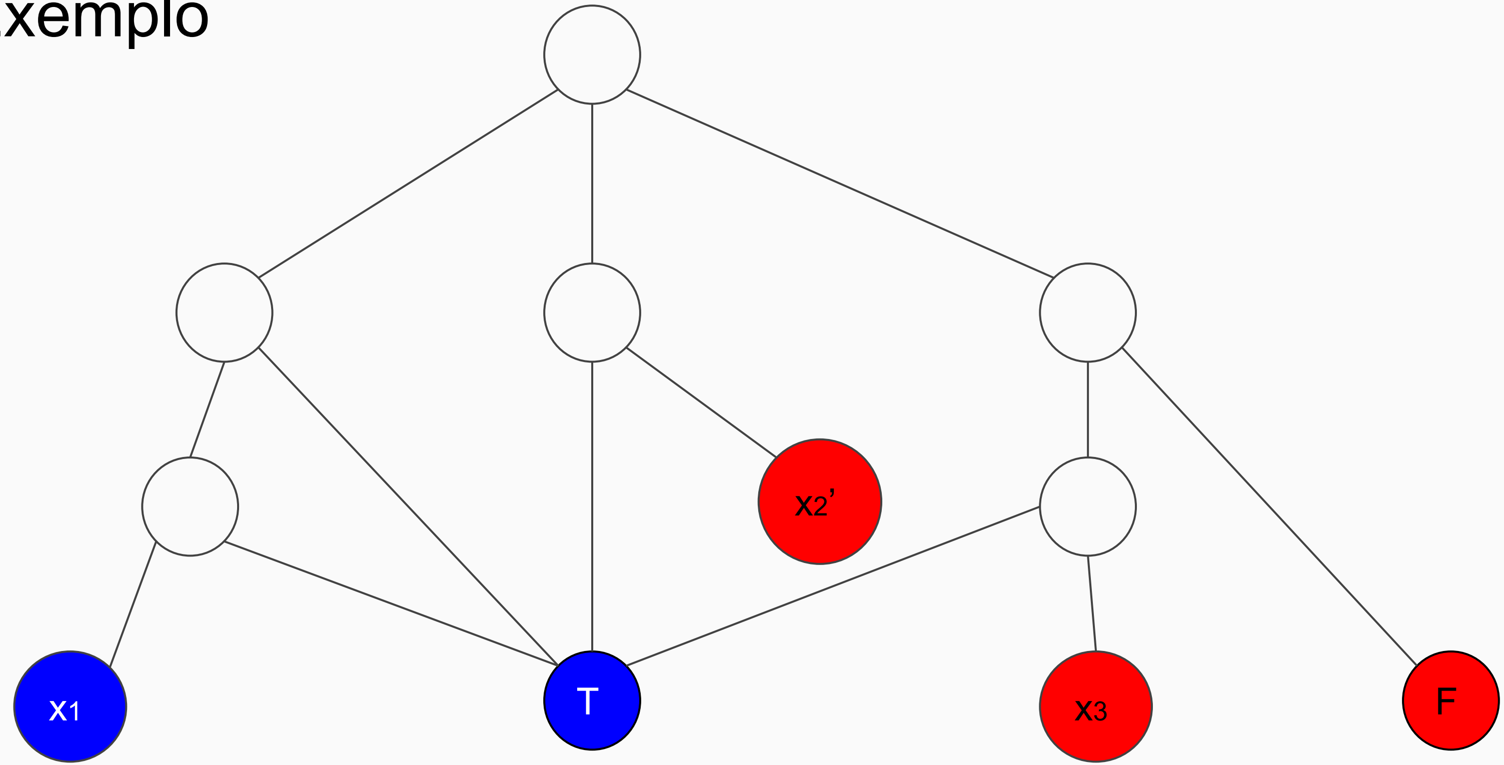


Prova NP-Completo



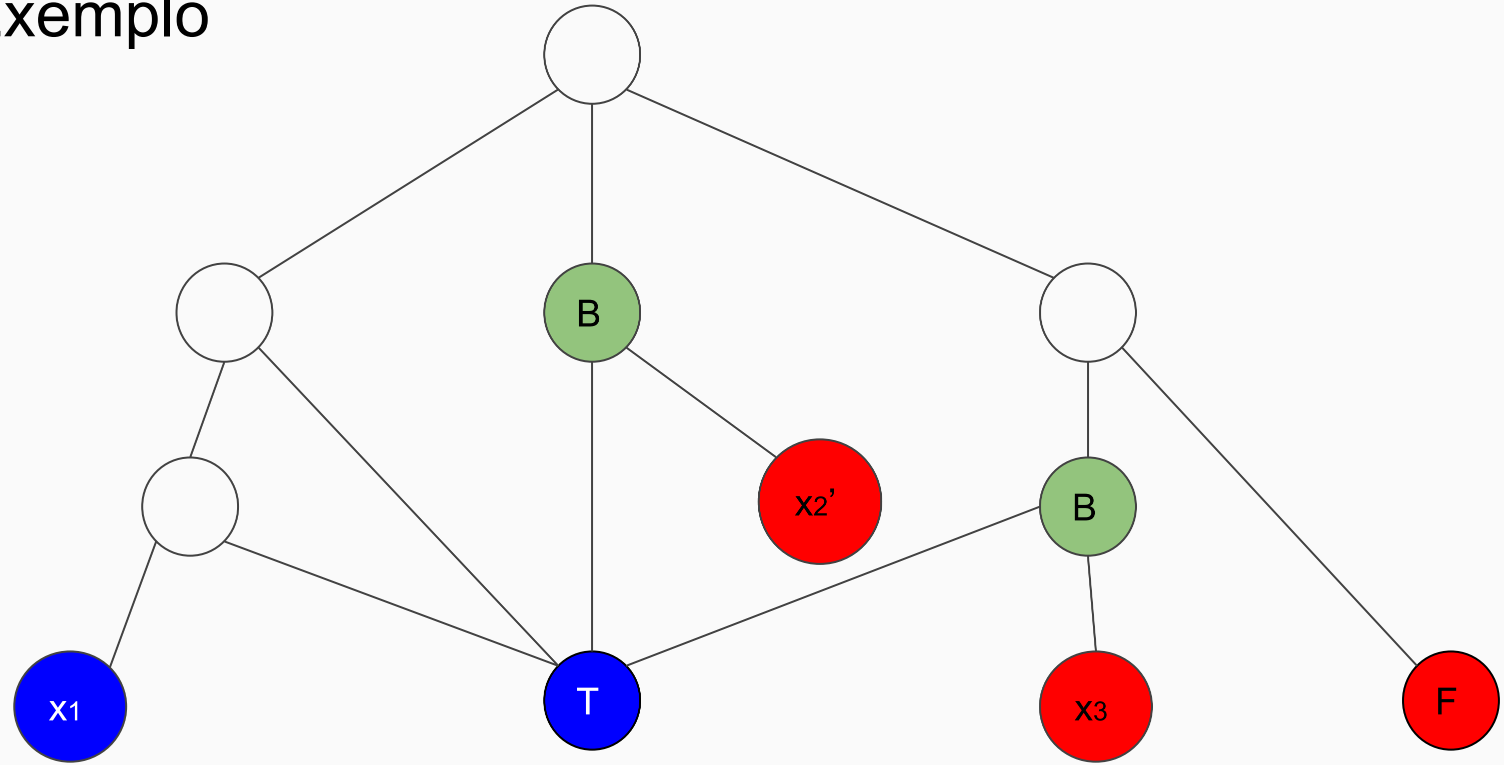
Prova NP-Completo

Exemplo



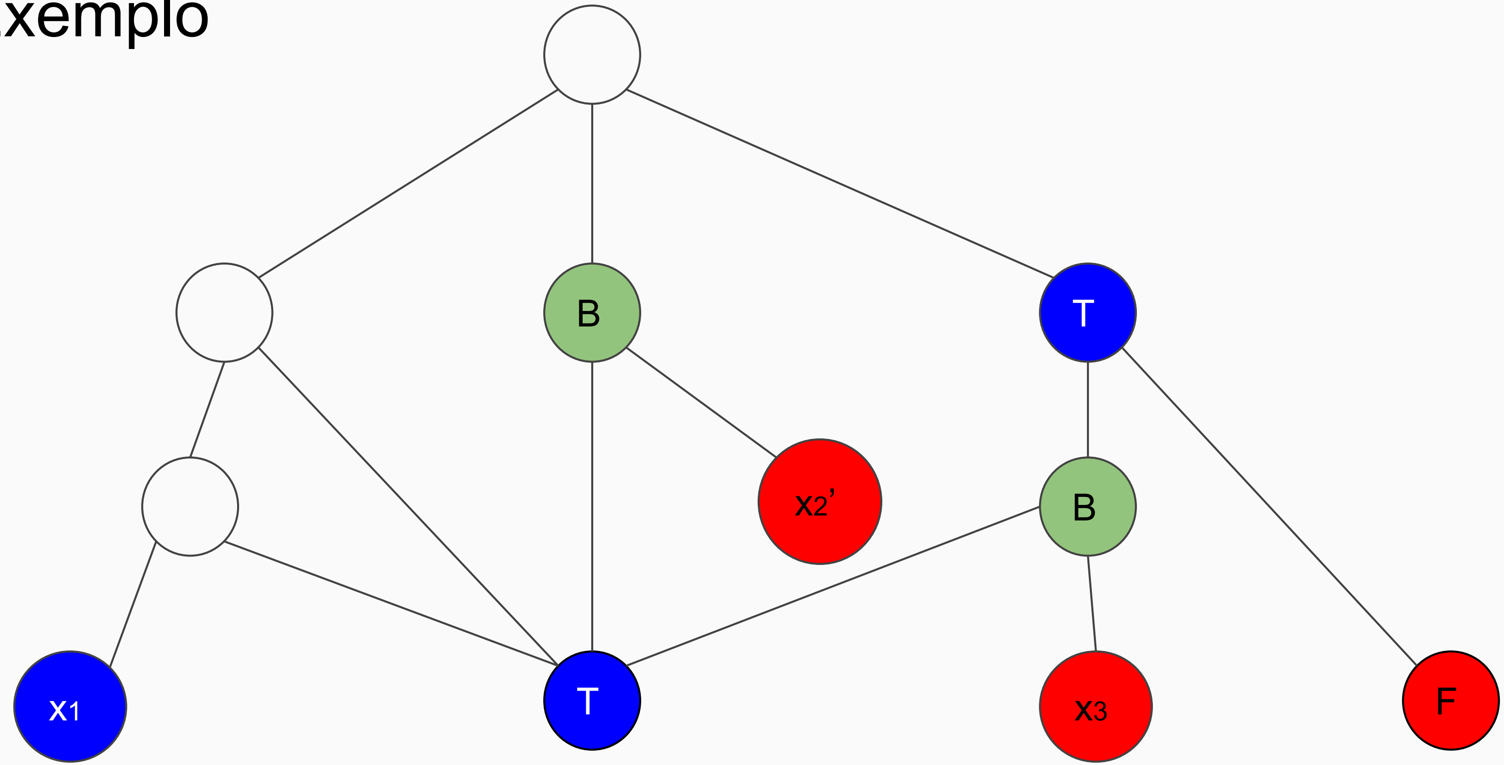
Prova NP-Completo

Exemplo



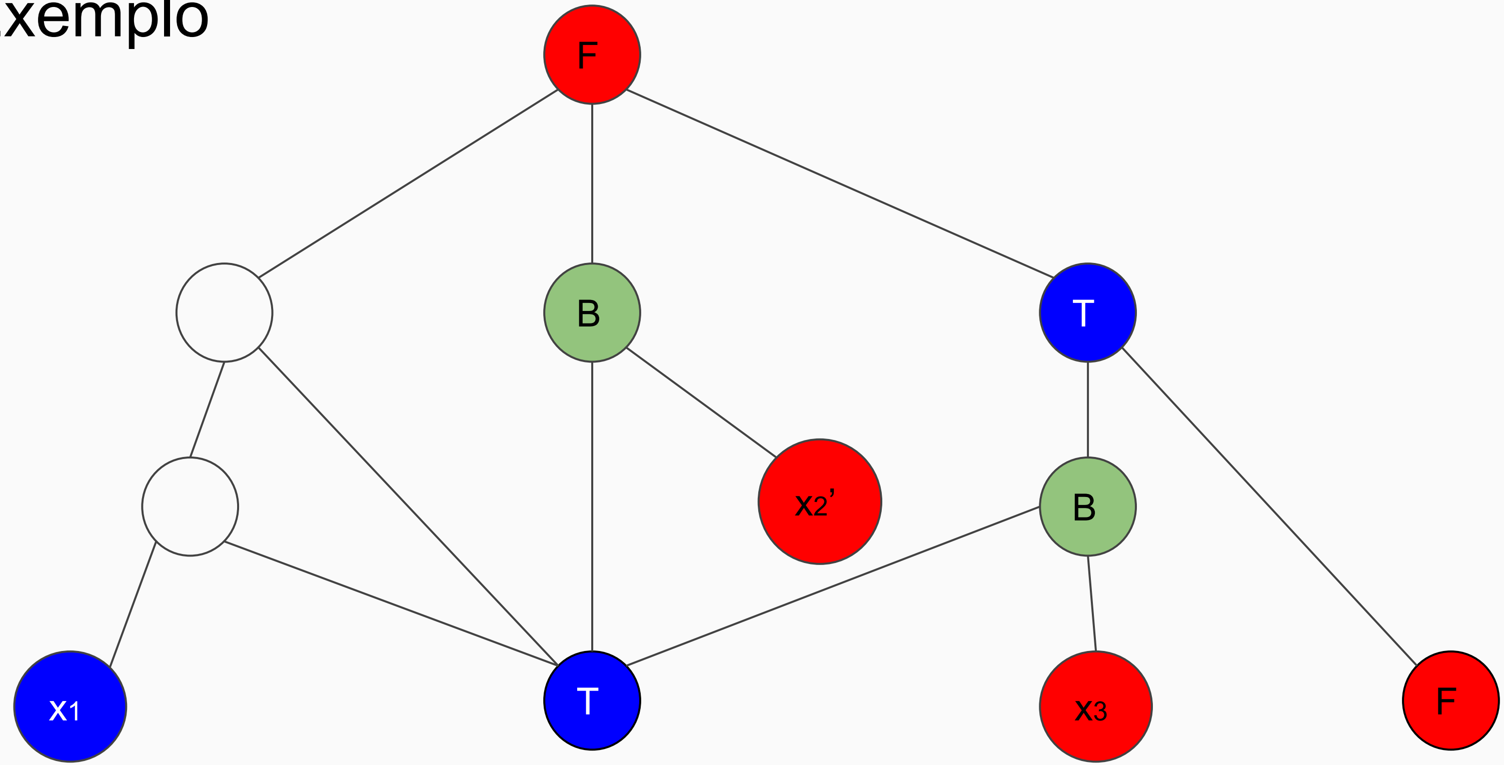
Prova NP-Completo

Exemplo



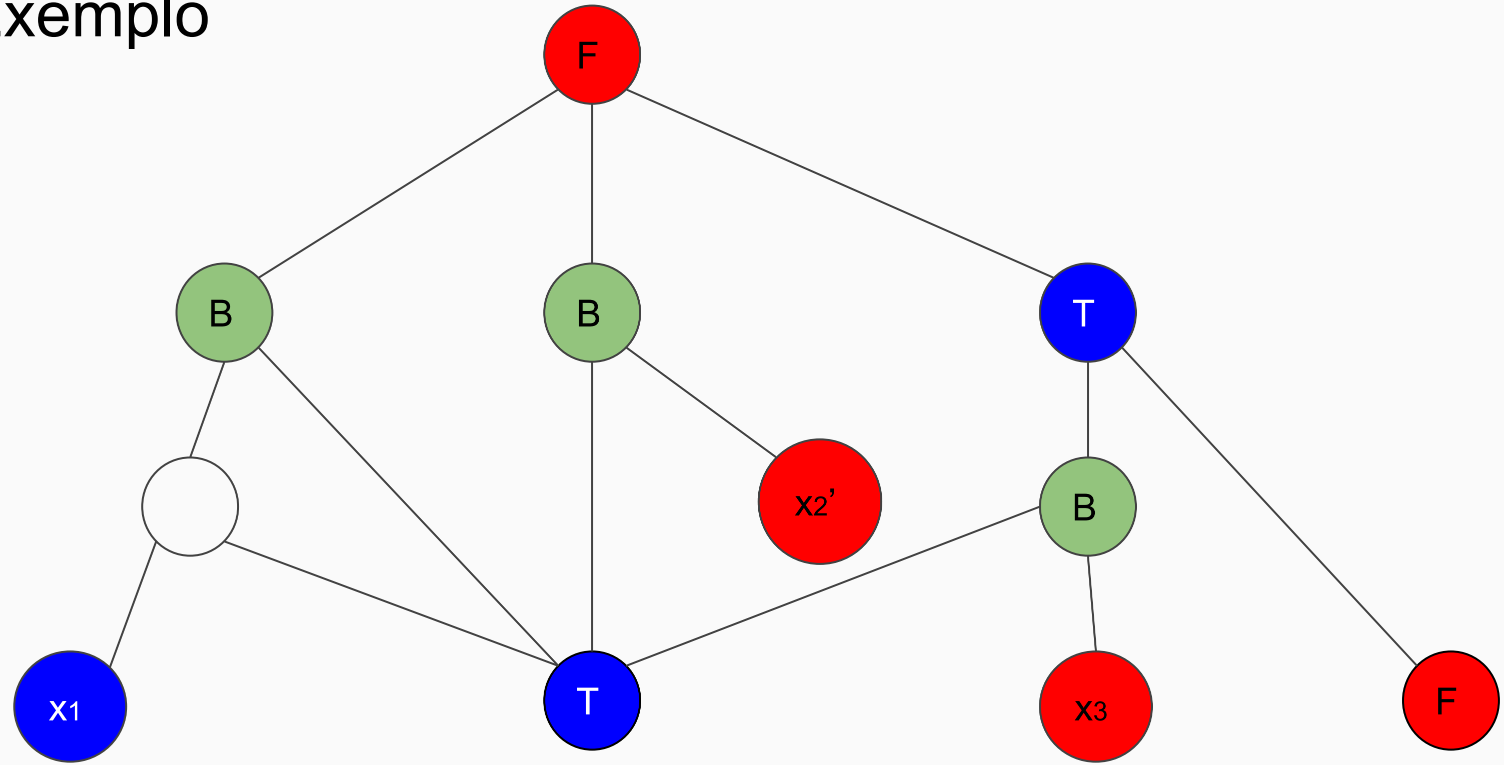
Prova NP-Completo

Exemplo



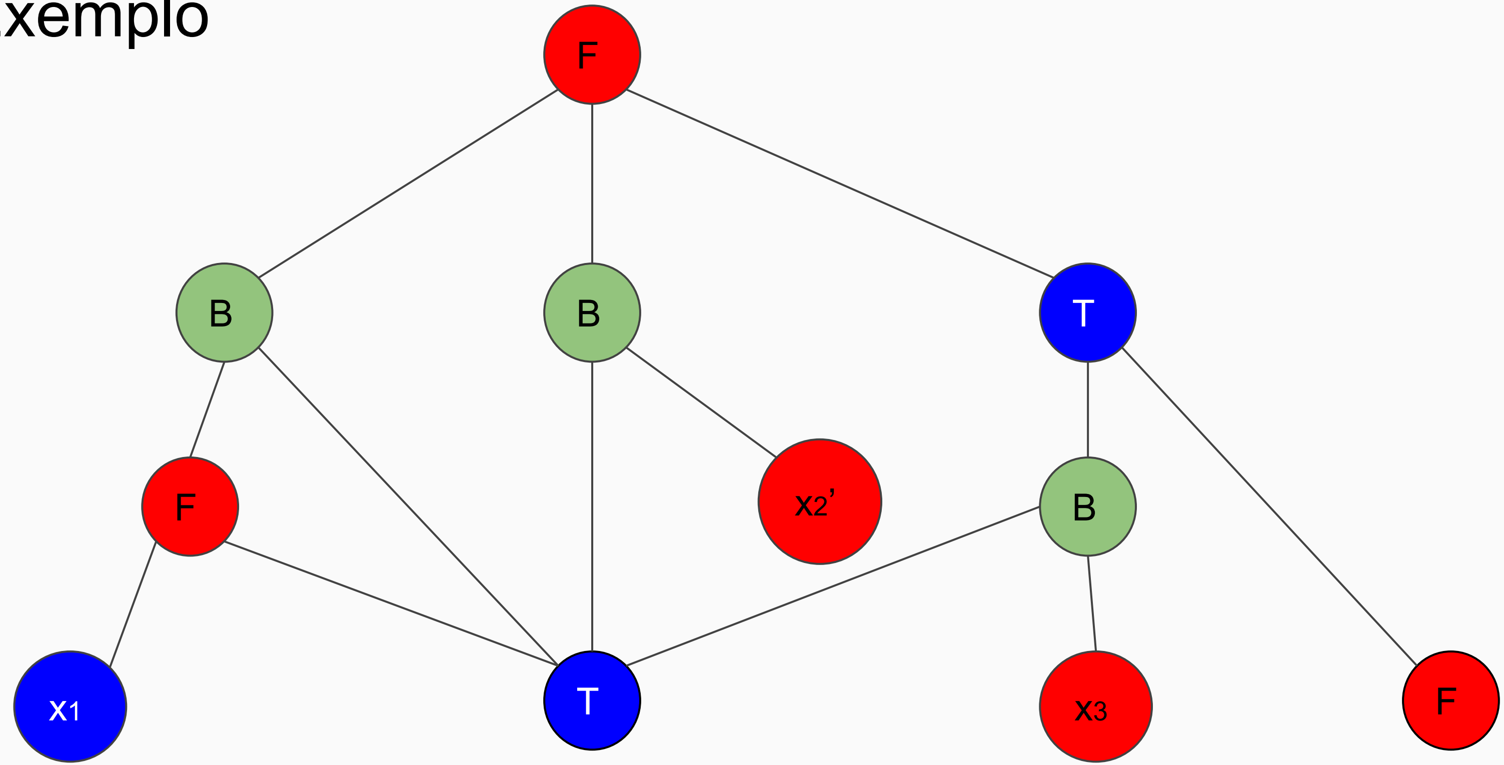
Prova NP-Completo

Exemplo

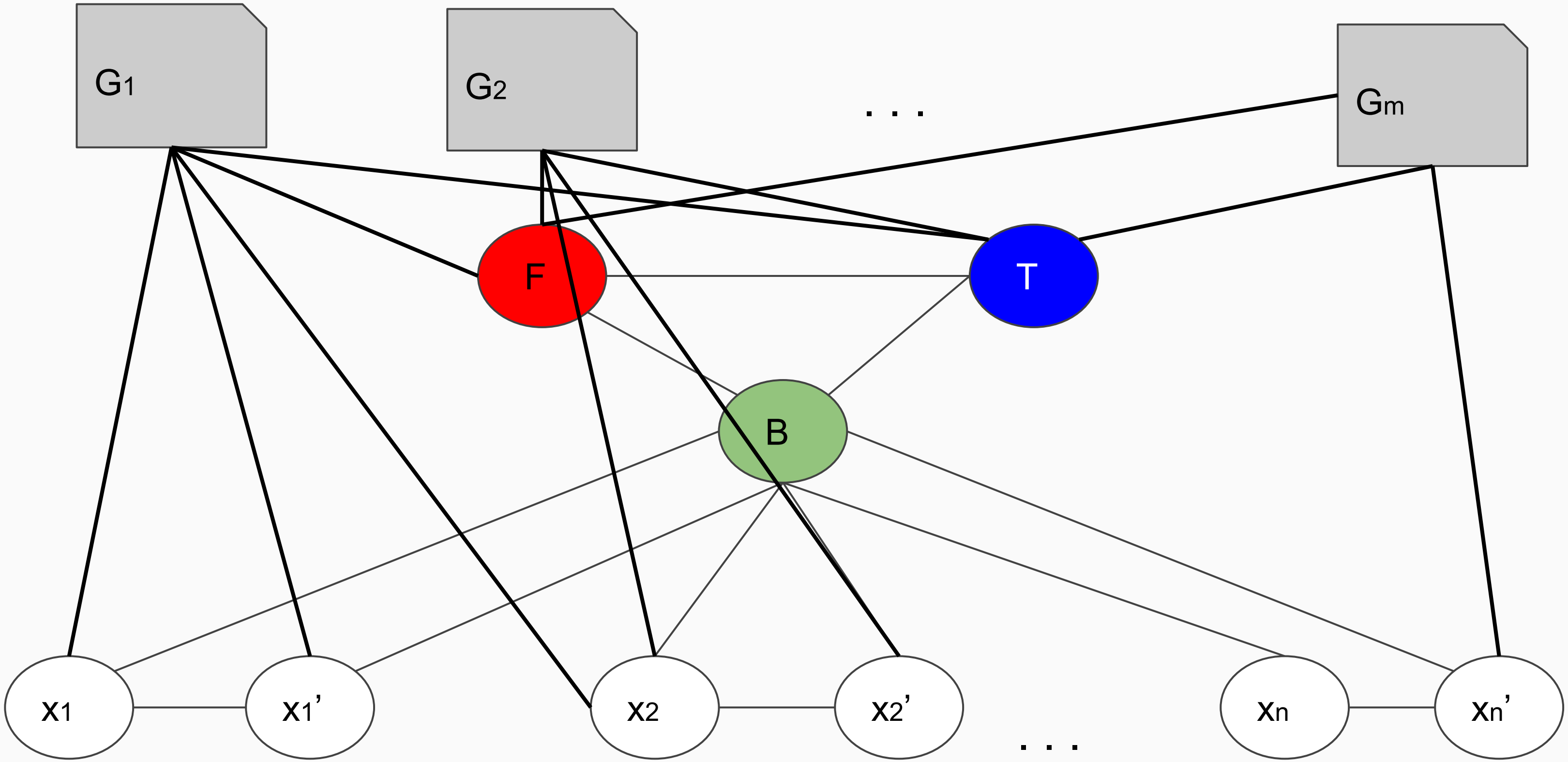


Prova NP-Completo

Exemplo

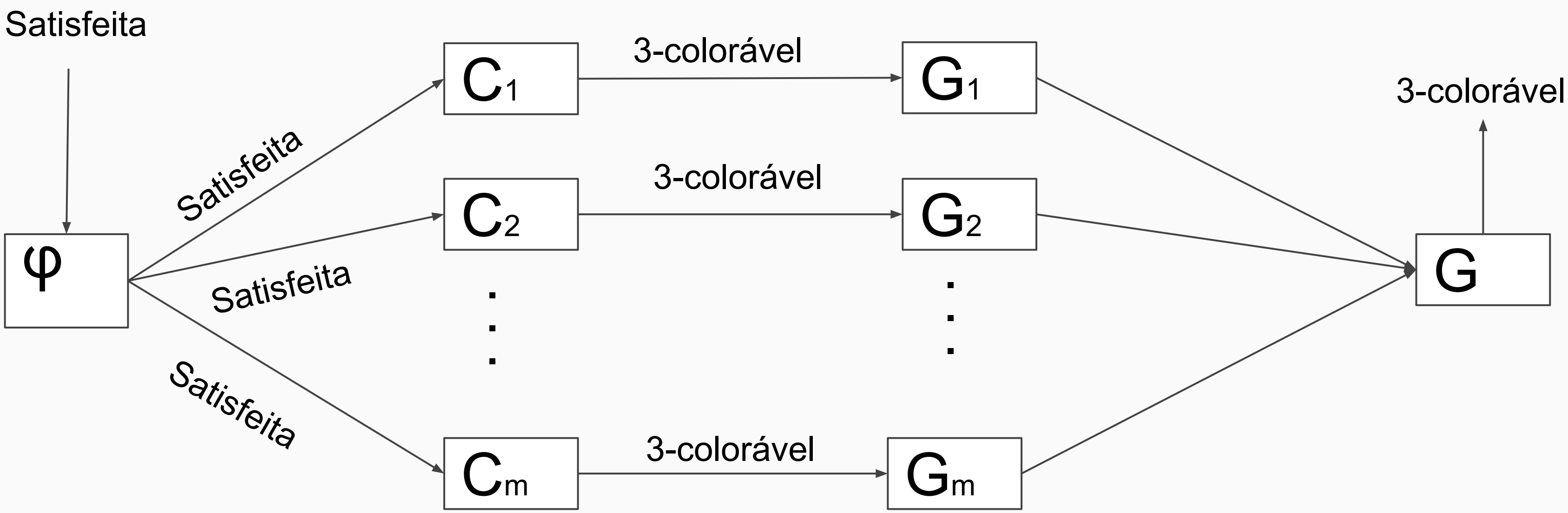


Prova NP-Completo



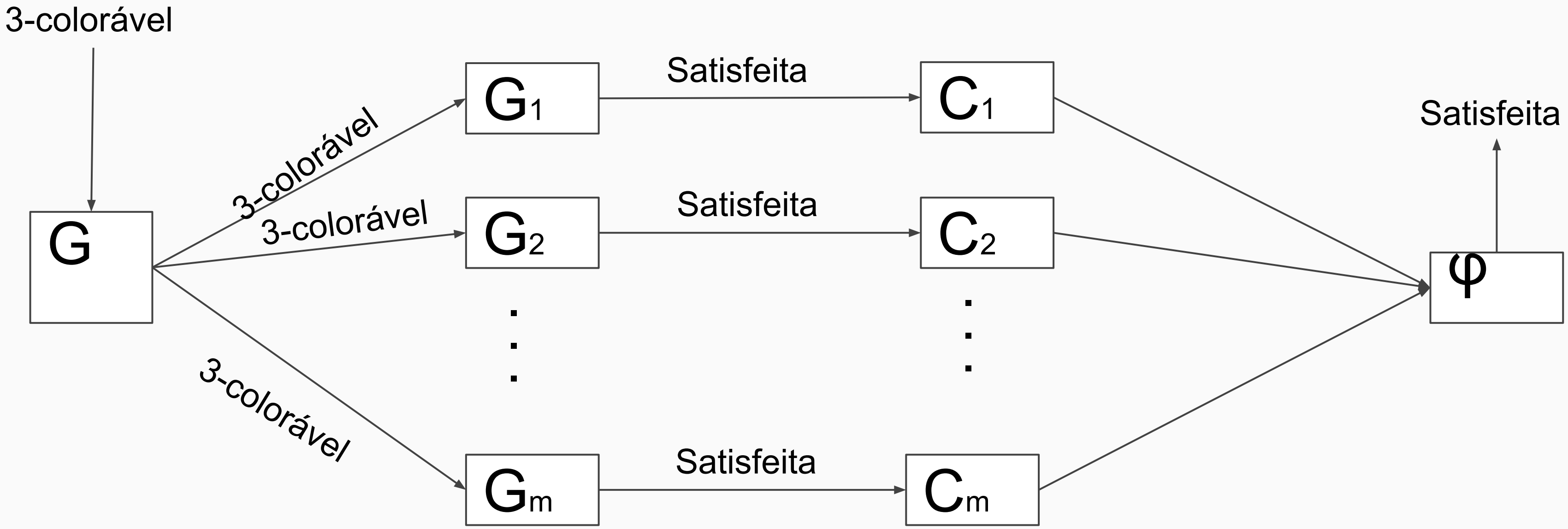
Prova NP-Completo

- Se a instância φ é satisfeita para uma dada valoração, então o grafo G é 3-colorável.



Prova NP-Completo

- Se o grafo G é 3-colorável, então a instância φ é satisfeita.



Algoritmo de Johnson

