Exemplos de uso de busca em profundidade

Karina Valdivia Delgado

kvd@usp.br

Agenda

- Ordenação topológica de um grafo acíclico orientado
- Localização de componentes fortemente conectadas de um grafo orientado

Busca em profundidade (Depth-first-search-DFS)

A ideia é prosseguir a busca sempre a partir do vértice descoberto mais recentemente, até que este não tenha mais vizinhos descobertos. Neste caso, volta-se na busca para o precursor desse vértice.

DFS devolve uma floresta.

Busca em profundidade

Para organizar o processo de busca os vértices são pintados:

- branco: não foram descobertos ainda
- cinza: são a fronteira. O vértice já foi descoberto mas ainda não examinamos os seus vizinhos.
- preto: são os vértices já descobertos e seus vizinhos já foram examinados.

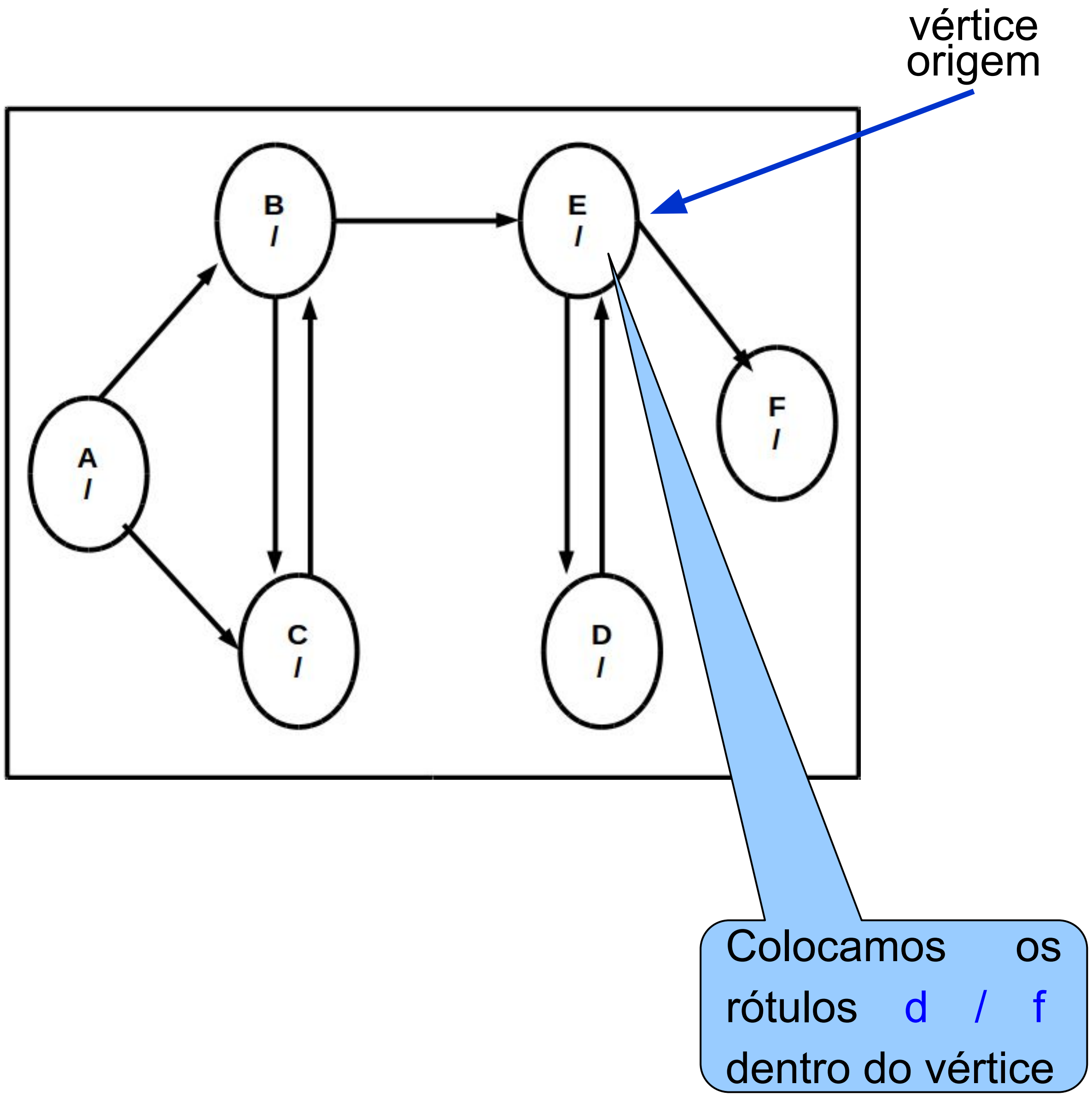
Busca em profundidade

Os vértices recebem 2 rótulos:

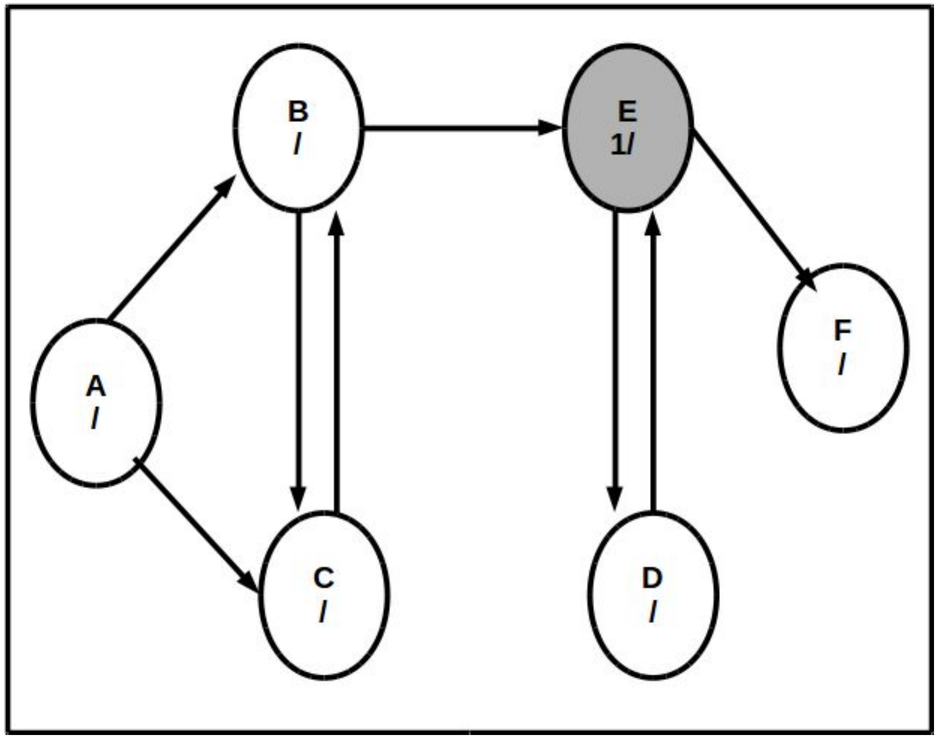
- d[.]: o momento em que o vértice foi descoberto (tornou-se cinza).
- f[.]: o momento em que examinamos os seus vizinhos (tornou-se preto).

O vértice é branco até d, cinza entre d e f e preto a partir de f.

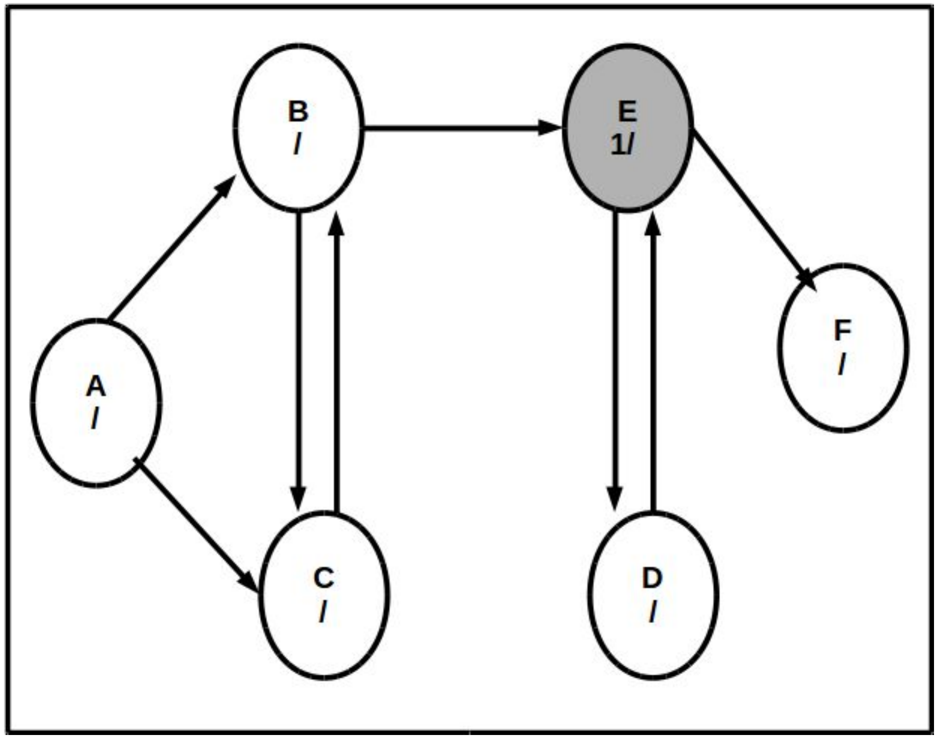
Busca em profundidade



Busca em profundidade

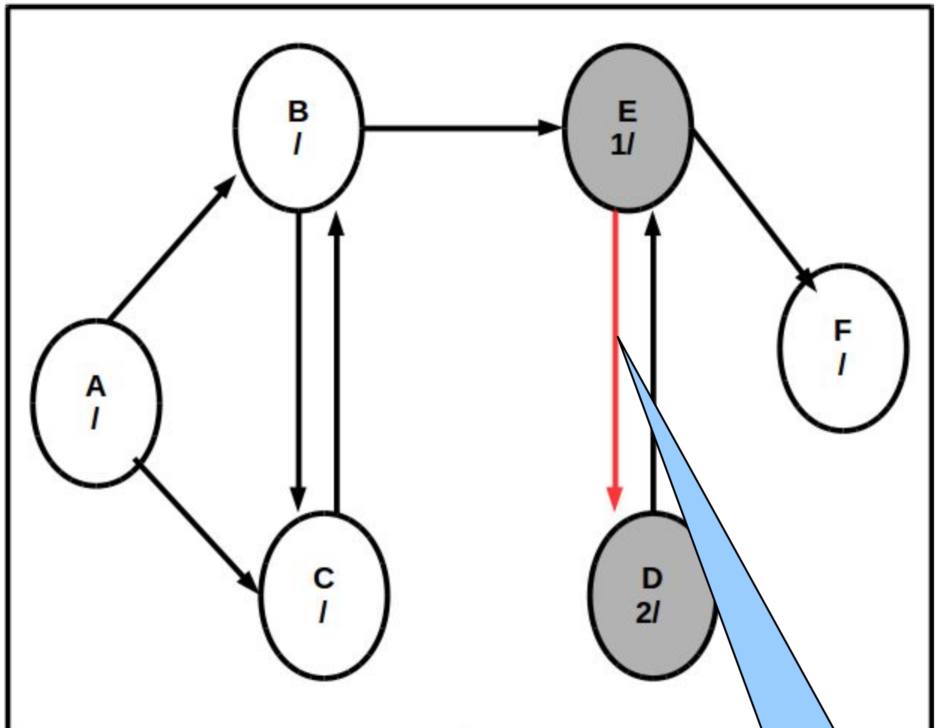


Busca em profundidade



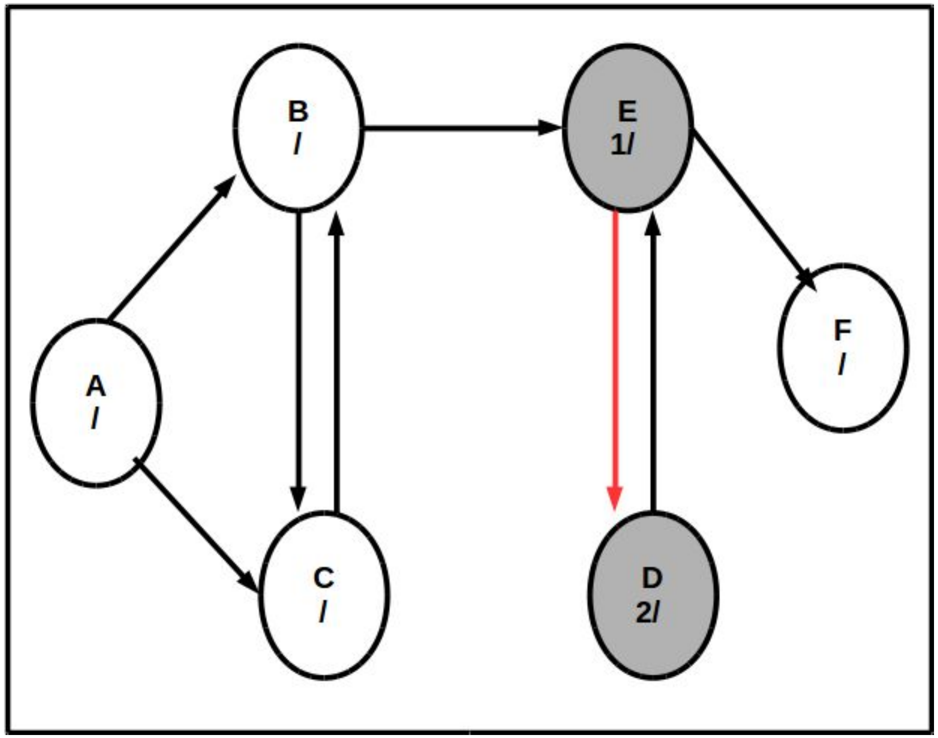
Existe algum vértice adjacente ao vértice que não tenha sido descoberto?

Busca em profundidade



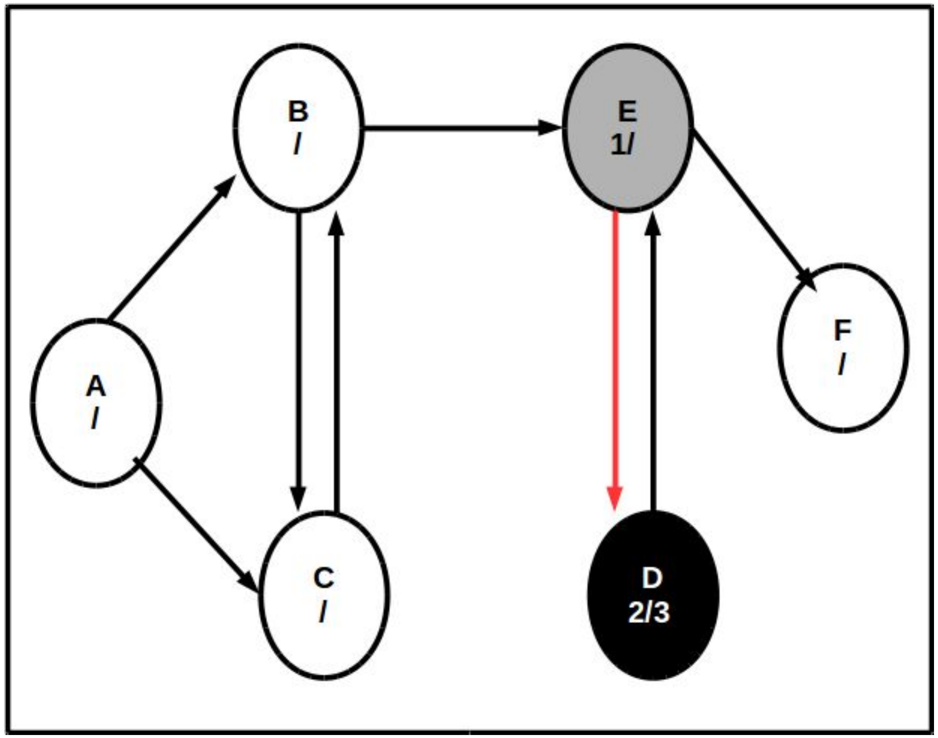
Aproveitamos de marcar quem é o pai

Busca em profundidade



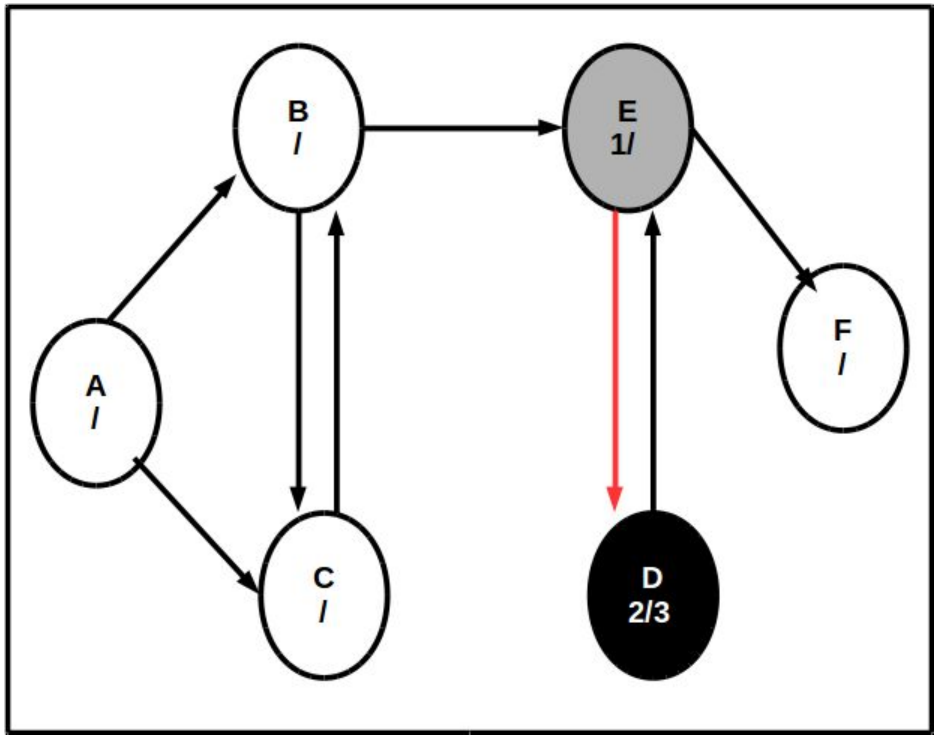
Existe algum vértice adjacente ao vértice que não tenha sido descoberto?

Busca em profundidade



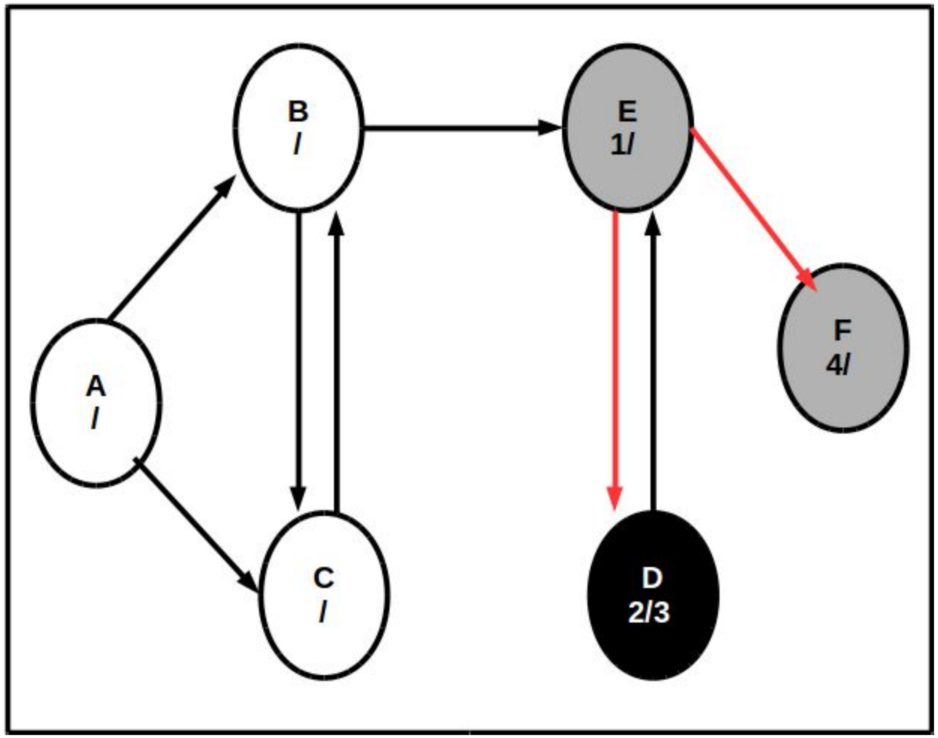
Não existe. Então, terminei com o vértice (pinta ele de preto)
Volta-se na busca para o precursor desse vértice.

Busca em profundidade

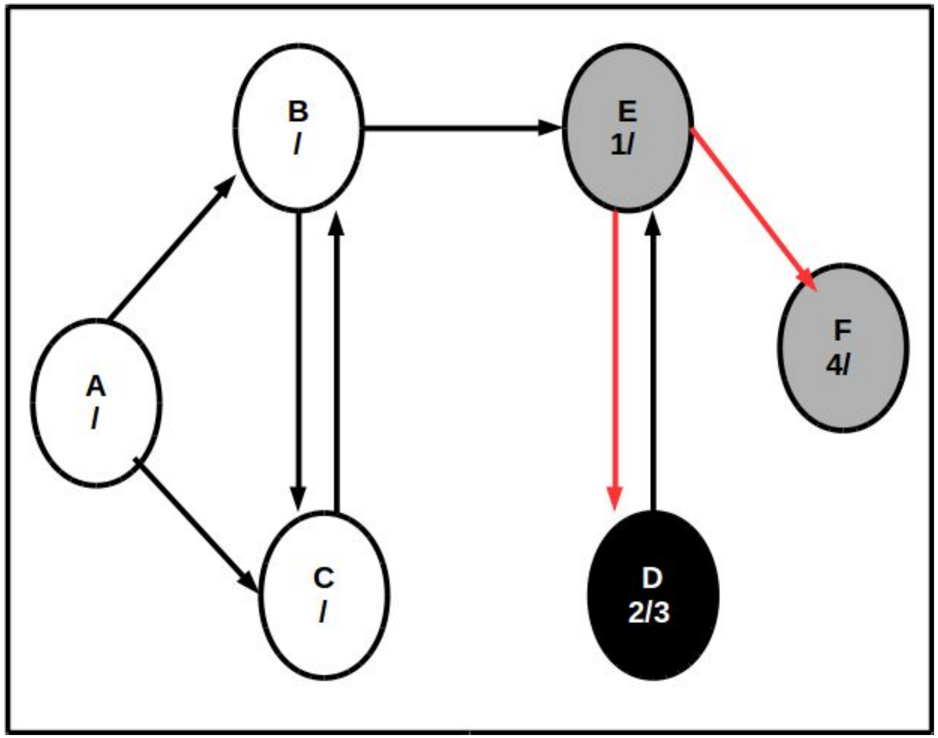


Existe algum vértice adjacente ao vértice que não tenha sido descoberto?

