

Algoritmos em Grafo

Busca em Profundidade

Prof. Dr. Luis Augusto Martins Pereira

- A **Busca em Profundidade** (Depth-First Search - DFS) é um método para percorrer recursivamente um grafo $G(V, E)$.
- **Estratégia do percurso:**
 - Sempre que visitamos um vértice s , visitamos seu primeiro vizinho t .
 - Mas antes de visitarmos outros vizinhos de s , visitamos o vizinho do t e assim por diante.

Exemplo de Busca em Profundidade



WIKIPÉDIA
A enciclopédia livre

Pesquisar na Wikipédia

Conteúdo

Início

[Definição Formal](#)

[Exemplo](#)

[Exemplos de Código](#)

[Aplicações](#)

[Referências](#)

[Ver também](#)

[Ligações externas](#)



WIKI LOVES EARTH 2022 NO BRASIL

Tem fotos de reservas naturais brasileiras?

Compartilhe e concorra a uma participação em uma expedição fotográfica no Brasil.

Busca em profundidade

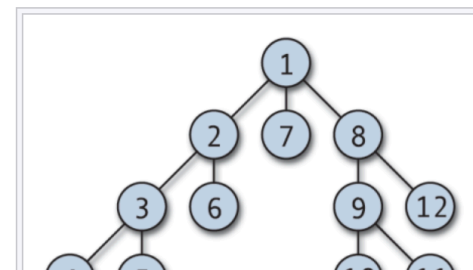
38 línguas [\[ocultar\]](#)

[Artigo](#) [Discussão](#) [Ler](#) [Editar](#) [Ver histórico](#)

Origem: Wikipédia, a enciclopédia livre.

Na teoria dos [grafos](#), **busca em profundidade** (ou busca em profundidade-primeiro, também conhecido em inglês por Depth-First Search - DFS) é um algoritmo usado para realizar uma [busca](#) ou travessia numa árvore, [estrutura de árvore](#) ou [grafo](#). Intuitivamente, o [algoritmo](#) começa num nó raiz (selecionando algum nó como sendo o raiz, no caso de um grafo) e explora tanto quanto possível cada um dos seus ramos, antes de retroceder([backtracking](#)).

Uma versão da busca em profundidade foi investigada no século XIX pelo [matemático francês Charles Pierre Trémaux](#)^[1] como estratégia para solucionar [labirintos](#).^{[2][3]}



Exemplo de Busca em Profundidade



Pesquisar na Wikipédia



WIKI LOVES EARTH 2022 NO BRASIL
Tem fotos de reservas naturais brasileiras?
Compartilhe e concorra a uma participação em uma expedição fotográfica no Brasil.

Teoria dos grafos

70 línguas [ocultar]

Artigo [Discussão](#) [Ler](#) [Editar](#) [Ver histórico](#)

Origem: Wikipédia, a enciclopédia livre.

([Redirecionado de Grafos](#))



Esta página [cita fontes](#), mas que **não cobrem todo o conteúdo**. Ajude a [inserir referências](#). Conteúdo não [verificável](#) pode ser [removido](#). — *Encontre fontes: Google (notícias, livros e acadêmico) (Agosto de 2013)*



Este artigo carece de reciclagem de acordo com o livro de estilo. Sinta-se livre para editá-la para que esta possa atingir um [nível de qualidade superior](#). (Fevereiro de 2013)


A **teoria dos grafos** ou **de grafos** é um ramo da [matemática](#) que estuda as relações entre os objetos de um determinado conjunto. Para tal são utilizadas estruturas chamadas de [grafos](#), $G(V, E)$, onde V é um conjunto



- Conteúdo
- Início
- Histórico
- Definições de grafos e dígrafos
- Representação gráfica
- Armazenamento de grafos em memória
- Definições de teoria dos grafos
 - Relações de incidência e de adjacência
 - Valência (Grau)
 - Passeio
 - Caminho
- Tipos de grafos

Exemplo de Busca em Profundidade

<<



WIKIPÉDIA
A enciclopédia livre

Página principal

Conteúdo destacado

Eventos atuais

Esplanada

Página aleatória

Portais

Informar um erro

Colaboração

Boas-vindas

Ajuda

Página de testes

Portal comunitário

Mudanças recentes

Manutenção

Criar página


Páginas novas

Contato


Donativos

Ferramentas

Páginas afluentes



Pesquisar na Wikipédia



WIKI LOVES EARTH 2022 NO BRASIL

Tem fotos de reservas naturais brasileiras?

Compartilhe e concorra a uma participação em uma expedição fotográfica no Brasil.

Matemática

Artigo

Discussão

Ler

Editar

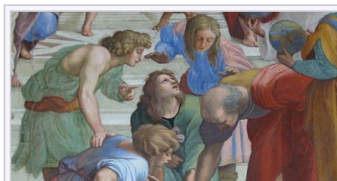
Ver histórico

Origem: Wikipédia, a enciclopédia livre.



Esta página **cita fontes**, mas que **não cobrem todo o conteúdo**. Ajude a **inserir referências**. Conteúdo não **verificável** pode ser **removido**. — *Encontre fontes: Google (notícias, livros e acadêmico) (Fevereiro de 2012)*

A **matemática** (dos termos **gregos** *μάθημα*, transliterado *máthēma*, 'ciência', 'conhecimento' ou 'aprendizagem';^[1] e *μαθηματικός*, transliterado *mathēmatikós*, 'inclinado a aprender') é a **ciência** do **raciocínio lógico** e **abstrato**, que estuda **quantidades** (**teoria dos números**), espaço e medidas (**geometria**), **estruturas**, **variações**^[2] e **estatística**.^{[3][4][5]} Não há, porém, uma definição consensual por parte da comunidade científica.^{[6][7]} O trabalho matemático consiste em procurar e relacionar padrões,^{[8][9]} de modo a formular



Algoritmos e Estrutura de Dados II

5

Controle do processo da busca

- **Pilha** (Last in, First out): guardar os nós visitados
- **Atributos dos nós:**
 - Cor :
 - BRANCO: não visitado;
 - CINZA: visitado pela primeira vez;
 - PRETO: lista de adjacência toda explorada.
 - Tempo de descoberta (d): registra quando um vértice é descoberto pela primeira vez;
 - Tempo de término (f): registra quando a busca termina de examinar a lista de adjacências de um vértice;
 - Predecessor (pred): vértice antecessor a um vértice no caminho explorado.