O algoritmo de Johnson foi o primeiro algoritmo a mostrar uma garantia de aproximação para o problema de coloração de grafos. Trata-se de um algoritmo guloso, cuja garantia de aproximação é de $O(n/\log(n))$ cores, para um grafo G qualquer com n vértices.

Algoritmo de Johnson

- O algoritmo funciona encontrando conjuntos independentes no grafo.

- Conjunto independente: conjunto de vértices em um grafo, tais que estes não possuam nenhuma aresta em comum.

- Cada vértice de um conjunto independente terá a mesma cor. Cada conjunto independente distinto possuirá uma cor distinta.

Algoritmo de Johnson(G):

$i \leftarrow 1$.

$U \leftarrow V$. (U é o conjunto de vértices ainda sem cor)

Enquanto $U \neq \varnothing$ faça:

  $W \leftarrow U$.

  Enquanto $W \neq \varnothing$ faça:

    Seja v o vértice com menor grau no subgrafo induzido por W.

    Colorir v com a cor i.

    $W \leftarrow W - \{v\} - N(v)$.

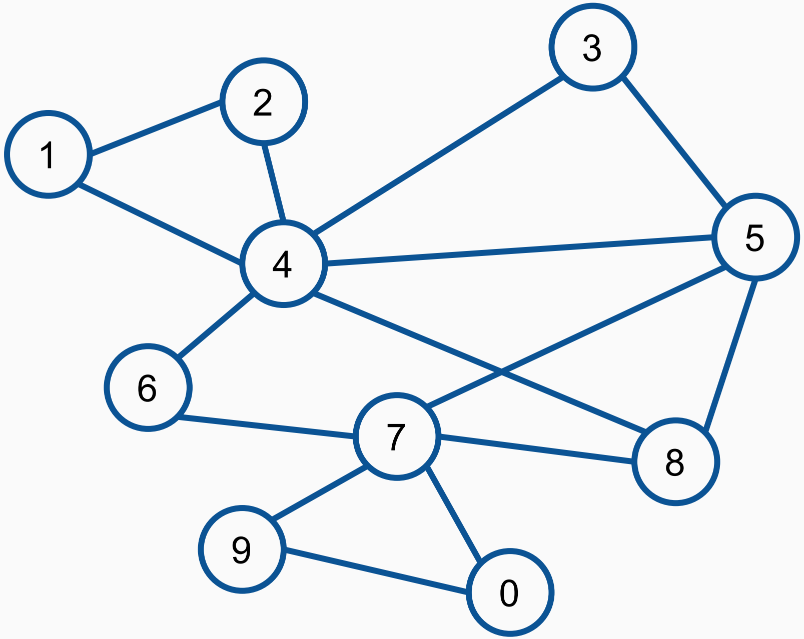
    $U \leftarrow U - \{v\}$.

  $i \leftarrow i + 1$.

Devolva a coloração.