Busca em profundidade

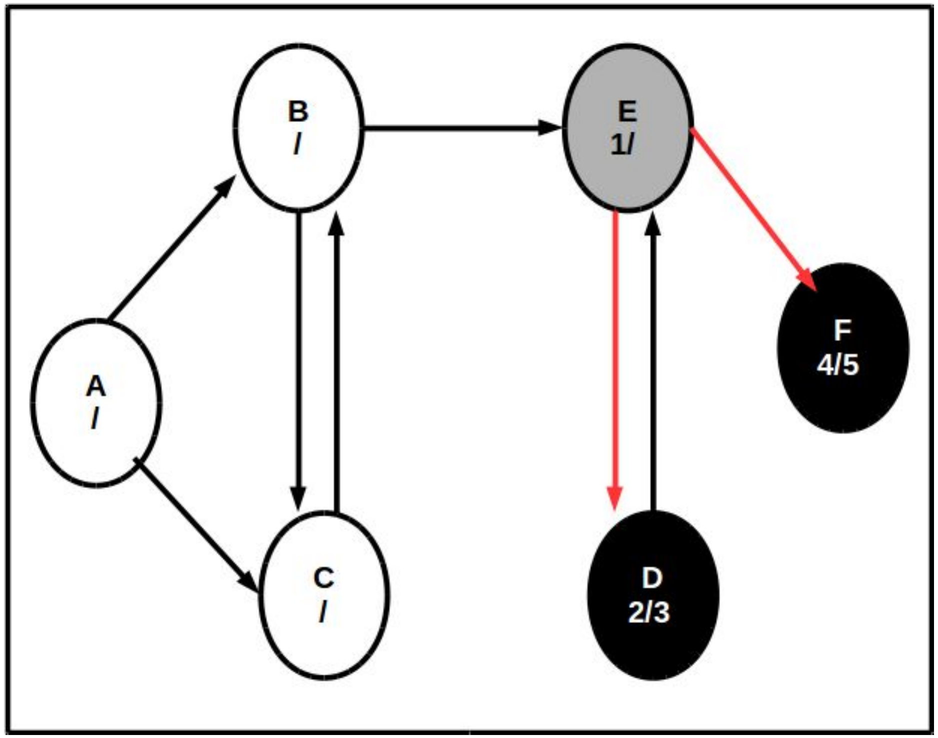


Busca em profundidade



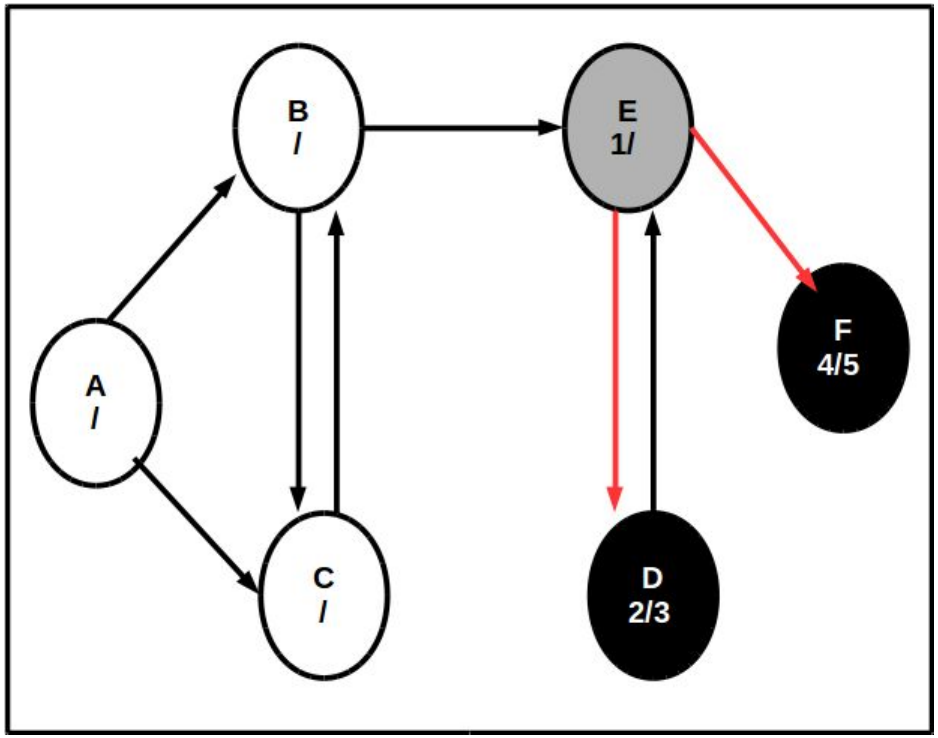
Existe algum vértice adjacente ao vértice que não tenha sido descoberto?

Busca em profundidade



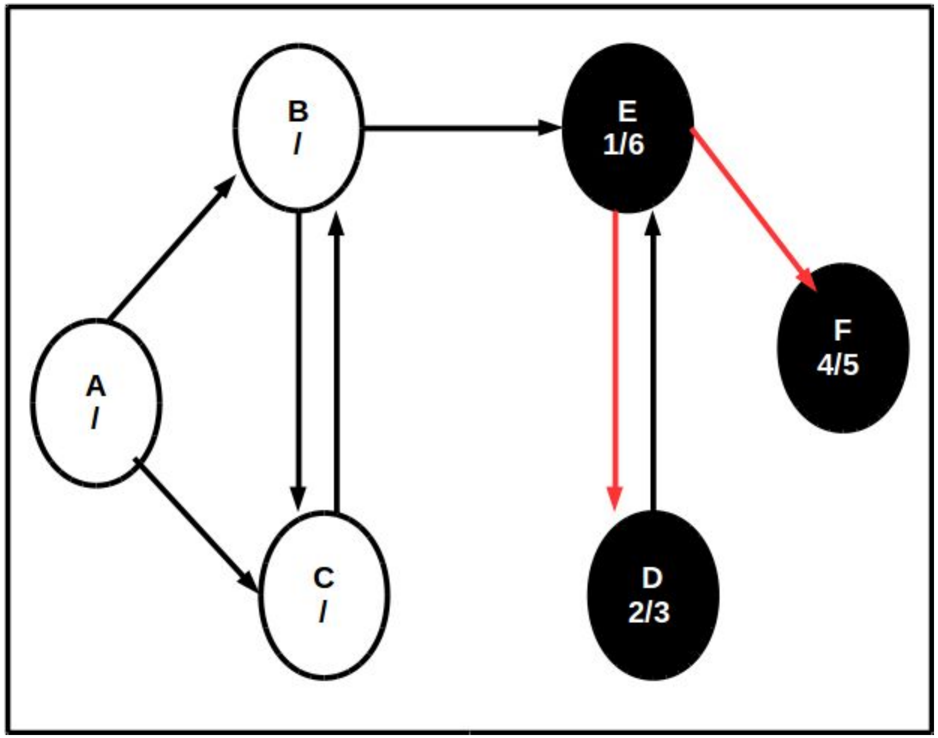
Não existe. Terminei!
Volta-se na busca para o precursor desse vértice.

Busca em profundidade



Existe algum vértice adjacente ao vértice que não tenha sido descoberto?

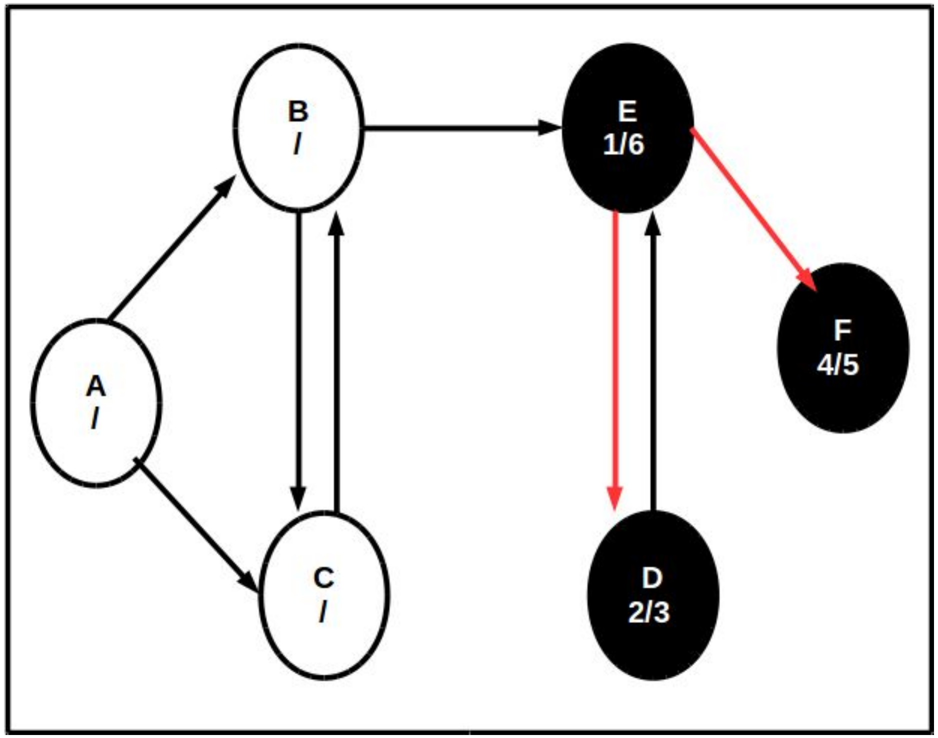
Busca em profundidade



Não existe. Terminei!
Volta-se na busca para o precursor desse vértice.

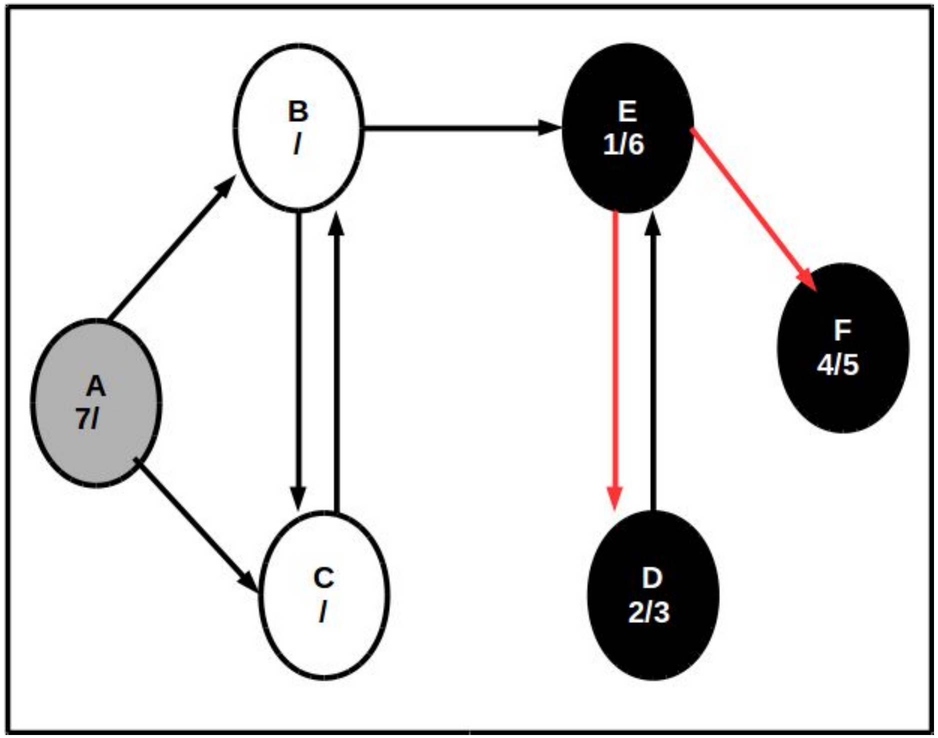
Busca em profundidade

vértice origem

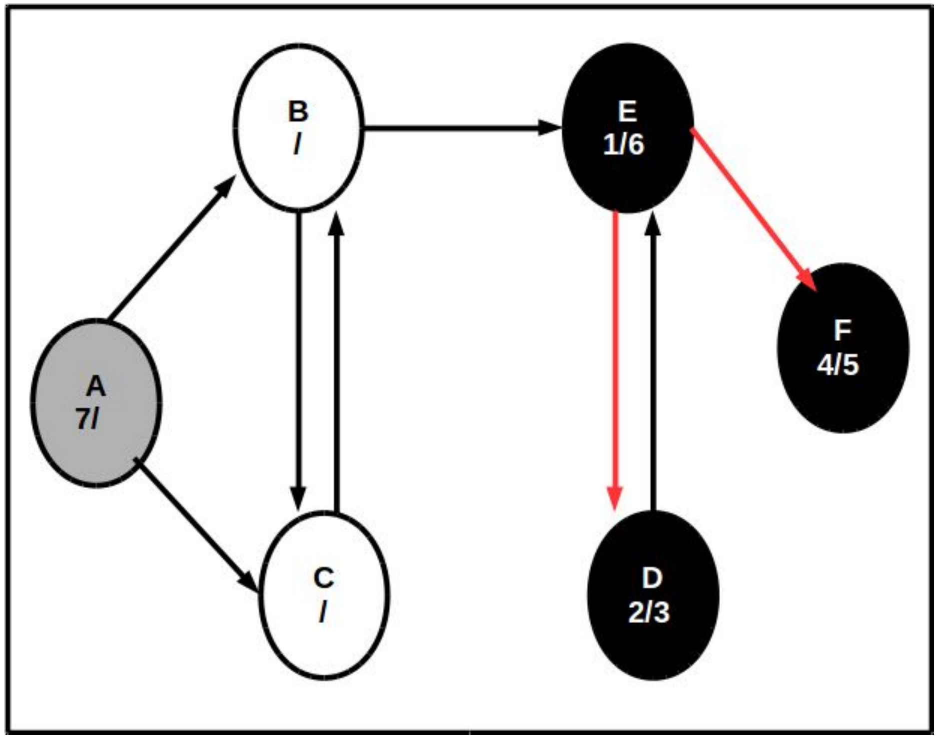


Busca em profundidade

vértice origem

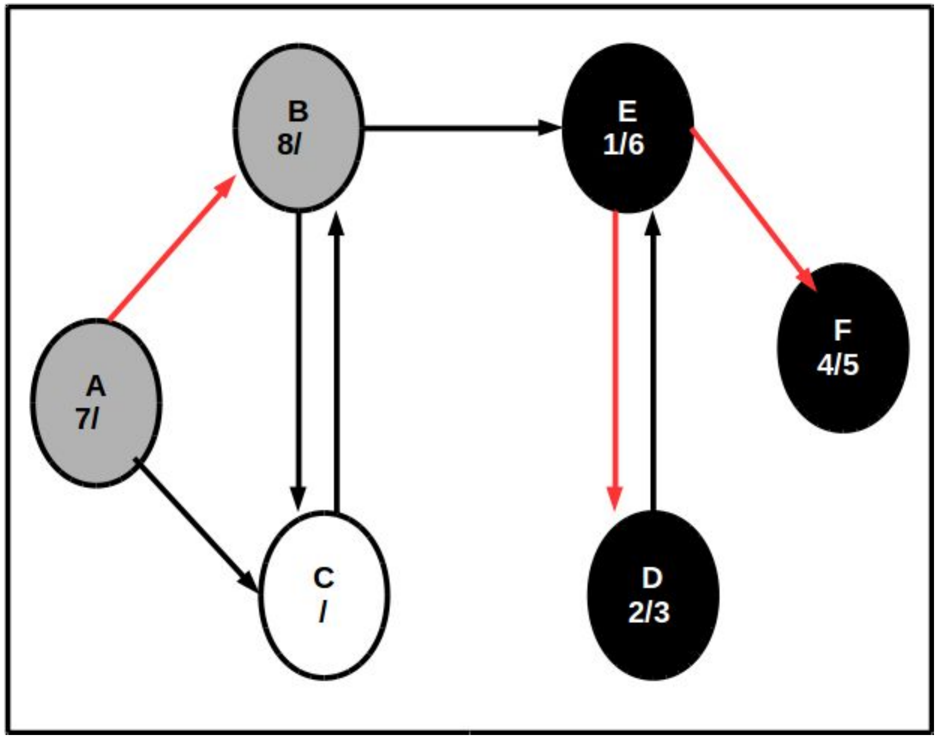


Busca em profundidade

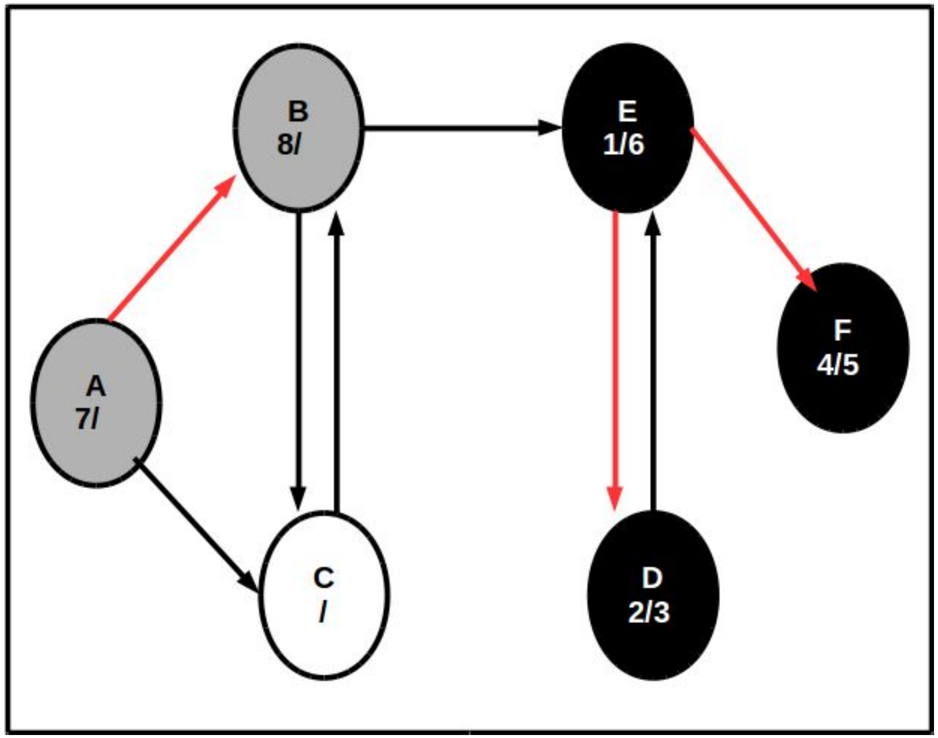


Existe algum vértice adjacente ao vértice que não tenha sido descoberto?

Busca em profundidade

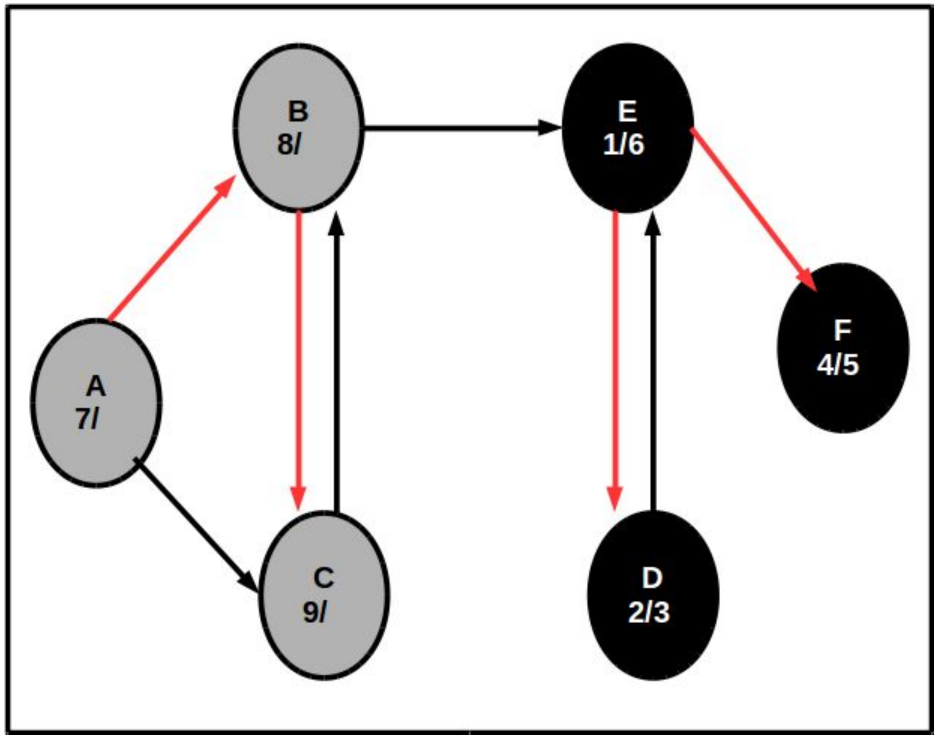


Busca em profundidade

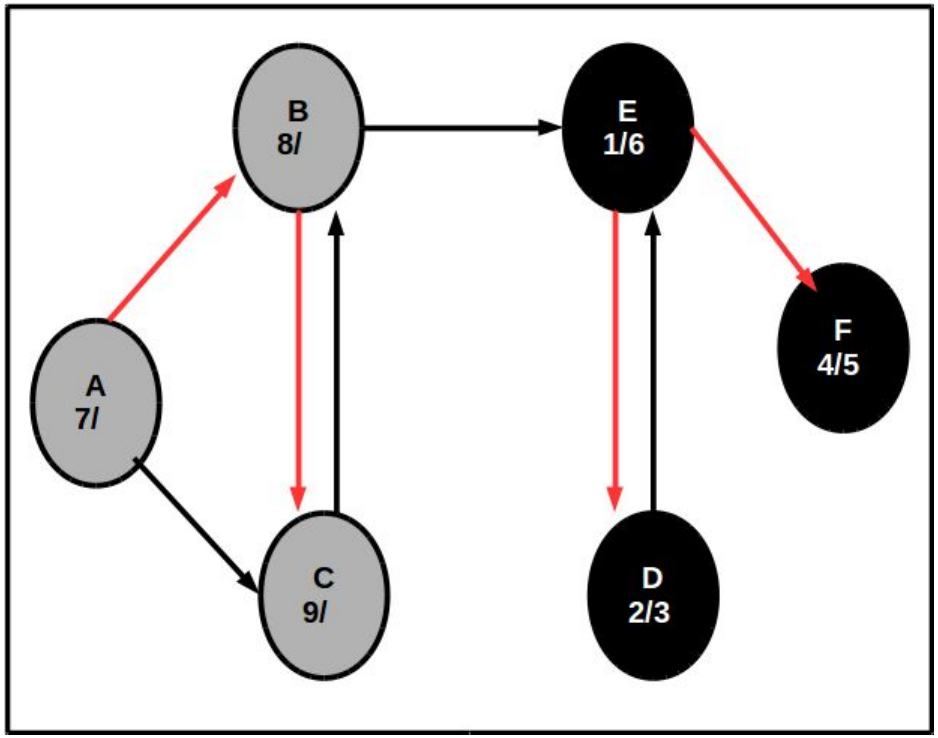


Existe algum vértice adjacente ao vértice que não tenha sido descoberto?