Pseudocódigo da Busca em Profundidade

```
1. Function DFS(G)
2.   For each u in V[G]
3.     u.cor = BRANCO
4.     u.pred = NIL
5.   Endfor
6.   tempo = 0
7.   For each u in V[G]
8.     if u.cor == BRANCO
9.       DFS-Visit(G,u)
10.    Endif
11.  Endfor
12.Endfunction
```

```
15.Function DFS-Visit(G,u)
16.  tempo = tempo + 1
17.  u.d = tempo
18.  u.cor = CINZA
19.
20.  For each v in G.Adj[u]
21.    if v.cor == BRANCO
22.      v.pred = u
23.      DFS-Visit(G,v)
24.    Endif
25.  Endfor
26.
27.  u.cor = PRETO
28.  tempo = tempo + 1
29.  u.f = tempo
30.Endfunction
```

Pseudocódigo da Busca em Profundidade

```
1. Function DFS(G)
2.   For each u in V[G]
3.     u.cor = BRANCO // Pinte todos os nós de BRANCO
4.     u.pred = NIL
5.   Endfor
6.   tempo = 0
7.   For each u in V[G]
8.     if u.cor == BRANCO
9.       DFS-Visit(G,u)
10.    Endif
11.  Endfor
12.Endfunction
```

```
15.Function DFS-Visit(G,u)
16.  tempo = tempo + 1
17.  u.d = tempo
18.  u.cor = CINZA
19.
20.  For each v in G.Adj[u]
21.    if v.cor == BRANCO
22.      v.pred = u
23.      DFS-Visit(G,v)
24.    Endif
25.  Endfor
26.
27.  u.cor = PRETO
28.  tempo = tempo + 1
29.  u.f = tempo
30.Endfunction
```


Pseudocódigo da Busca em Profundidade

```
1. Function DFS(G)
2.   For each u in V[G]
3.     u.cor = BRANCO // Pinte todos os nós de BRANCO
4.     u.pred = NIL // Predecessor de todos os "u" para NIL
5.   Endfor
6.   tempo = 0
7.   For each u in V[G]
8.     if u.cor == BRANCO
9.       DFS-Visit(G,u)
10.    Endif
11.  Endfor
12.Endfunction
```

```
15.Function DFS-Visit(G,u)
16.  tempo = tempo + 1
17.  u.d = tempo
18.  u.cor = CINZA
19.
20.  For each v in G.Adj[u]
21.    if v.cor == BRANCO
22.      v.pred = u
23.      DFS-Visit(G,v)
24.    Endif
25.  Endfor
26.
27.  u.cor = PRETO
28.  tempo = tempo + 1
29.  u.f = tempo
30.Endfunction
```

Pseudocódigo da Busca em Profundidade

```
1. Function DFS(G)
2.   For each u in V[G]
3.     u.cor = BRANCO // Pinte todos os nós de BRANCO
4.     u.pred = NIL // Predecessor de todos os "u" para NIL
5.   Endfor
6.   tempo = 0 // Inicializando o contador de tempo
7.   For each u in V[G]
8.     if u.cor == BRANCO
9.       DFS-Visit(G,u)
10.    Endif
11.  Endfor
12.Endfunction
```

```
15.Function DFS-Visit(G,u)
16.  tempo = tempo + 1
17.  u.d = tempo
18.  u.cor = CINZA
19.
20.  For each v in G.Adj[u]
21.    if v.cor == BRANCO
22.      v.pred = u
23.      DFS-Visit(G,v)
24.    Endif
25.  Endfor
26.
27.  u.cor = PRETO
28.  tempo = tempo + 1
29.  u.f = tempo
30.Endfunction
```

Pseudocódigo da Busca em Profundidade

```
1. Function DFS(G)
2.   For each u in V[G]
3.     u.cor = BRANCO // Pinte todos os nós de BRANCO
4.     u.pred = NIL // Predecessor de todos os "u" para NIL
5.   Endfor
6.   tempo = 0 // Inicializando o contador de tempo
7.   For each u in V[G]
8.     if u.cor == BRANCO // Verifique se "u" não foi visitado
9.       DFS-Visit(G,u)
10.    Endif
11.  Endfor
12.Endfunction
```

```
15.Function DFS-Visit(G,u)
16.  tempo = tempo + 1
17.  u.d = tempo
18.  u.cor = CINZA
19.
20.  For each v in G.Adj[u]
21.    if v.cor == BRANCO
22.      v.pred = u
23.      DFS-Visit(G,v)
24.    Endif
25.  Endfor
26.
27.  u.cor = PRETO
28.  tempo = tempo + 1
29.  u.f = tempo
30.Endfunction
```

Pseudocódigo da Busca em Profundidade

```
1. Function DFS(G)
2.   For each u in V[G]
3.     u.cor = BRANCO // Pinte todos os nós de BRANCO
4.     u.pred = NIL // Predecessor de todos os "u" para NIL
5.   Endfor
6.   tempo = 0 // Inicializando o contador de tempo
7.   For each u in V[G]
8.     if u.cor == BRANCO // Verifique se "u" não foi visitado
9.       DFS-Visit(G,u) // Chame DFS-Visit em "u"
10.    Endif
11.  Endfor
12.Endfunction
```

```
15.Function DFS-Visit(G,u)
16.  tempo = tempo + 1
17.  u.d = tempo
18.  u.cor = CINZA
19.
20.  For each v in G.Adj[u]
21.    if v.cor == BRANCO
22.      v.pred = u
23.      DFS-Visit(G,v)
24.    Endif
25.  Endfor
26.
27.  u.cor = PRETO
28.  tempo = tempo + 1
29.  u.f = tempo
30.Endfunction
```

Pseudocódigo da Busca em Profundidade

```
1. Function DFS(G)
2.   For each u in V[G]
3.     u.cor = BRANCO // Pinte todos os nós de BRANCO
4.     u.pred = NIL // Predecessor de todos os "u" para NIL
5.   Endfor
6.   tempo = 0 // Inicializando o contador de tempo
7.   For each u in V[G]
8.     if u.cor == BRANCO // Verifique se "u" não foi visitado
9.       DFS-Visit(G,u) // Chame DFS-Visit em "u"
10.    Endif
11.  Endfor
12.Endfunction
```

```
15.Function DFS-Visit(G,u)
16.  tempo = tempo + 1 // incremente a variável de tempo
17.  u.d = tempo
18.  u.cor = CINZA
19.
20.  For each v in G.Adj[u]
21.    if v.cor == BRANCO
22.      v.pred = u
23.      DFS-Visit(G,v)
24.    Endif
25.  Endfor
26.
27.  u.cor = PRETO
28.  tempo = tempo + 1
29.  u.f = tempo
30.Endfunction
```