Algoritmo de Johnson

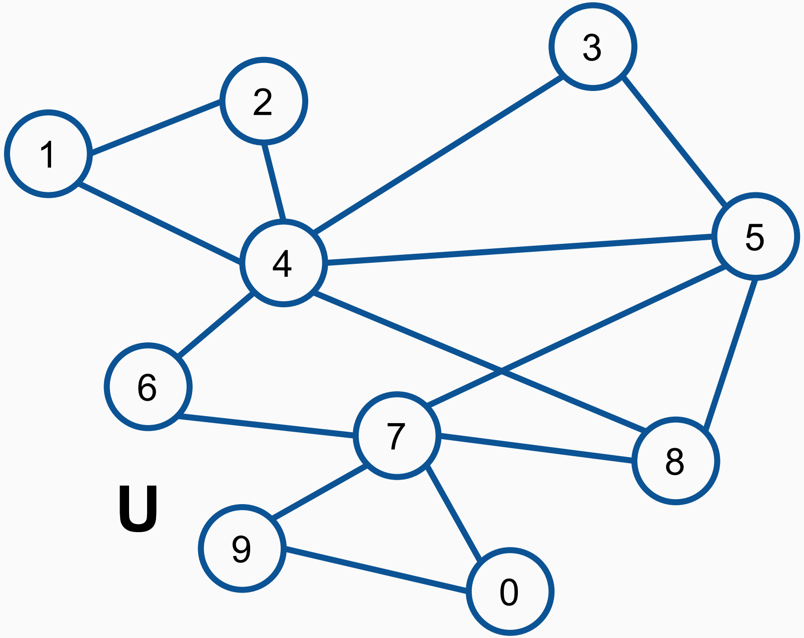
Instância de exemplo

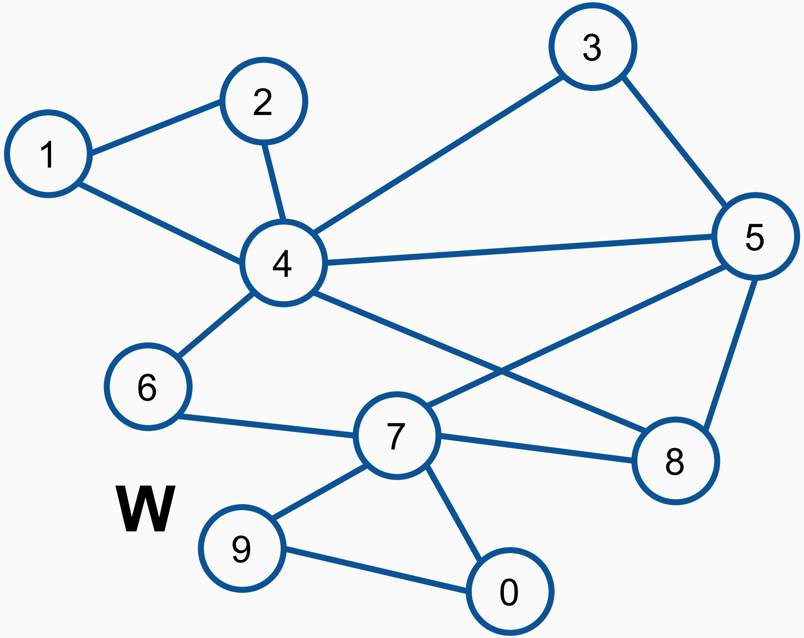


Algoritmo de Johnson

$i \leftarrow 1$
$U \leftarrow V$

$U \neq \emptyset$? Sim
$W \leftarrow U$





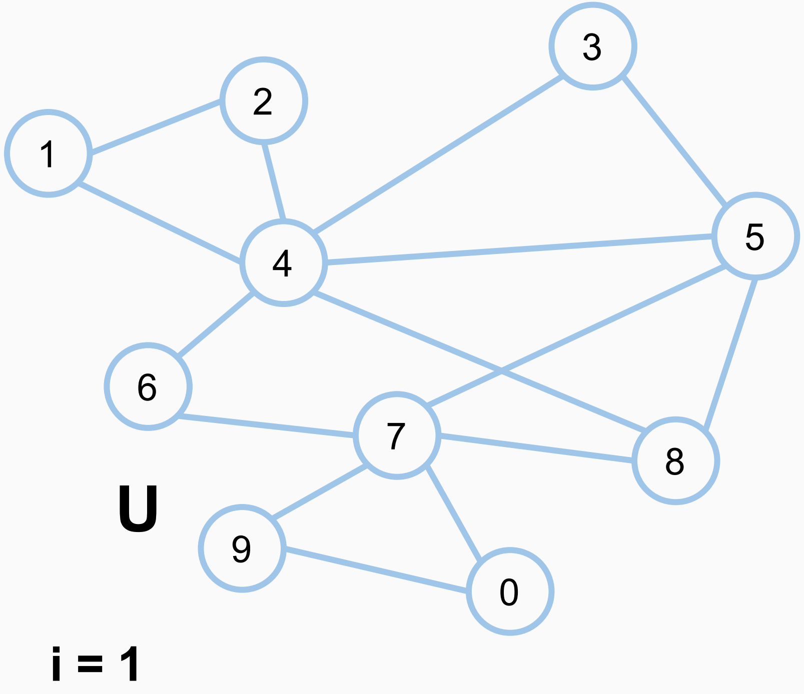
i = 1

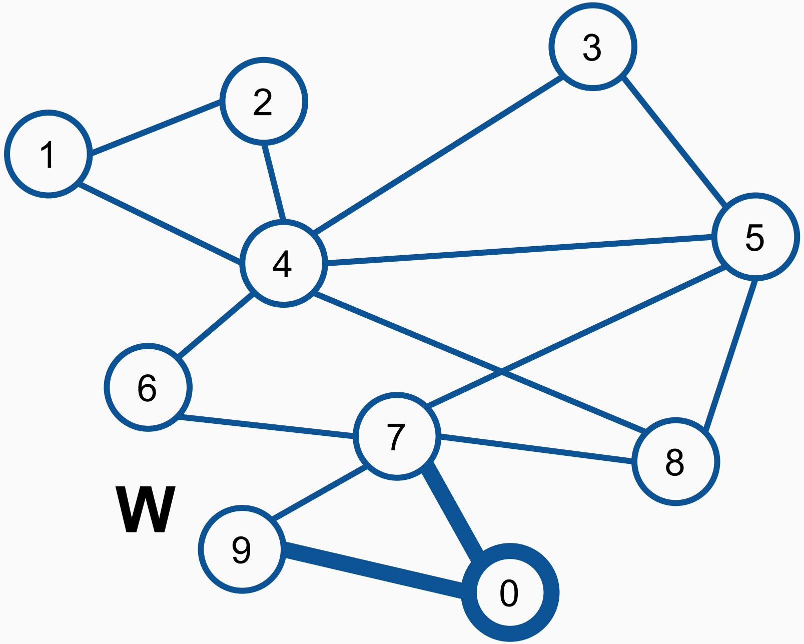
Cores: 0

Algoritmo de Johnson

W ≠ Ø? Sim

Obter o vértice de menor grau no subgrafo induzido por W

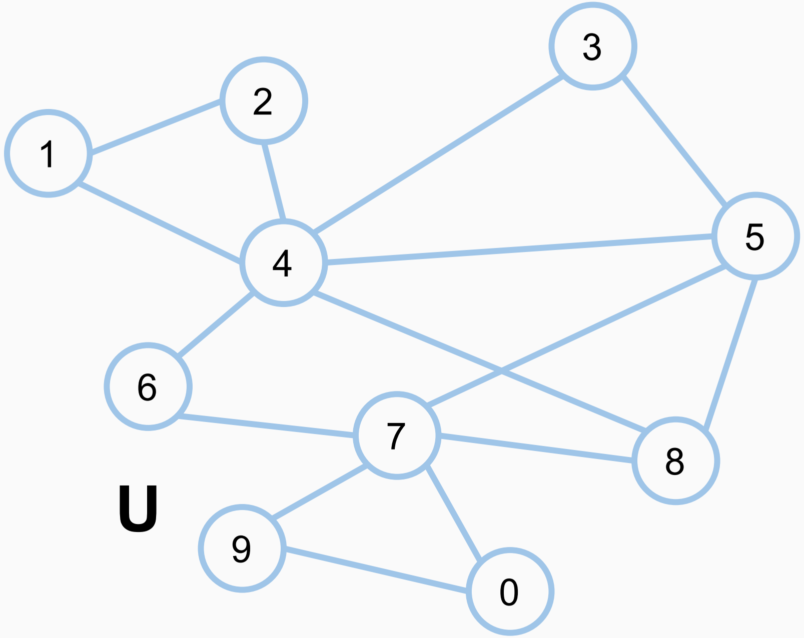


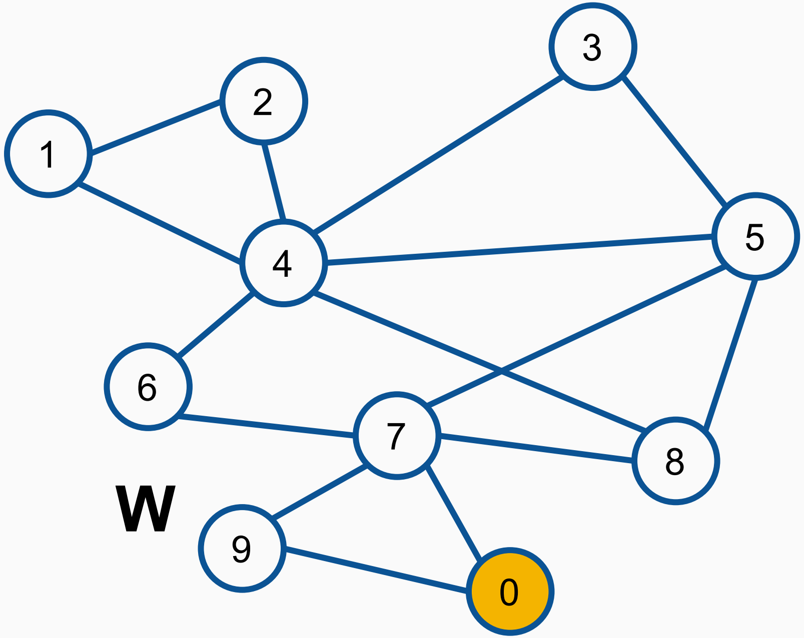


Cores: 0

Algoritmo de Johnson

Colorir v com a cor i





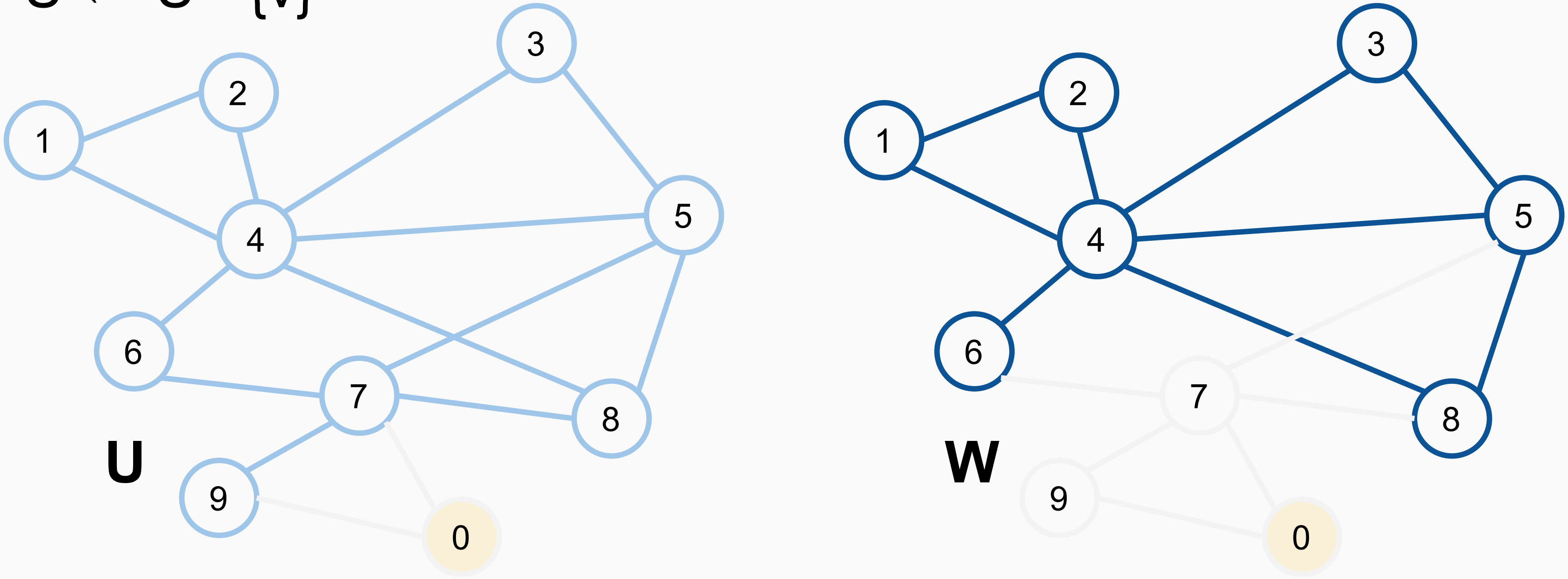
i = 1

Cores: 1

Algoritmo de Johnson

$W \leftarrow W - \{v\} - N(v)$

$U \leftarrow U - \{v\}$



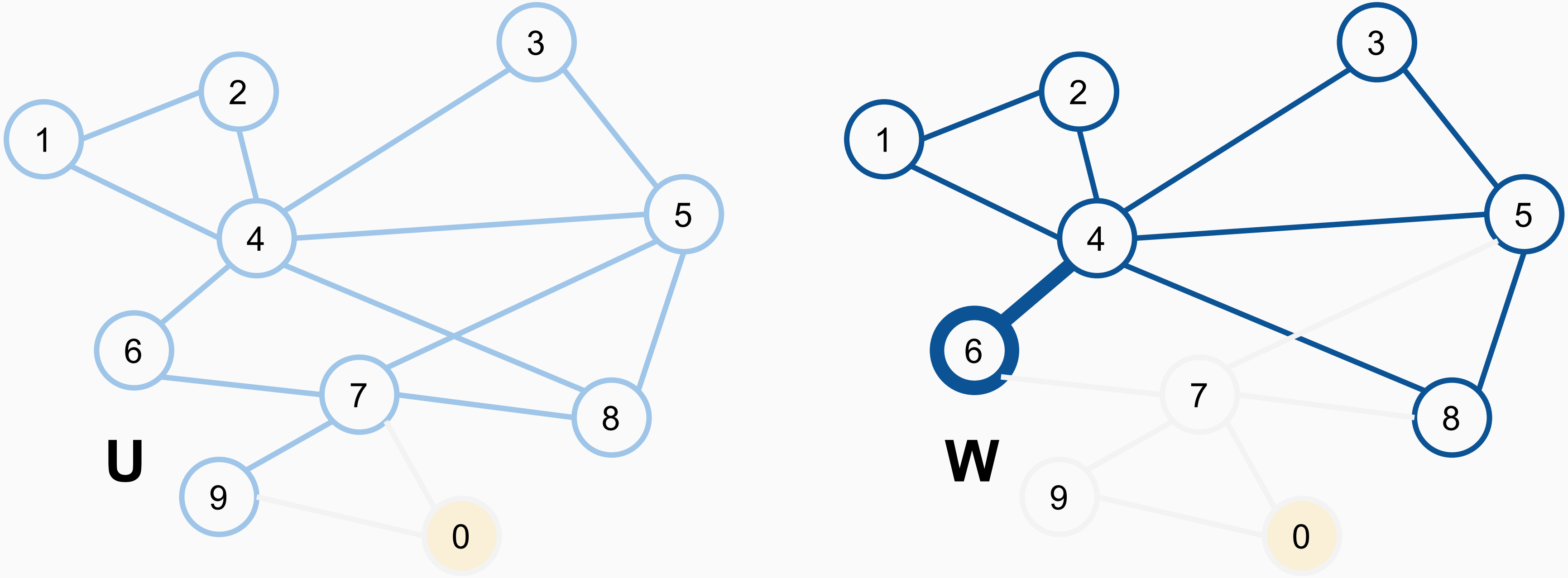
i = 1

Cores: 1

Algoritmo de Johnson

W ≠ Ø? Sim

Obter o vértice de menor grau no subgrafo induzido por W

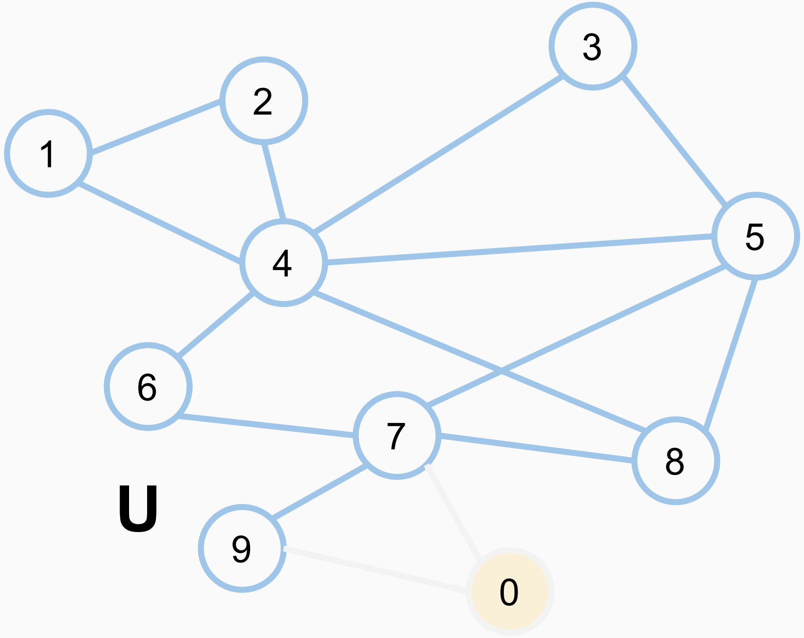


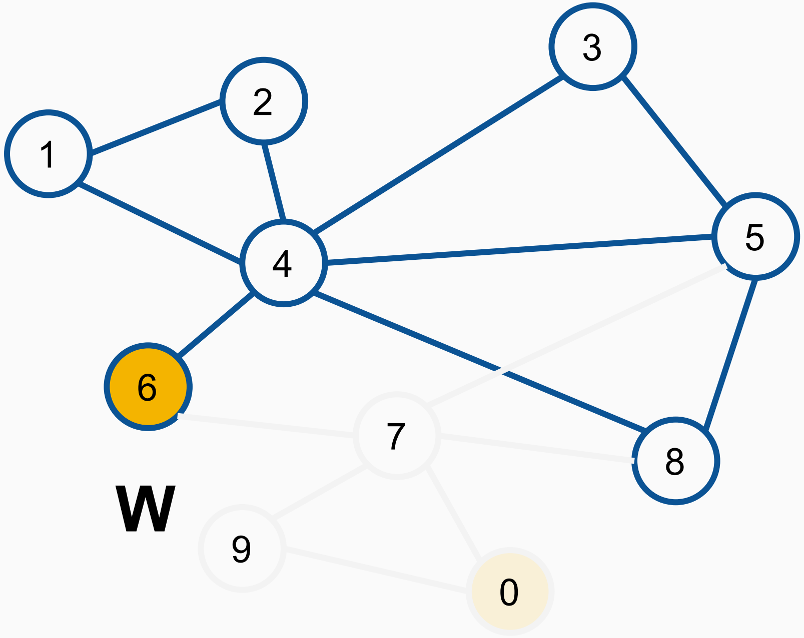
i = 1

Cores: 1

Algoritmo de Johnson

Colorir v com a cor i





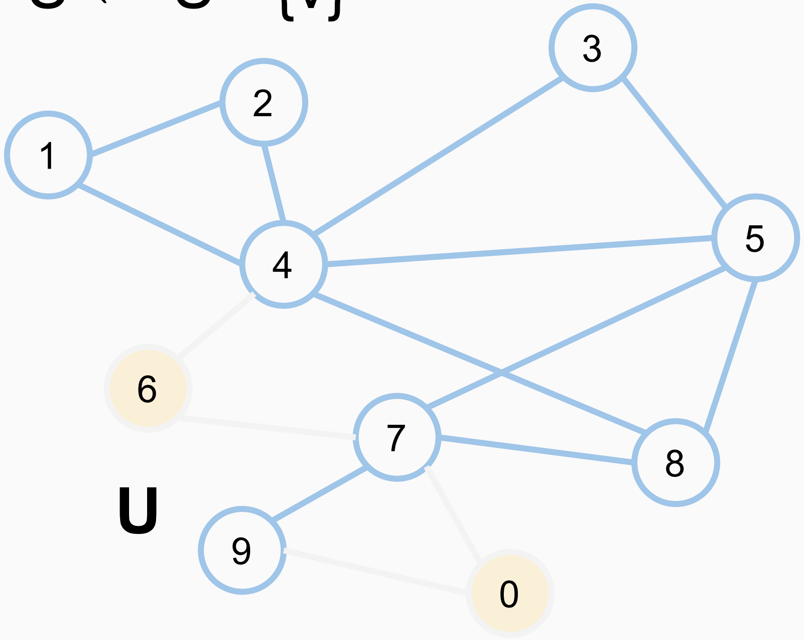
i = 1

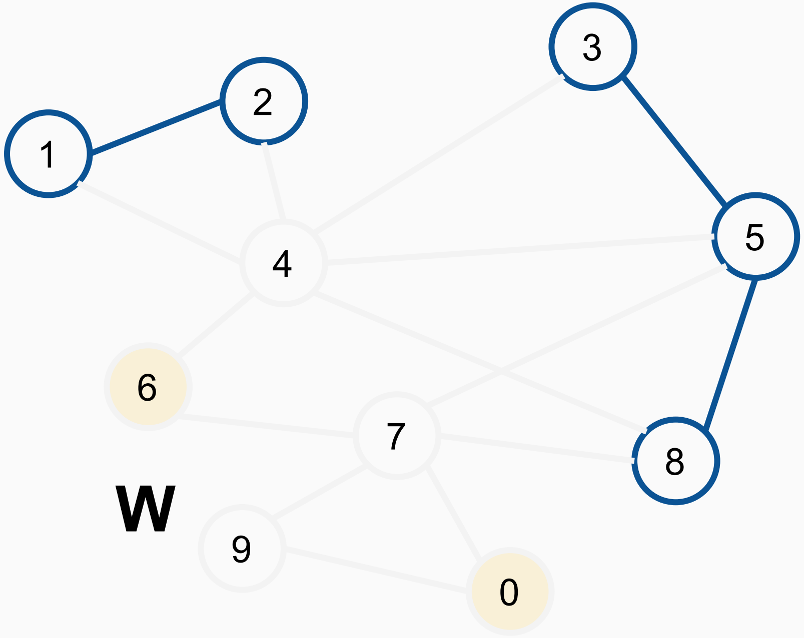
Cores: 1

Algoritmo de Johnson

$W \leftarrow W - \{v\} - N(v)$

$U \leftarrow U - \{v\}$