Busca em profundidade

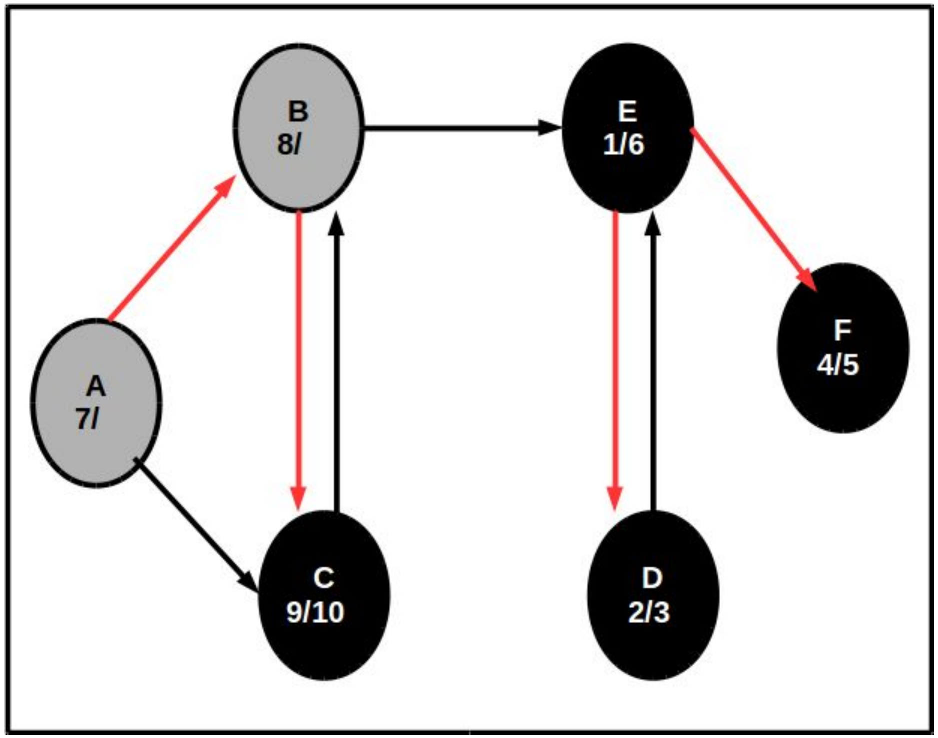


Busca em profundidade



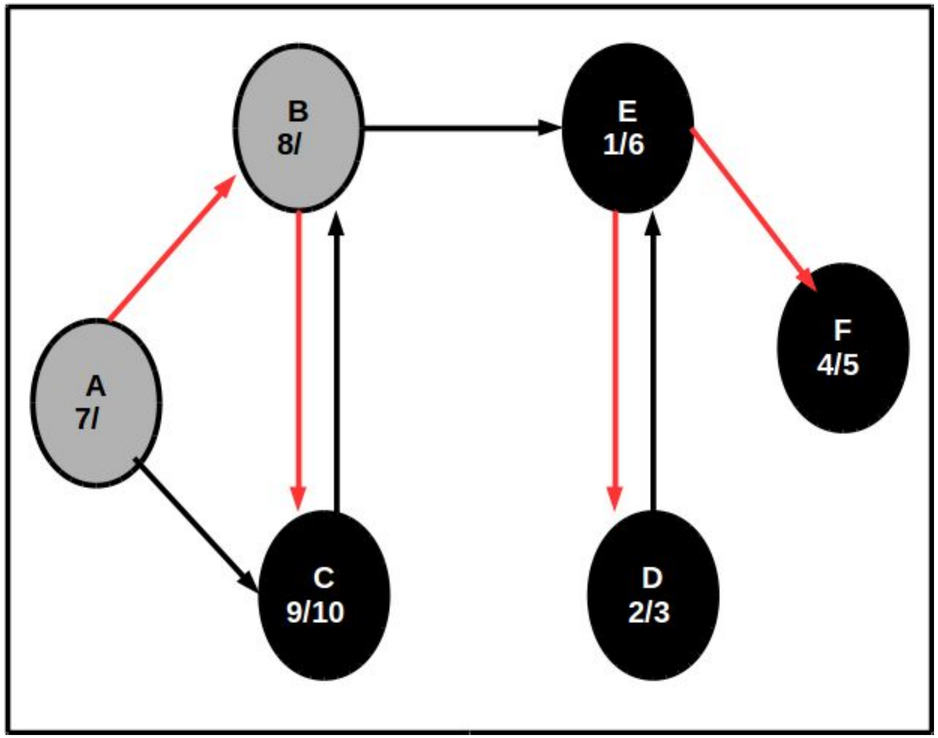
Existe algum vértice adjacente ao vértice que não tenha sido descoberto?

Busca em profundidade



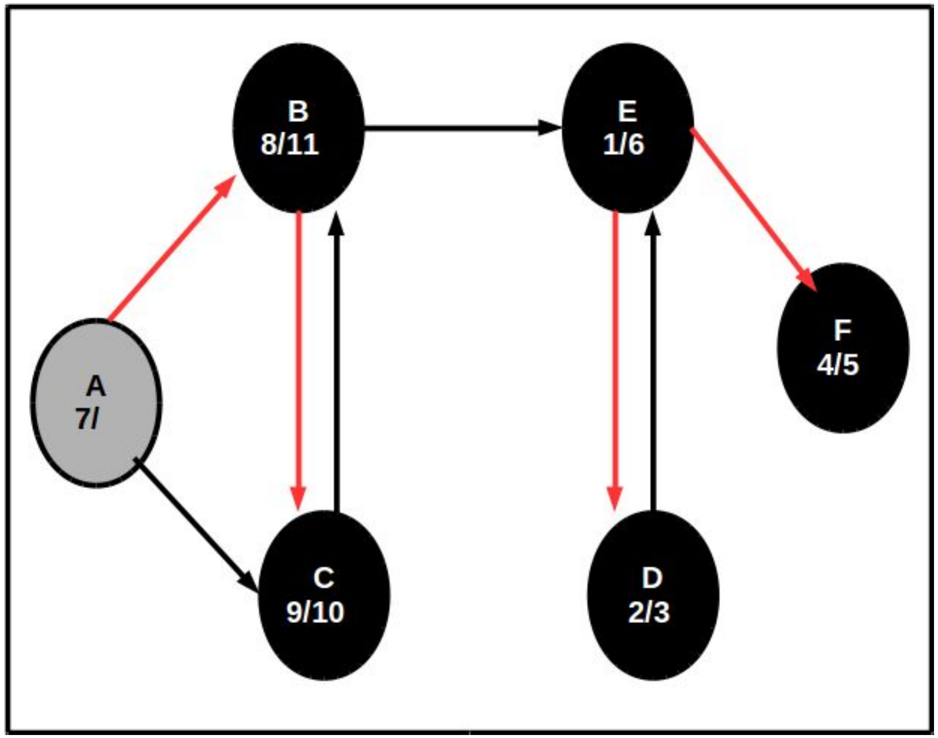
Não existe. Terminei!
Volta-se na busca para o precursor desse vértice.

Busca em profundidade



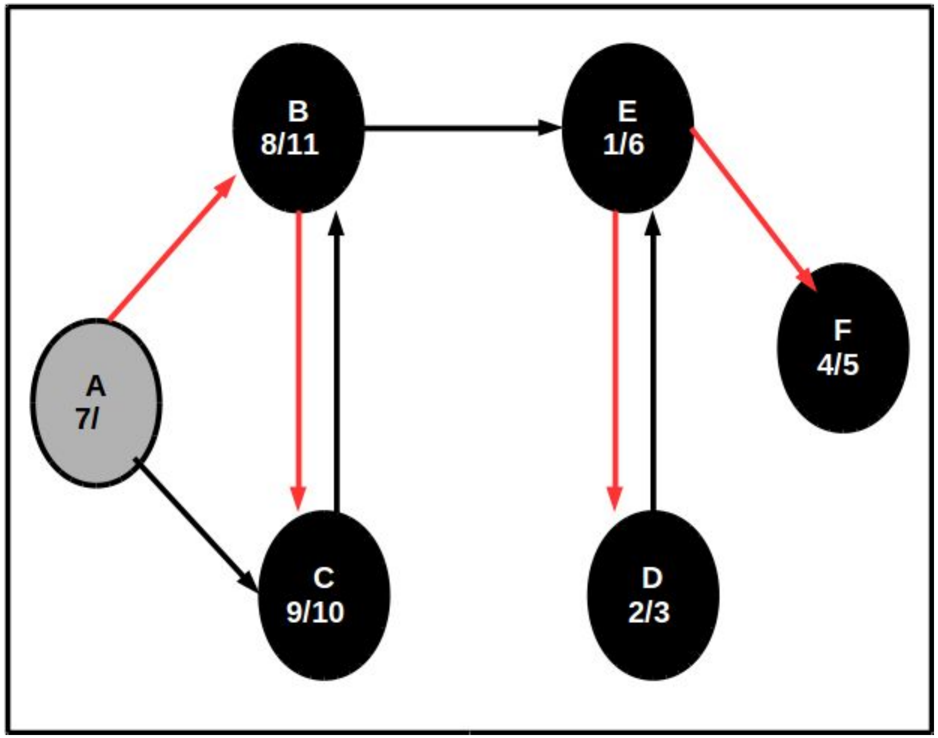
Existe algum vértice adjacente ao vértice que não tenha sido descoberto?

Busca em profundidade



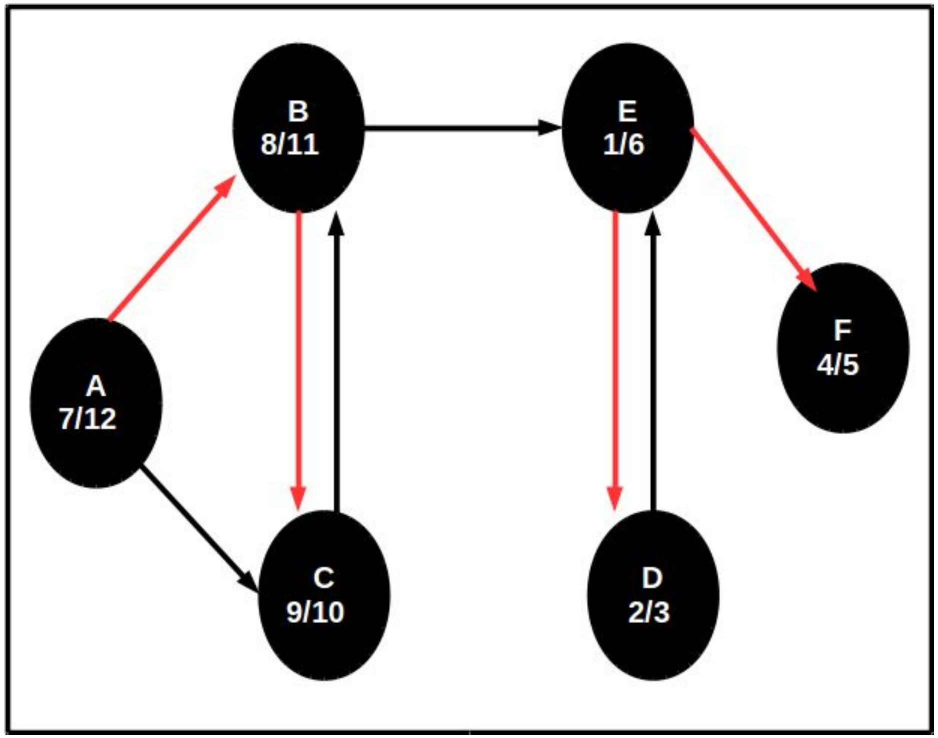
Não existe. Terminei!
Volta-se na busca para o precursor desse vértice.

Busca em profundidade



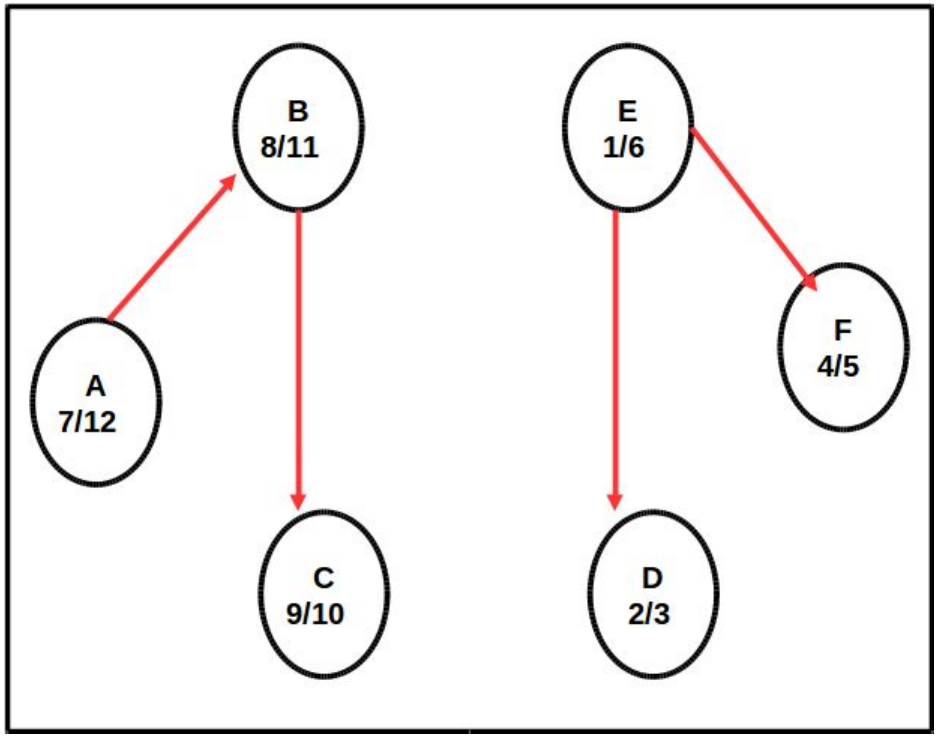
Existe algum vértice adjacente ao vértice que não tenha sido descoberto?

Busca em profundidade



Não existe. Terminei!

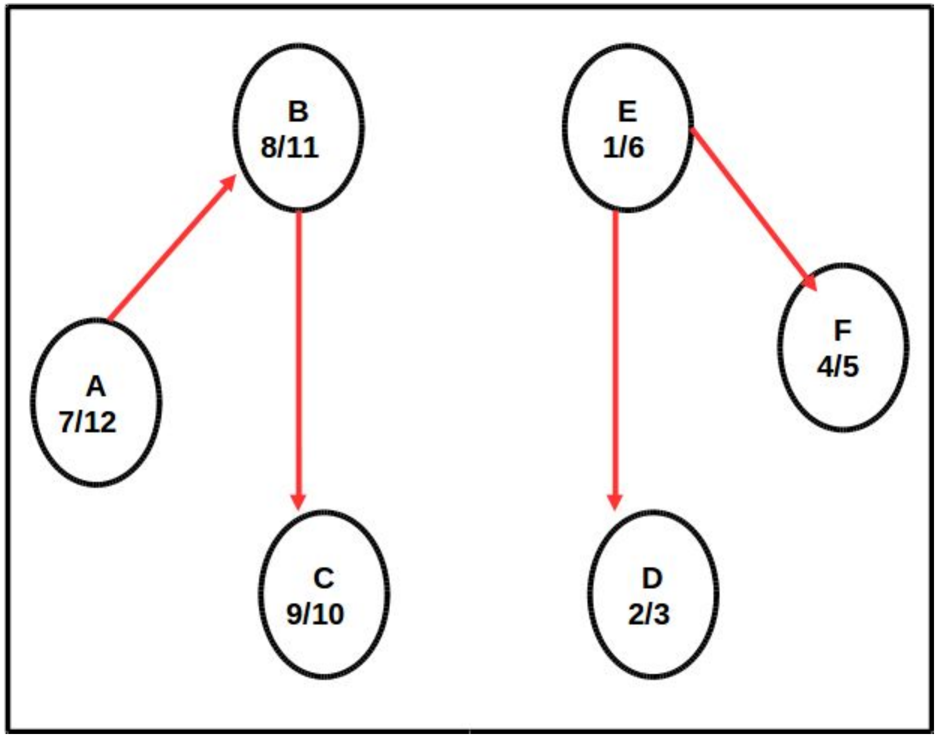
Busca em profundidade



Resultado: uma floresta com duas árvores

Busca em profundidade

vértice	π
A	NIL
B	A
C	B



vértice	π
E	NIL
D	E
F	E

Resultado: uma floresta com duas árvores

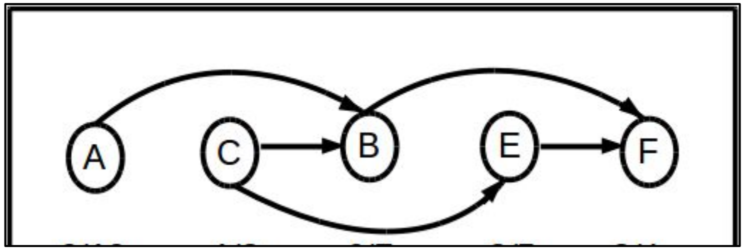
Uso de busca em profundidade para:

- Ordenação topológica de um grafo acíclico orientado
- Localização de componentes fortemente conectadas de um grafo orientado

Ordenação topológica

Uma ordenação topológica de um grafo acíclico orientado G(V,A) é uma ordenação linear de todos os seus vértices, tal que:

- se G contém uma aresta (u,v), então u aparece antes de v na ordenação.



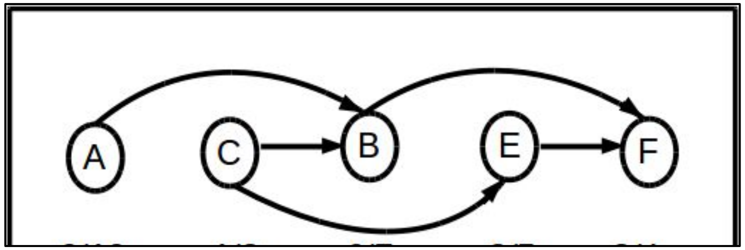
Ordenação topológica

Grafos acíclicos orientados são usados para indicar precedência entre eventos ou tarefas. Uma ordenação topológica é uma sequência válida de tarefas.

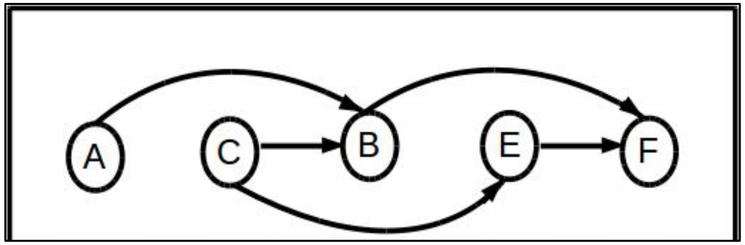
Exemplo:

- Vértices: disciplinas de um curso
- Arcos: pré-requisitos entre as disciplinas.

Uma ordenação topológica é uma sequência válida para cursar as disciplinas.



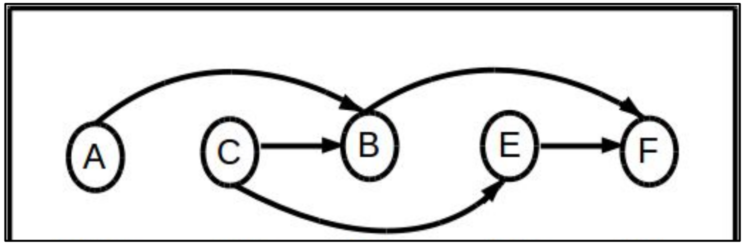
Ordenação topológica



1

F não é prerrequisito de nenhuma outra disciplina, posso deixar F para cursar no final