Pseudocódigo da Busca em Profundidade

```
1. Function DFS(G)
2.   For each u in V[G]
3.     u.cor = BRANCO // Pinte todos os nós de BRANCO
4.     u.pred = NIL // Predecessor de todos os "u" para NIL
5.   Endfor
6.   tempo = 0 // Inicializando o contador de tempo
7.   For each u in V[G]
8.     if u.cor == BRANCO // Verifique se "u" não foi visitado
9.       DFS-Visit(G,u) // Chame DFS-Visit em "u"
10.    Endif
11.  Endfor
12.Endfunction
```

```
15.Function DFS-Visit(G,u)
16.  tempo = tempo + 1 // incremente a variável de tempo
17.  u.d = tempo // Atualize o tempo de descoberta
18.  u.cor = CINZA
19.
20.  For each v in G.Adj[u]
21.    if v.cor == BRANCO
22.      v.pred = u
23.      DFS-Visit(G,v)
24.    Endif
25.  Endfor
26.
27.  u.cor = PRETO
28.  tempo = tempo + 1
29.  u.f = tempo
30.Endfunction
```

Pseudocódigo da Busca em Profundidade

```
1. Function DFS(G)
2.   For each u in V[G]
3.     u.cor = BRANCO // Pinte todos os nós de BRANCO
4.     u.pred = NIL // Predecessor de todos os "u" para NIL
5.   Endfor
6.   tempo = 0 // Inicializando o contador de tempo
7.   For each u in V[G]
8.     if u.cor == BRANCO // Verifique se "u" não foi visitado
9.       DFS-Visit(G,u) // Chame DFS-Visit em "u"
10.    Endif
11.  Endfor
12.Endfunction
```

```
15.Function DFS-Visit(G,u)
16.  tempo = tempo + 1 // incremente a variável de tempo
17.  u.d = tempo // Atualize o tempo de descoberta
18.  u.cor = CINZA // Pinte v de cinza (descoberto)
19.
20.  For each v in G.Adj[u]
21.    if v.cor == BRANCO
22.      v.pred = u
23.      DFS-Visit(G,v)
24.    Endif
25.  Endfor
26.
27.  u.cor = PRETO
28.  tempo = tempo + 1
29.  u.f = tempo
30.Endfunction
```

Pseudocódigo da Busca em Profundidade

```
1. Function DFS(G)
2.   For each u in V[G]
3.     u.cor = BRANCO // Pinte todos os nós de BRANCO
4.     u.pred = NIL // Predecessor de todos os "u" para NIL
5.   Endfor
6.   tempo = 0 // Inicializando o contador de tempo
7.   For each u in V[G]
8.     if u.cor == BRANCO // Verifique se "u" não foi visitado
9.       DFS-Visit(G,u) // Chame DFS-Visit em "u"
10.    Endif
11.  Endfor
12.Endfunction

15.Function DFS-Visit(G,u)
16.  tempo = tempo + 1 // incremente a variável de tempo
17.  u.d = tempo // Atualize o tempo de descoberta
18.  u.cor = CINZA // Pinte v de cinza (descoberto)
19.
20.  For each v in G.Adj[u] // Todos os nós "v" adjacentes ao nó "u"
21.    if v.cor == BRANCO
22.      v.pred = u
23.      DFS-Visit(G,v)
24.    Endif
25.  Endfor
26.
27.  u.cor = PRETO
28.  tempo = tempo + 1
29.  u.f = tempo
30.Endfunction
```


Pseudocódigo da Busca em Profundidade

```
1. Function DFS(G)
2.   For each u in V[G]
3.     u.cor = BRANCO // Pinte todos os nós de BRANCO
4.     u.pred = NIL // Predecessor de todos os "u" para NIL
5.   Endfor
6.   tempo = 0 // Inicializando o contador de tempo
7.   For each u in V[G]
8.     if u.cor == BRANCO // Verifique se "u" não foi visitado
9.       DFS-Visit(G,u) // Chame DFS-Visit em "u"
10.    Endif
11.  Endfor
12.Endfunction

15.Function DFS-Visit(G,u)
16.  tempo = tempo + 1 // incremente a variável de tempo
17.  u.d = tempo // Atualize o tempo de descoberta
18.  u.cor = CINZA // Pinte v de cinza (descoberto)
19.
20.  For each v in G.Adj[u] // Todos os nós "v" adjacentes ao nó "u"
21.    if v.cor == BRANCO // Verifique se "v" não foi visitado
22.      v.pred = u
23.      DFS-Visit(G,v)
24.    Endif
25.  Endfor
26.
27.  u.cor = PRETO
28.  tempo = tempo + 1
29.  u.f = tempo
30.Endfunction
```

Pseudocódigo da Busca em Profundidade

```
1. Function DFS(G)
2.   For each u in V[G]
3.     u.cor = BRANCO // Pinte todos os nós de BRANCO
4.     u.pred = NIL // Predecessor de todos os "u" para NIL
5.   Endfor
6.   tempo = 0 // Inicializando o contador de tempo
7.   For each u in V[G]
8.     if u.cor == BRANCO // Verifique se "u" não foi visitado
9.       DFS-Visit(G,u) // Chame DFS-Visit em "u"
10.    Endif
11.  Endfor
12.Endfunction

15.Function DFS-Visit(G,u)
16.  tempo = tempo + 1 // incremente a variável de tempo
17.  u.d = tempo // Atualize o tempo de descoberta
18.  u.cor = CINZA // Pinte v de cinza (descoberto)
19.
20.  For each v in G.Adj[u] // Todos os nós "v" adjacentes ao nó "u"
21.    if v.cor == BRANCO // Verifique se "v" não foi visitado
22.      v.pred = u // Atualize predecessor
23.      DFS-Visit(G,v)
24.    Endif
25.  Endfor
26.
27.  u.cor = PRETO
28.  tempo = tempo + 1
29.  u.f = tempo
30.Endfunction
```

Pseudocódigo da Busca em Profundidade

```
1. Function DFS(G)
2.   For each u in V[G]
3.     u.cor = BRANCO // Pinte todos os nós de BRANCO
4.     u.pred = NIL // Predecessor de todos os "u" para NIL
5.   Endfor
6.   tempo = 0 // Inicializando o contador de tempo
7.   For each u in V[G]
8.     if u.cor == BRANCO // Verifique se "u" não foi visitado
9.       DFS-Visit(G,u) // Chame DFS-Visit em "u"
10.    Endif
11.  Endfor
12. Endfunction

15. Function DFS-Visit(G,u)
16.   tempo = tempo + 1 // incremente a variável de tempo
17.   u.d = tempo // Atualize o tempo de descoberta
18.   u.cor = CINZA // Pinte v de cinza (descoberto)
19.
20.   For each v in G.Adj[u] // Todos os nós "v" adjacentes ao nó "u"
21.     if v.cor == BRANCO // Verifique se "v" não foi visitado
22.       v.pred = u // Atualize predecessor
23.       DFS-Visit(G,v) // Chame recursivamente DFS-Visit em "v"
24.     Endif
25.   Endfor
26.
27.   u.cor = PRETO
28.   tempo = tempo + 1
29.   u.f = tempo
30. Endfunction
```

