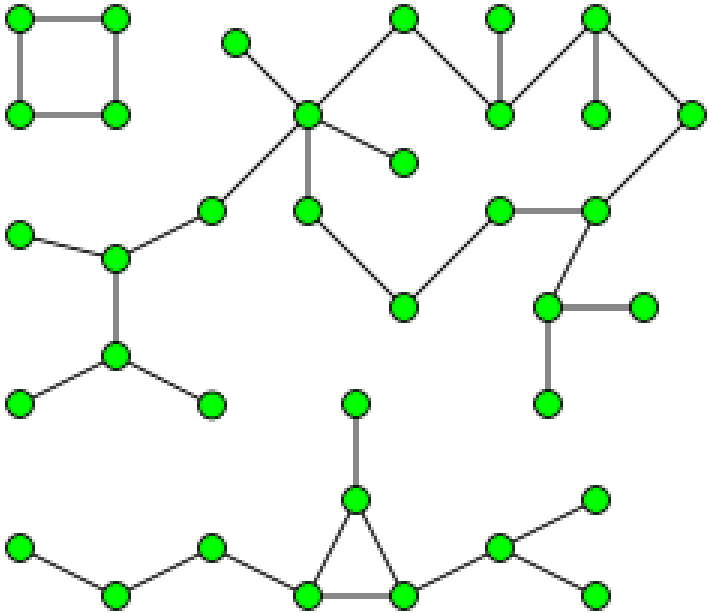
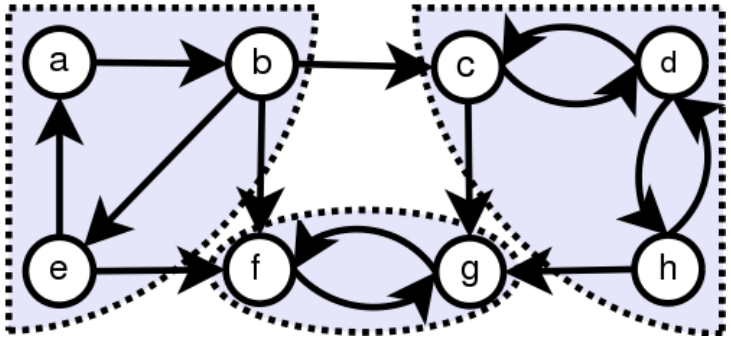
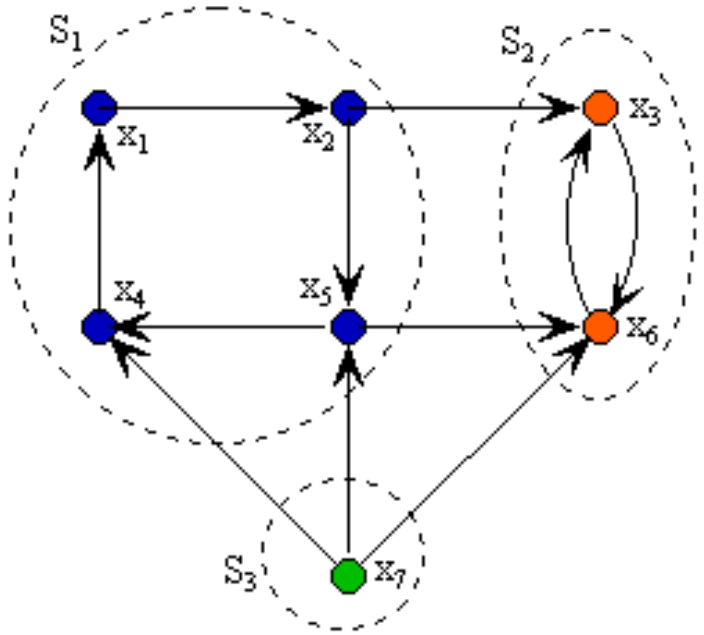