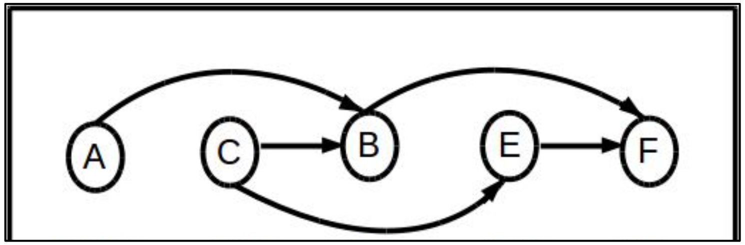
Ordenação topológica



1

O único com grau de saída 0 é F

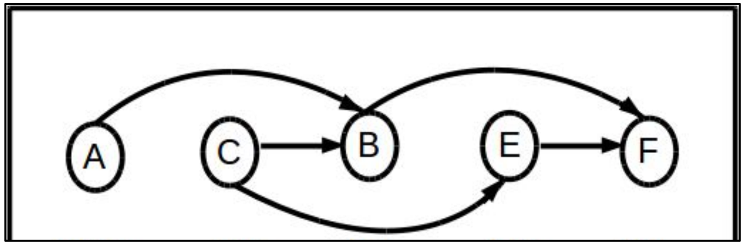
Ordenação topológica



1 2

O único com grau de saída 0 é F

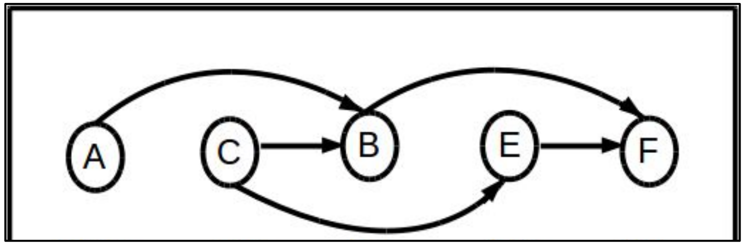
Ordenação topológica



1 2 3

O único com grau de saída 0 é F

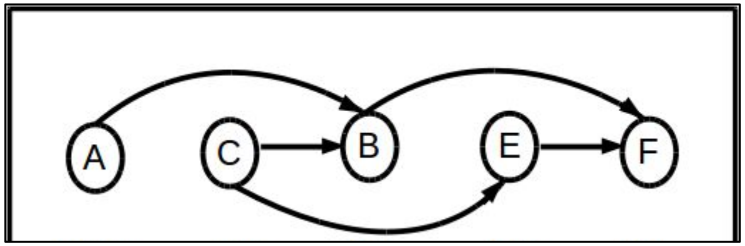
Ordenação topológica



1 2 3/4

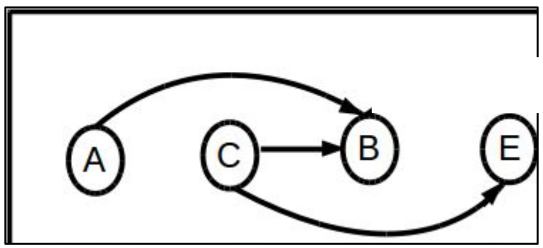
O único com grau de saída 0 é F

Ordenação topológica



1 2 3/4

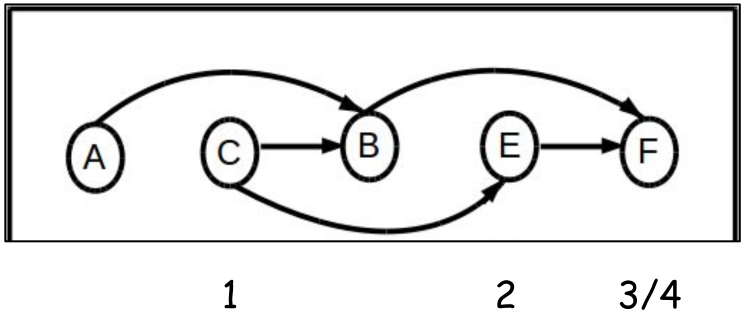
O único com grau de saída 0 é F



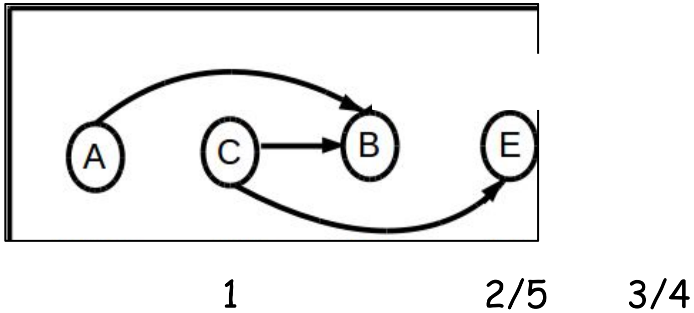
1 2 3/4

Um dos nós que tem grau de saída 0 é E

Ordenação topológica

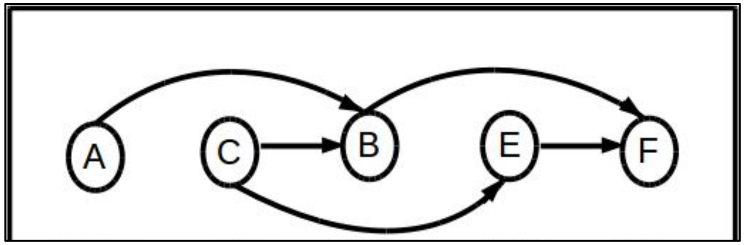


O único com grau de saída 0 é F



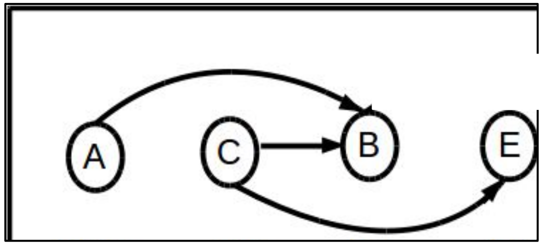
Um dos nós que tem grau de saída 0 é E

Ordenação topológica



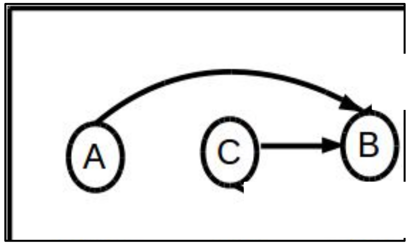
1 2 3/4

O único com grau de saída 0 é F



1 2/5 3/4

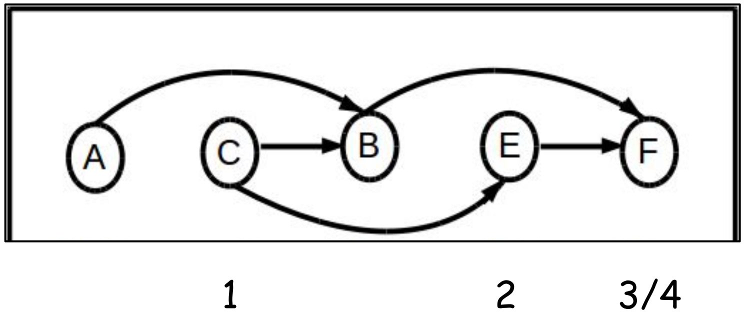
Um dos nós que tem grau de saída 0 é E



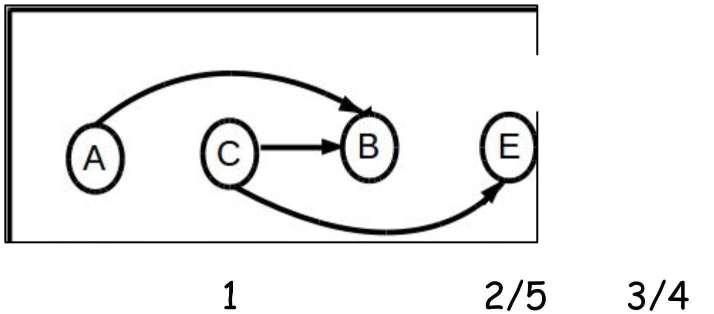
1 2/5 3/4

O único com grau de saída 0 é B

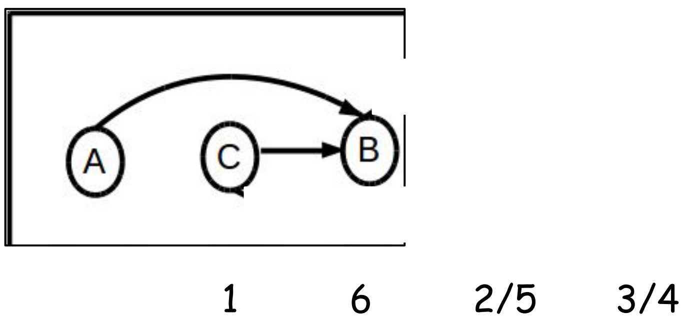
Ordenação topológica



O único com grau de saída 0 é F

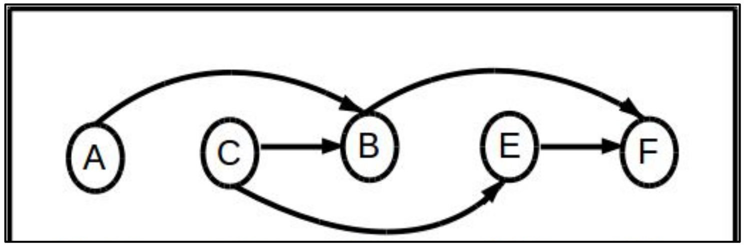


Um dos nós que tem grau de saída 0 é E



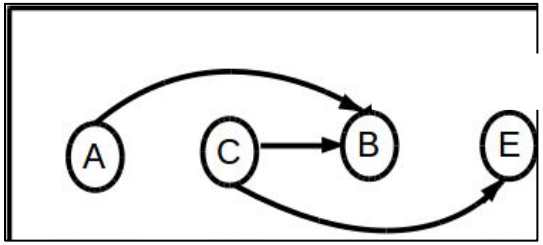
O único com grau de saída 0 é B

Ordenação topológica



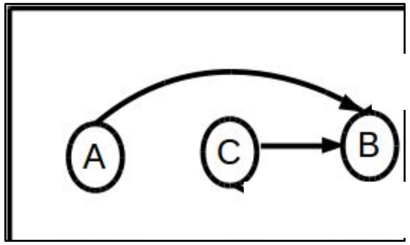
1 2 3/4

O único com grau de saída 0 é F



1 2/5 3/4

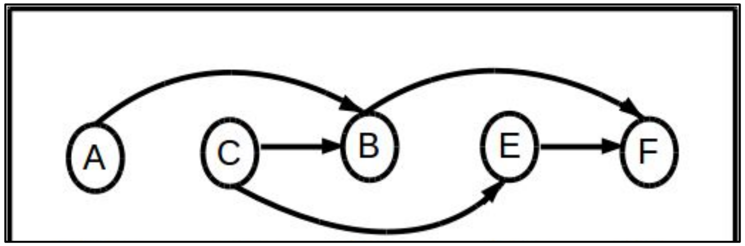
Um dos nós que tem grau de saída 0 é E



1 6/7 2/5 3/4

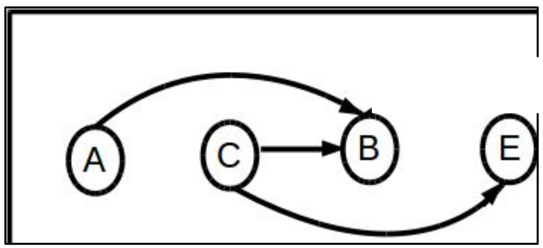
O único com grau de saída 0 é B

Ordenação topológica



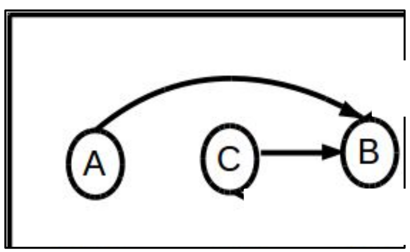
1 2 3/4

O único com grau de saída 0 é F



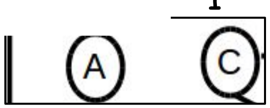
1 2/5 3/4

Um dos nós que tem grau de saída 0 é E



1 6/7 2/5 3/4

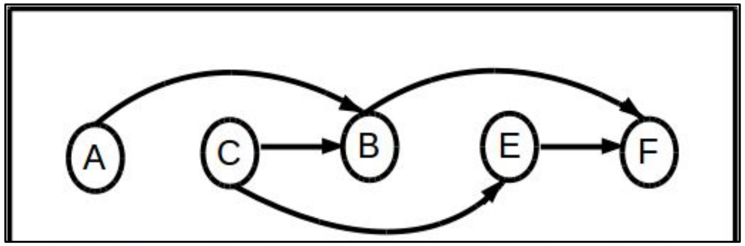
O único com grau de saída 0 é B



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

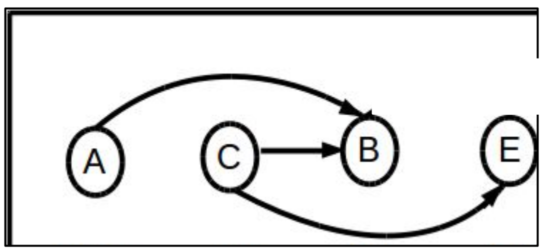
Um dos nós que tem grau de saída 0 é C

Ordenação topológica



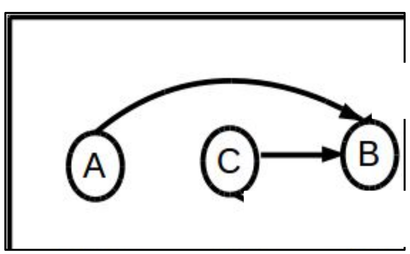
1 2 3/4

O único com grau de saída 0 é F



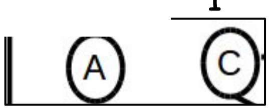
1 2/5 3/4

Um dos nós que tem grau de saída 0 é E



O único com grau de saída 0 é B

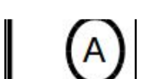
1 6/7 2/5 3/4



Um dos nós que tem grau de saída 0 é C

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

9/10

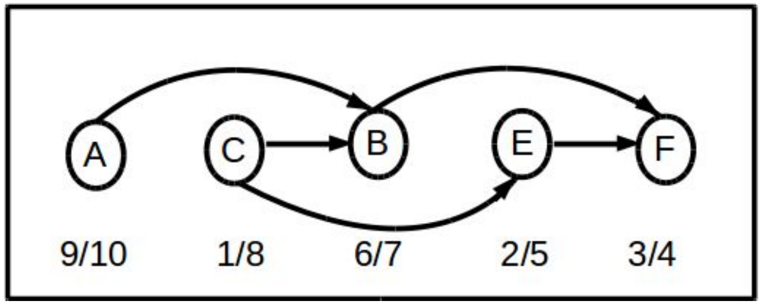


O único com grau de saída 0 é A

Ordenação topológica

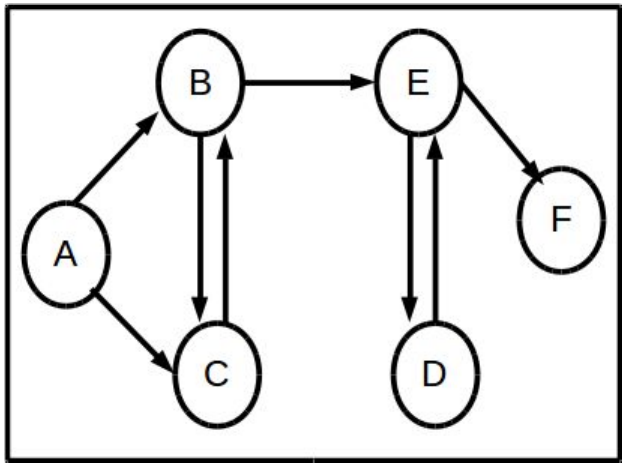
OrdenaçãoTopologica (V, A)

1. Chamar DFS (V, A), quando o vértice é colorido de preto, inserí-lo à frente de uma lista ligada
2. Devolva a lista ligada de vértices



Localização de componentes fortemente conectadas

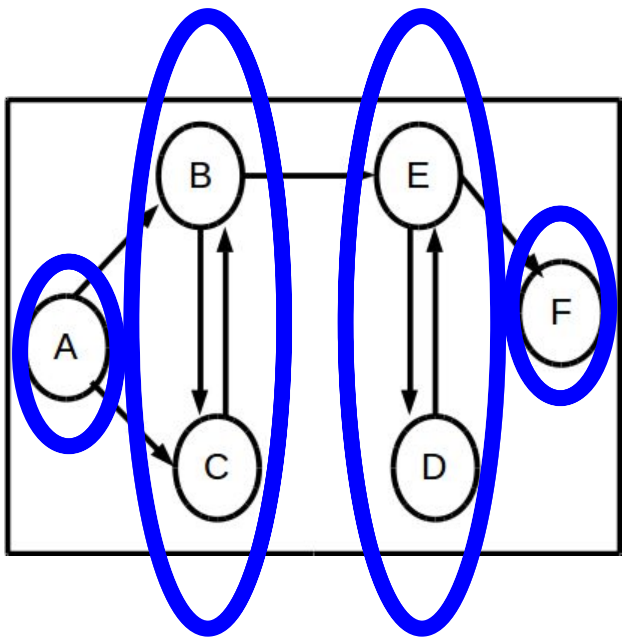
As componentes fortemente conectadas de um grafo direcionado são conjuntos de vértices sob a relação “são mutuamente alcançáveis”.

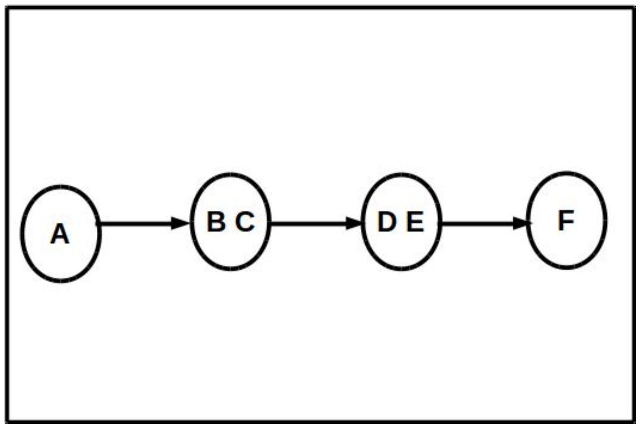


Localização de componentes fortemente conectadas

As componentes fortemente conectadas de um grafo direcionado são conjuntos de vértices sob a relação "são mutuamente alcançáveis".

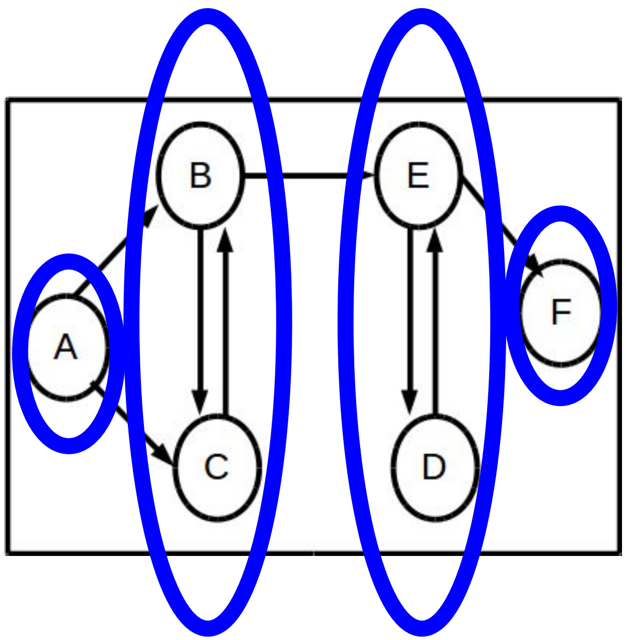
O grafo de componentes é um grafo acíclico orientado



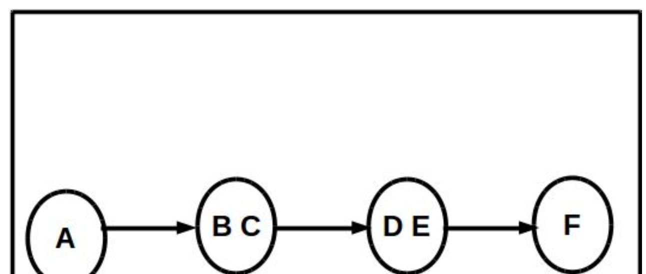


Localização de componentes fortemente conectadas

As componentes fortemente conectadas de um grafo direcionado são conjuntos de vértices sob a relação "são mutuamente alcançáveis".



O grafo de componentes
é um grafo acíclico orientado



$G_{SCC} = (V_{SCC}, A_{SCC})$
V_{SCC} tem um vértice para cada SCC do grafo G
A_{SCC} contém uma aresta se existe uma aresta correspondente entre os SCC's do grafo G

Localização de componentes fortemente conectadas

Algoritmos:

- R. Tarjan. *Depth first search and linear graph algorithms*. SIAM Journal on computing. 1 (2): 146-160, 1972.
- S. R. Kosarayu (não publicado) e M. Sharir. *A strong-connectivity algorithm and its applications in data flow analysis.* Computers & Mathematics with Applications. 7: 67-72, 1981.
- H. N. Gabow. *Path-based depth-first search for strong and biconnected components.* Information Processing Letters. 74: 107-114, 2000

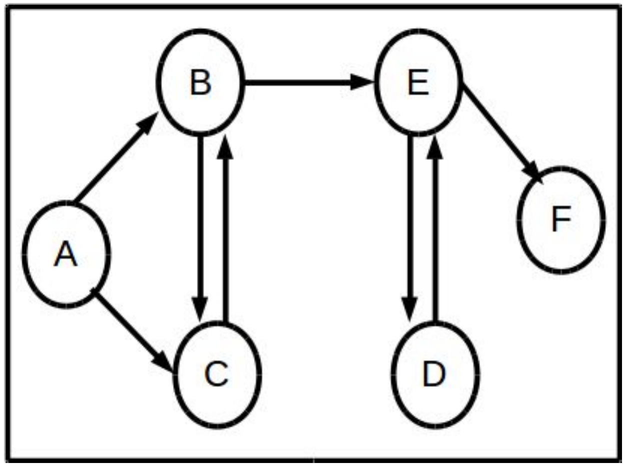
Localização de componentes fortemente conectadas: Kosarayu

- Podemos pensar que cada vez que é executado DFS-Visit(u) na linha 7 do algoritmo DFS (V,A) temos um novo componente forte do grafo.
- Infelizmente, isso só é verdade se o vértice inicial u escolhido em cada chamada for escolhido de uma maneira especial.
- Para fazer essa escolha especial, o algoritmo de Kosaraju começa por colocar os vértices numa certa ordem

https://www.ime.usp.br/~pf/algoritmos_para_grafos/aulas/kosaraju.html

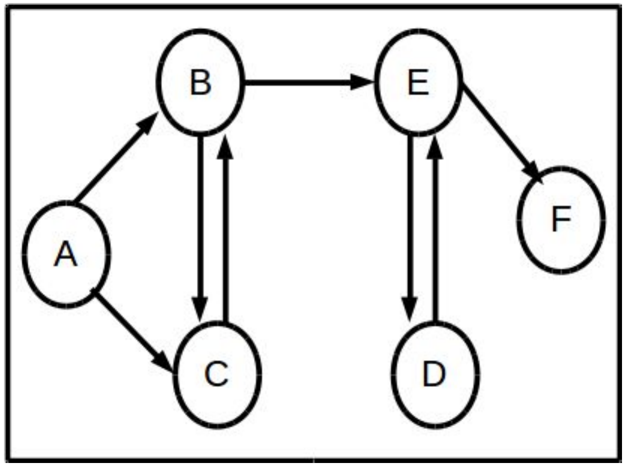
Localização de componentes fortemente conectadas: Kosarayu

Se escolhemos por exemplo a seguinte ordem:
cada vez que é executado DFS-Visit(u)



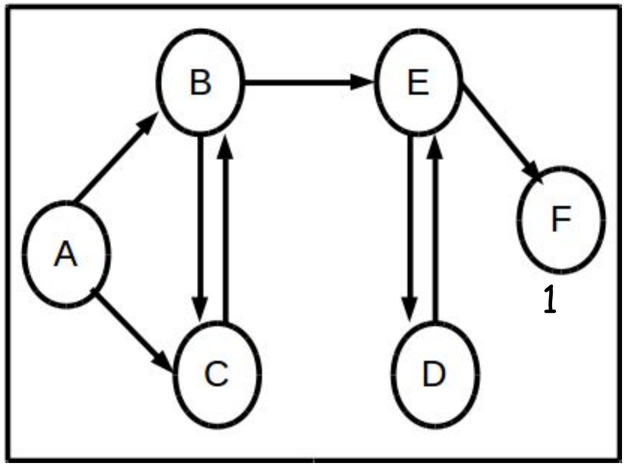
Localização de componentes fortemente conectadas: Kosarayu

Se escolhemos por exemplo a seguinte ordem: F E D B C A
cada vez que é executado DFS-Visit(u)



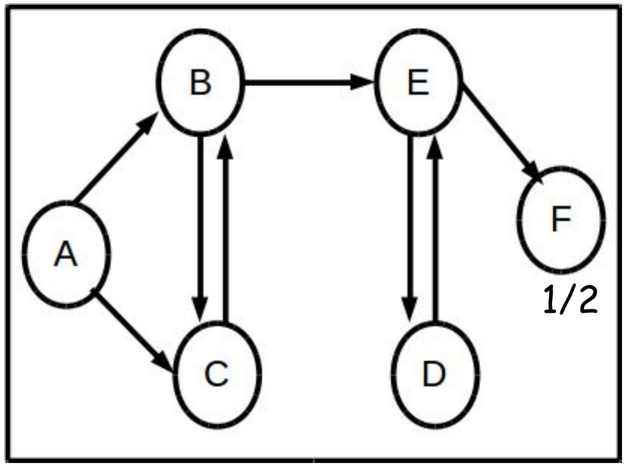
Localização de componentes fortemente conectadas: Kosarayu

Se escolhemos por exemplo a seguinte ordem: F E D B C A
cada vez que é executado DFS-Visit(u)



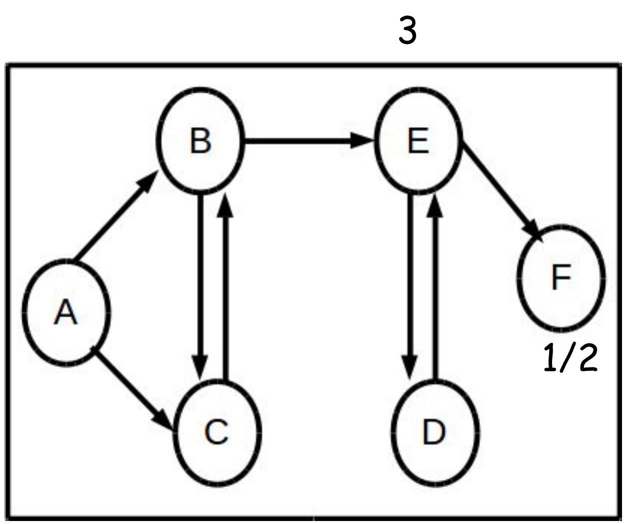
Localização de componentes fortemente conectadas: Kosarayu

Se escolhemos por exemplo a seguinte ordem: F E D B C A
cada vez que é executado DFS-Visit(u)



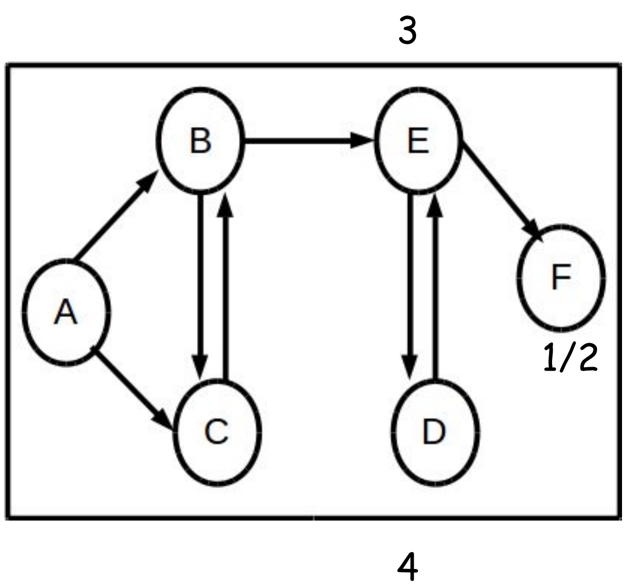
Localização de componentes fortemente conectadas: Kosarayu

Se escolhemos por exemplo a seguinte ordem: F E D B C A
cada vez que é executado DFS-Visit(u)