Pseudocódigo da Busca em Profundidade

```
1. Function DFS(G)
2.   For each u in V[G]
3.     u.cor = BRANCO // Pinte todos os nós de BRANCO
4.     u.pred = NIL // Predecessor de todos os "u" para NIL
5.   Endfor
6.   tempo = 0 // Inicializando o contador de tempo
7.   For each u in V[G]
8.     if u.cor == BRANCO // Verifique se "u" não foi visitado
9.       DFS-Visit(G,u) // Chame DFS-Visit em "u"
10.    Endif
11.  Endfor
12. Endfunction

15. Function DFS-Visit(G,u)
16.   tempo = tempo + 1 // incremente a variável de tempo
17.   u.d = tempo // Atualize o tempo de descoberta
18.   u.cor = CINZA // Pinte v de cinza (descoberto)
19.
20.   For each v in G.Adj[u] // Todos os nós "v" adjacentes ao nó "u"
21.     if v.cor == BRANCO // Verifique se "v" não foi visitado
22.       v.pred = u // Atualize predecessor
23.       DFS-Visit(G,v) // Chame recursivamente DFS-Visit em "v"
24.     Endif
25.   Endfor
26.
27.   u.cor = PRETO // Pinte "u" de preto (não há mais vizinhos para serem descobertos)
28.   tempo = tempo + 1
29.   u.f = tempo
30. Endfunction
```

Pseudocódigo da Busca em Profundidade

```
1. Function DFS(G)
2.   For each u in V[G]
3.     u.cor = BRANCO // Pinte todos os nós de BRANCO
4.     u.pred = NIL // Predecessor de todos os "u" para NIL
5.   Endfor
6.   tempo = 0 // Inicializando o contador de tempo
7.   For each u in V[G]
8.     if u.cor == BRANCO // Verifique se "u" não foi visitado
9.       DFS-Visit(G,u) // Chame DFS-Visit em "u"
10.    Endif
11.  Endfor
12. Endfunction

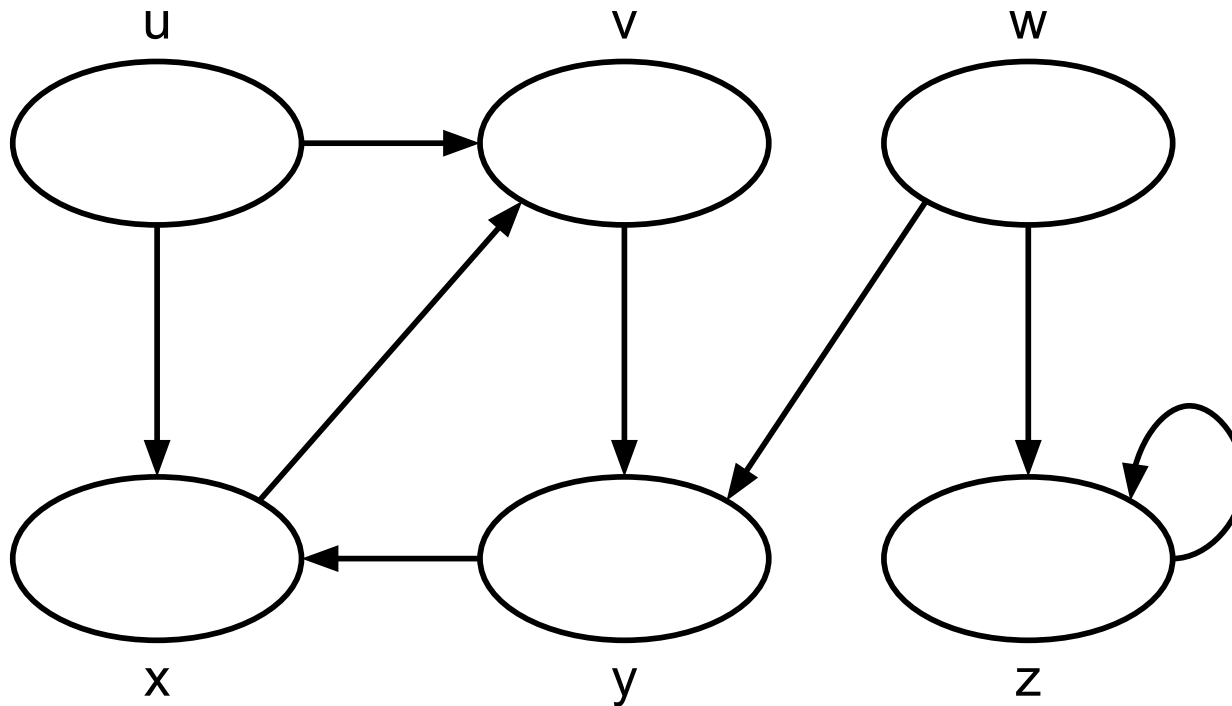
15. Function DFS-Visit(G,u)
16.   tempo = tempo + 1 // incremente a variável de tempo
17.   u.d = tempo // Atualize o tempo de descoberta
18.   u.cor = CINZA // Pinte v de cinza (descoberto)
19.
20.   For each v in G.Adj[u] // Todos os nós "v" adjacentes ao nó "u"
21.     if v.cor == BRANCO // Verifique se "v" não foi visitado
22.       v.pred = u // Atualize predecessor
23.       DFS-Visit(G,v) // Chame recursivamente DFS-Visit em "v"
24.     Endif
25.   Endfor
26.
27.   u.cor = PRETO // Pinte "u" de preto (não há mais vizinhos para serem descobertos)
28.   tempo = tempo + 1 // incremente a variável de tempo
29.   u.f = tempo
30. Endfunction
```

Pseudocódigo da Busca em Profundidade

```
1. Function DFS(G)
2.   For each u in V[G]
3.     u.cor = BRANCO // Pinte todos os nós de BRANCO
4.     u.pred = NIL // Predecessor de todos os "u" para NIL
5.   Endfor
6.   tempo = 0 // Inicializando o contador de tempo
7.   For each u in V[G]
8.     if u.cor == BRANCO // Verifique se "u" não foi visitado
9.       DFS-Visit(G,u) // Chame DFS-Visit em "u"
10.    Endif
11.  Endfor
12. Endfunction

15. Function DFS-Visit(G,u)
16.   tempo = tempo + 1 // incremente a variável de tempo
17.   u.d = tempo // Atualize o tempo de descoberta
18.   u.cor = CINZA // Pinte v de cinza (descoberto)
19.
20.   For each v in G.Adj[u] // Todos os nós "v" adjacentes ao nó "u"
21.     if v.cor == BRANCO // Verifique se "v" não foi visitado
22.       v.pred = u // Atualize predecessor
23.       DFS-Visit(G,v) // Chame recursivamente DFS-Visit em "v"
24.     Endif
25.   Endfor
26.
27.   u.cor = PRETO // Pinte "u" de preto (não há mais vizinhos para serem descobertos)
28.   tempo = tempo + 1 // incremente a variável de tempo
29.   u.f = tempo // Atualize o tempo final
30. Endfunction
```