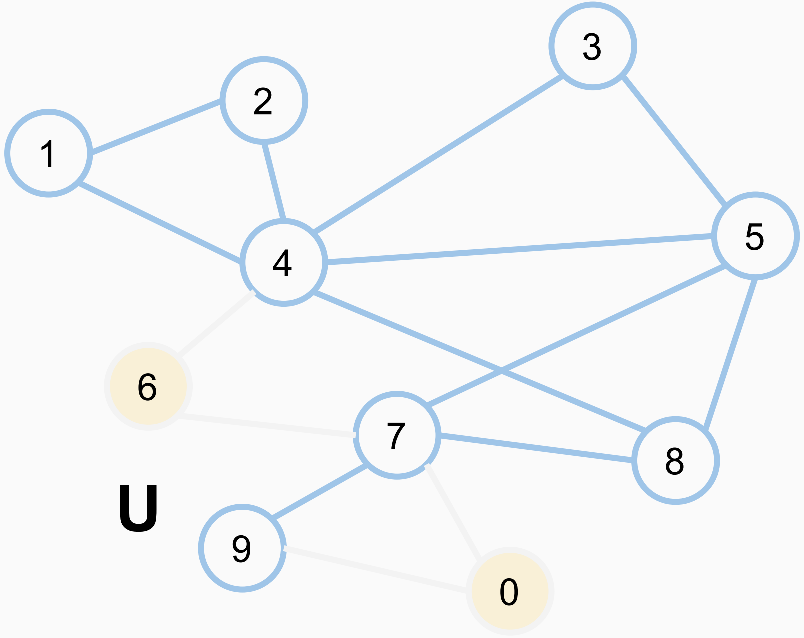
i = 1

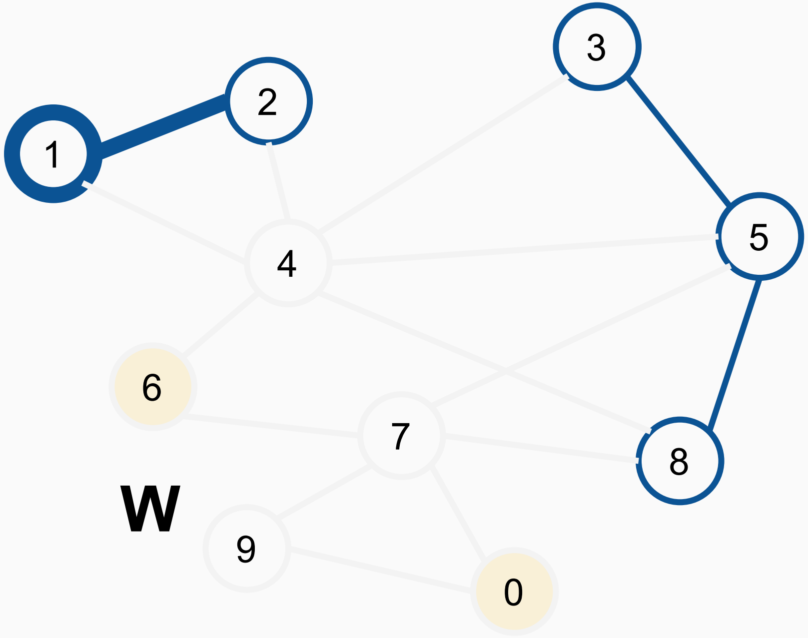
Cores: 1

Algoritmo de Johnson

W ≠ Ø? Sim

Obter o vértice de menor grau no subgrafo induzido por W



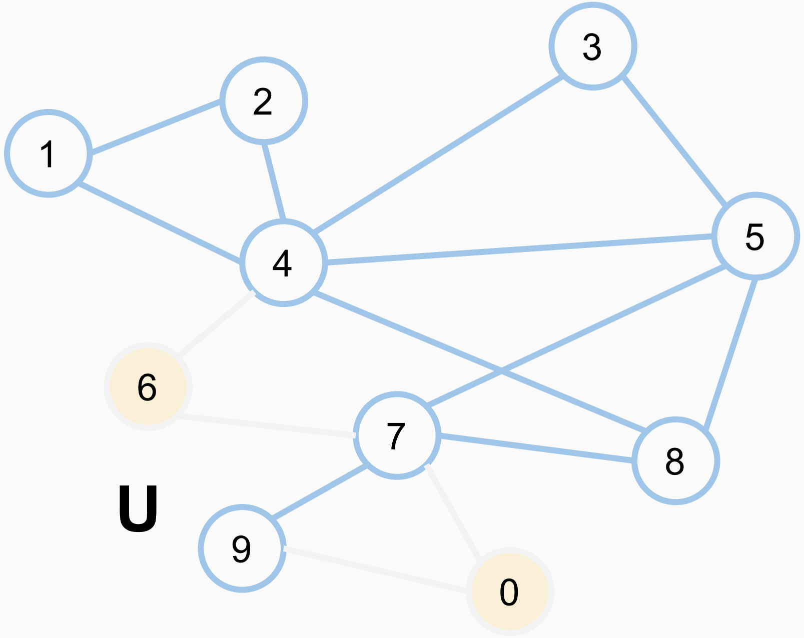


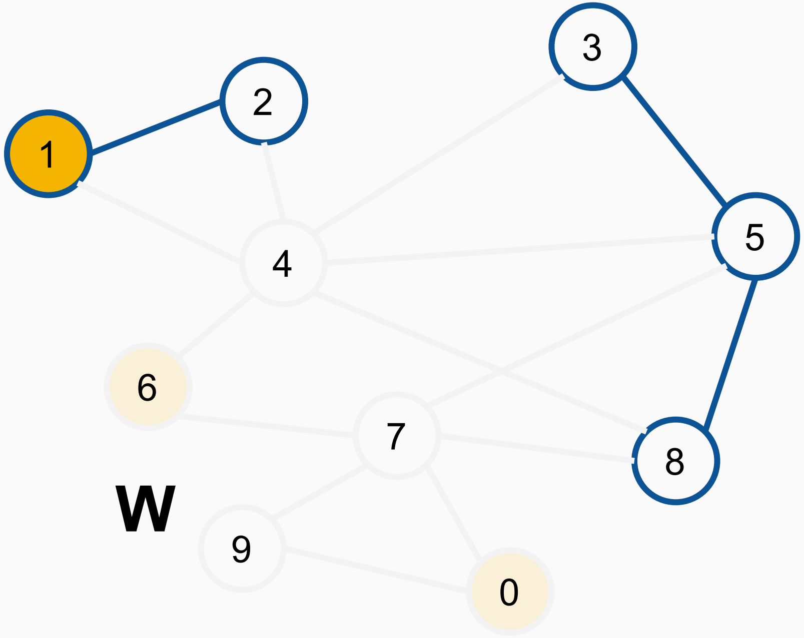
i = 1

Cores: 1

Algoritmo de Johnson

Colorir v com a cor i



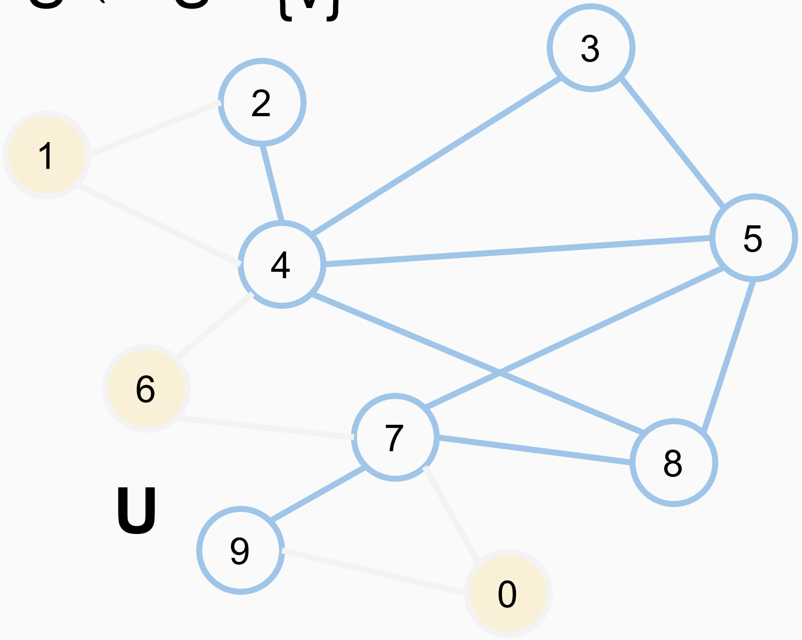


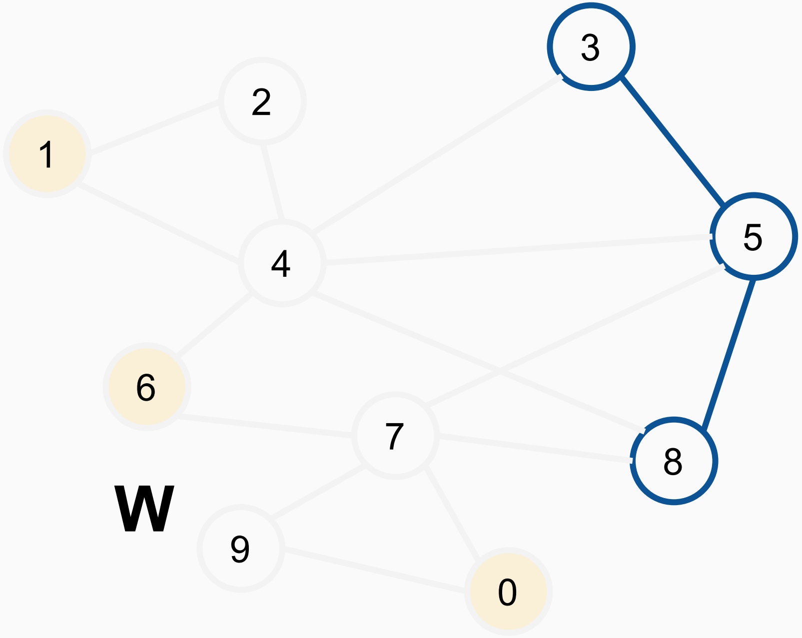
i = 1

Cores: 1

Algoritmo de Johnson

$W \leftarrow W - \{v\} - N(v)$
$U \leftarrow U - \{v\}$





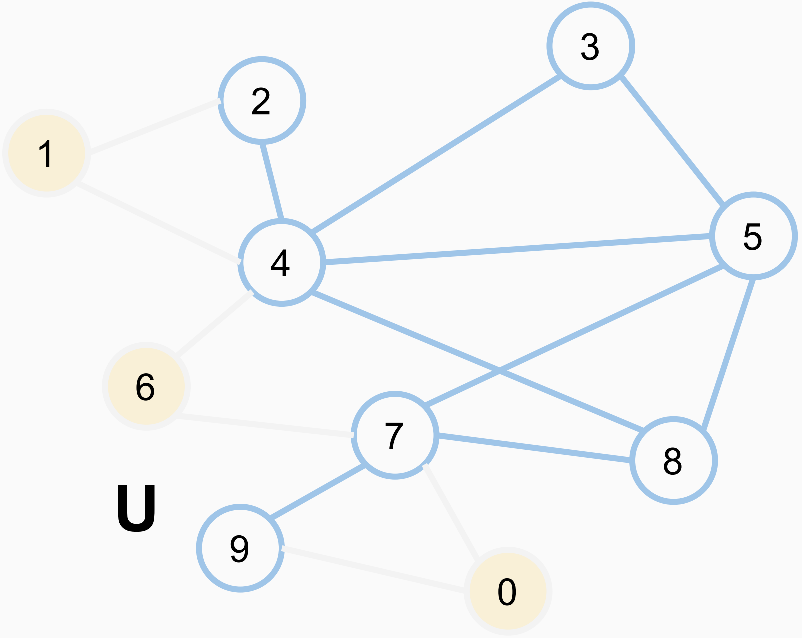
i = 1

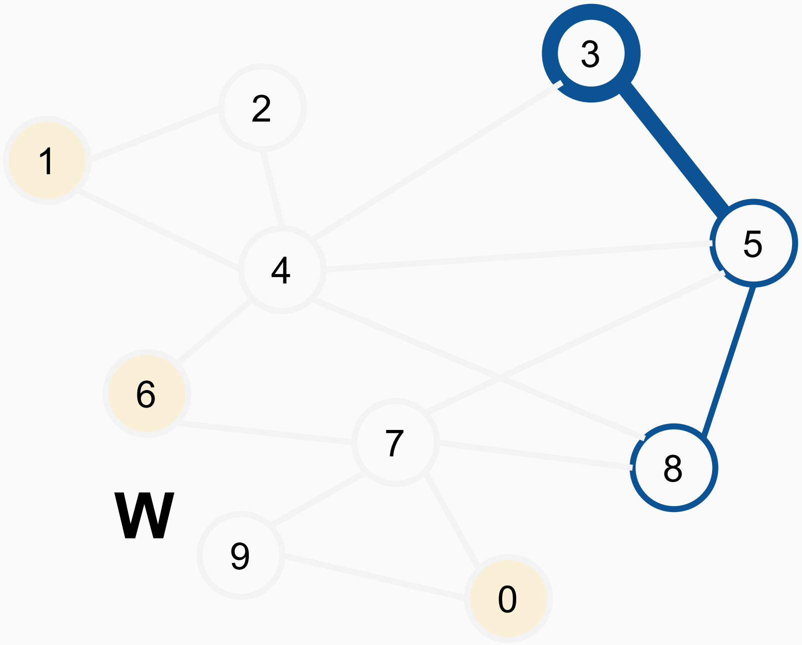
Cores: 1

Algoritmo de Johnson

W ≠ Ø? Sim

Obter o vértice de menor grau no subgrafo induzido por W



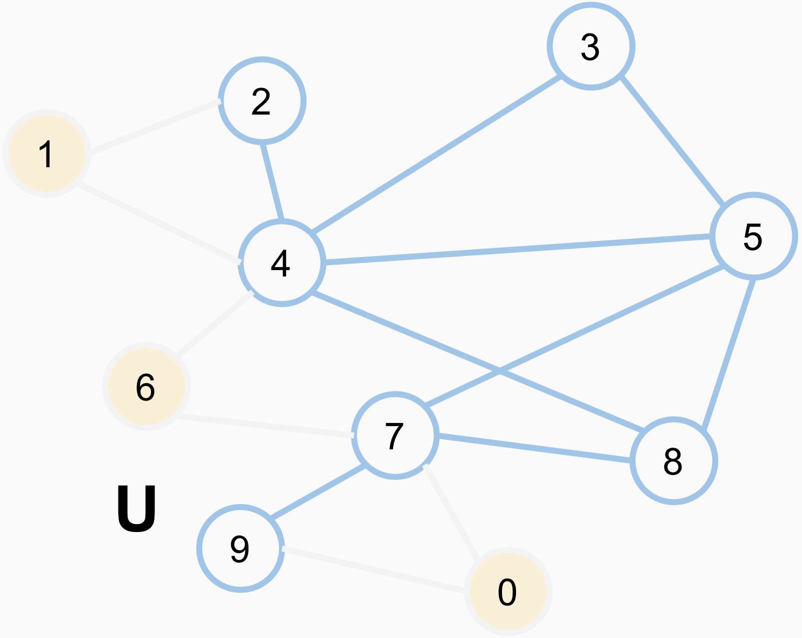


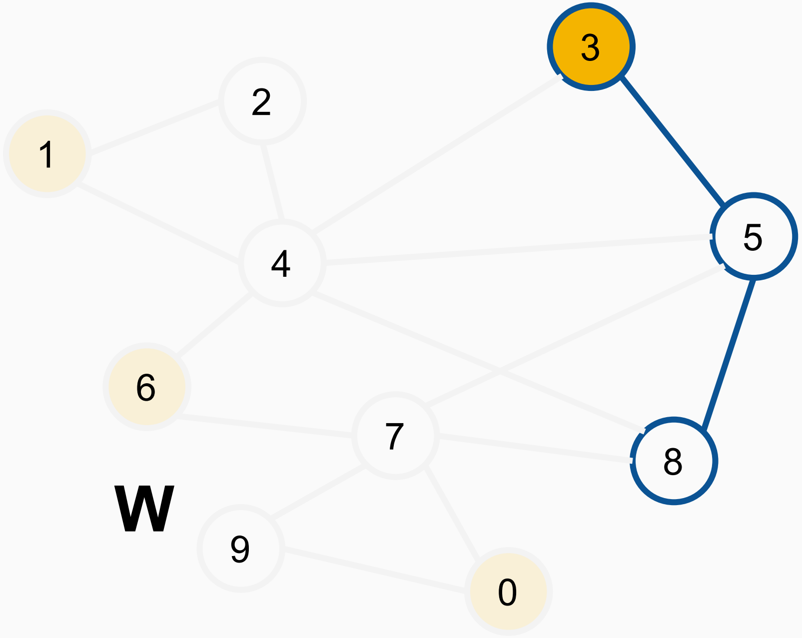
i = 1

Cores: 1

Algoritmo de Johnson

Colorir v com a cor i



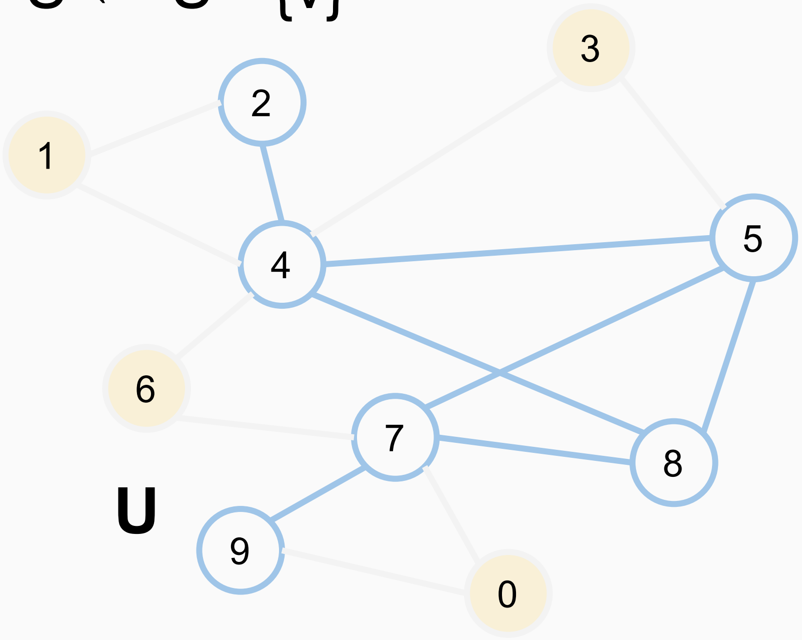


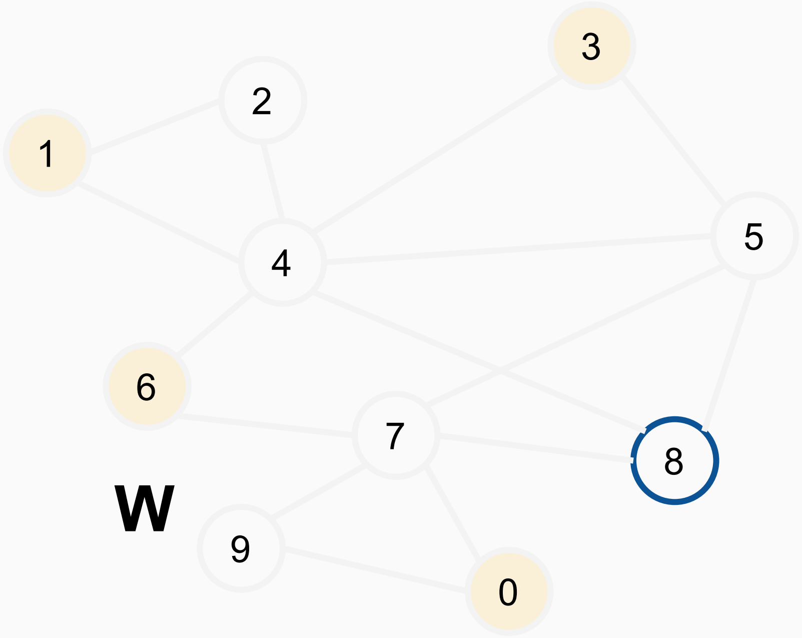
i = 1

Cores: 1

Algoritmo de Johnson

$W \leftarrow W - \{v\} - N(v)$
$U \leftarrow U - \{v\}$



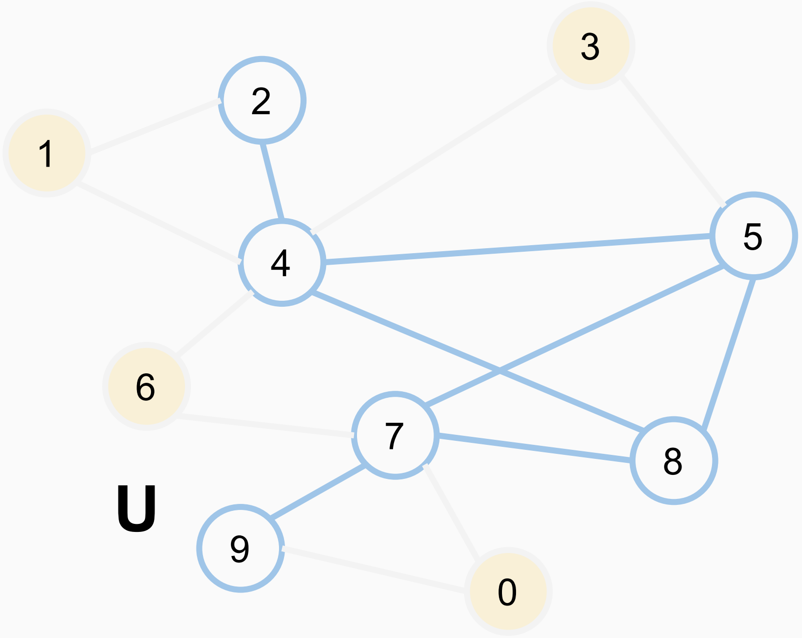


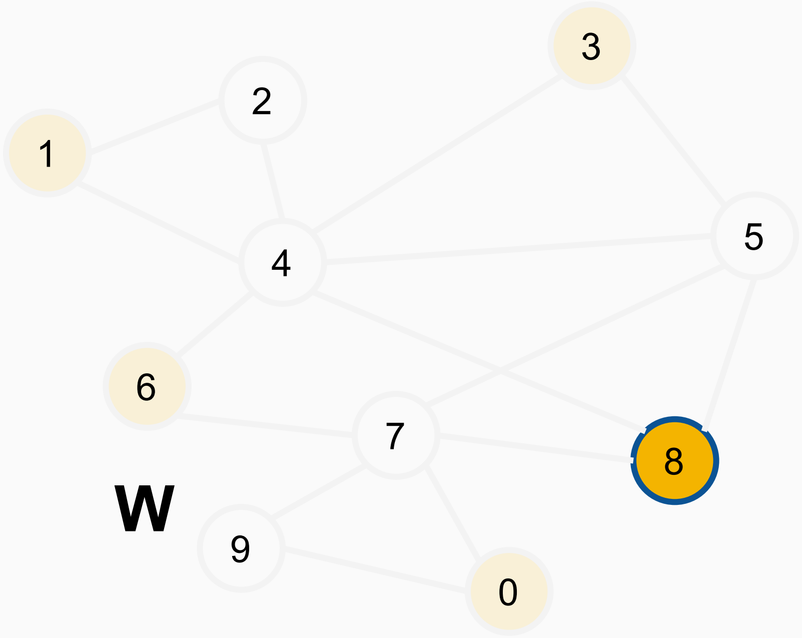
i = 1

Cores: 1

Algoritmo de Johnson

Por último...