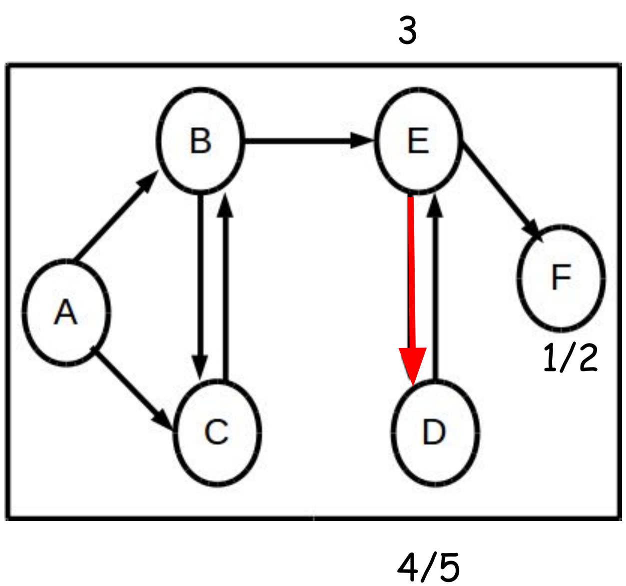
Localização de componentes fortemente conectadas: Kosarayu

Se escolhemos por exemplo a seguinte ordem: F E D B C A
cada vez que é executado DFS-Visit(u)



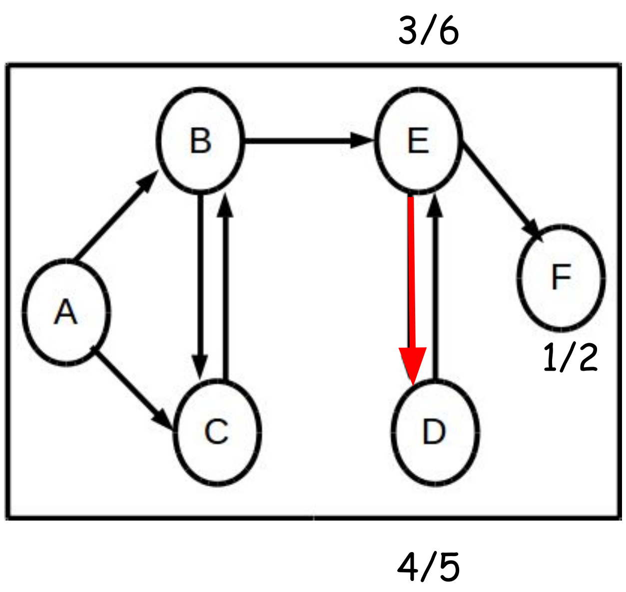
Localização de componentes fortemente conectadas: Kosarayu

Se escolhemos por exemplo a seguinte ordem: F E D B C A
cada vez que é executado DFS-Visit(u)



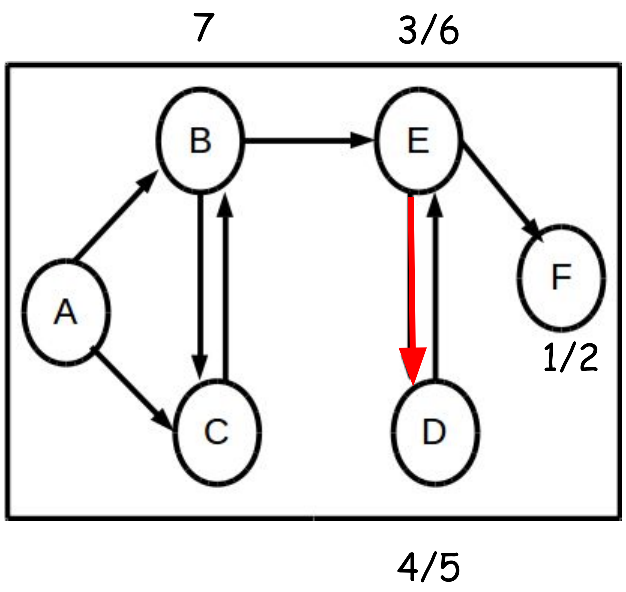
Localização de componentes fortemente conectadas: Kosarayu

Se escolhemos por exemplo a seguinte ordem: F E D B C A
cada vez que é executado DFS-Visit(u)



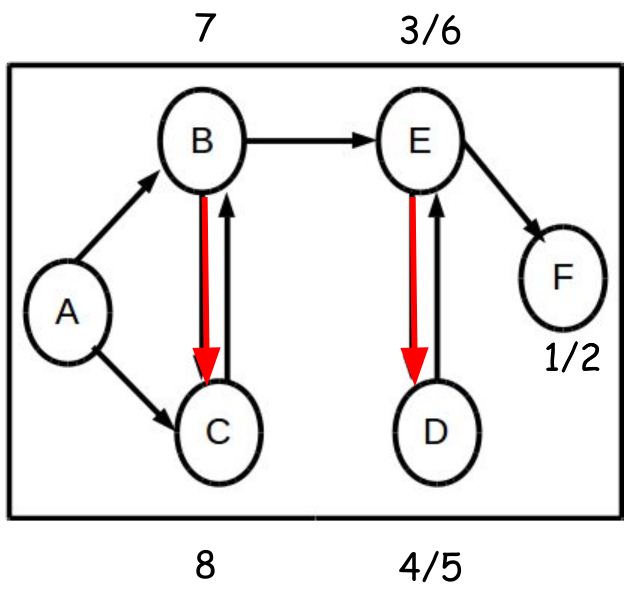
Localização de componentes fortemente conectadas: Kosarayu

Se escolhemos por exemplo a seguinte ordem: F E D B C A
cada vez que é executado DFS-Visit(u)



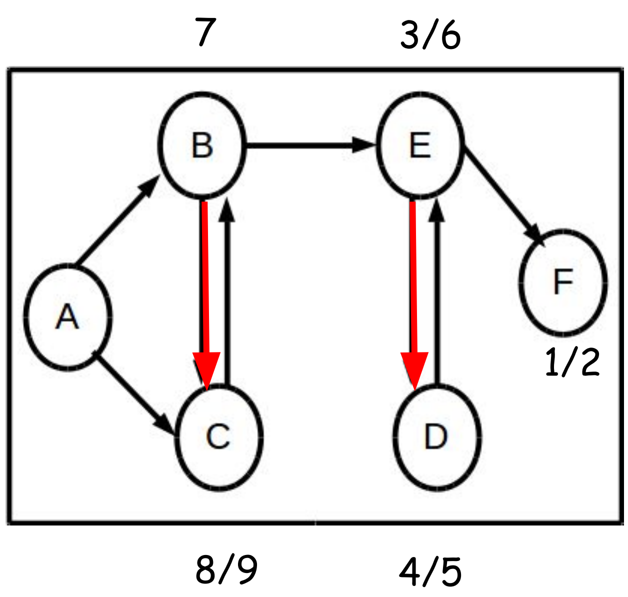
Localização de componentes fortemente conectadas: Kosarayu

Se escolhemos por exemplo a seguinte ordem: F E D B C A
cada vez que é executado DFS-Visit(u)



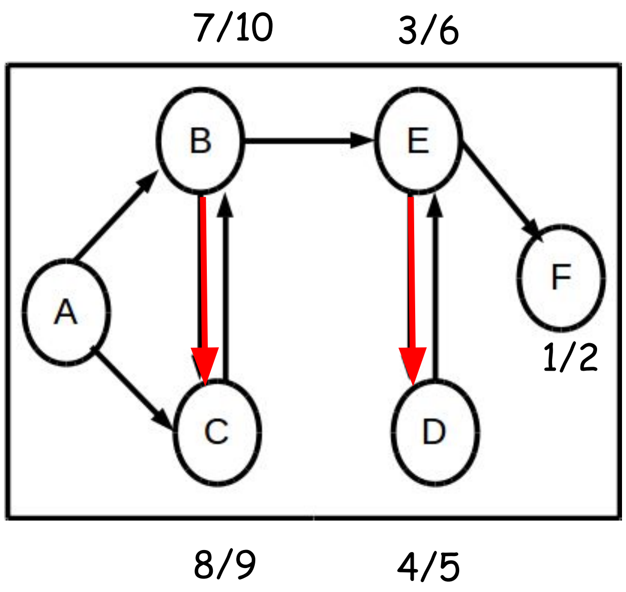
Localização de componentes fortemente conectados: Kosarayu

Se escolhemos por exemplo a seguinte ordem: F E D B C A
cada vez que é executado DFS-Visit(u)



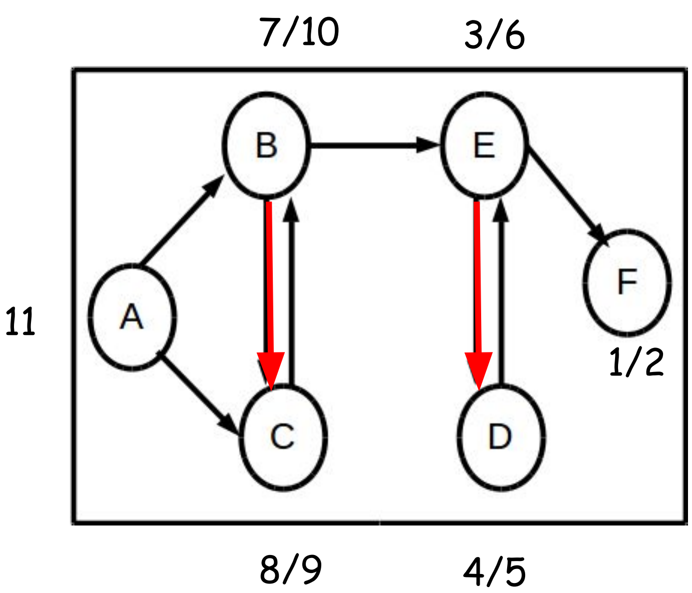
Localização de componentes fortemente conectadas: Kosarayu

Se escolhemos por exemplo a seguinte ordem: F E D B C A
cada vez que é executado DFS-Visit(u)



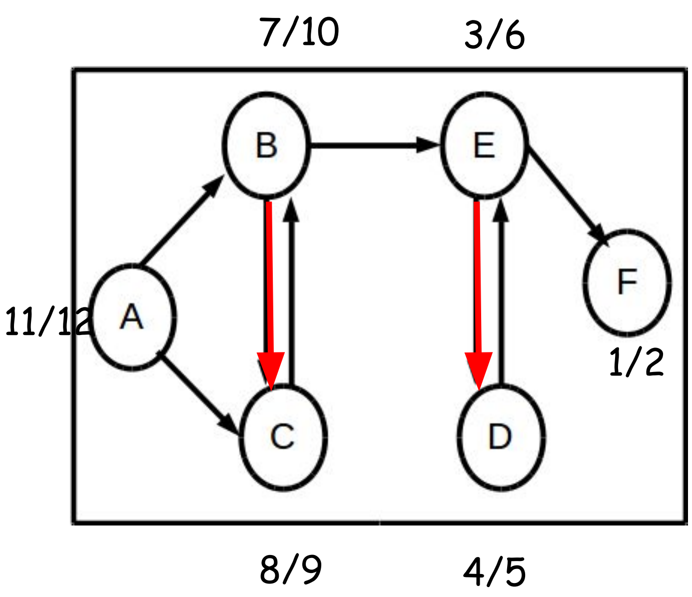
Localização de componentes fortemente conectados: Kosarayu

Se escolhemos por exemplo a seguinte ordem: F E D B C A
cada vez que é executado DFS-Visit(u)



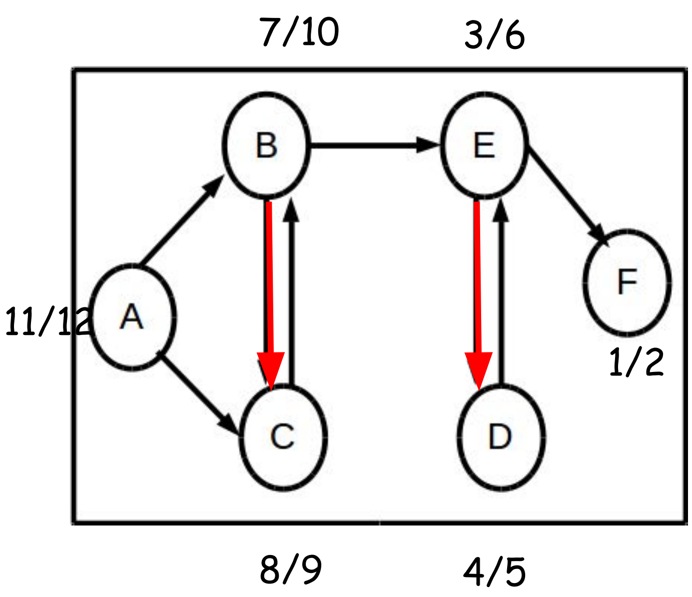
Localização de componentes fortemente conectadas: Kosarayu

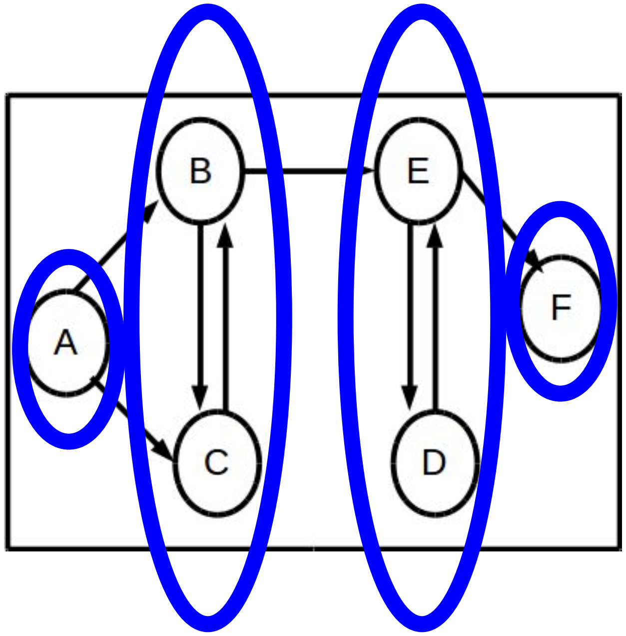
Se escolhemos por exemplo a seguinte ordem: F E D B C A
cada vez que é executado DFS-Visit(u)



Localização de componentes fortemente conectadas: Kosarayu

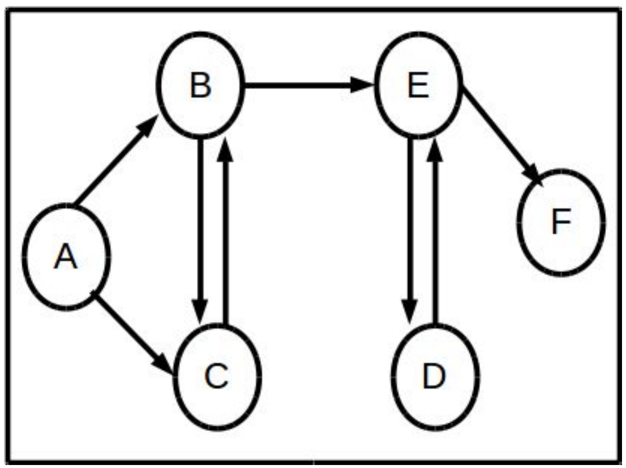
Se escolhemos por exemplo a seguinte ordem: F E D B C A
cada vez que é executado DFS-Visit(u)

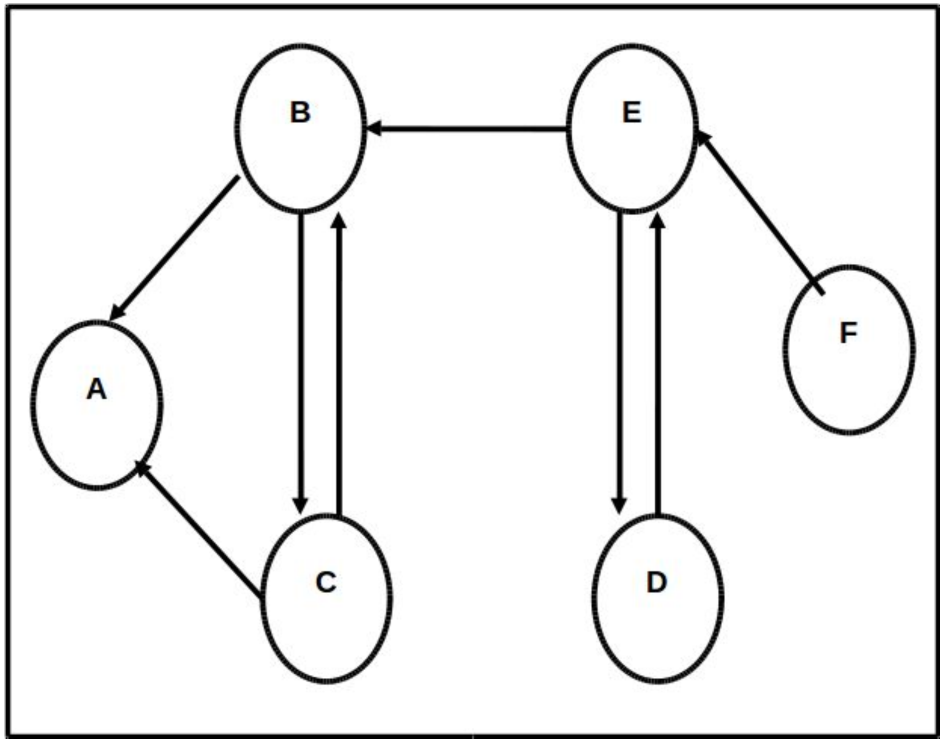




Localização de componentes fortemente conectadas: Kosarayu

Se escolhemos por exemplo a seguinte ordem: F E D B C A
cada vez que é executado DFS-Visit(u)
Como encontramos essa ordem? Criar o grafo G^T





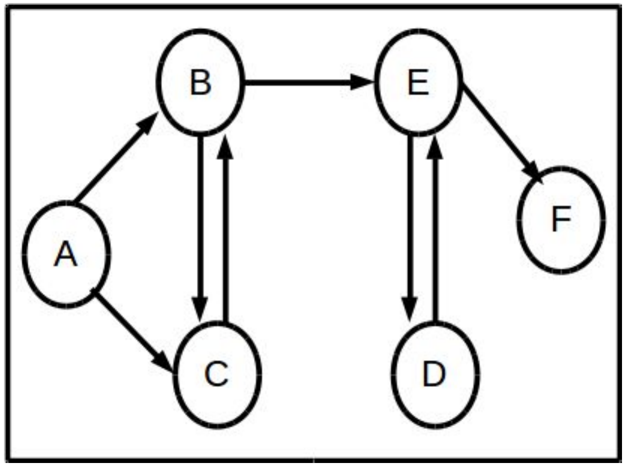
Arestas são invertidas
$G^T=(V,A^T)$

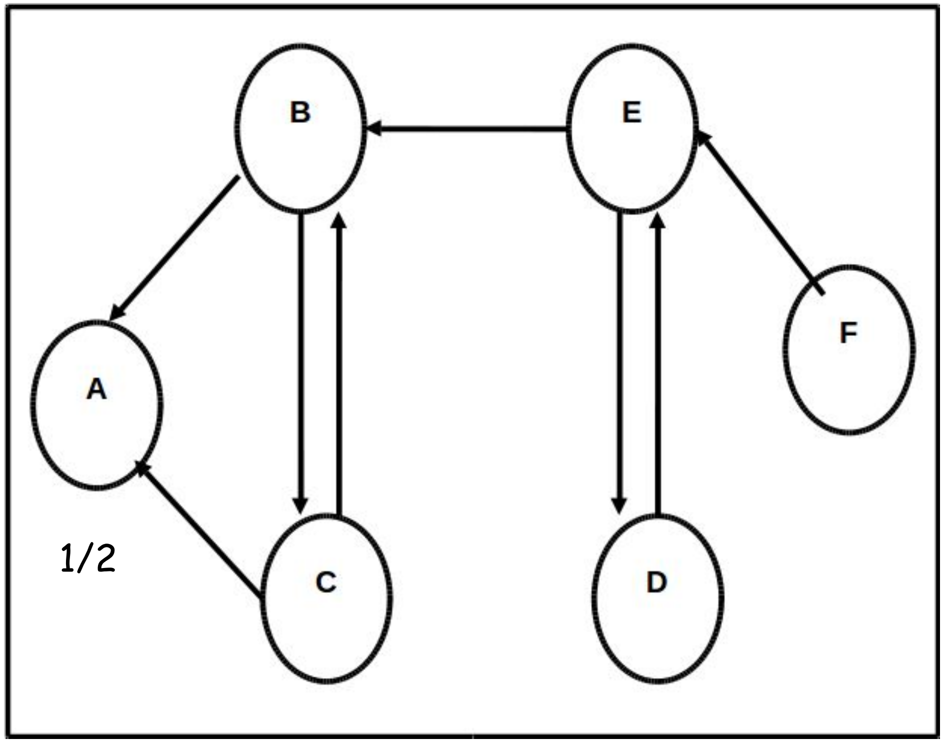
Localização de componentes fortemente conectadas: Kosarayu

Se escolhemos por exemplo a seguinte ordem: F E D B C A

cada vez que é executado DFS-Visit(u)

Como encontramos essa ordem? Criar o grafo G^T





Arestas são invertidas

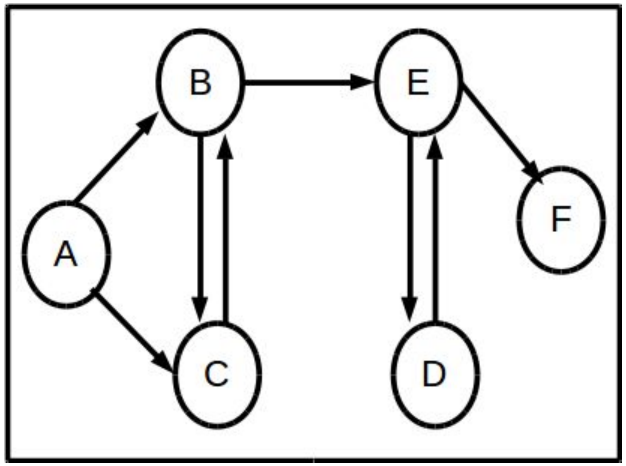
$G^T=(V,A^T)$

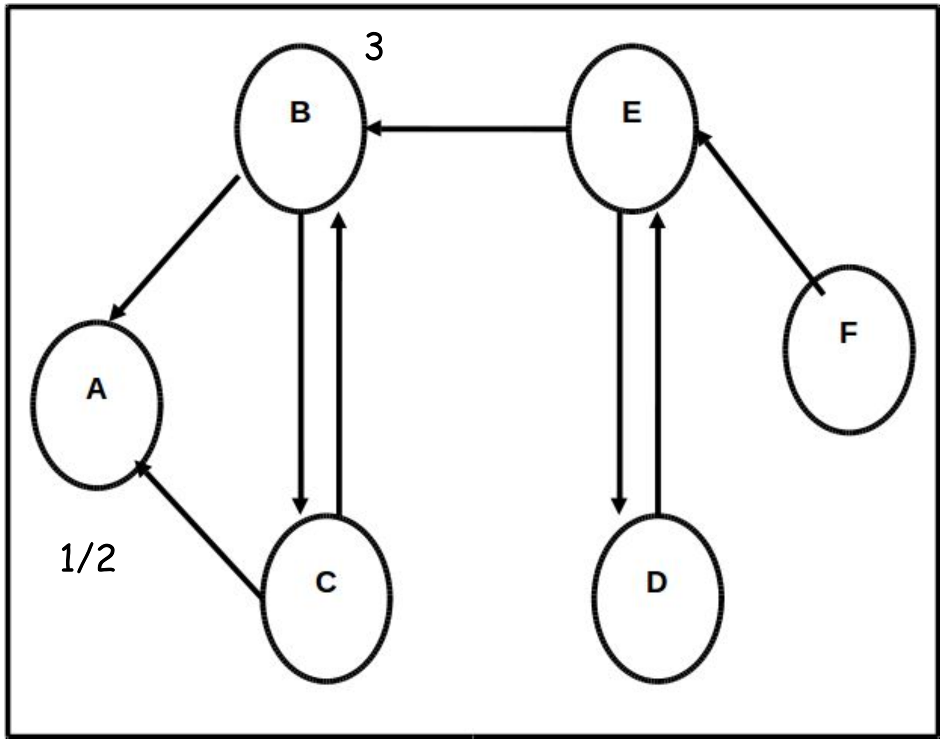
Localização de componentes fortemente conectadas: Kosarayu