Simulação da Busca em Profundidade

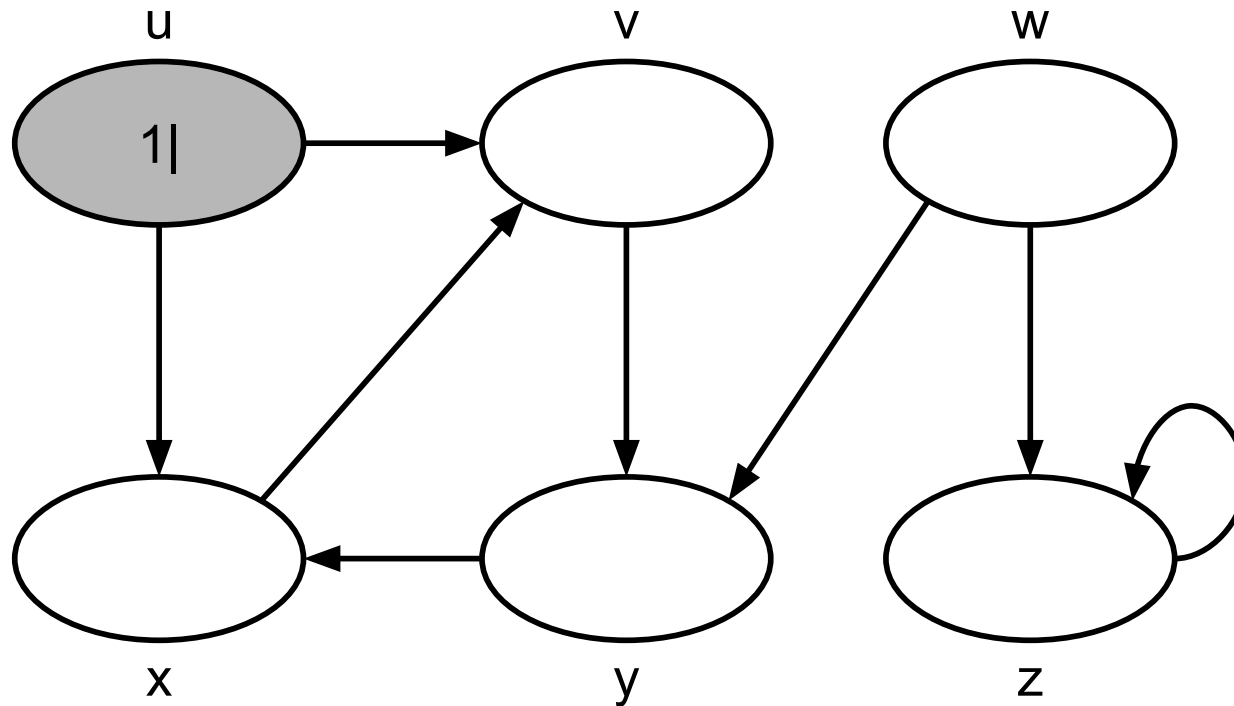


```
1.  Function DFS(G)
2.    For each u in V[G]
3.      u.cor = BRANCO
4.      u.pred = NIL
5.    Endfor
6.    tempo = 0
7.    For each u in V[G]
8.      if u.cor == BRANCO
9.        DFS-Visit(G,u)
10.     Endif
11.  Endfor
12. Endfunction

15. Function DFS-Visit(G,u)
16.   tempo = tempo + 1
17.   u.d = tempo
18.   u.cor = CINZA
19.   For each v in G.Adj[u]
20.     if v.cor == BRANCO
21.       v.pred = u
22.       DFS-Visit(G,v)
23.     Endif
24.   Endfor
25.   u.cor = PRETO
26.   u.f = tempo + 1
27.   u.f = tempo
28. Endfunction

34.   u.f = tempo
35. endfunction
```

Simulação da Busca em Profundidade

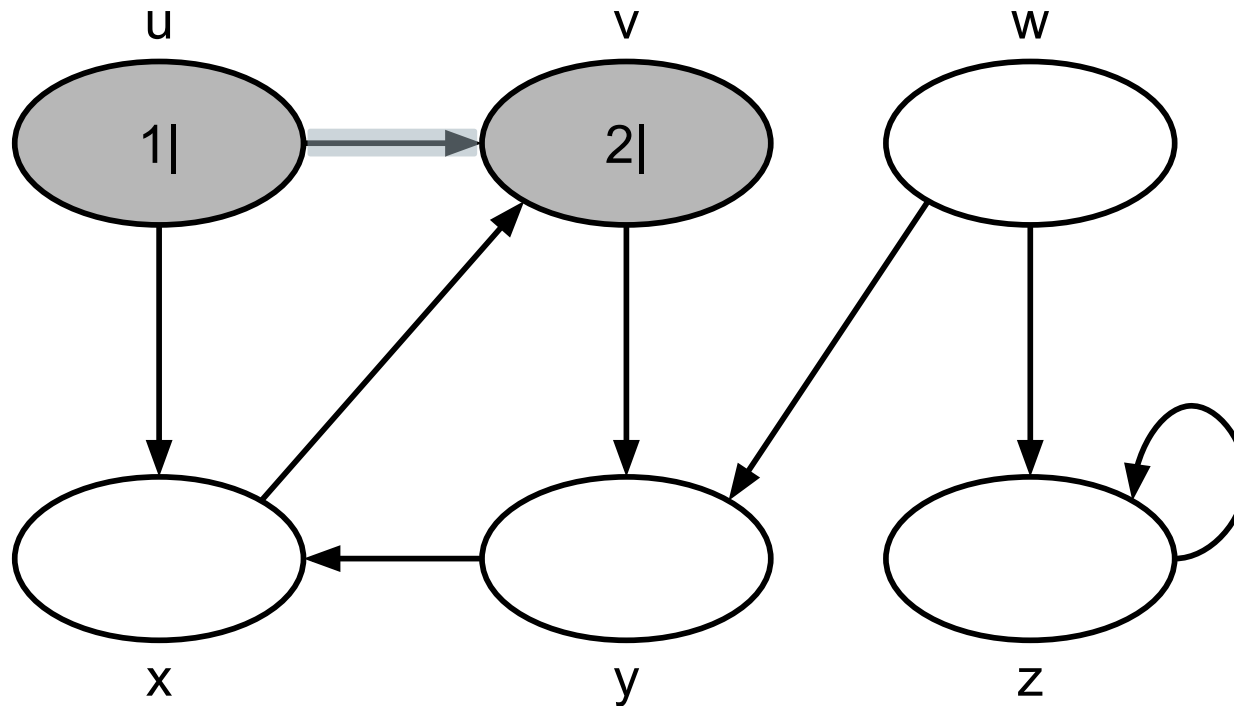


```
1. Function DFS(G)
2.   For each u in V[G]
3.     u.cor = BRANCO
4.     u.pred = NIL
5.   Endfor
6.   tempo = 0
7.   For each u in V[G]
8.     if u.cor == BRANCO
9.       DFS-Visit(G,u)
10.    Endif
11.  Endfor
12. Endfunction

15. Function DFS-Visit(G,u)
16.   tempo = tempo + 1
17.   u.d = tempo
18.   u.cor = CINZA
19.   For each v in G.Adj[u]
20.     if v.cor == BRANCO
21.       v.pred = u
22.       DFS-Visit(G,v)
23.     Endif
24.   Endfor
25.   u.cor = PRETO
26.   tempo = tempo + 1
27.   u.f = tempo
28. Endfunction

34.   u.f = tempo
35. endfunction
```