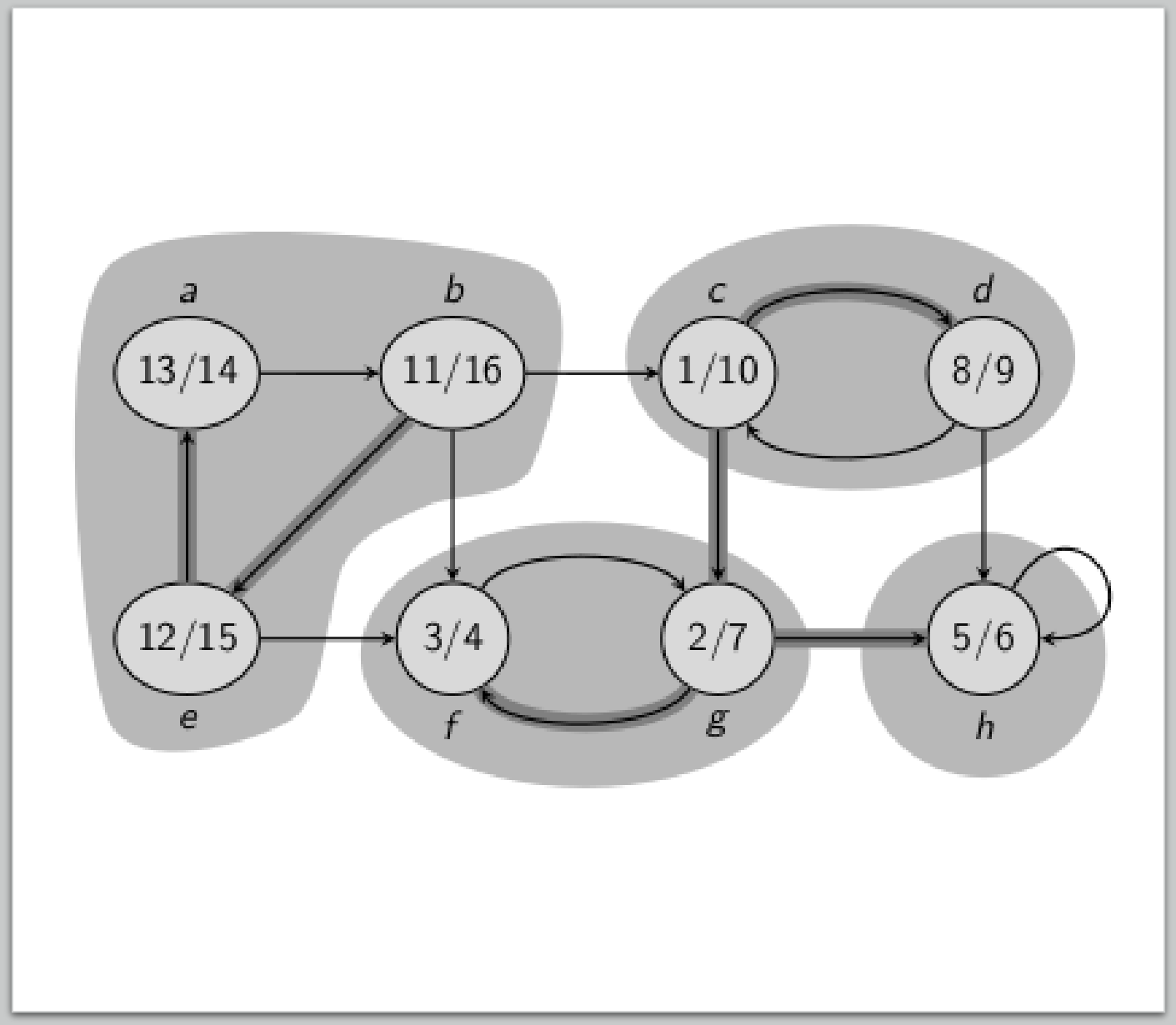
Componentes Fortemente Conexas

Teoria dos Grafos

Andrei Luiz Satoshi Kawamoto

Componentes Fortemente Conexas

- Um **grafo fortemente conexo** é aquele no qual existe 1 caminho entre quaisquer pares de vértices
- Uma componente fortemente conexa (**Strongly Connected Component - SCC)** é um subgrafo máximo que mantém essa propriedade



Aplicações

Componentes fortemente conexas podem ser usadas para identificar grupos de pessoas que são mais proximamente relacionados em conjuntos grandes de dados (como por exemplo, redes sociais).

Isso pode, por exemplo, servir para:

- recomendar pessoas que poderiam ser conhecidas
- recomendar interesses em comum que provavelmente existem.