Se escolhemos por exemplo a seguinte ordem: F E D B C A

cada vez que é executado DFS-Visit(u)

Como encontramos essa ordem? Criar o grafo G^T





Arestas são invertidas

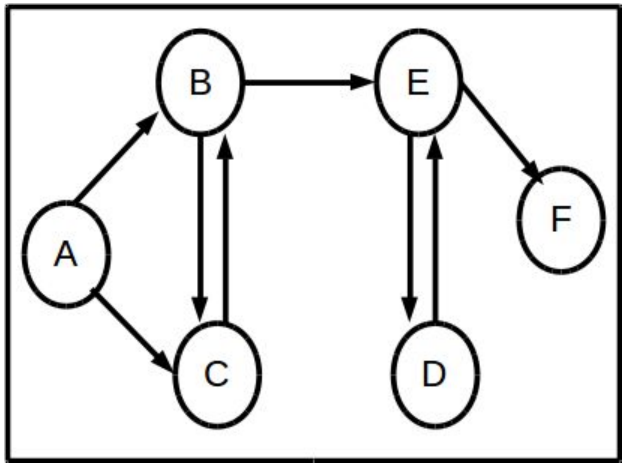
$G^T=(V,A^T)$

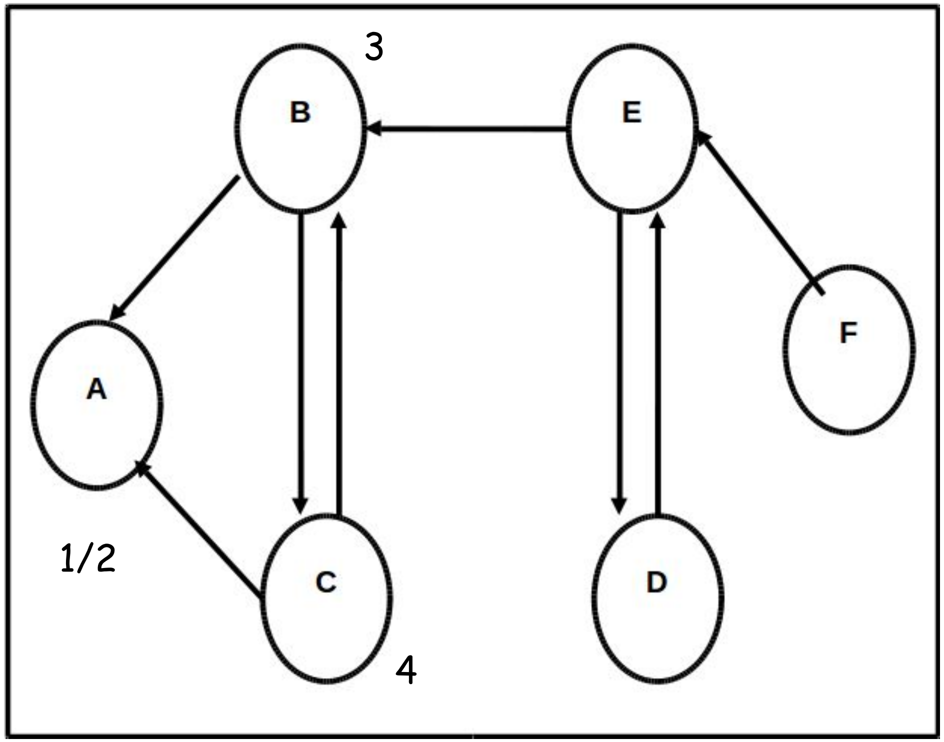
Localização de componentes fortemente conectadas: Kosarayu

Se escolhemos por exemplo a seguinte ordem: F E D B C A

cada vez que é executado DFS-Visit(u)

Como encontramos essa ordem? Criar o grafo G^T





Arestas são invertidas

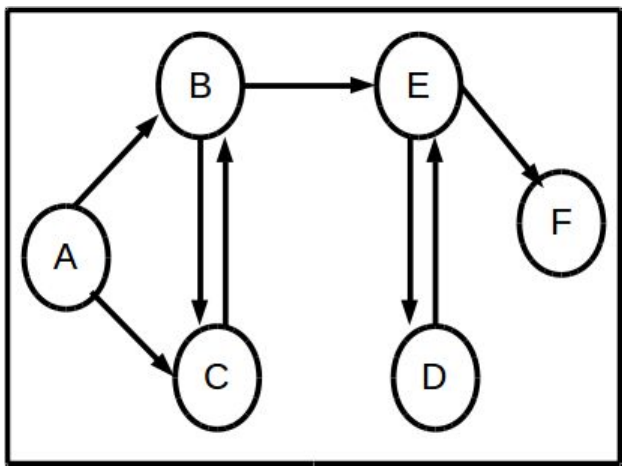
$G^T=(V,A^T)$

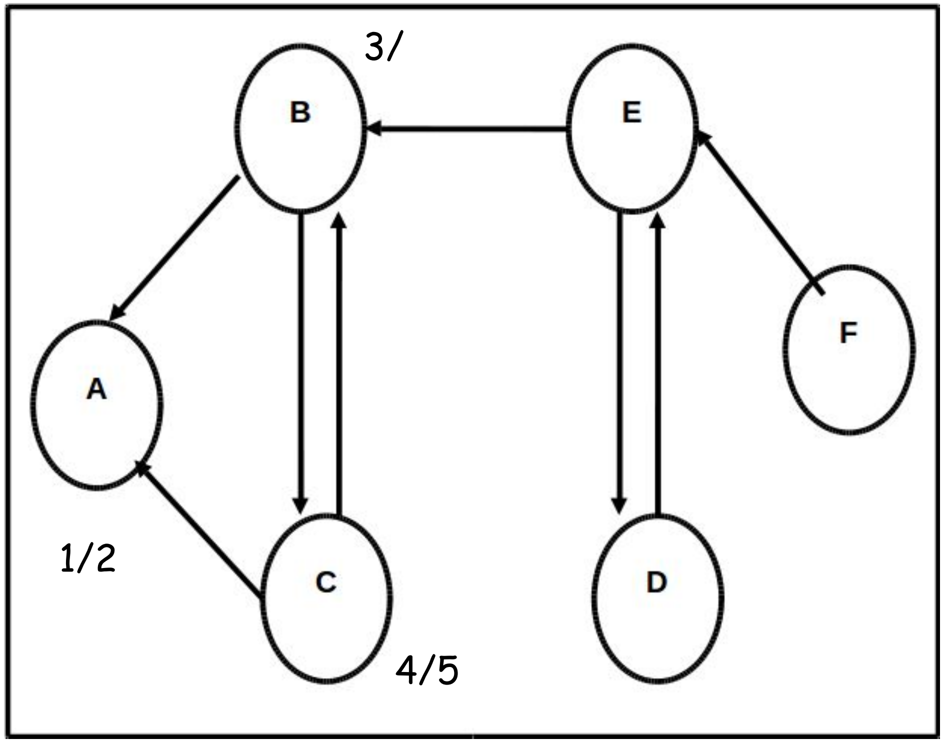
Localização de componentes fortemente conectadas: Kosarayu

Se escolhemos por exemplo a seguinte ordem: F E D B C A

cada vez que é executado DFS-Visit(u)

Como encontramos essa ordem? Criar o grafo G^T





Arestas são invertidas

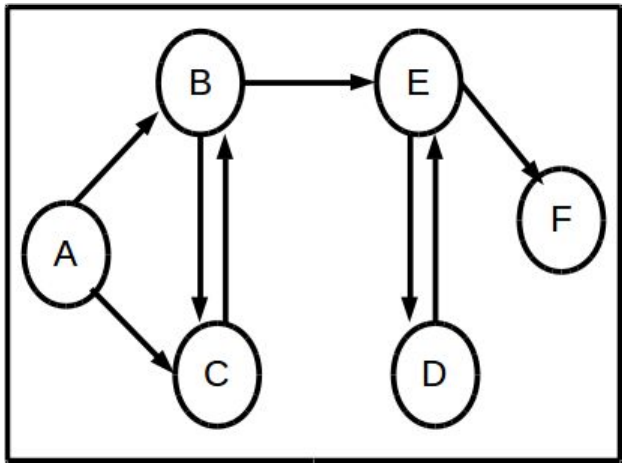
$G^T=(V,A^T)$

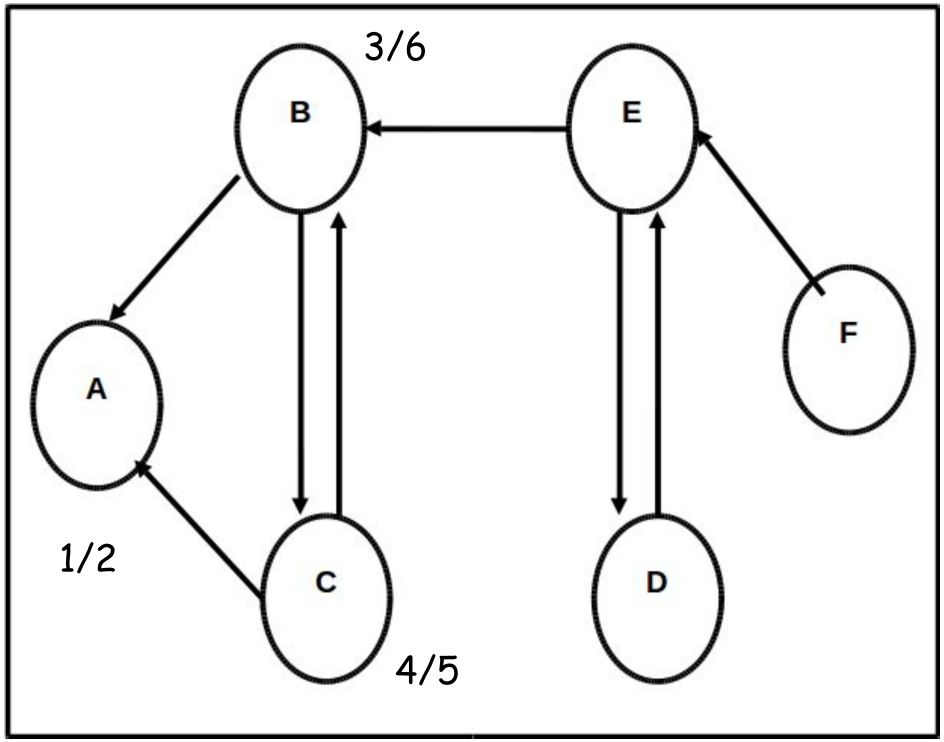
Localização de componentes fortemente conectadas: Kosarayu

Se escolhemos por exemplo a seguinte ordem: F E D B C A

cada vez que é executado DFS-Visit(u)

Como encontramos essa ordem? Criar o grafo G^T





Arestas são invertidas

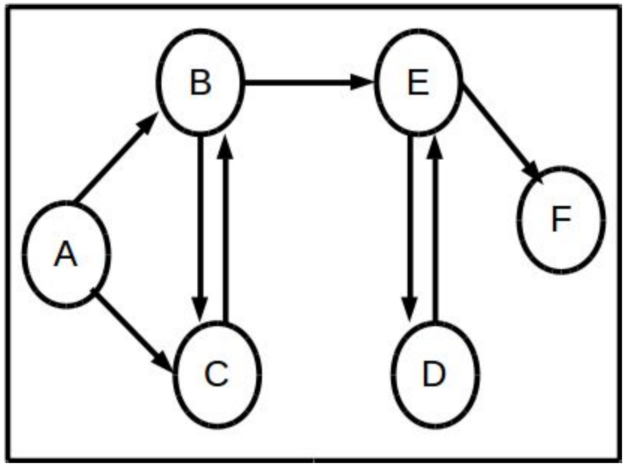
$G^T=(V,A^T)$

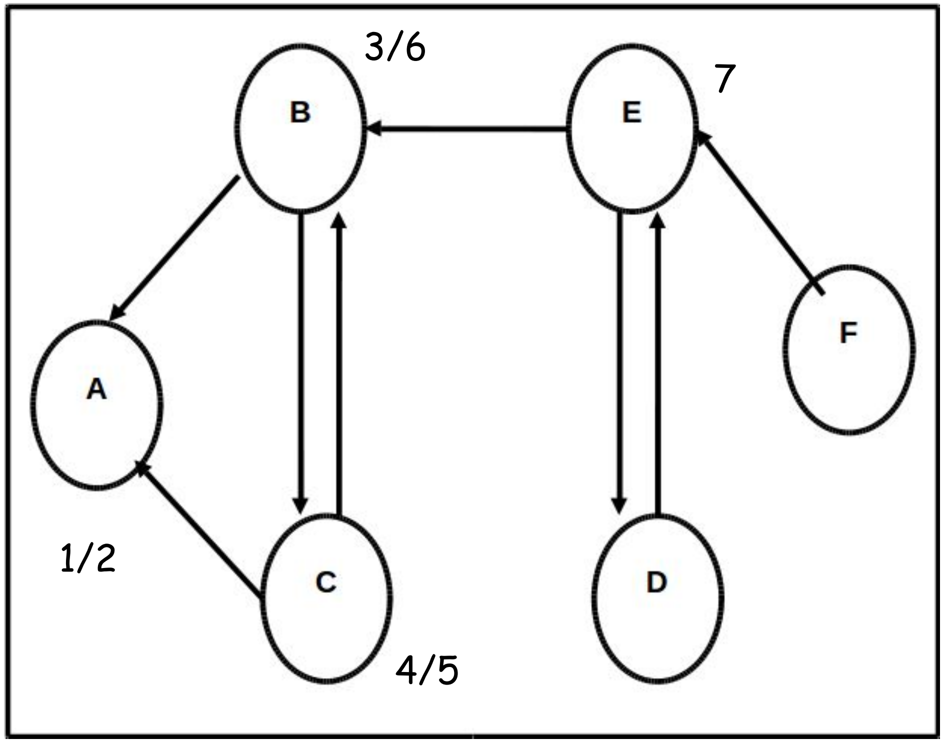
Localização de componentes fortemente conectadas: Kosarayu

Se escolhemos por exemplo a seguinte ordem: F E D B C A

cada vez que é executado DFS-Visit(u)

Como encontramos essa ordem? Criar o grafo G^T





Arestas são invertidas

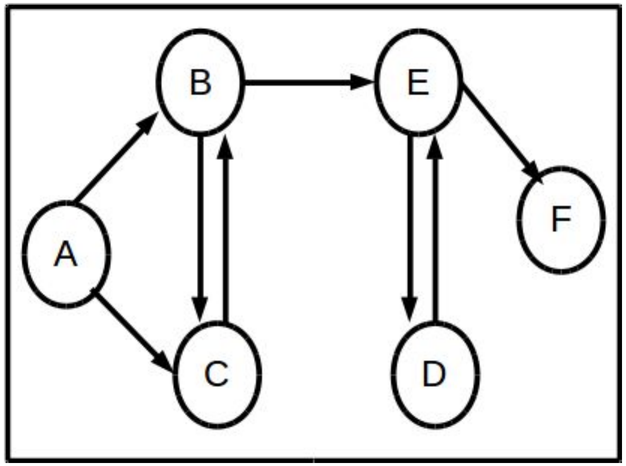
$G^T=(V,A^T)$

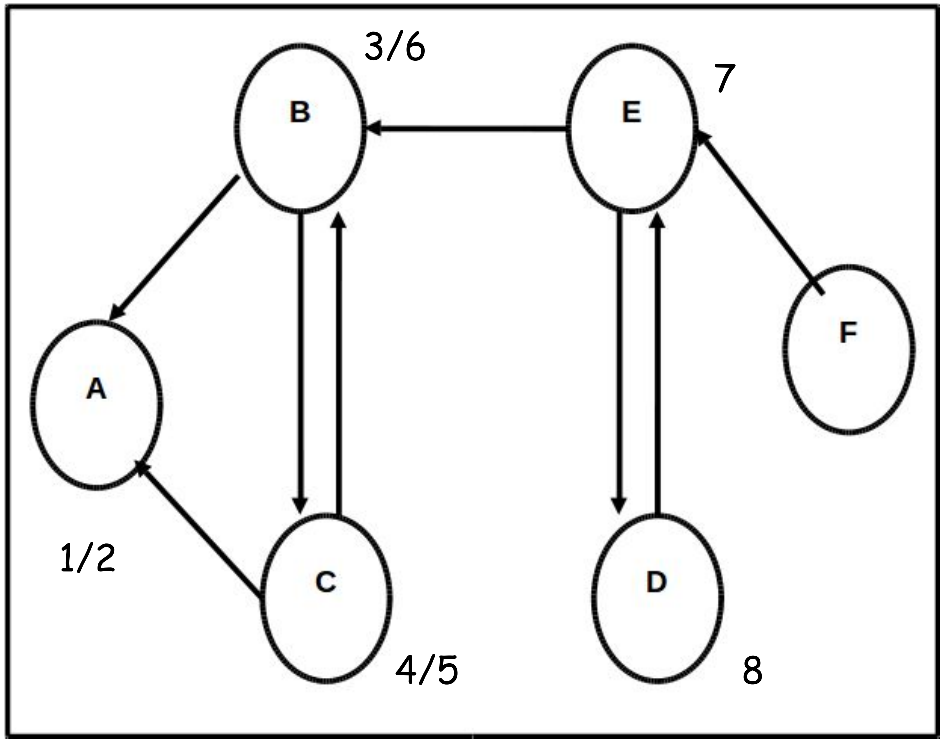
Localização de componentes fortemente conectadas: Kosarayu

Se escolhemos por exemplo a seguinte ordem: F E D B C A

cada vez que é executado DFS-Visit(u)

Como encontramos essa ordem? Criar o grafo G^T





Arestas são invertidas

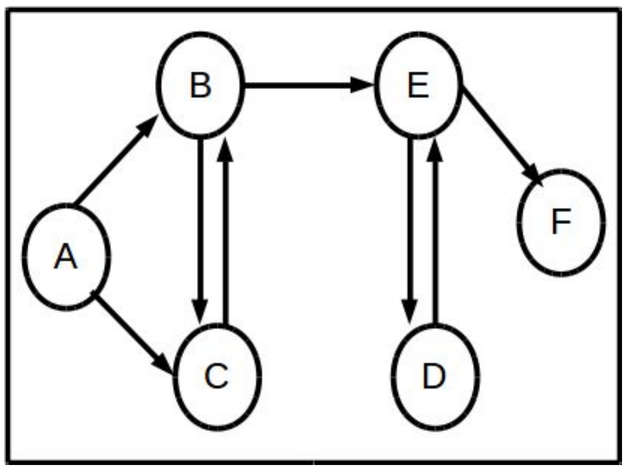
$G^T=(V,A^T)$

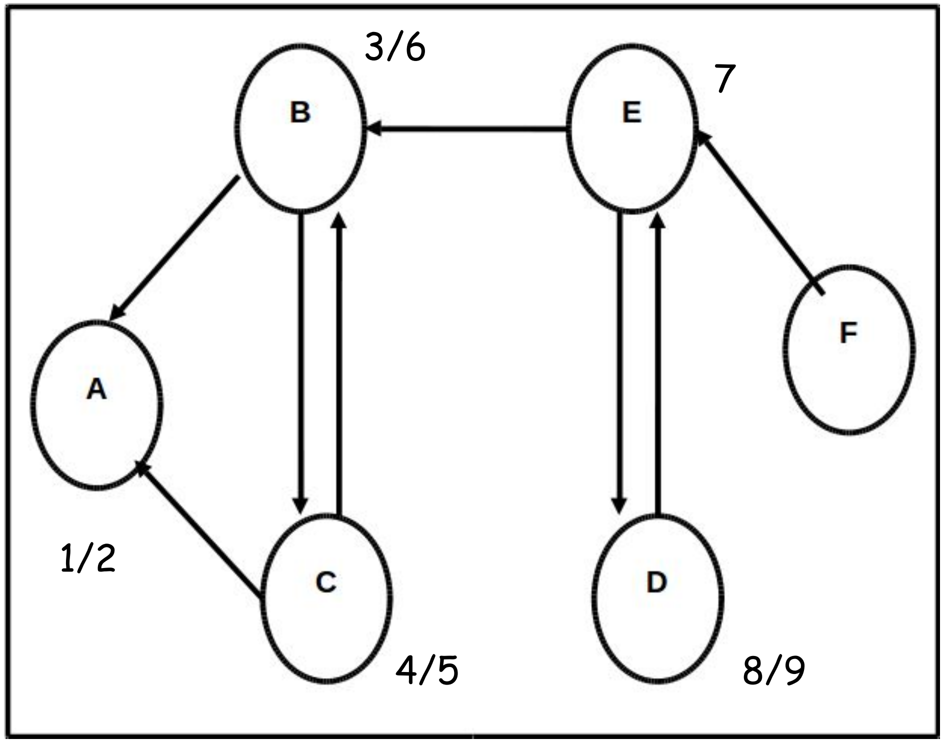
Localização de componentes fortemente conectadas: Kosarayu

Se escolhemos por exemplo a seguinte ordem: F E D B C A

cada vez que é executado DFS-Visit(u)

Como encontramos essa ordem? Criar o grafo G^T





Arestas são invertidas

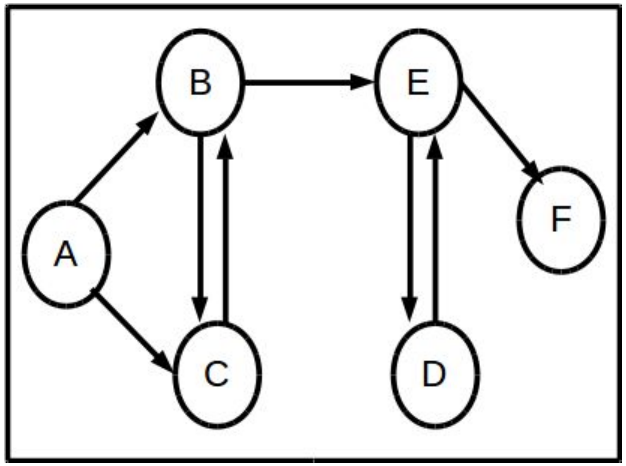
$G^T=(V,A^T)$

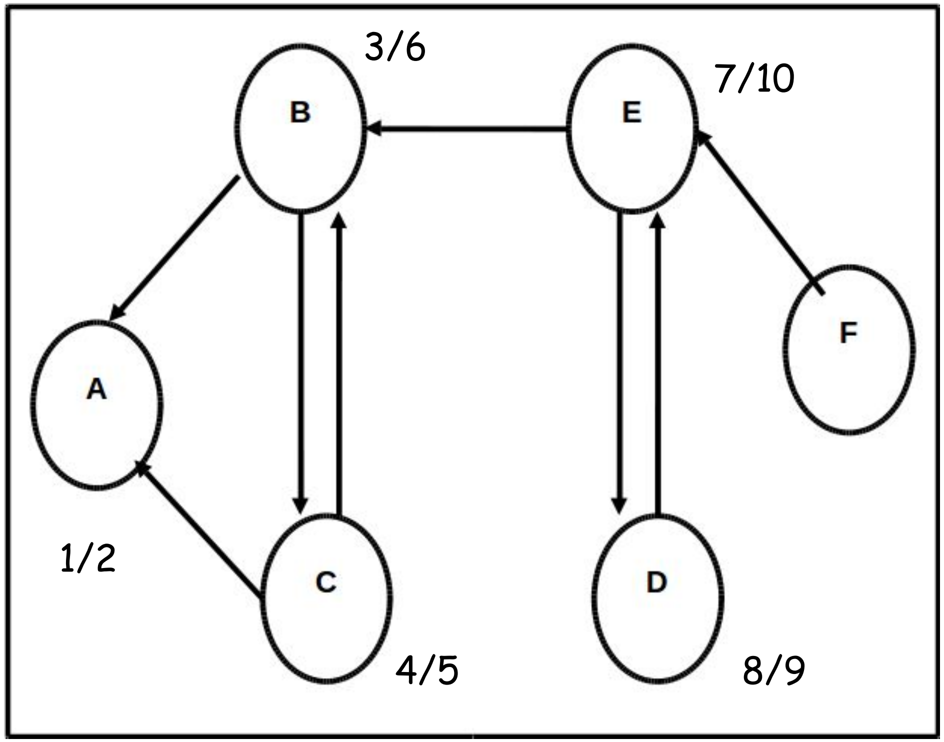
Localização de componentes fortemente conectadas: Kosarayu