Simulação da Busca em Profundidade

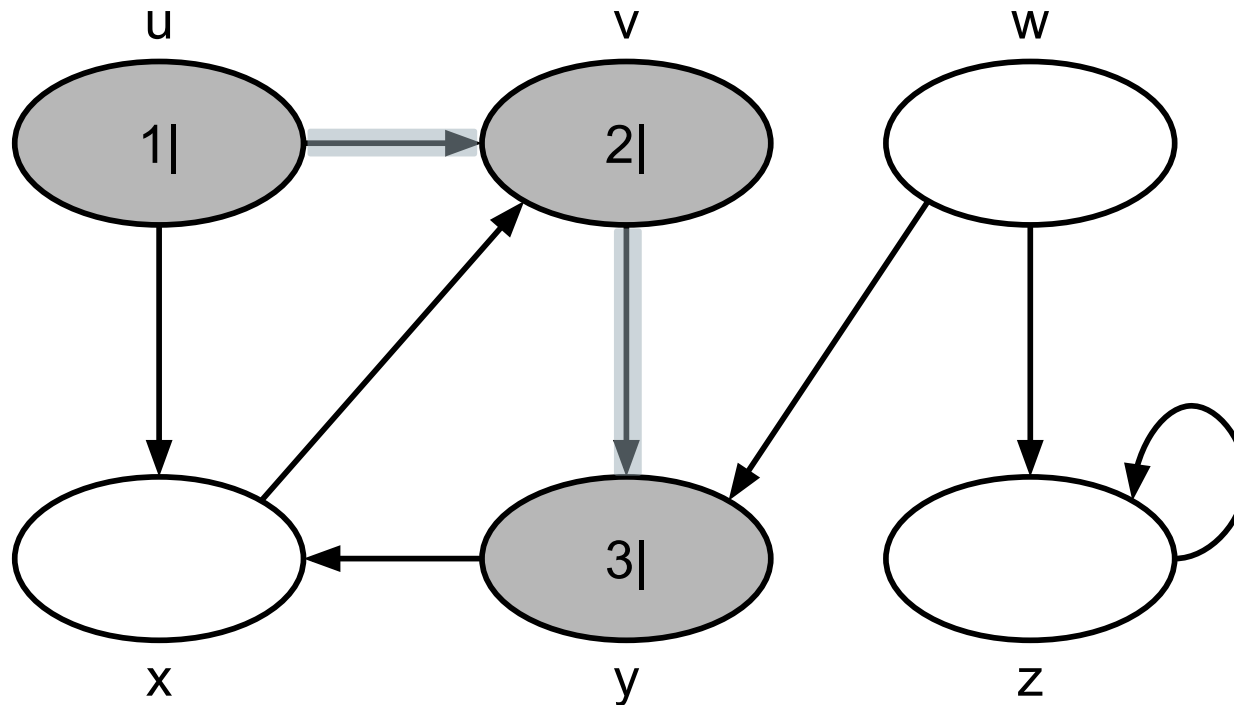


```
1.  Function DFS(G)
2.    For each u in V[G]
3.      u.cor = BRANCO
4.      u.pred = NIL
5.    Endfor
6.    tempo = 0
7.    For each u in V[G]
8.      if u.cor == BRANCO
9.        DFS-Visit(G,u)
10.     Endif
11.  Endfor
12. Endfunction

15. Function DFS-Visit(G,u)
16.   tempo = tempo + 1
17.   u.d = tempo
18.   u.cor = CINZA
19.
20.   For each v in G.Adj[u]
21.     if v.cor == BRANCO
22.       v.pred = u
23.       DFS-Visit(G,v)
24.     Endif
25.   Endfor
26.
27.   u.cor = PRETO
28.   tempo = tempo + 1
29.   u.f = tempo
30. Endfunction

34.   u.f = tempo
35. endfunction
```