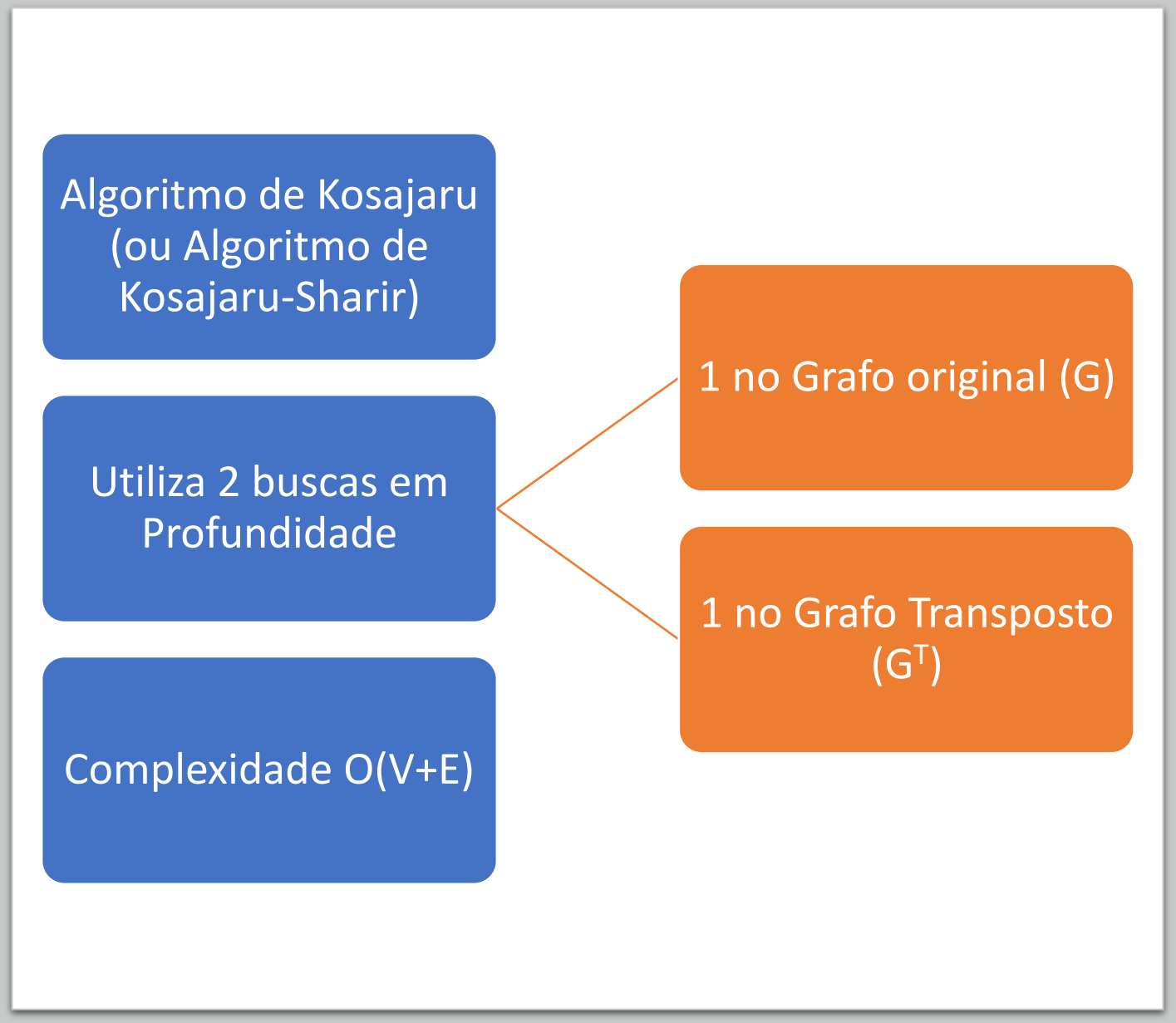
Outras Aplicações

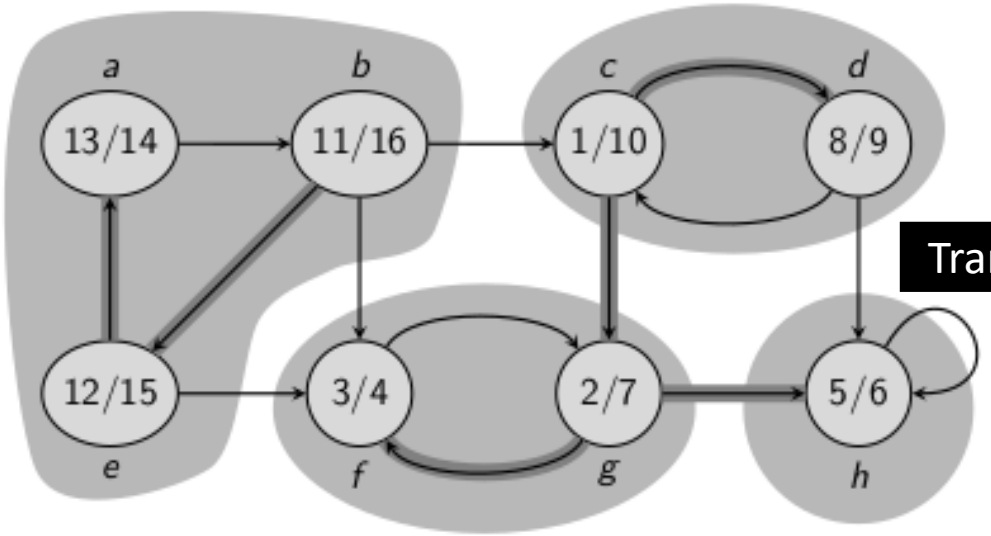
Páginas Web ser consideradas como nós, e os links para outras páginas como vértices, então seria possível descobrir conjuntos de páginas que não se relacionam

Encontrar dependências cíclicas em um programa, ou seja, a inclusão de 1 biblioteca ou função vai requerer a inclusão de outras

Identificar autores/pesquisadores que costumam publicar em conjunto e então identificar áreas de interesse em grupos de pesquisa.