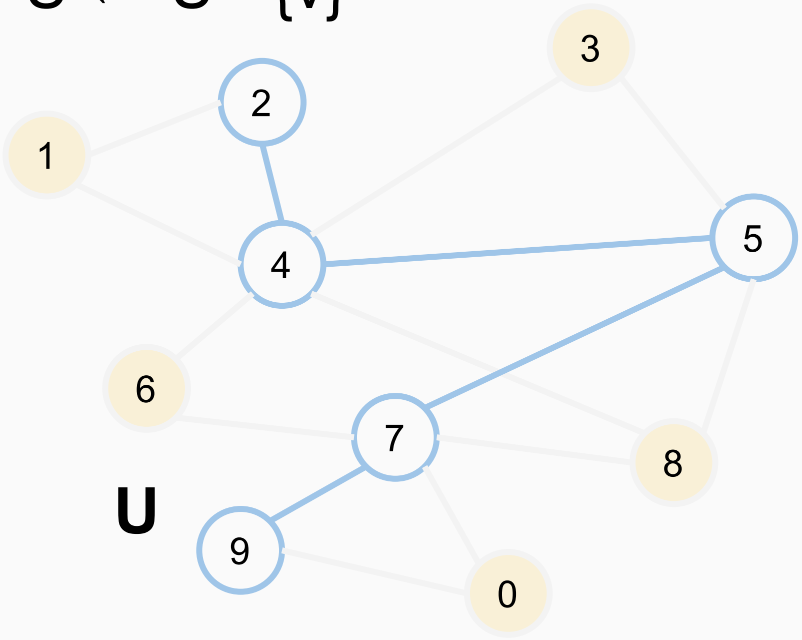
i = 1

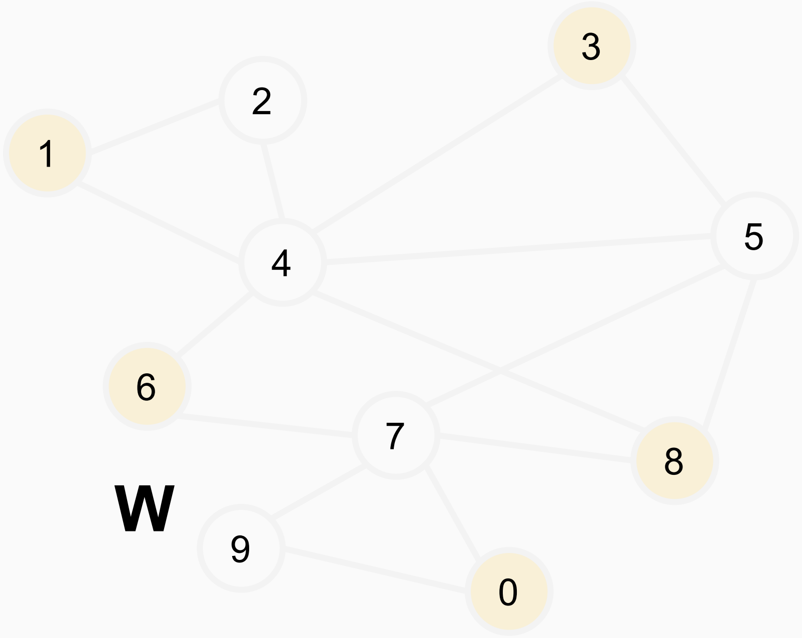
Cores: 1

Algoritmo de Johnson

$W \leftarrow W - \{v\} - N(v)$

$U \leftarrow U - \{v\}$





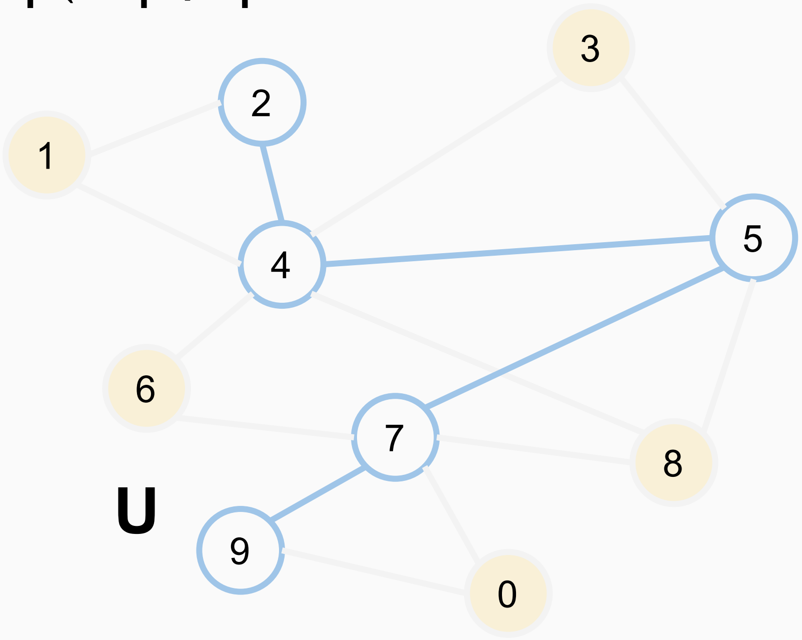
i = 1

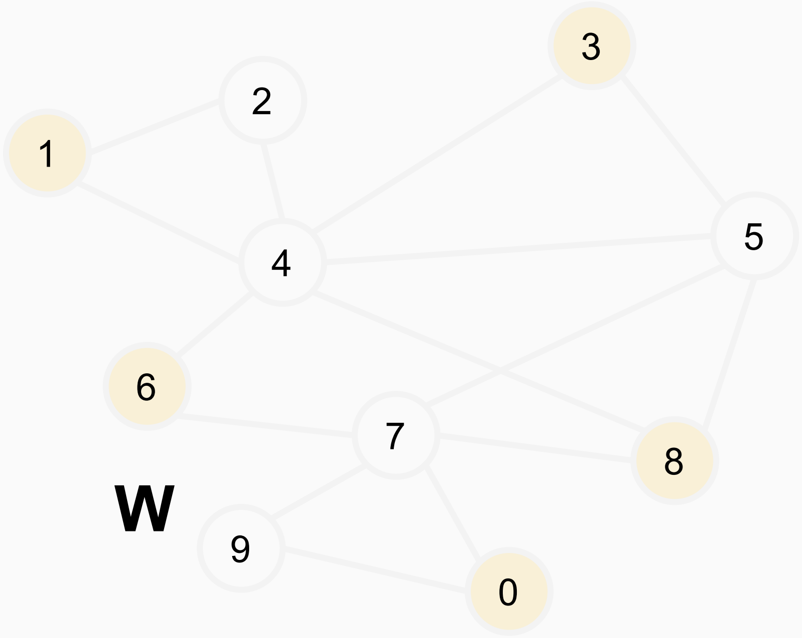
Cores: 1

Algoritmo de Johnson

W ≠ Ø? Não

i ← i + 1



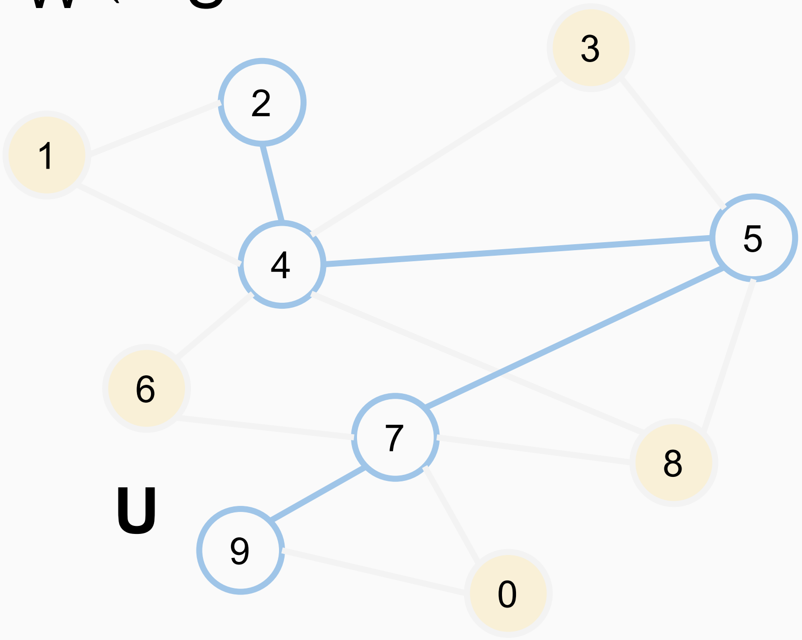


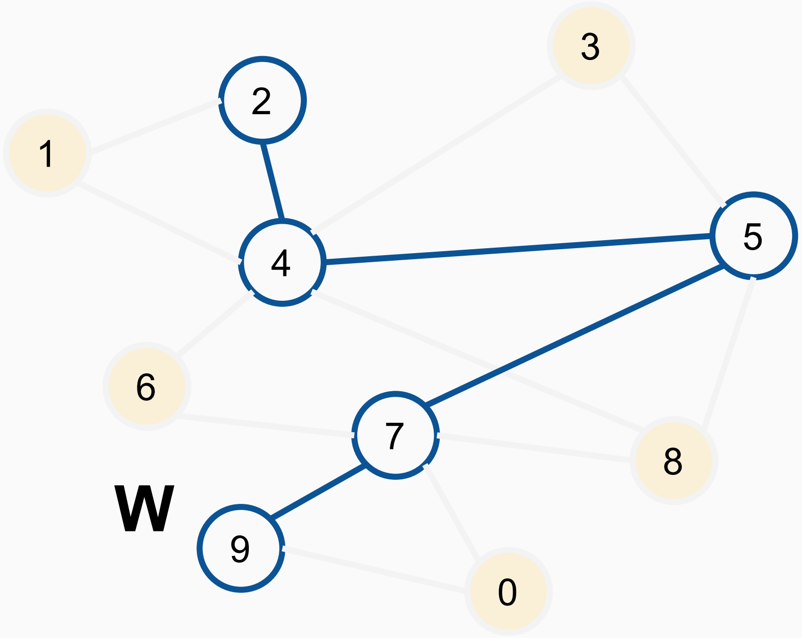
i = 2

Cores: 1

Algoritmo de Johnson

$U \neq \emptyset$? Sim
$W \leftarrow U$





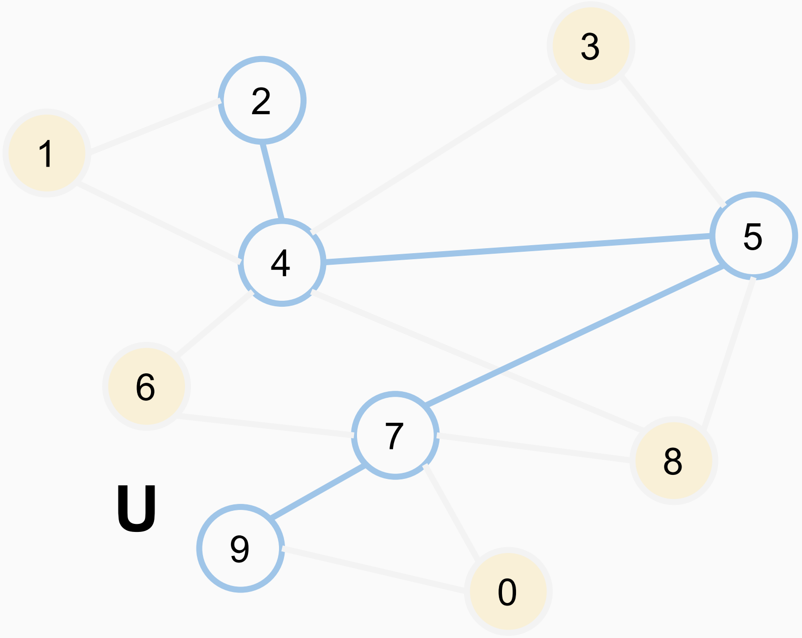
i = 2

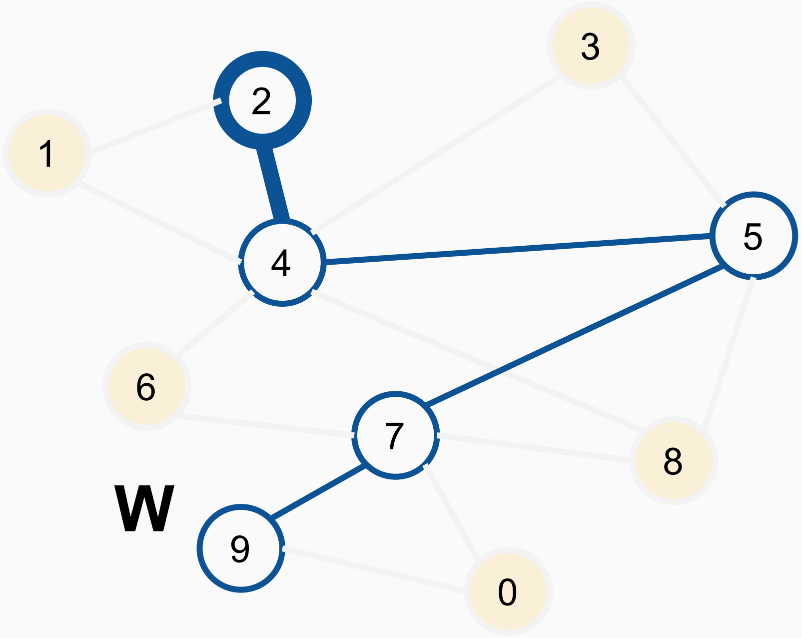
Cores: 1

Algoritmo de Johnson

W ≠ Ø? Sim

Obter o vértice de menor grau no subgrafo induzido por W



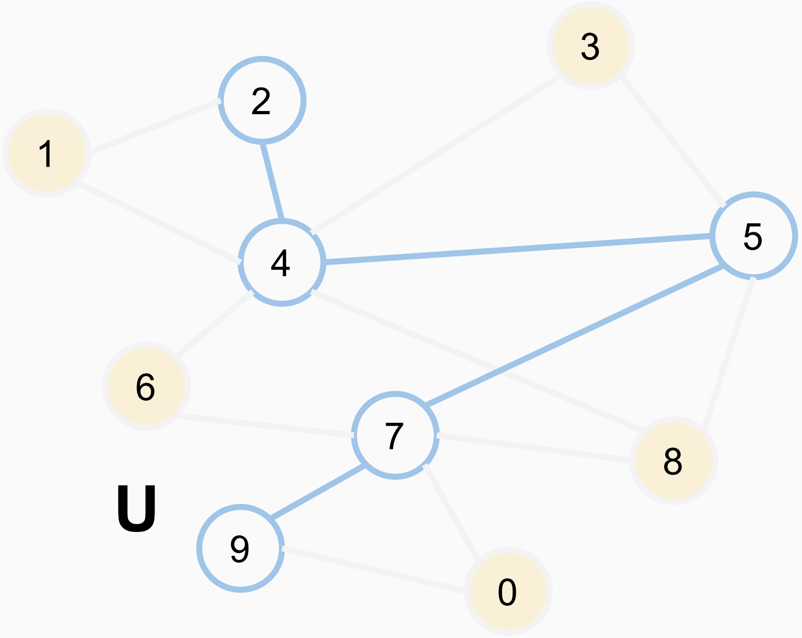


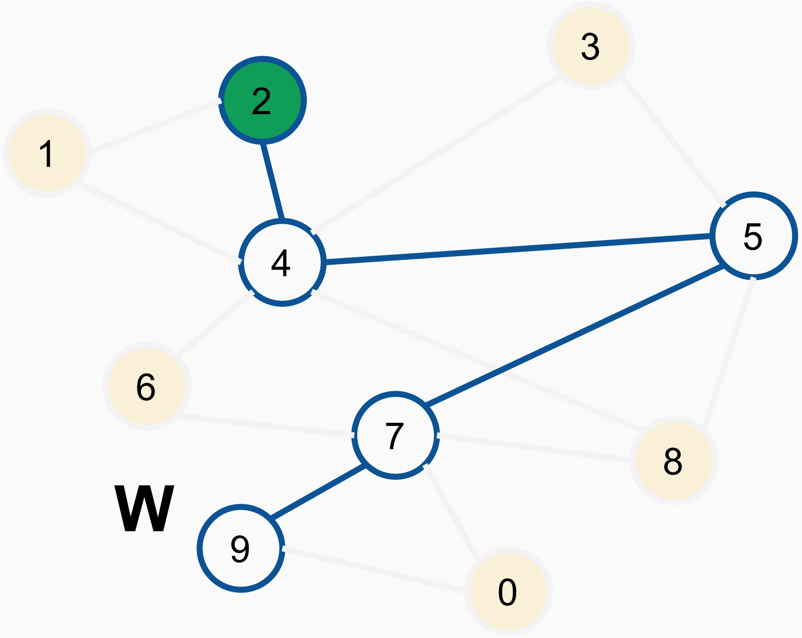
i = 2

Cores: 1

Algoritmo de Johnson

Colorir v com a cor i



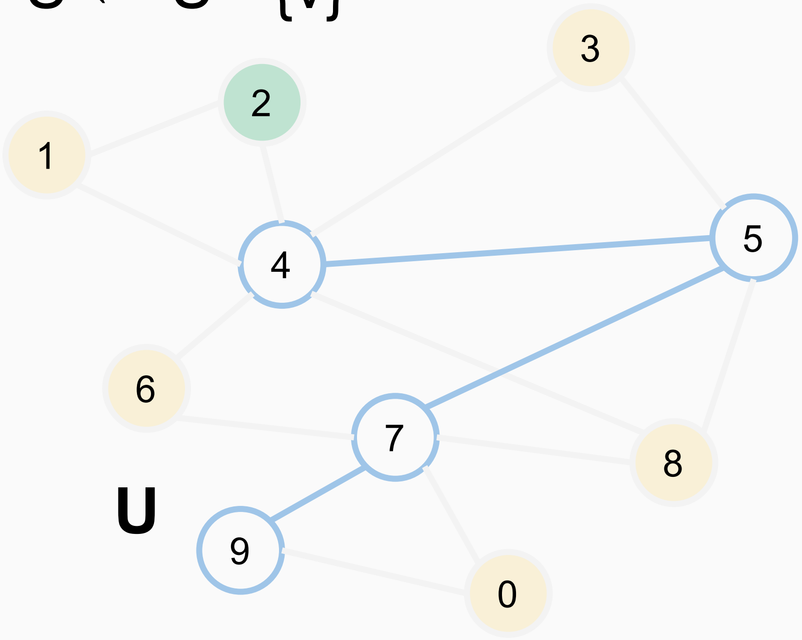


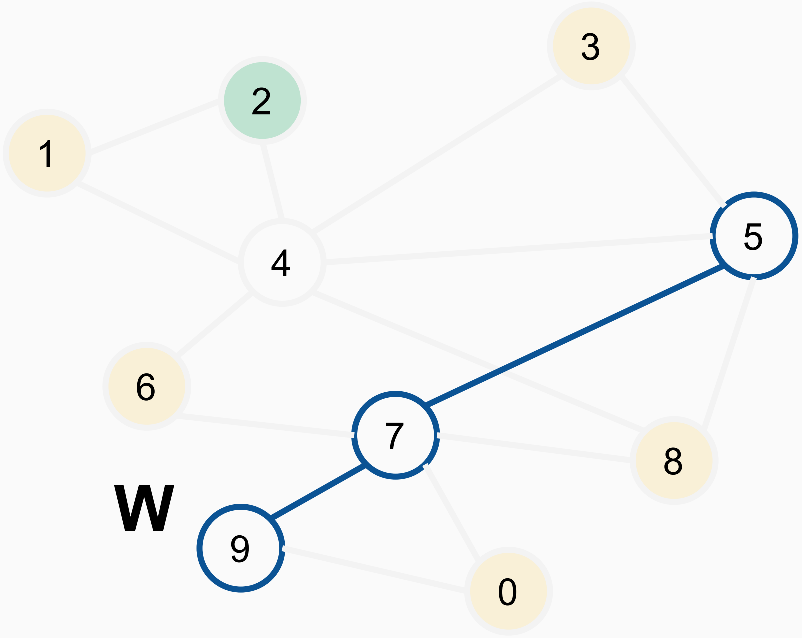
i = 2

Cores: 2

Algoritmo de Johnson

W ← W - {v} - N(v)
U ← U - {v}



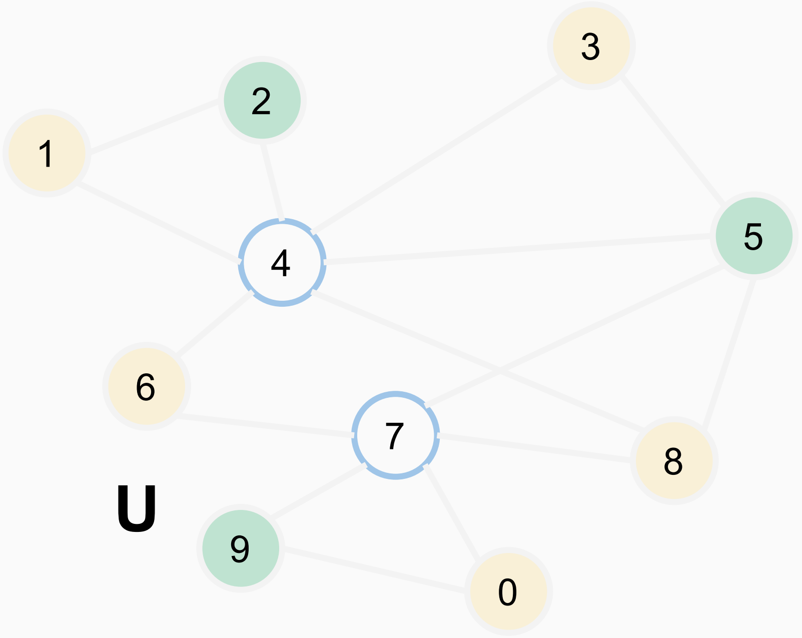


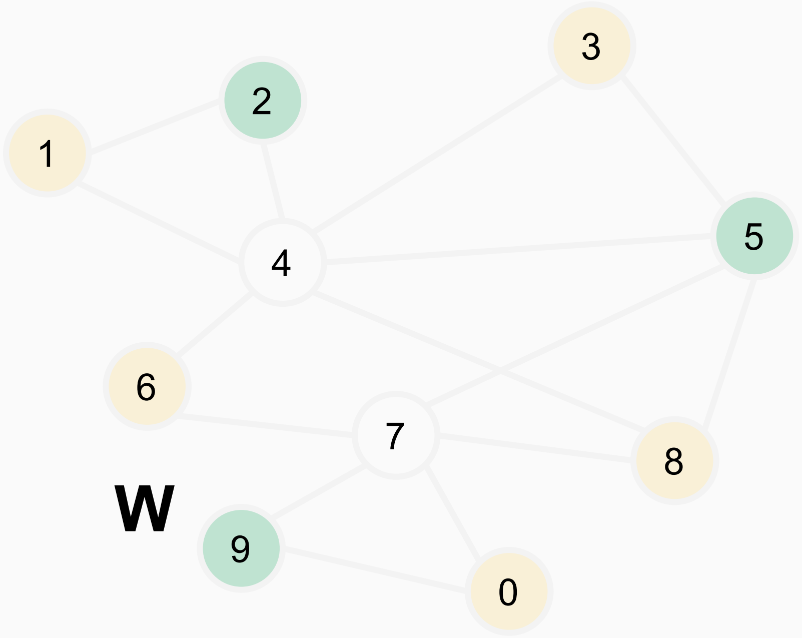
i = 2

Cores: 2

Algoritmo de Johnson

Continuando para esse W...



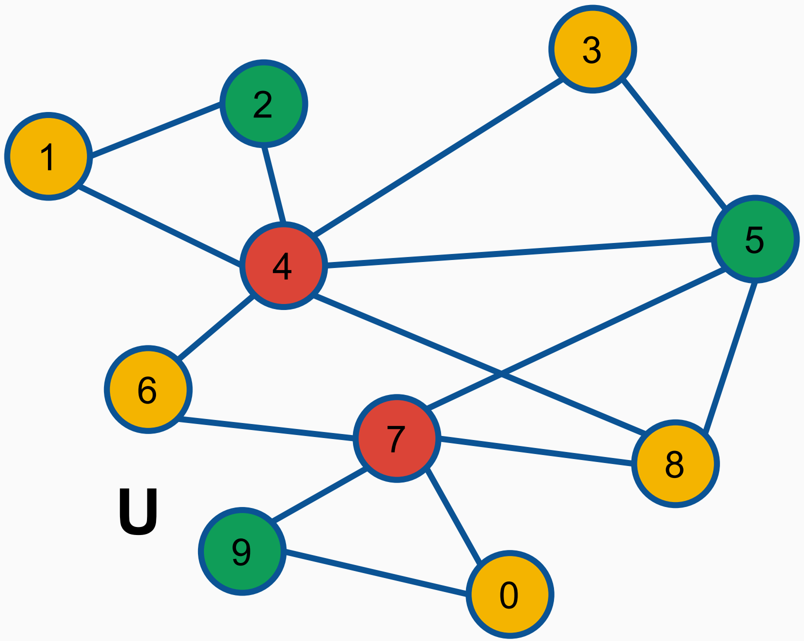


i = 2

Cores: 2

Algoritmo de Johnson

Terminando a execução...



i = 3

Cores: 3

Prova da Aproximação do Algoritmo de Johnson

Teorema 3.1: Um grafo k-colorável possui um conjunto independente σ de tamanho $\lceil |V|/k \rceil$.

Teorema 3.2: Cada vértice v do conjunto independente σ possui $|N(v)| \leq V - \lceil |V|/k \rceil$ vizinhos.

Algoritmo de Johnson(G):

$i \leftarrow 1$.

$U \leftarrow V$. (U é o conjunto de vértices ainda sem cor)

Enquanto $U \neq \varnothing$ faça:

  $W \leftarrow U$.

  Enquanto $W \neq \varnothing$ faça:

    Seja v o vértice com menor grau no subgrafo induzido por W.

    Colorir v com a cor i.

    $W \leftarrow W - \{v\} - N(v)$.

    $U \leftarrow U - \{v\}$.

  $i \leftarrow i + 1$.

Devolva a coloração.

Prova da Aproximação do Algoritmo de Johnson

$i \leftarrow 1$.

$U \leftarrow V$. (U é o conjunto de vértices ainda sem cor)

Enquanto $U \neq \varnothing$ faça:

    $W \leftarrow U$.

    Enquanto $W \neq \varnothing$ faça:

        Seja v o vértice com menor grau no subgrafo induzido por W.

        Colorir v com a cor i.

        $W \leftarrow W - \{v\} - N(v)$.

        $U \leftarrow U - \{v\}$.

    $i \leftarrow i + 1$.

Devolva a coloração.

Seja H este subgrafo, de forma que W seja o conjunto de vértices de H.

Como H é subgrafo de um grafo k-colorável, H também é k-colorável.

Prova da Aproximação do Algoritmo de Johnson