Simulação da Busca em Profundidade

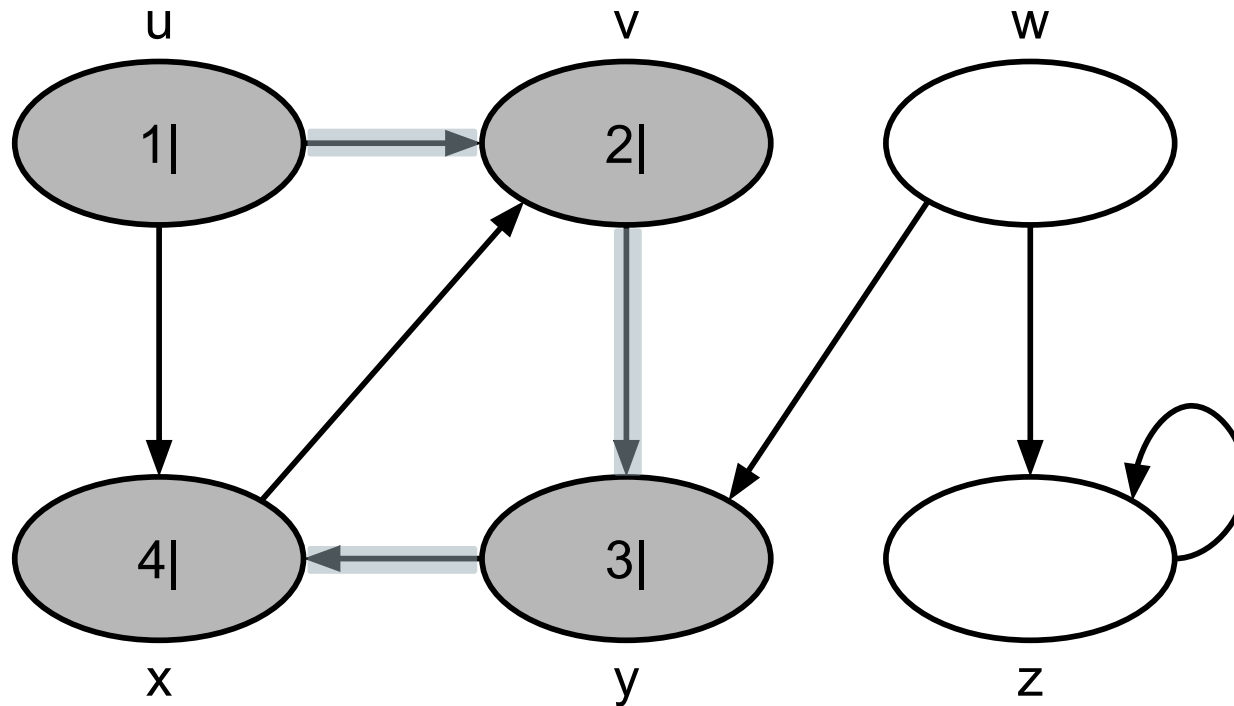


```
1.  Function DFS(G)
2.    For each u in V[G]
3.      u.cor = BRANCO
4.      u.pred = NIL
5.    Endfor
6.    tempo = 0
7.    For each u in V[G]
8.      if u.cor == BRANCO
9.        DFS-Visit(G,u)
10.     Endif
11.   Endfor
12. Endfunction

15. Function DFS-Visit(G,u)
16.   tempo = tempo + 1
17.   u.d = tempo
18.   u.cor = CINZA
19.   For each v in G.Adj[u]
20.     if v.cor == BRANCO
21.       v.pred = u
22.       DFS-Visit(G,v)
23.     Endif
24.   Endfor
25.   u.cor = PRETO
26.   u.f = tempo
27. Endfunction

34.   u.f = tempo
35. endfunction
```

Simulação da Busca em Profundidade

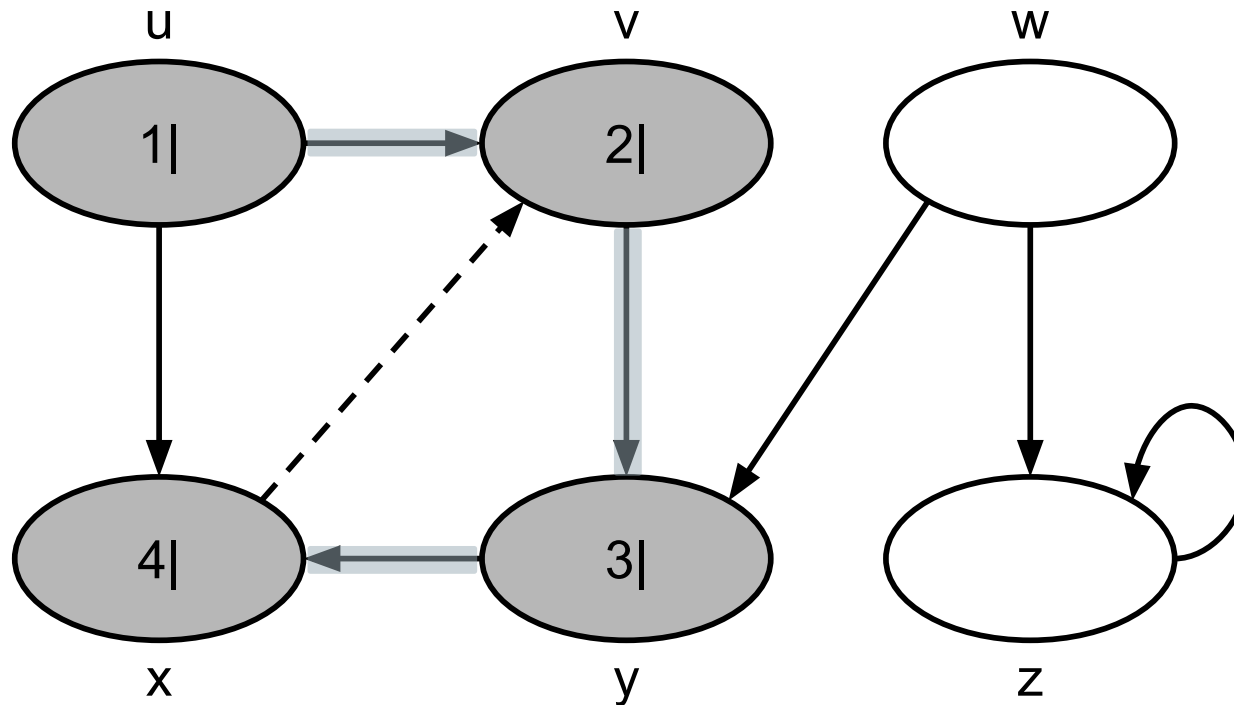


```
1.  Function DFS(G)
2.    For each u in V[G]
3.      u.cor = BRANCO
4.      u.pred = NIL
5.    Endfor
6.    tempo = 0
7.    For each u in V[G]
8.      if u.cor == BRANCO
9.        DFS-Visit(G,u)
10.     Endif
11.  Endfor
12. Endfunction

15. Function DFS-Visit(G,u)
16.   tempo = tempo + 1
17.   u.d = tempo
18.   u.cor = CINZA
19.   For each v in G.Adj[u]
20.     if v.cor == BRANCO
21.       v.pred = u
22.       DFS-Visit(G,v)
23.     Endif
24.   Endfor
25.   u.cor = PRETO
26.   u.f = tempo
27. Endfunction

34.   u.f = tempo
35. endfunction
```