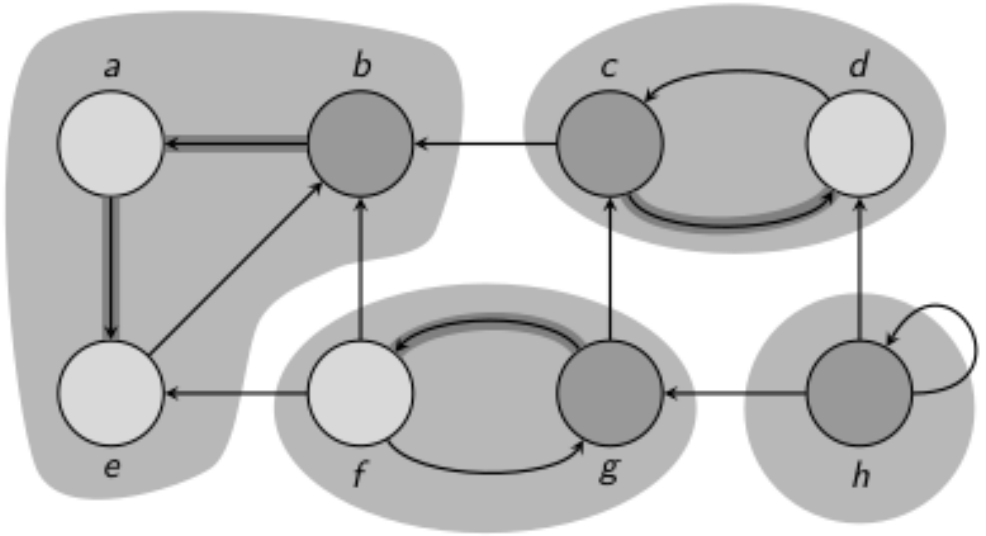
Algoritmo para descoberta de SCC





Grafo G

Transposição



Grafo G^T

O que é um Grafo Transposto?

- R: Grafo com todas as arestas invertidas

Kosajaru - Algoritmo

Algoritmo: **Identificação de componentes fortemente conexas**

1: function Strongly-Connected-Components(G)

2: chamar DFS(G) para calcular os tempos de término u:f, para cada vértice u

3: calcular G^T

4: chamar DFS(G^T) mas, no laco principal de DFS, considerar os vértices em ordem decrescente de u:f (como uma pilha)