Dos teoremas 3.1 e 3.2, temos que H possui um conjunto independente de tamanho pelo menos

$|W|/k$,

sendo que cada vértice neste conjunto possui grau no máximo

$|W| - |W|/k = |W|(k-1)/k$

Logo, o grau mínimo de H é no máximo $|W|(k-1)/k$, e assim, pelo menos

$|W| - |W|(k-1)/k = |W|/k$

vértices estarão em W no começo da próxima iteração.

Prova da Aproximação do Algoritmo de Johnson

O laço interno só termina quando W fica vazio, assim, pelo menos $\lceil \log_k |W| \rceil$ iterações devem ser executadas:

Iteração 0: $|W|$
Iteração 1: $|W|/k$
Iteração 2: $|W|/k/k = |W|/k^2$
Iteração 3: $|W|/k^3$
…
Iteração n: $|W|/k^n = 1$

Para $k^n = |W|$, precisamos de $n = \log_k |W|$ iterações.

Prova da Aproximação do Algoritmo de Johnson

Deste modo, no final do laço interno, teremos $|\{v \mid v \in W \land cor(v) = i\}| \geq \lceil \log_k |W| \rceil$.

Ou seja, o número de vértices coloridos com a cor i será pelo menos $\lceil \log_k |U| \rceil$, sendo U o conjunto de vértices sem cor antes do começo do laço.

Prova da Aproximação do Algoritmo de Johnson

Para completar a prova, precisamos analisar o tamanho de U no começo de uma iteração qualquer do laço externo.

Quando $|U| \geq |V|/\log_k|V|$, temos que
$\lceil \log_k|U| \rceil \geq \log_k|U| \geq \log_k(|V|/\log_k|V|) > \log_k\sqrt{|V|} = ½ \log_k|V|$.

Assim, o tamanho de U diminui pelo menos $½ \log_k|V|$ a cada iteração. Desta maneira, quando $|U|$ se tornar menor que $|V|/\log_k|V|$, o algoritmo não terá usado mais do que $2|V|/\log_k|V|$ cores.

Prova da Aproximação do Algoritmo de Johnson

Para completar a prova, precisamos analisar o tamanho de U no começo de uma iteração qualquer do laço externo.

Quando $|U| < |V|/\log_k|V|$, é claro que o algoritmo utilizará no máximo $|V|/\log_k|V|$ cores. Segue então que o algoritmo de Johnson utiliza no máximo $3|V|/\log_k|V|$ cores.

Para o nosso caso, k=3, portanto o algoritmo de Johnson usa no máximo **$3|V|/\log_3|V|$ cores**.

Algoritmo de Wigderson

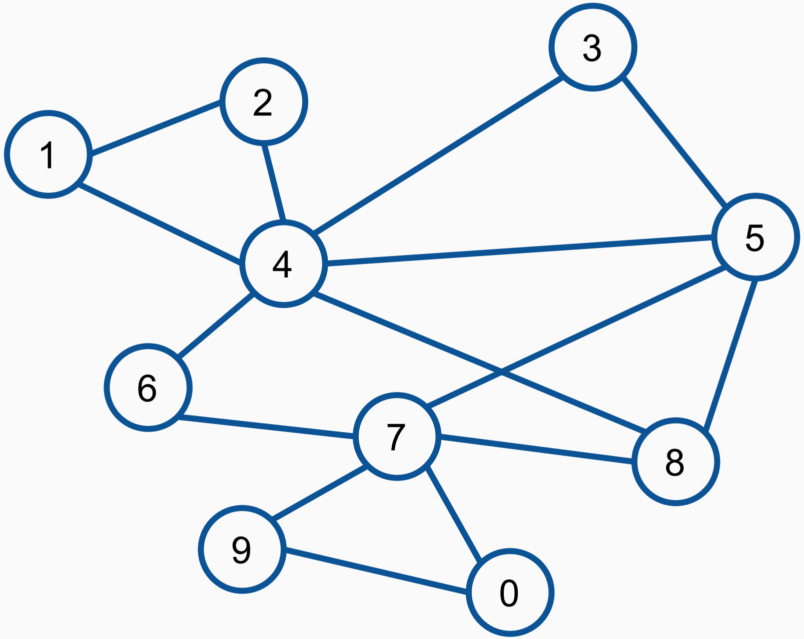
Teorema 4.1: Para um grafo 3-colorável, a vizinhança de cada vértice pode ser colorida com duas cores.

Teorema 4.2: Qualquer grafo cujo vértice de maior grau é $\Delta(G)$ pode ser colorido com no máximo $\Delta(G)+1$ cores em tempo polinomial.

Teorema 4.3: Todo grafo bipartido pode ser colorido em tempo polinomial.

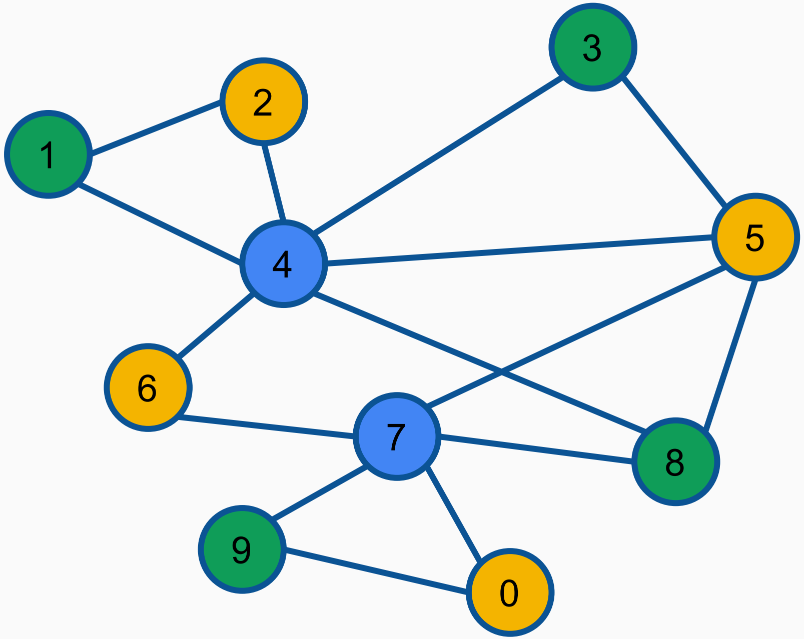
Algoritmo de Wigderson

Instância de exemplo

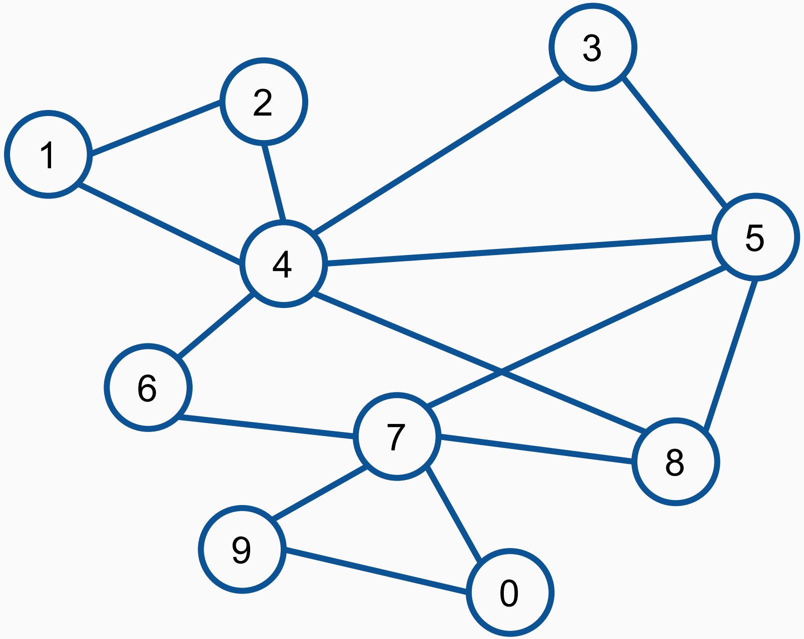


Algoritmo de Wigderson

Coloração ótima



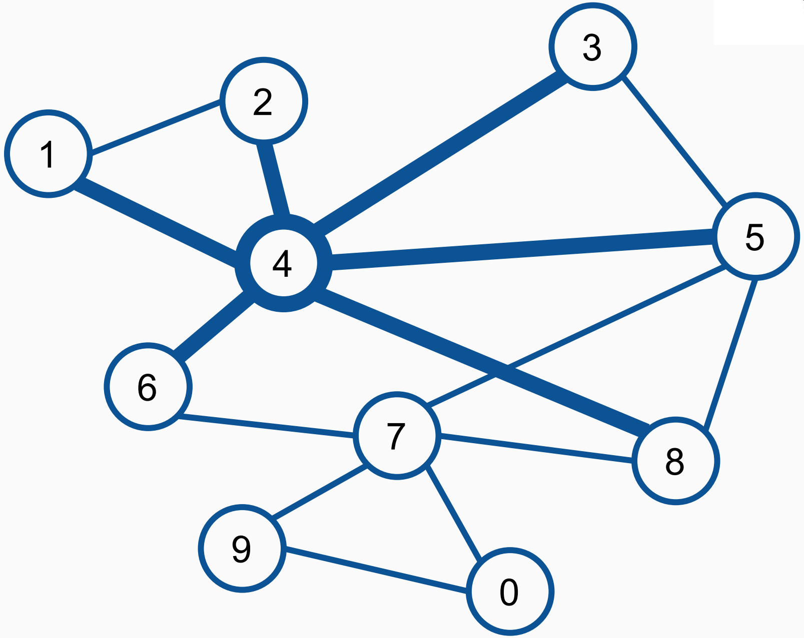
Algoritmo de Wigderson



Cores: 0

Algoritmo de Wigderson

$\Delta(G) \geq \lceil \sqrt{n} \rceil$?