Se escolhemos por exemplo a seguinte ordem: F E D B C A

cada vez que é executado DFS-Visit(u)

Como encontramos essa ordem? Criar o grafo G^T





Arestas são invertidas

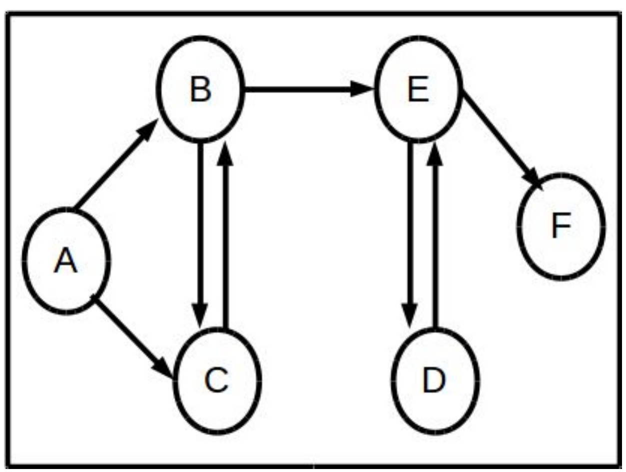
$G^T=(V,A^T)$

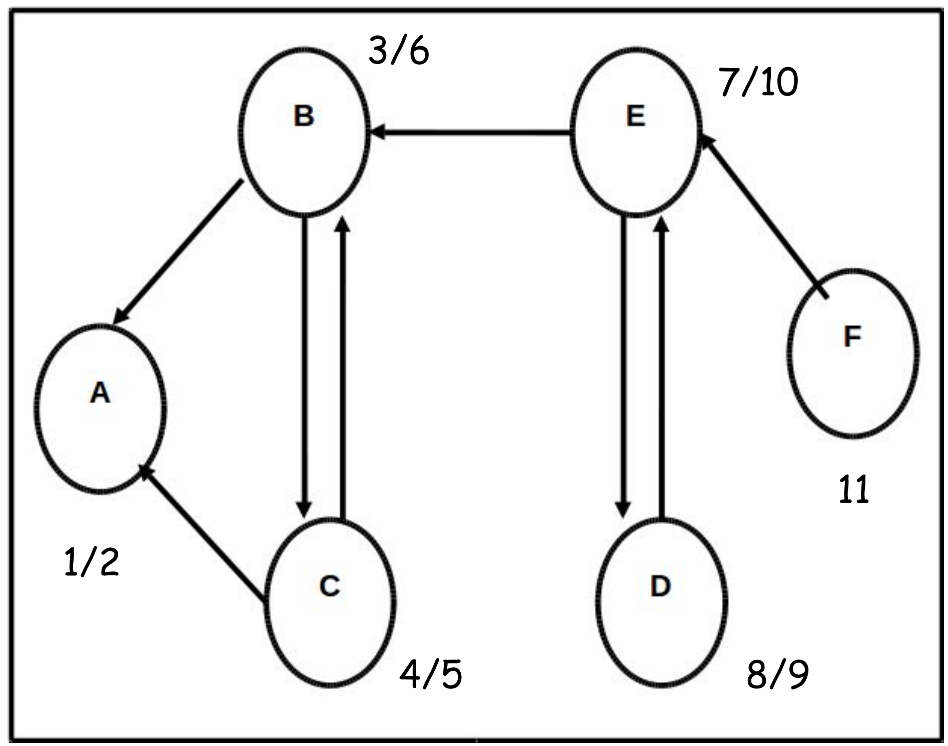
Localização de componentes fortemente conectadas: Kosarayu

Se escolhemos por exemplo a seguinte ordem: F E D B C A

cada vez que é executado DFS-Visit(u)

Como encontramos essa ordem? Criar o grafo G^T





Arestas são invertidas

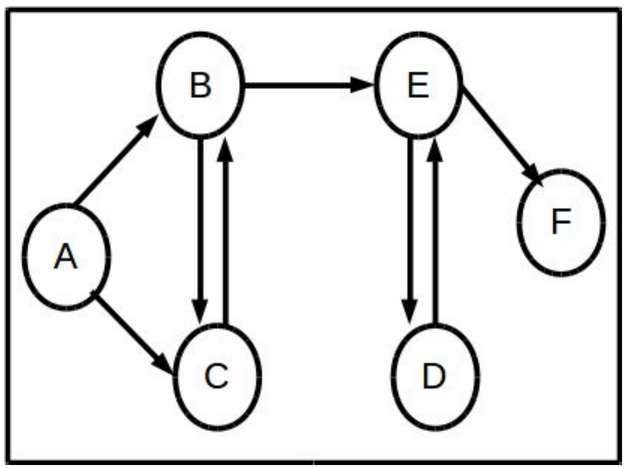
$G^T=(V,A^T)$

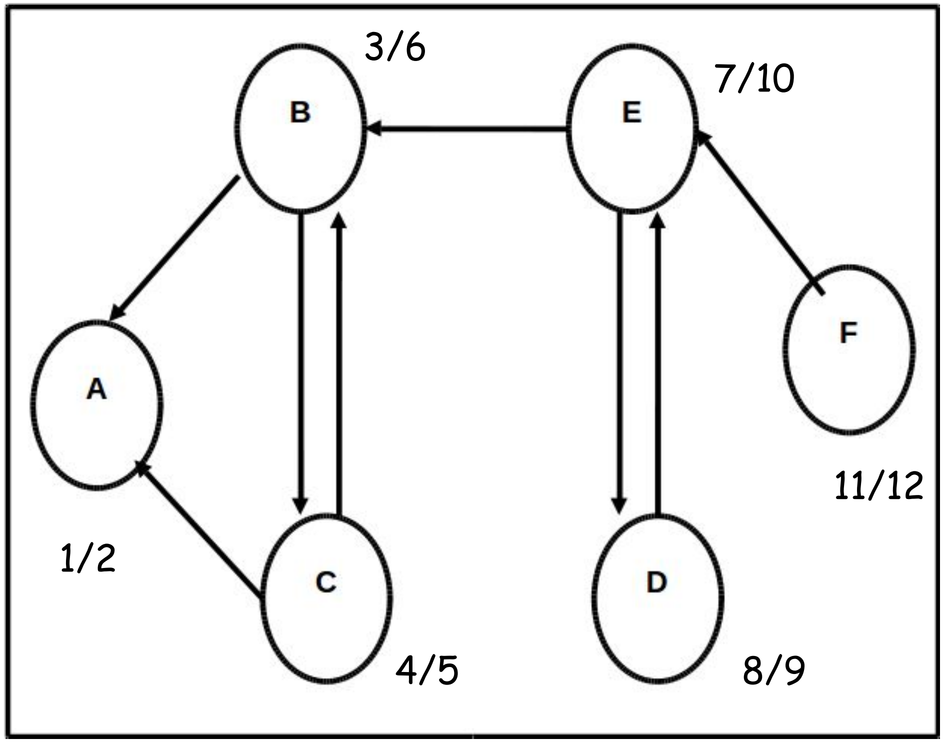
Localização de componentes fortemente conectadas: Kosarayu

Se escolhemos por exemplo a seguinte ordem: F E D B C A

cada vez que é executado DFS-Visit(u)

Como encontramos essa ordem? Criar o grafo G^T





Arestas são invertidas

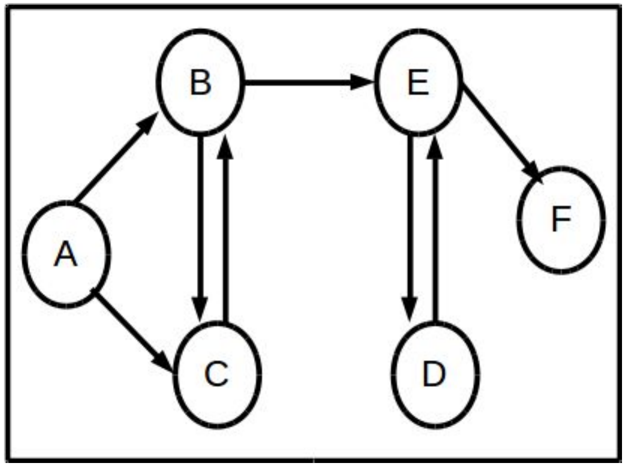
$G^T=(V,A^T)$

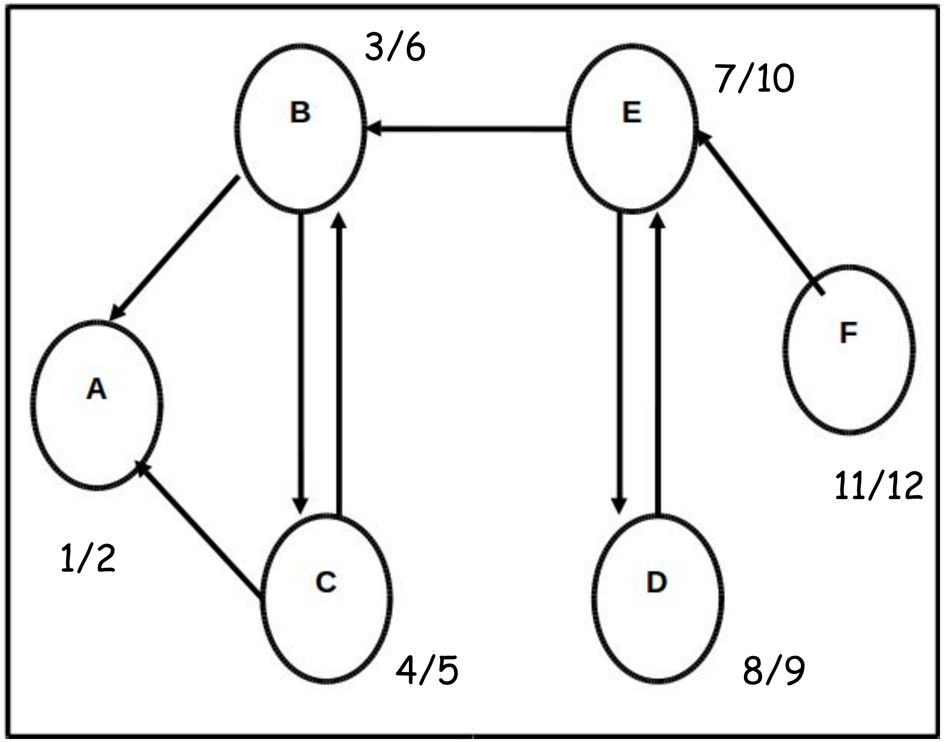
Localização de componentes fortemente conectadas: Kosarayu

Se escolhemos por exemplo a seguinte ordem: F E D B C A

cada vez que é executado DFS-Visit(u)

Como encontramos essa ordem? Criar o grafo G^T e coloco em ordem decrescente de f[u]





Arestas são invertidas

$G^T=(V,A^T)$

Localização de componentes fortemente conectadas: Kosarayu

SCCs (V, A)

1. Calcular A^T
2. Chamar DFS (V, A^T) para calcular f[u]
3. Chamar DFS (V, A) considerando no laço principal os vértices em ordem decrescente de f (calculados na linha 2)
4. Devolva os vértices de cada árvore resultante da linha 3 como uma componente fortemente conectada separada

Localização de componentes fortemente conectadas: Kosarayu

SCCs (V, A)

1. Calcular A^T
2. Chamar DFS (V, A^T) para calcular f[u]
3. Chamar DFS (V, A) considerando no laço principal os vértices em ordem decrescente de f (calculados na linha 2)
4. Devolva os vértices de cada árvore resultante da linha 3 como uma componente fortemente conectada separada

E se fazemos o DFS primeiro no grafo original para calcular f e depois processamos o grafo reverso na ordem inversa de f. Será que conseguimos também encontrar as componentes fortemente conectadas?

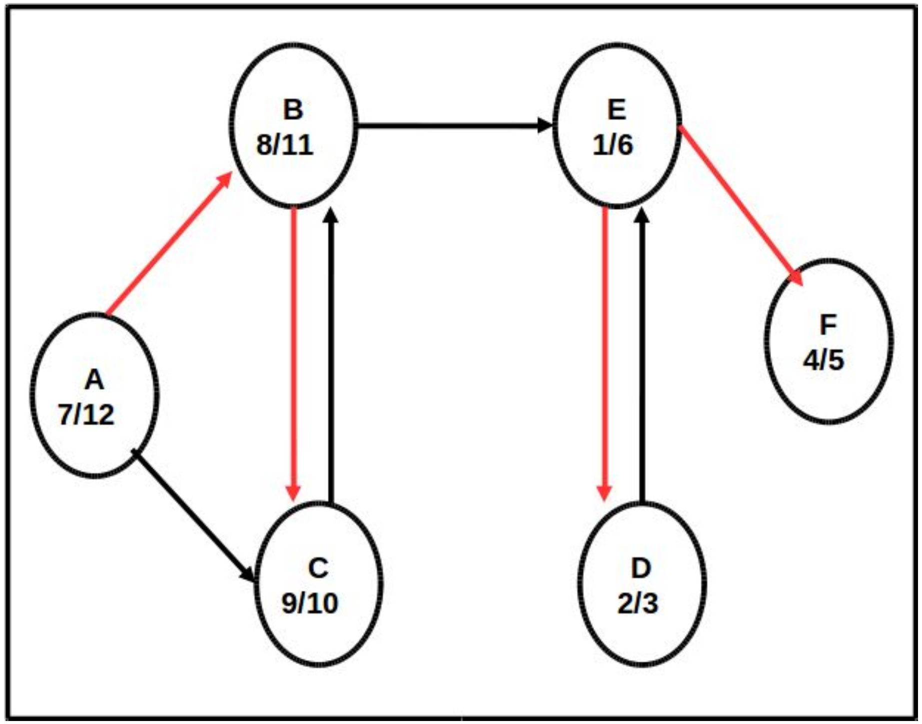
Localização de componentes fortemente conectadas: Kosarayu

SCCs (V, A)

1. Calcular A^T
2. Chamar DFS (V, A^T) para calcular f[u]
3. Chamar DFS (V, A) considerando no laço principal os vértices em ordem decrescente de f (calculados na linha 2)
4. Devolva os vértices de cada árvore resultante da linha 3 como uma componente fortemente conectada separada

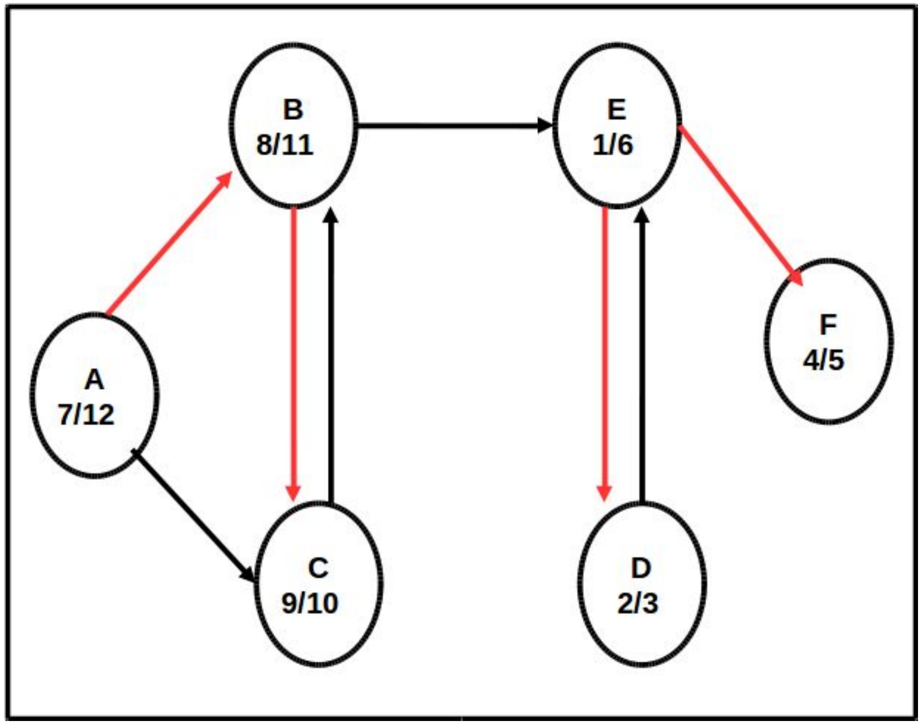
Pode-se mostrar que o grafo das componentes fortes de G é o transposto do grafo das componentes fortes de G^T

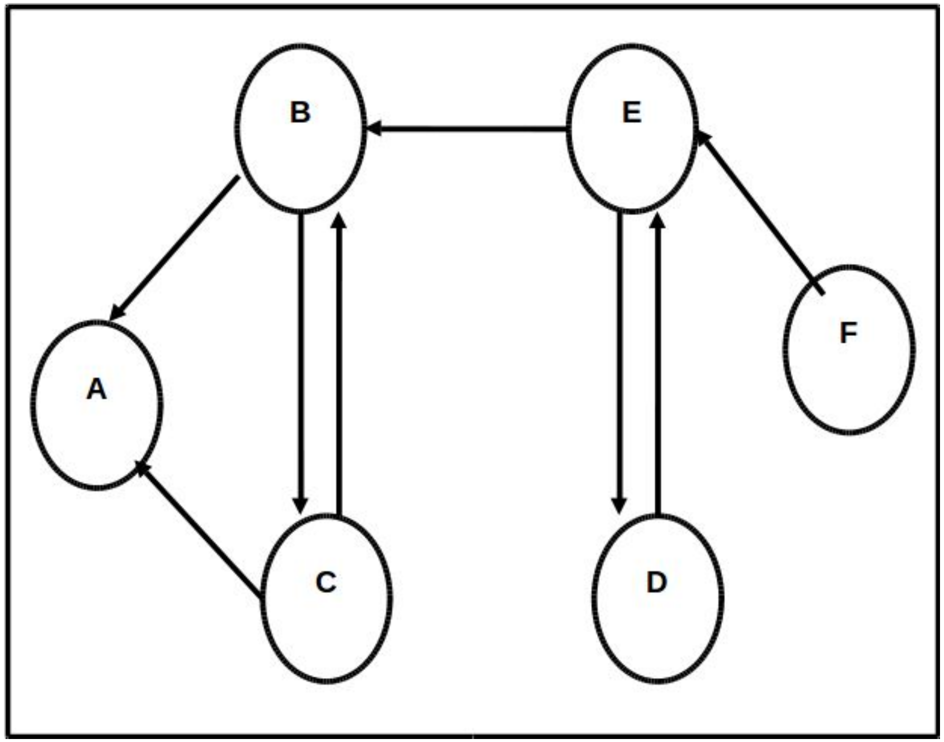
Localização de componentes fortemente conectadas: Kosarayu



Busca em profundidade no grafo original G(V,A)

Localização de componentes fortemente conectadas: Kosarayu

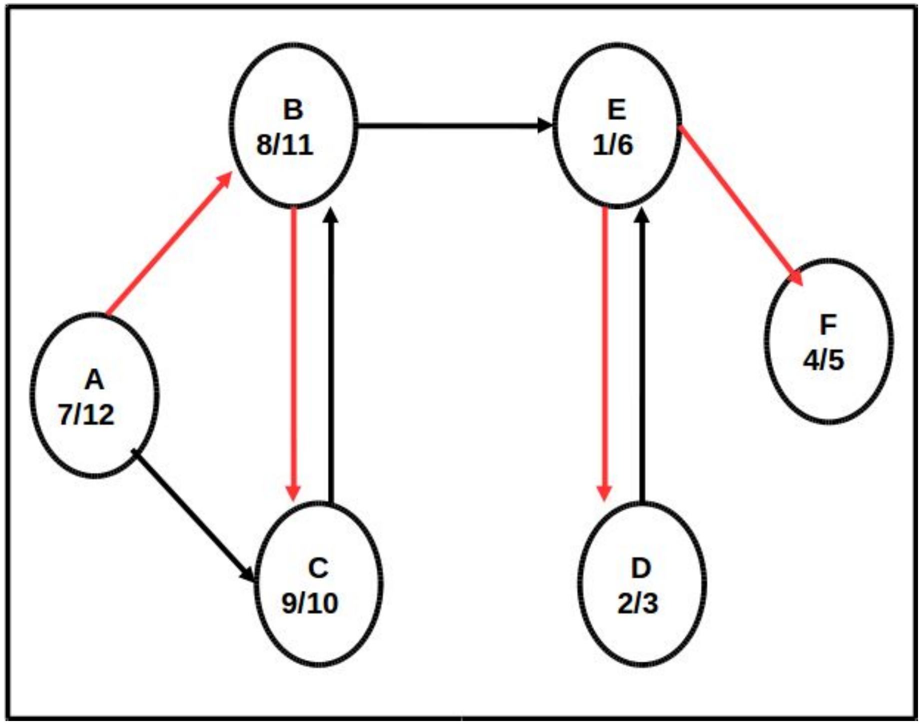




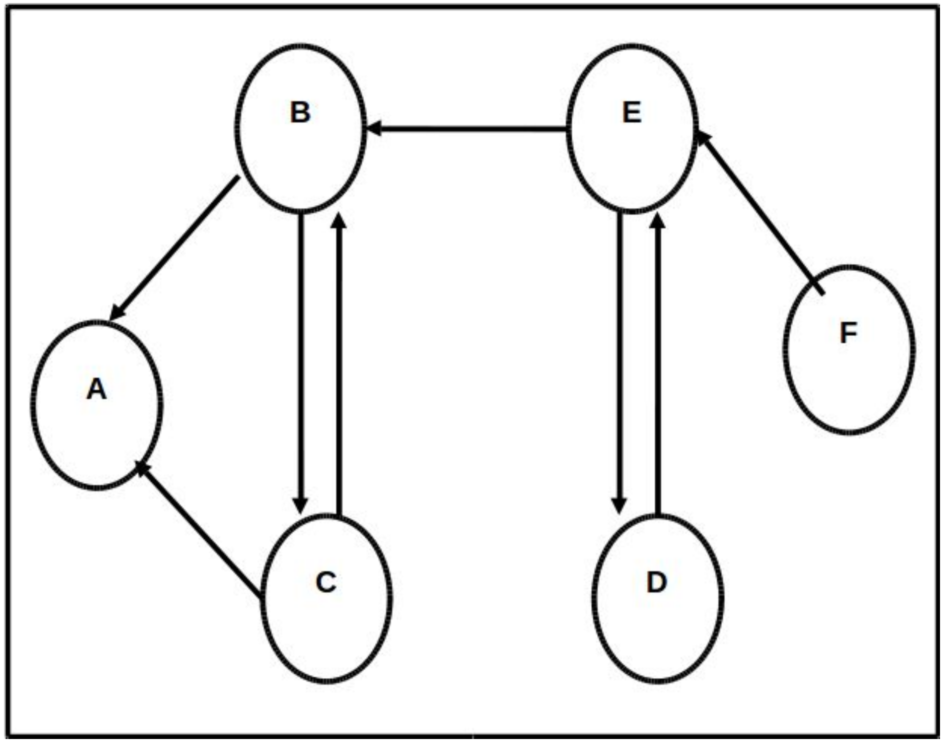
Busca em profundidade no grafo original G(V,A)