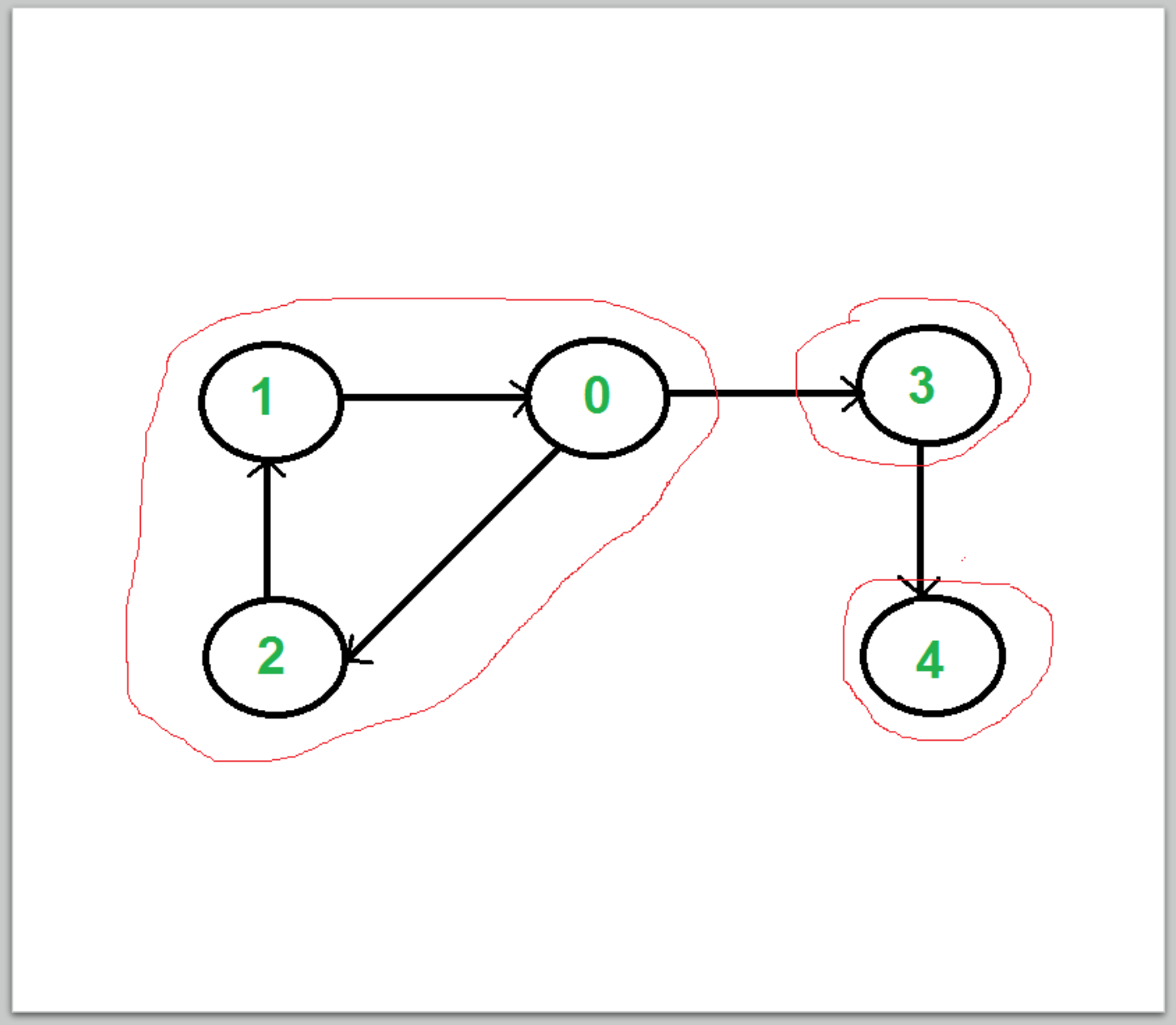
5: dar saída aos vértices de cada árvore na floresta em profundidade formada na linha 4 como uma componente fortemente conexa separada