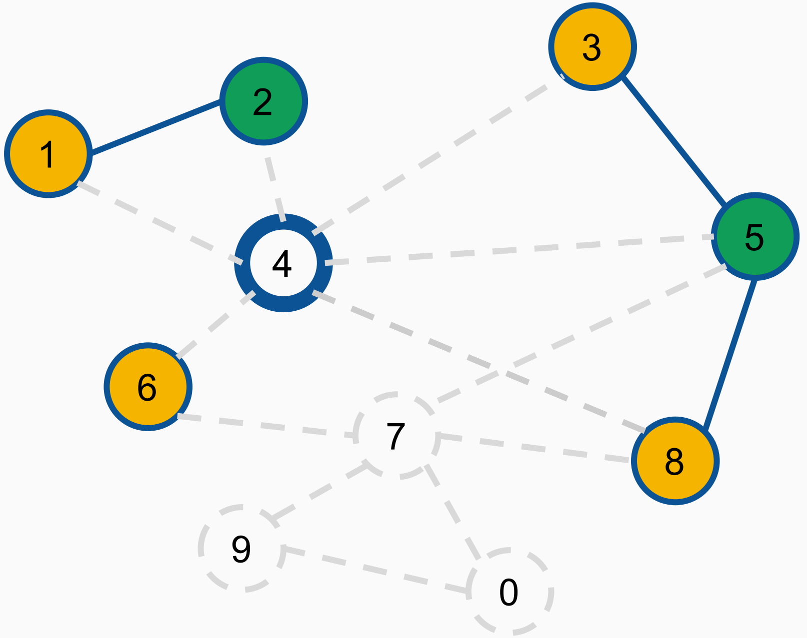


i = 1

Cores: 0

Algoritmo de Wigderson

Obter o subgrafo **H** induzido da vizinhança do vértice de maior grau e colorir com as cores **i** e **i+1**

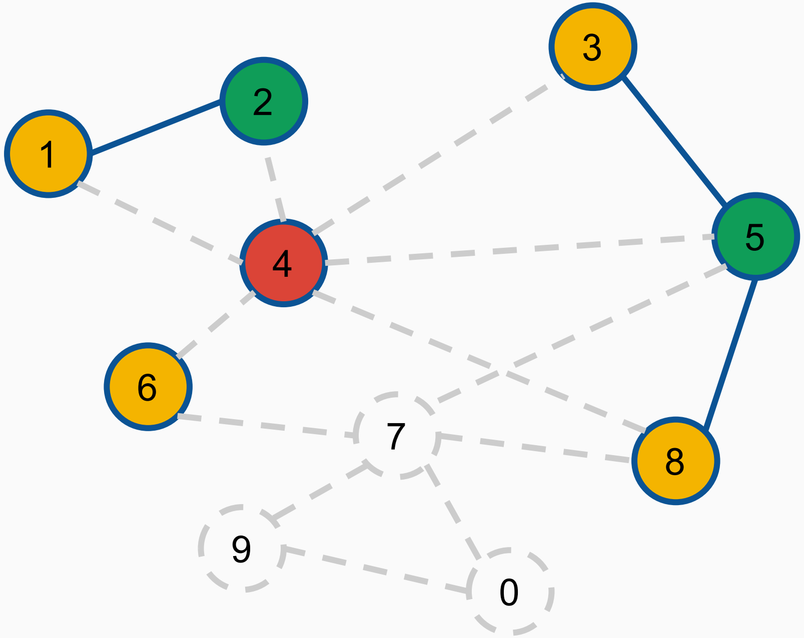


i = 1

Cores: 2

Algoritmo de Wigderson

Colorir o vértice de maior grau com a cor **i+2**

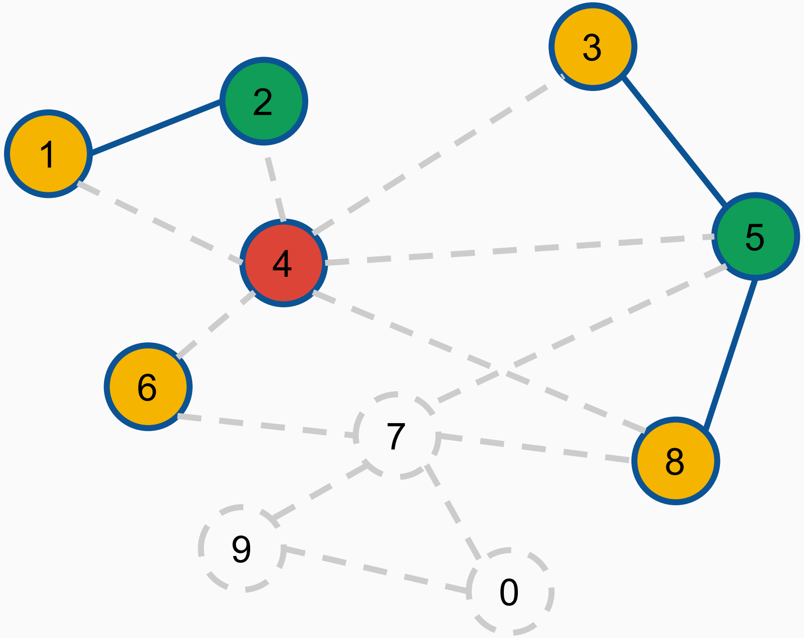


i = 1

Cores: 3

Algoritmo de Wigderson

$i \leftarrow i + 2$

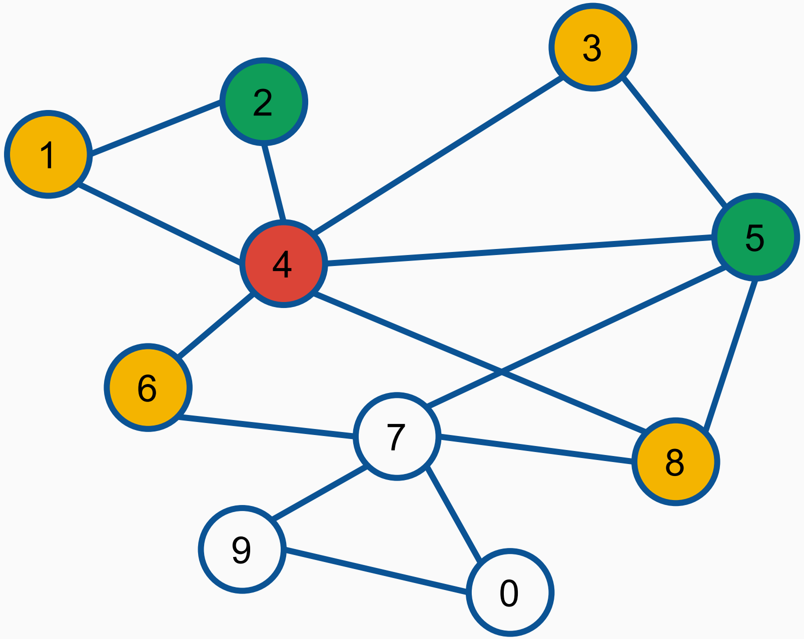


i = 3

Cores: 3

Algoritmo de Wigderson

Remove, do grafo G, o vértice de maior grau e seu vizinhos

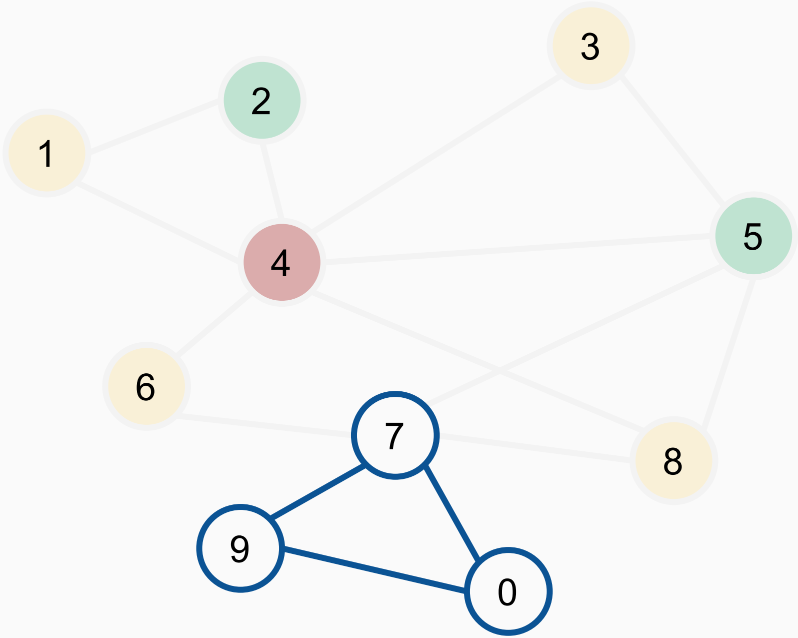


i = 3

Cores: 3

Algoritmo de Wigderson

Remove, do grafo G, o vértice de maior grau e seu vizinhos

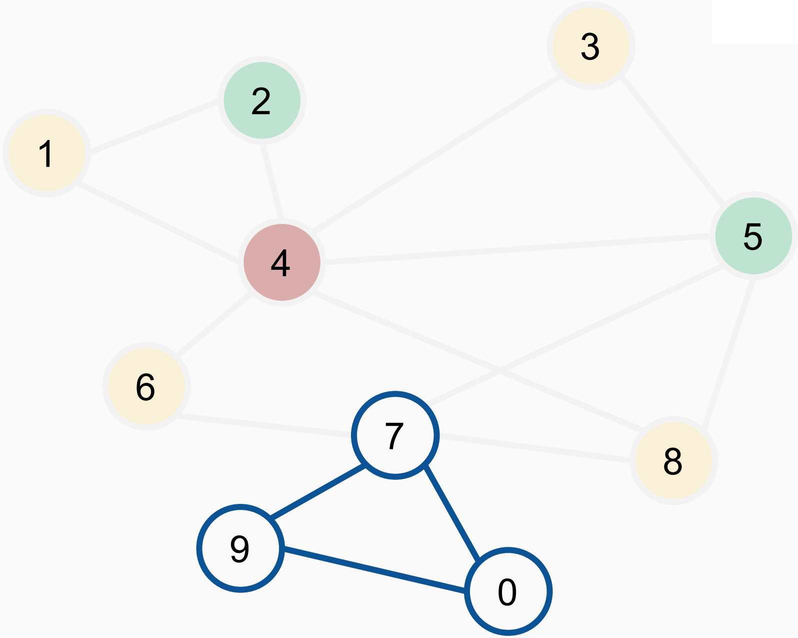


i = 3

Cores: 3

Algoritmo de Wigderson

$$\Delta(G) \geq \lceil \sqrt{n} \rceil \, ?$$

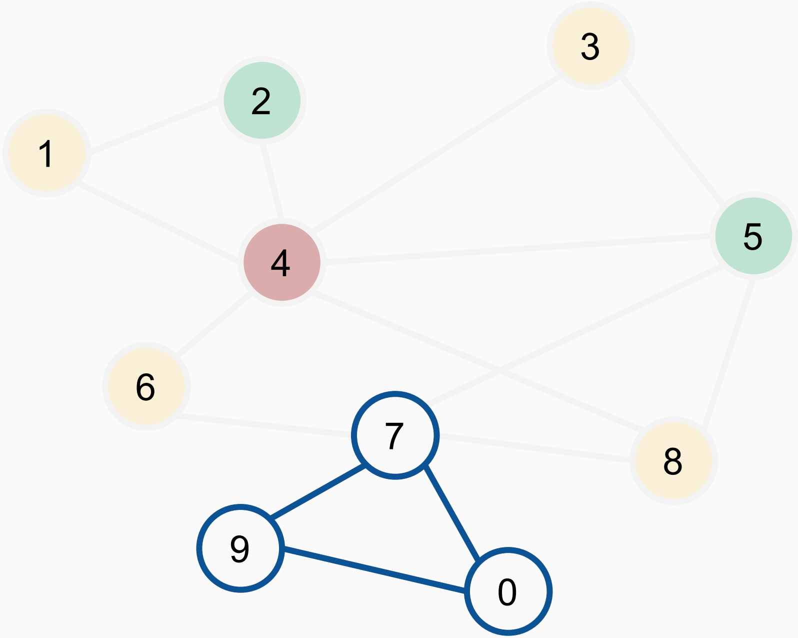


i = 3

Cores: 3

Algoritmo de Wigderson

Colorir os vértices restantes com $\Delta(G)+1$ cores

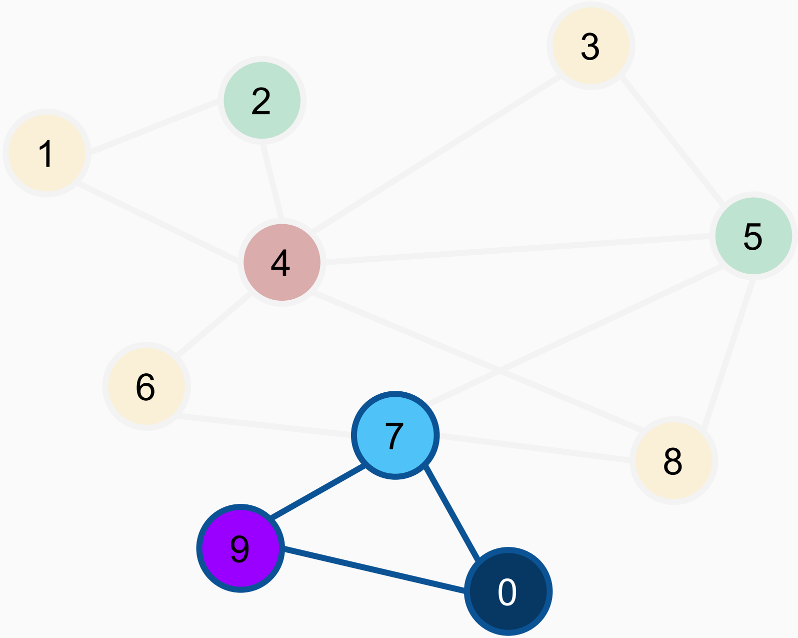


i = 3

Cores: 3

Algoritmo de Wigderson

Colorir os vértices restantes com $\Delta(G)+1$ cores

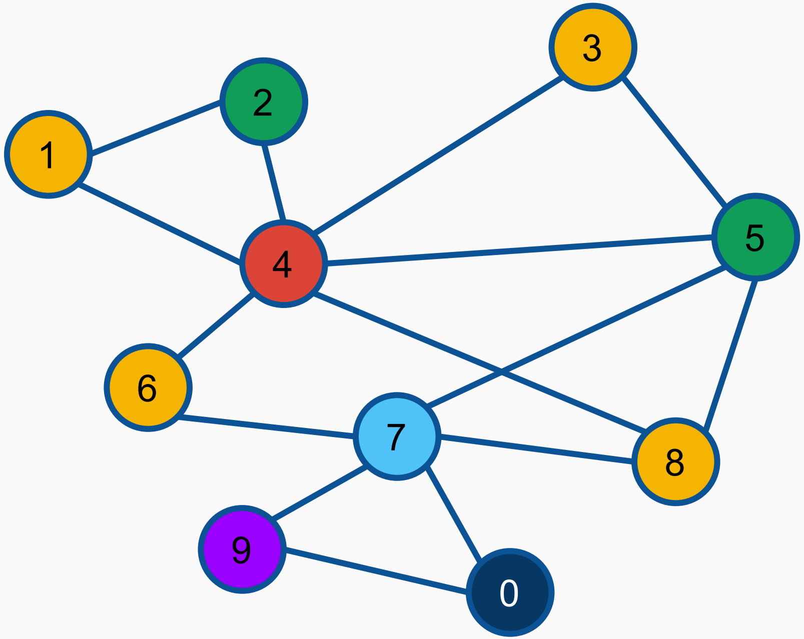


i = 3

Cores: 6

Algoritmo de Wigderson

Resultado



i = 3

Cores: 6

Algoritmo de Wigderson(G):

$n \leftarrow |V|$.

$i \leftarrow 1$.

Enquanto $\Delta(G) \geq \lceil \sqrt{n} \rceil$ faça:

    Seja v um vértice com o maior grau em G.

    $H \leftarrow$ o subgrafo de G induzido por $N_G(v)$.

    Colorir H com duas cores i e $i+1$.

    Colorir v com a cor $i+2$.

    $i \leftarrow i+2$.

    $G \leftarrow$ o subgrafo de G sem os vértices $N_G(v) \cup \{v\}$.

Colorir o grafo G com no máximo $\Delta(G)+1$ cores usando o teorema 5.2.

Devolva a coloração.

Prova da Aproximação do Algoritmo de Wigderson

$n \leftarrow |V|$.

$i \leftarrow 1$.

Enquanto $\Delta(G) \geq \lceil \sqrt{n} \rceil$ faça:

    Seja v um vértice com o maior grau em G.

    $H \leftarrow$ o subgrafo de G induzido por $N_G(v)$.

    Colorir H com duas cores i e $i+1$.

    Colorir v com a cor $i+2$.

    $i \leftarrow i+2$.

    $G \leftarrow$ o subgrafo de G sem os vértices $N_G(v) \cup \{v\}$.

Colorir o grafo G com no máximo $\Delta(G)+1$ cores usando o teorema 5.2.

Devolva a coloração.

Prova da Aproximação do Algoritmo de Wigderson

```
n ← |V| .
i ← 1 .
Enquanto Δ(G) ≥ ⌈√n⌉ faça:
        Seja v um vértice com o maior grau em G .
        H ← o subgrafo de G induzido por N_G(v) .
        Colorir H com duas cores i e i+1 .
        Colorir v com a cor i+2 .
        i ← i+2 .
        G ← o subgrafo de G sem os vértices N_G(v) ∪ {v} .
Colorir o grafo G com no máximo Δ(G)+1 cores usando o teorema 5.2.
Devolva a coloração.
```

Para cada iteração são usadas 2 cores

Prova da Aproximação do Algoritmo de Wigderson

1 $n \leftarrow |V|$.

2 $i \leftarrow 1$.

3 Enquanto $\Delta(G) \geq \lceil \sqrt{n} \rceil$ faça:

4 Seja v um vértice com o maior grau em G.

5 $H \leftarrow$ o subgrafo de G induzido por $N_G(v)$.

6 Colorir H com duas cores i e $i+1$.

7 Colorir v com a cor $i+2$.

8 $i \leftarrow i+2$.

9 $G \leftarrow$ o subgrafo de G sem os vértices $N_G(v) \cup \{v\}$.

10 Colorir o grafo G com no máximo $\Delta(G)+1$ cores usando o teorema 5.2.

11 Devolva a coloração.

Linhas 3 até 9 são executadas no máximo √n vezes

Prova da Aproximação do Algoritmo de Wigderson

$n \leftarrow |V|$.

$i \leftarrow 1$.

Enquanto $\Delta(G) \geq \lceil \sqrt{n} \rceil$ faça:

  Seja v um vértice com o maior grau em G .

  $H \leftarrow$ o subgrafo de G induzido por $N_G(v)$.

  Colorir H com duas cores i e $i+1$.

  Colorir v com a cor $i+2$.

  $i \leftarrow i+2$.

  $G \leftarrow$ o subgrafo de G sem os vértices $N_G(v) \cup \{v\}$.

Colorir o grafo G com no máximo $\Delta(G)+1$ cores usando o teorema 5.2.

Devolva a coloração.

Serão usadas no máximo $2\sqrt{n}$ cores

Prova da Aproximação do Algoritmo de Wigderson

$n \leftarrow |V|$.

$i \leftarrow 1$.

Enquanto $\Delta(G) \geq \lceil \sqrt{n} \rceil$ faça:

  Seja v um vértice com o maior grau em G.

  $H \leftarrow$ o subgrafo de G induzido por $N_G(v)$.

  Colorir H com duas cores i e $i+1$.

  Colorir v com a cor $i+2$.

  $i \leftarrow i+2$.

  $G \leftarrow$ o subgrafo de G sem os vértices $N_G(v) \cup \{v\}$.

Colorir o grafo G com no máximo $\Delta(G)+1$ cores usando o teorema 5.2.

Devolva a coloração.

Quando o grafo tiver vértices com grau menor que √n, eles serão coloridos com no máximo

Δ(G)+1 cores

Prova da Aproximação do Algoritmo de Wigderson

$n \leftarrow |V|$.

$i \leftarrow 1$.

Enquanto $\Delta(G) \geq \lceil \sqrt{n} \rceil$ faça:

  Seja v um vértice com o maior grau em G.