Arestas são invertidas $G^T=(V,A^T)$

Localização de componentes fortemente conectadas: Kosarayu

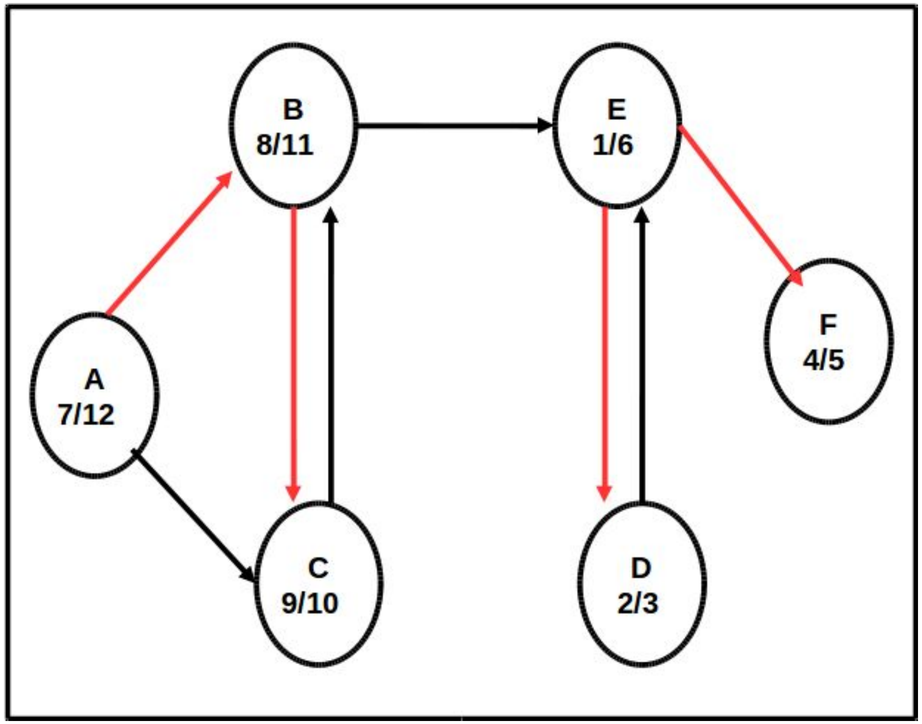


Busca em profundidade no grafo original G(V,A)

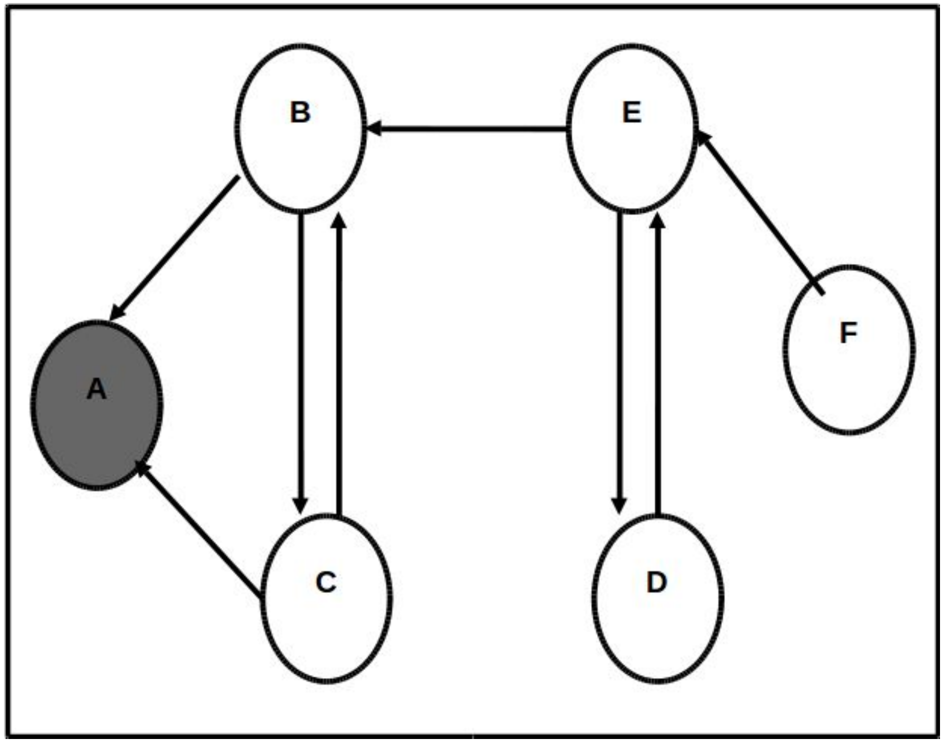


Busca em profundidade no grafo $G^T=(V,A^T)$ seguindo a ordenação topológica da primeira busca (A B C E F D)

Localização de componentes fortemente conectadas: Kosarayu

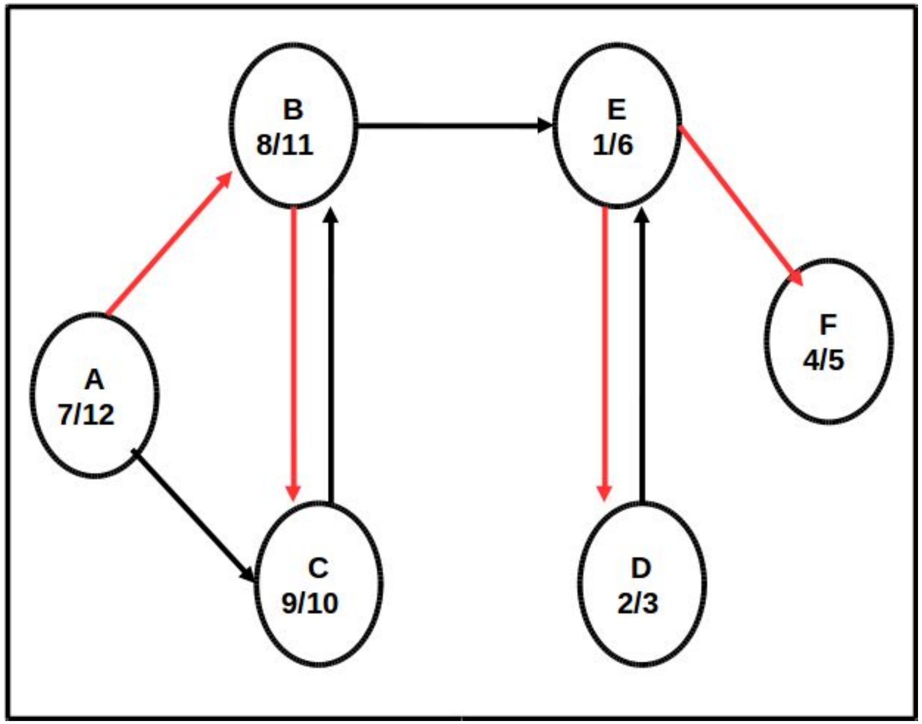


Busca em profundidade no grafo original G(V,A)

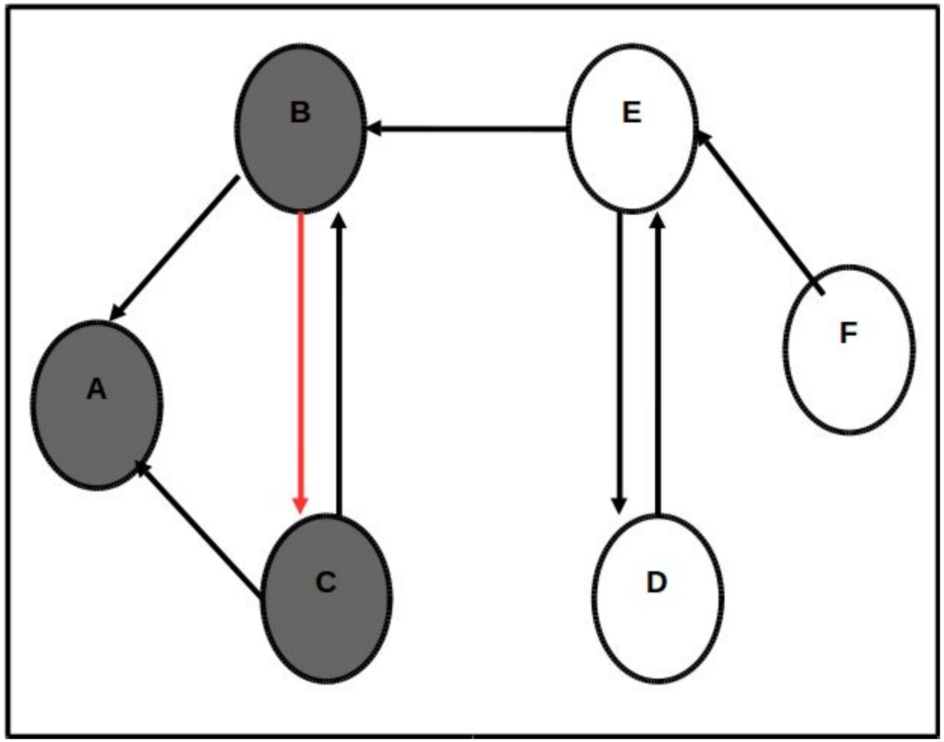


Busca em profundidade no grafo $G^T=(V,A^T)$ seguindo a ordenação topológica da primeira busca (A B C E F D)

Localização de componentes fortemente conectadas: Kosarayu

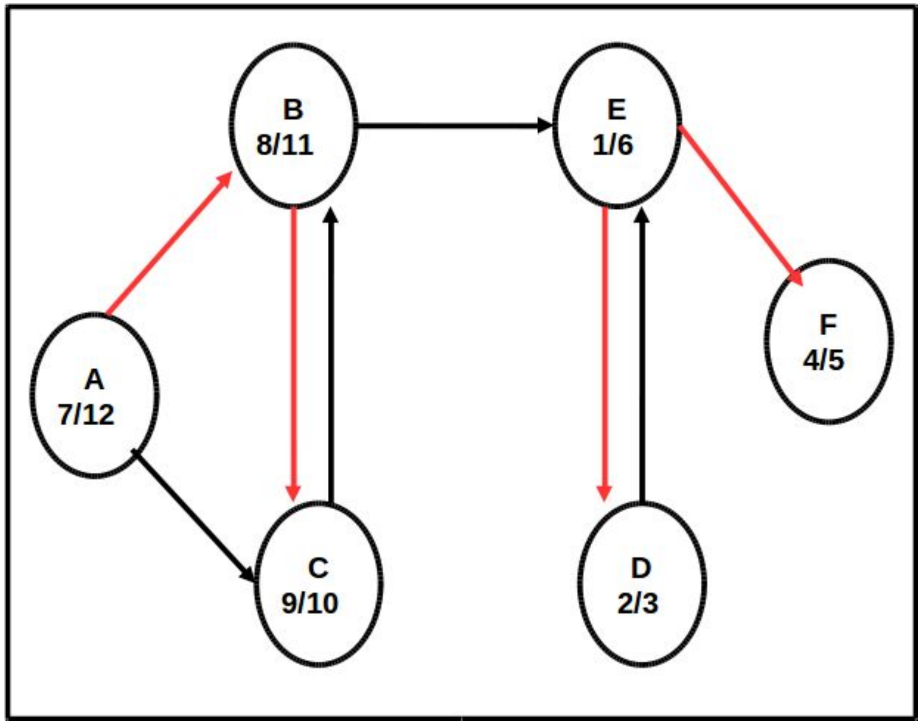


Busca em profundidade no grafo original G(V,A)

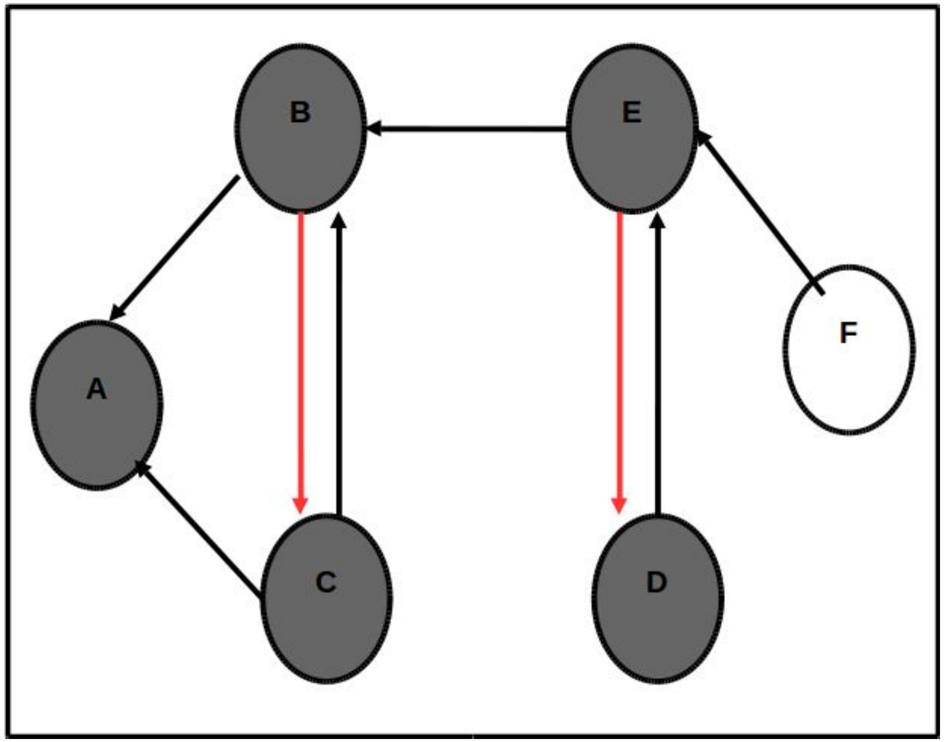


Busca em profundidade no grafo $G^T=(V,A^T)$ seguindo a ordenação topológica da primeira busca (A B C E F D)

Localização de componentes fortemente conectadas: Kosarayu

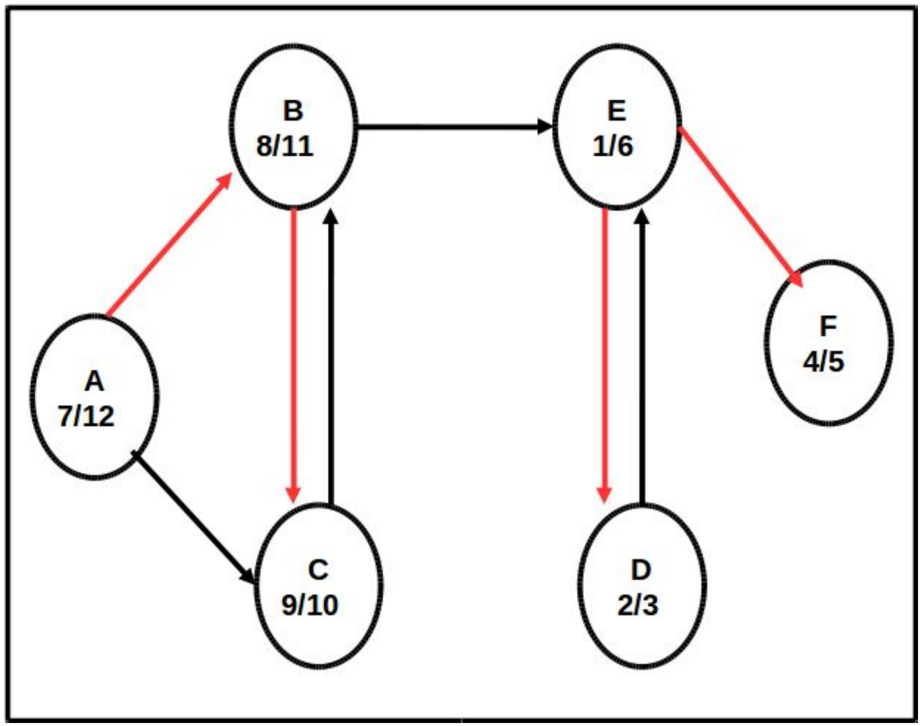


Busca em profundidade no grafo original G(V,A)

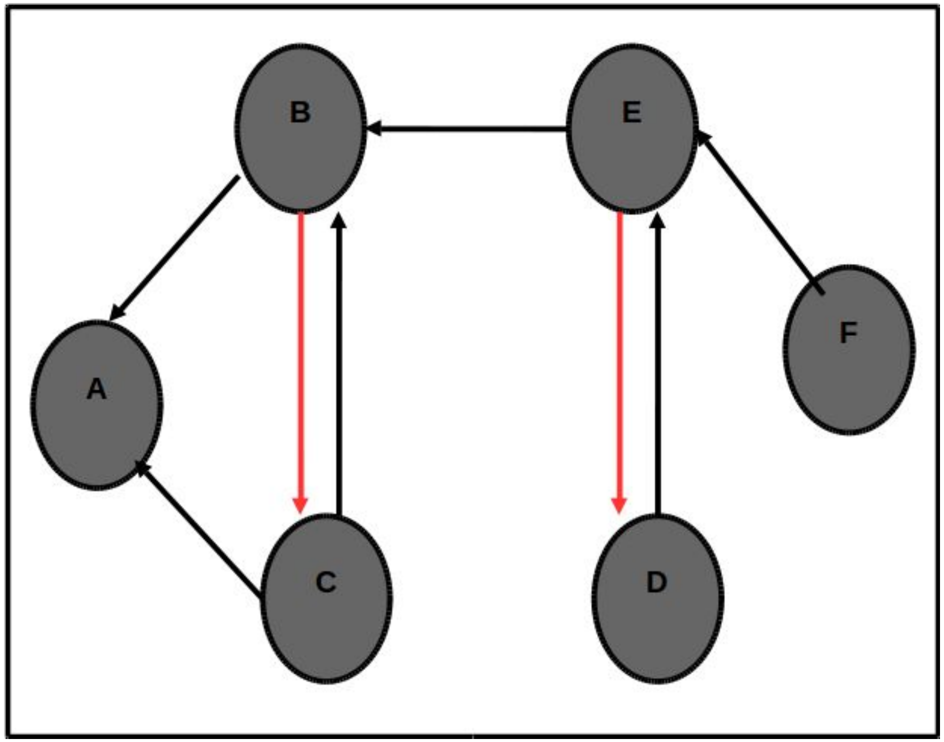


Busca em profundidade no grafo $G^T=(V,A^T)$ seguindo a ordenação topológica da primeira busca (A B C E F D)

Localização de componentes fortemente conectadas: Kosarayu

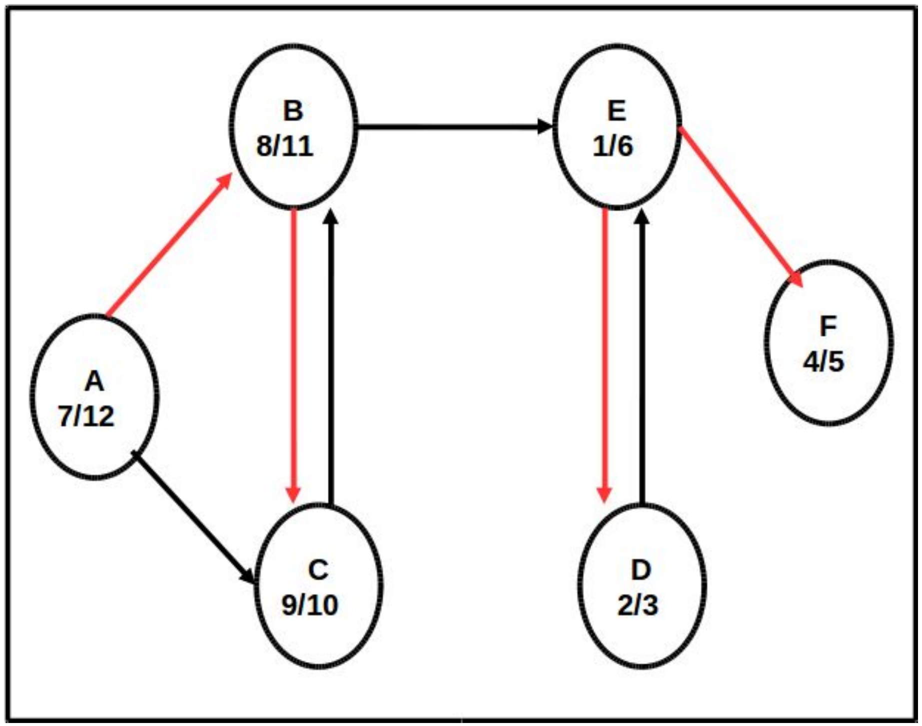


Busca em profundidade no grafo original G(V,A)

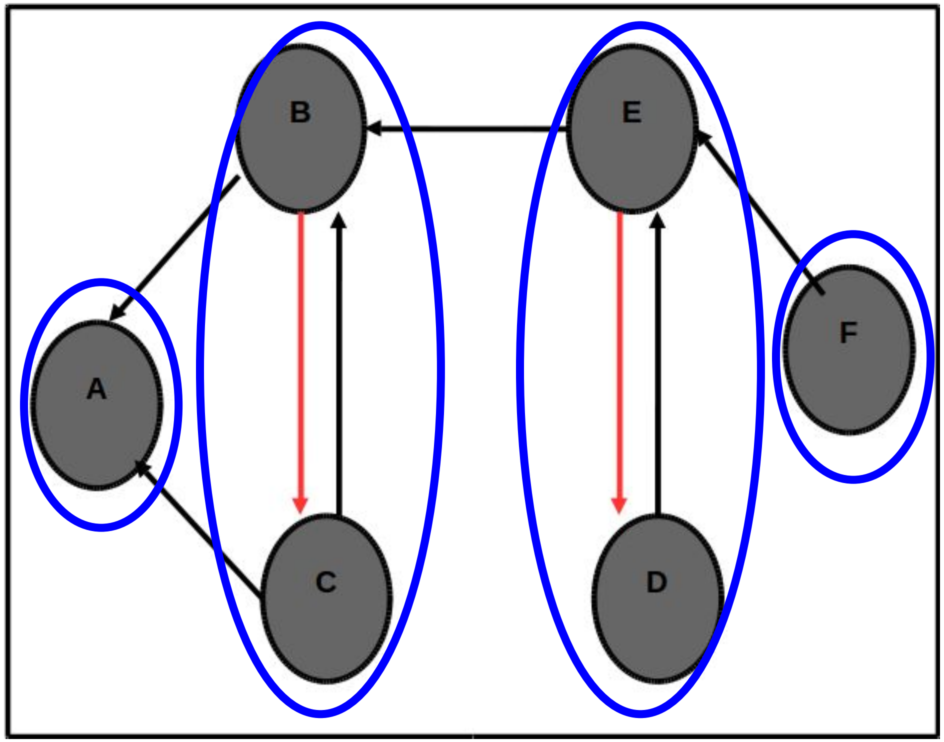


Busca em profundidade no grafo $G^T=(V,A^T)$ seguindo a ordenação topológica da primeira busca (A B C E F D)

Localização de componentes fortemente conectadas: Kosarayu



Busca em profundidade no grafo original G(V,A)



Busca em profundidade no grafo $G^T=(V,A^T)$ seguindo a ordenação topológica da primeira busca (A B C E F D)

Localização de componentes fortemente conectadas: Kosarayu

SCCs (V, A)

1. Chamar DFS (V, A) para calcular f[u]
2. Calcular A^T
3. Chamar DFS (V, A^T) considerando no laço principal os vértices em ordem decrescente de f (calculados na linha 1)
4. Devolva os vértices de cada árvore resultante da linha 3 como uma componente fortemente conectada separada