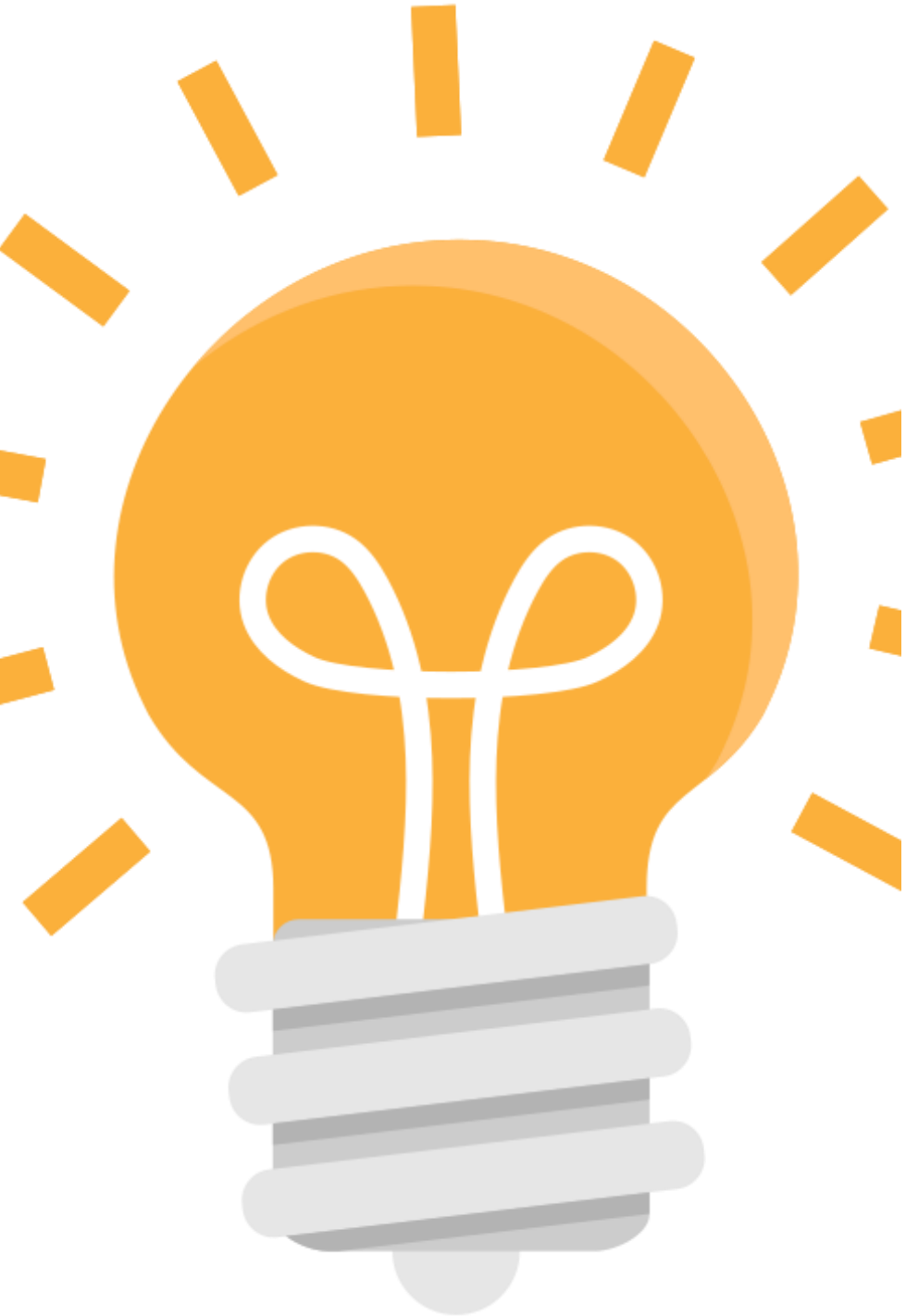
Por que Funciona?

- Em um grafo qualquer, ao fazermos apenas 1 Busca em Profundidade
  - Se houver 1 única componente fortemente conexa, o resultado sempre é uma árvore
  - Se houver mais de uma SCC, o resultado pode variar de acordo com o vértice escolhido como inicial*

*Observe o Exemplo

- Se selecionarmos como vértice inicial 0, 1 ou 2, teremos uma árvore.
- Se selecionarmos o 3 ou 4, teremos uma floresta
- Para obter as SCCs corretamente, deveríamos iniciar no vértice 4 (que é sorvedouro), em seguida iniciar pelo 3, que é o próximo sorvedouro e finalmente qualquer um dos vértices restantes (0, 1 ou 2)
- Infelizmente não há maneira de estabelecer o jeito ‘correto’ de selecionar os vértices