  $H \leftarrow$ o subgrafo de G induzido por $N_G(v)$.

  Colorir H com duas cores i e $i+1$.

  Colorir v com a cor $i+2$.

  $i \leftarrow i+2$.

  $G \leftarrow$ o subgrafo de G sem os vértices $N_G(v) \cup \{v\}$.

Colorir o grafo G com no máximo $\Delta(G)+1$ cores usando o teorema 5.2.

Devolva a coloração.

Assim, o algoritmo de Wigderson, para colorir um grafo 3-colorável, emprega no máximo <u>3√n</u> cores

Implementação

Os algoritmos de Johnson e Wigderson podem ser implementados com complexidade linear

Acesso em tempo constante ao vértice de menor ou maior grau logo após a remoção de vértices e arestas do grafo

Implementação

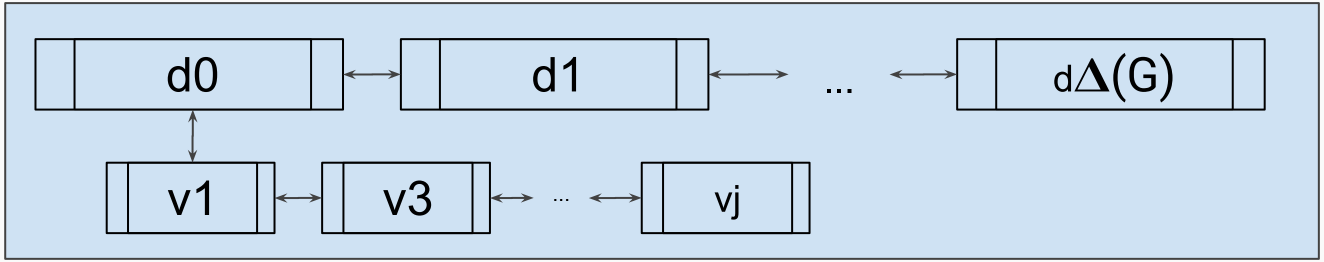
Os algoritmos de Johnson e Wigderson podem ser implementados com complexidade linear

Acesso em tempo constante ao vértice de menor ou maior grau logo após a remoção de vértices e arestas do grafo

Implementação

Os algoritmos de Johnson e Wigderson podem ser implementados com complexidade linear

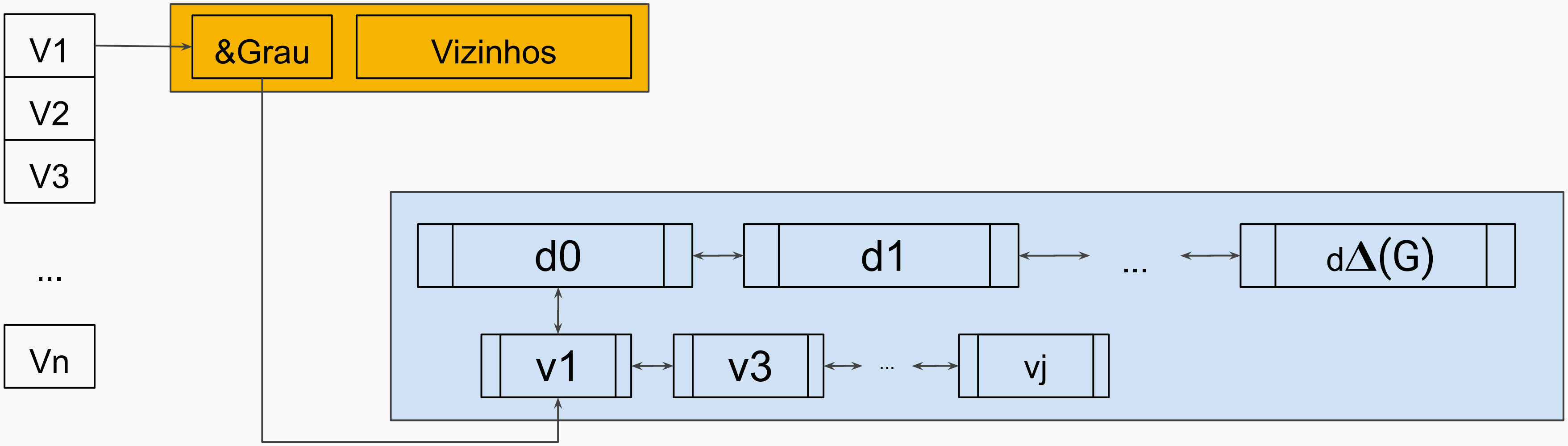
Acesso em tempo constante ao vértice de menor ou maior grau logo após a remoção de vértices e arestas do grafo



Implementação

Os algoritmos de Johnson e Wigderson podem ser implementados com complexidade linear

Acesso em tempo constante ao vértice de menor ou maior grau logo após a remoção de vértices e arestas do grafo



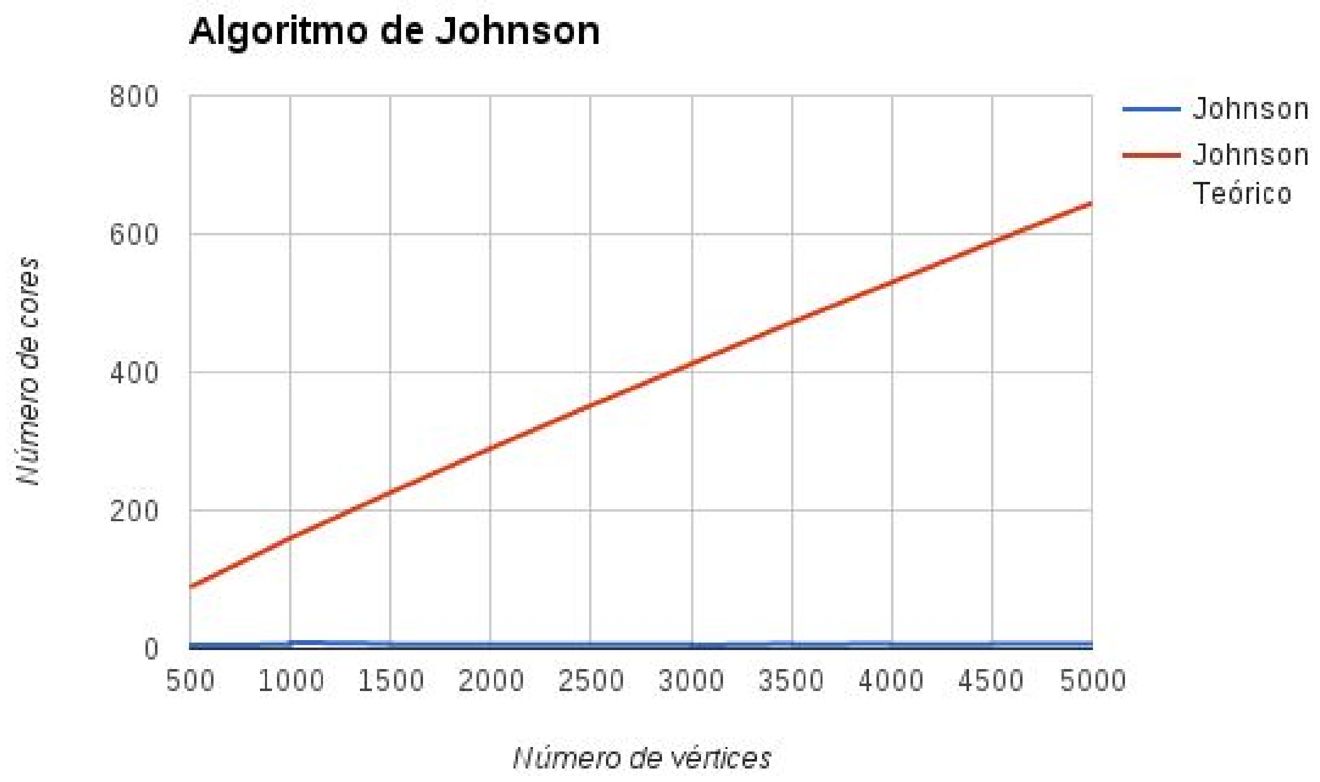
Resultados

Vértices (|V|)	Arestas (|E|)	# instâncias
500	3318 ± 31.7	5
1000	6651 ± 92.5	5
1500	14952 ± 130.8	5
2000	26562 ±115.9	5
2500	41532 ± 127.2	5
3000	44926 ± 125.4	5
3500	48992.8 ± 200	5
4000	64033 ± 258.2	5
4500	67581 ± 242.4	5
5000	83531 ± 223.7	5

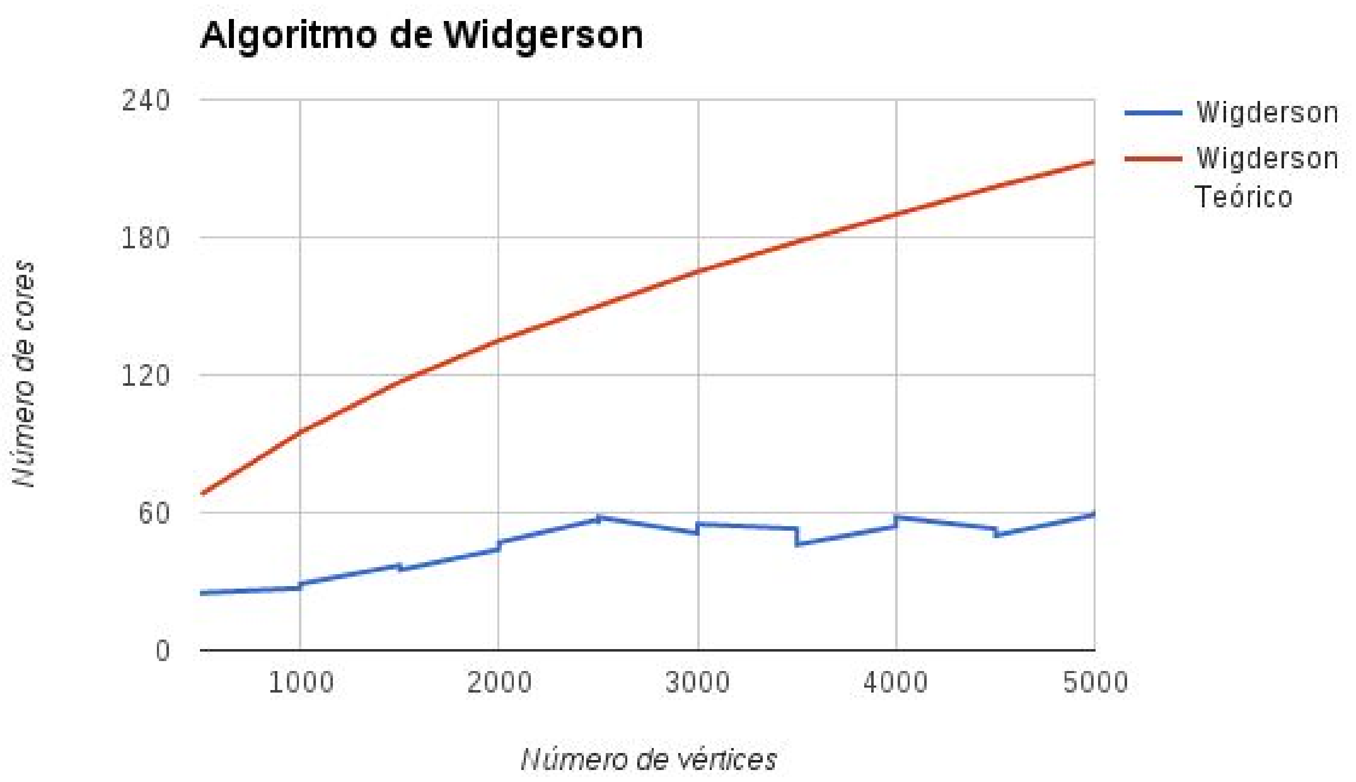
Resultados


Comportamento teórico dos algoritmos
200
150
100
50
0
10
50
100
500
Número de vértices
Johnson
Wigderson

Resultados

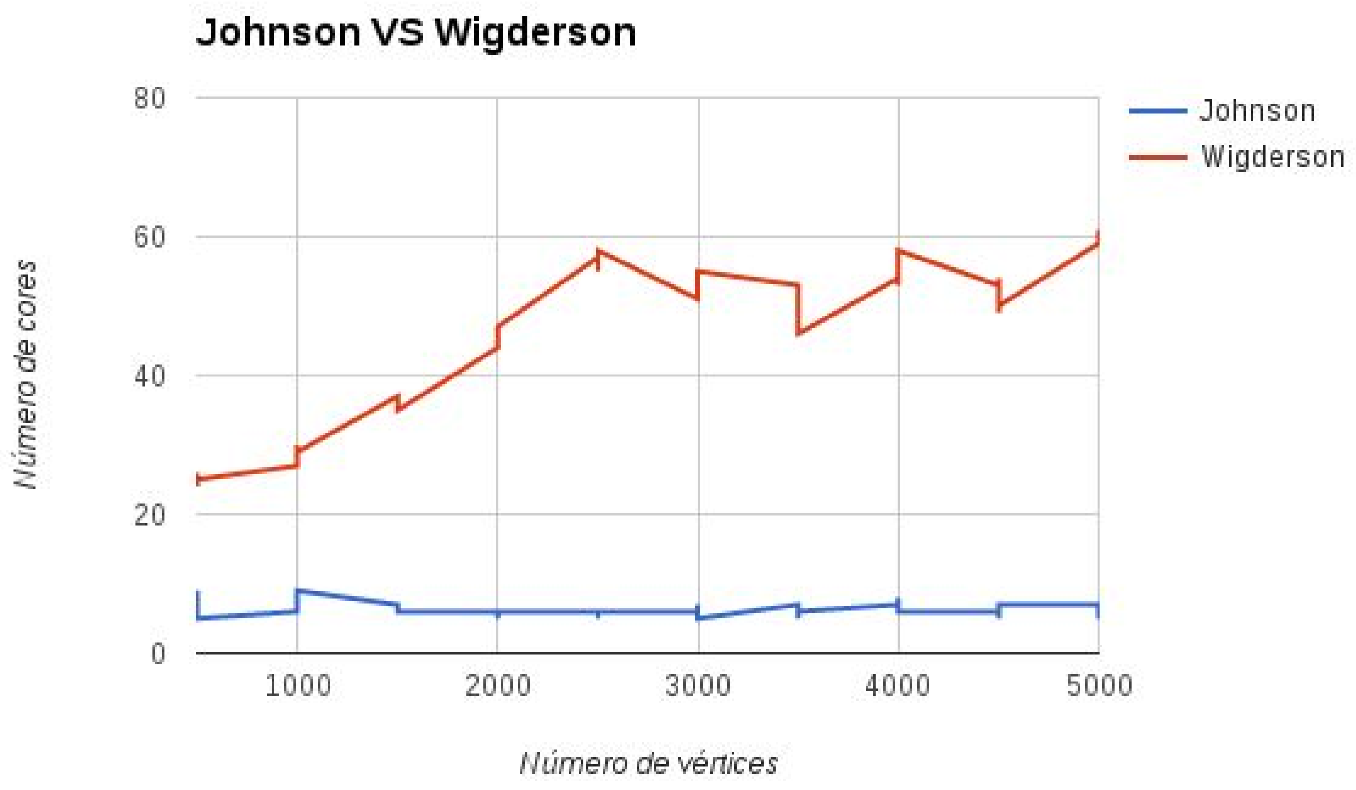

Algoritmo de Johnson
800
600
400
200
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
Número de cores
Número de vértices
Johnson
Johnson Teórico

Resultados



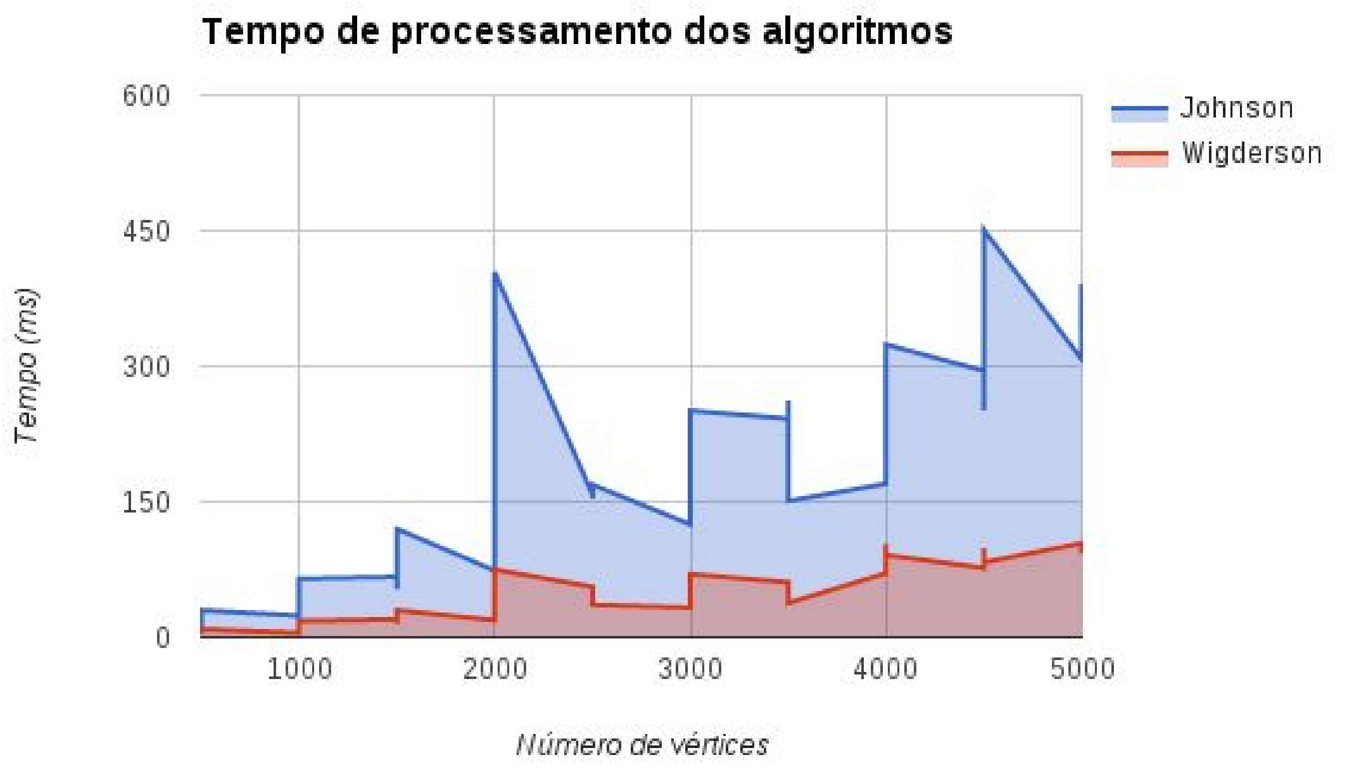
Algoritmo de Widgerson
240
180
120
60
0
1000
2000
3000
4000
5000
Número de cores
Número de vértices
Wigderson
Wigderson Teórico

Resultados



Johnson VS Wigderson
Johnson
Wigderson
Número de cores
Número de vértices
0
20
40
60
80
1000
2000
3000
4000
5000

Resultados



Tempo de processamento dos algoritmos
Johnson
Wigderson
600
450
300
150
0
Tempo (ms)
1000
2000
3000
4000
5000
Número de vértices

Conclusões

O algoritmo de Wigderson, apesar de possuir uma garantia de aproximação melhor que o algoritmo de Johnson, apresentou, em média, piores resultados de coloração.

É importante que uma análise dos algoritmos seja feita antes da escolha de algum deles, uma vez que confiar nas garantias de aproximação nem sempre é a melhor opção.

Dúvidas?