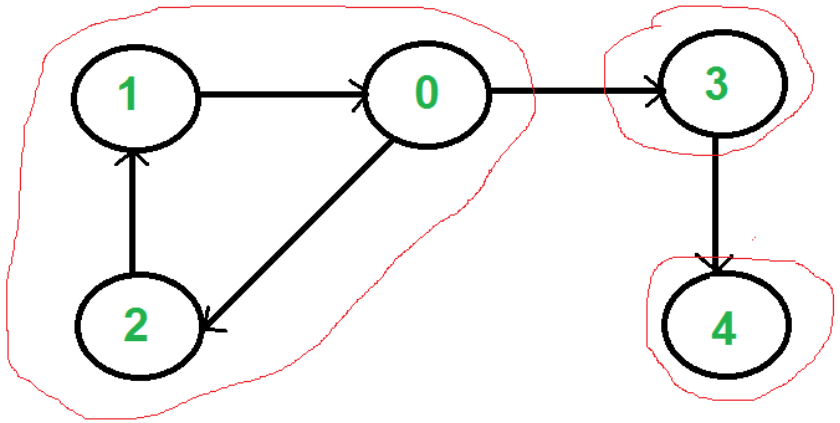


Por que funciona?

- No entanto, se fizermos uma DFS e armazenar vértices de acordo com seus tempos de acabamento (ou seja, uma ordenação topológica), garantiremos que o tempo de finalização de um vértice que se conecta a outros SCCs (diferente da sua própria SCC) seja sempre maior do que o tempo final de vértices no outro SCC

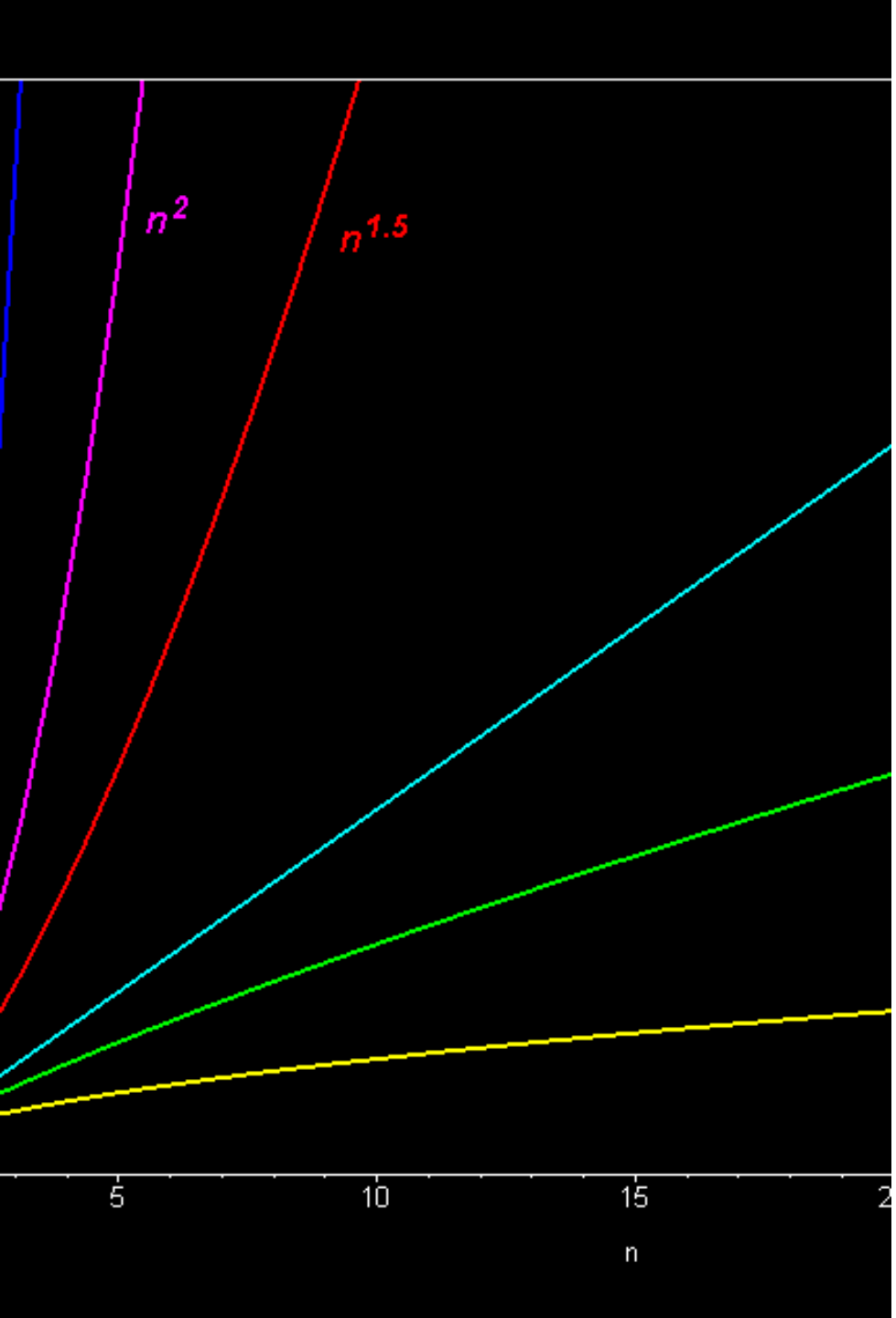
Por que Funciona?

- No exemplo, o tempo de término de 0 é **sempre** maior que o do 3 e do 4 (independentemente da seqüência de vértices considerados para DFS).
- O mesmo não ocorre com outros vértices, por exemplo, os tempos de acabamento de 1 e 2 podem ser menores ou maiores que 3 e 4, dependendo da sequência de vértices considerados para DFS.
- Para usar essa propriedade, fazemos o percurso DFS do grafo completo e empilhe cada vértice concluído. Nessa pilha, 3 sempre aparece após 4 e 0 aparecem após 3 e 4.



Por que funciona?

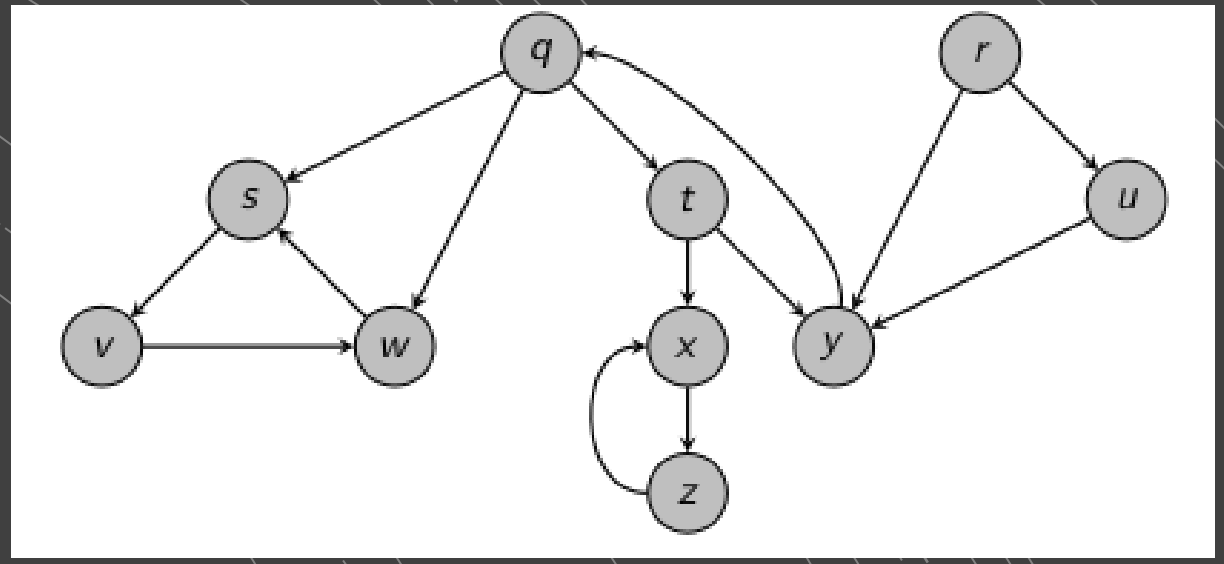
- Ao inverter o grafo, todas as arestas que conectam 2 componentes são invertidas
- Assim, o SCC {0, 1, 2} torna-se sorvedouro e o SCC {4} se torna fonte.
- Como discutimos, na pilha, sempre temos 0 antes de 3 e 4.
- Então, se fizermos uma DFS do grafo invertido usando seqüências de vértices na pilha, processamos vértices do sorvedouro para a fonte (no gráfico invertido).
- Isso é o que queríamos alcançar e isso é tudo necessário para imprimir SCCs um a um.



Complexidade

- Tempo para Busca em Profundidade O(V+E)
- Tempo para inverter as arestas O(E)
- Tempo para busca em profundidade a fim de obter as SCCs O(V+E)

- O(V+E)



Questões para reflexão

- O número de componentes fortemente conexas de um grafo pode mudar se uma nova aresta e adicionada ou removida? Que tipo de aresta pode ser responsável por isso?
- Use o algoritmo para identificar SCCs no grafo