Simulação da Busca em Profundidade

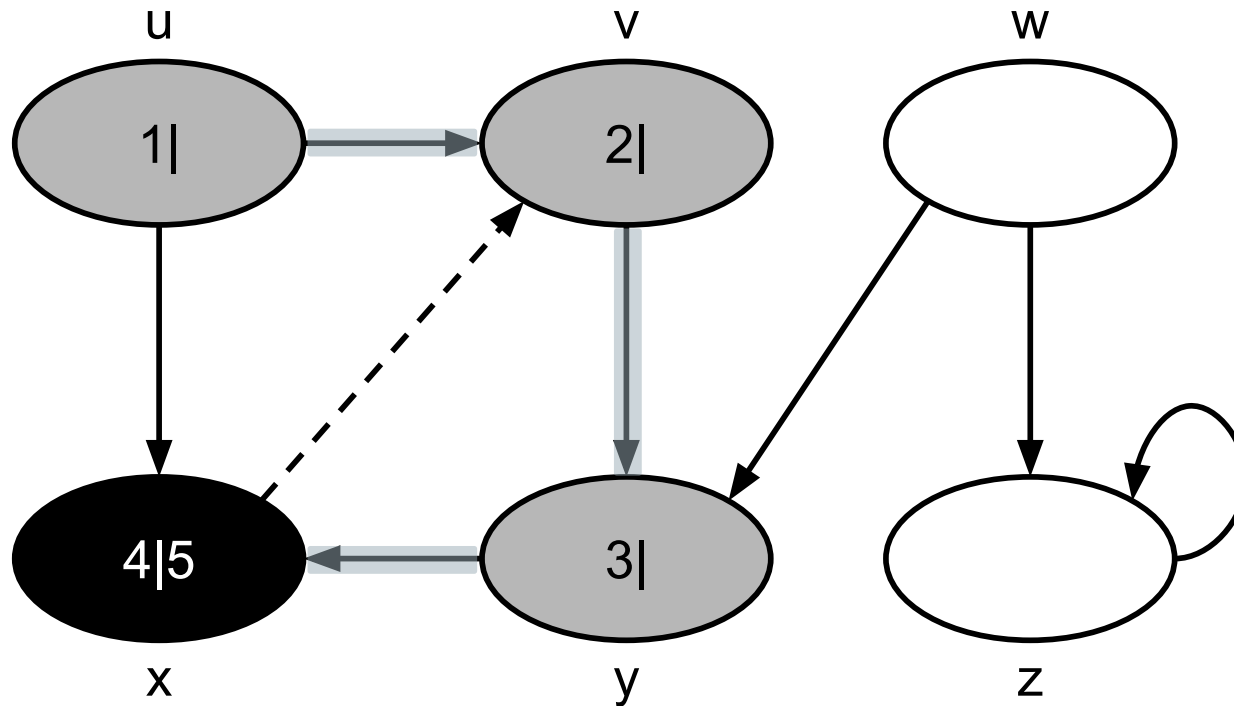


```
1. Function DFS(G)
2.   For each u in V[G]
3.     u.cor = BRANCO
4.     u.pred = NIL
5.   Endfor
6.   tempo = 0
7.   For each u in V[G]
8.     if u.cor == BRANCO
9.       DFS-Visit(G,u)
10.    Endif
11.  Endfor
12. Endfunction

15. Function DFS-Visit(G,u)
16.   tempo = tempo + 1
17.   u.d = tempo
18.   u.cor = CINZA
19.   For each v in G.Adj[u]
20.     if v.cor == BRANCO
21.       v.pred = u
22.       DFS-Visit(G,v)
23.     Endif
24.   Endfor
25.   u.cor = PRETO
26.   tempo = tempo + 1
27.   u.f = tempo
28. Endfunction

34.   u.f = tempo
35. endfunction
```

Simulação da Busca em Profundidade

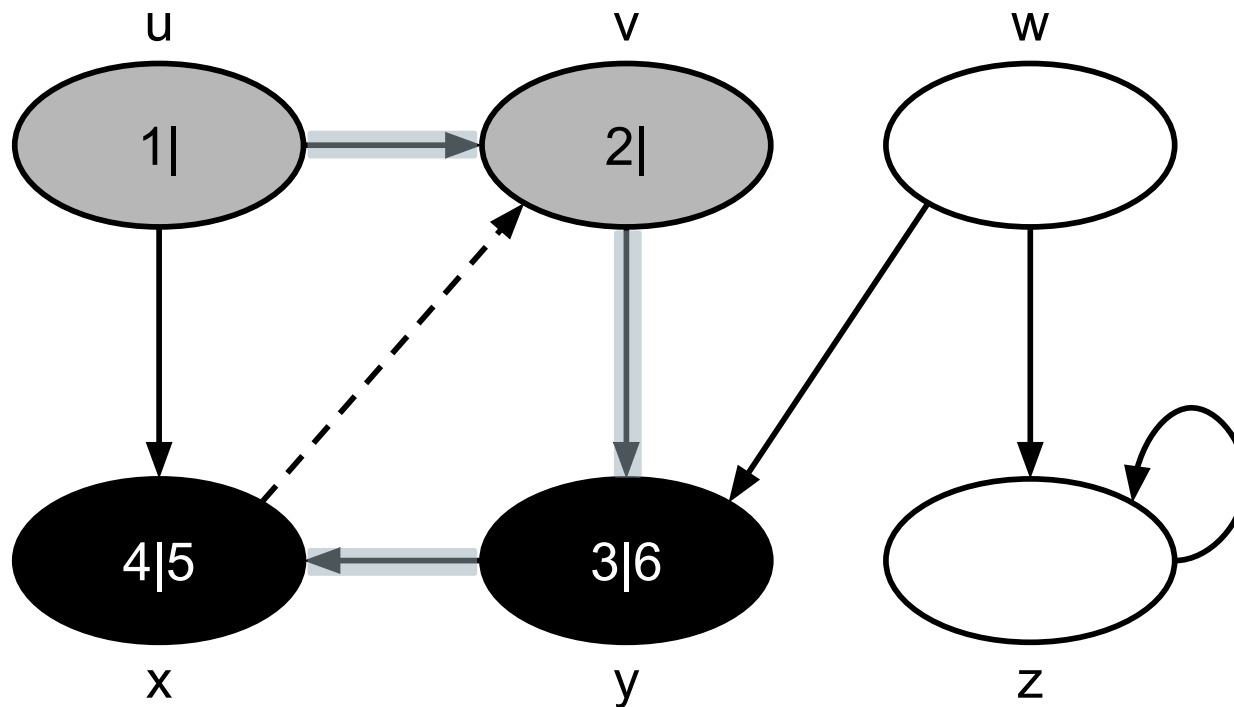


```
1. Function DFS(G)
2.   For each u in V[G]
3.     u.cor = BRANCO
4.     u.pred = NIL
5.   Endfor
6.   tempo = 0
7.   For each u in V[G]
8.     if u.cor == BRANCO
9.       DFS-Visit(G,u)
10.    Endif
11.  Endfor
12. Endfunction

15. Function DFS-Visit(G,u)
16.   tempo = tempo + 1
17.   u.d = tempo
18.   u.cor = CINZA
19.   For each v in G.Adj[u]
20.     if v.cor == BRANCO
21.       v.pred = u
22.       DFS-Visit(G,v)
23.     Endif
24.   Endfor
25.   u.cor = PRETO
26.   u.f = tempo
27. Endfunction

34.   u.f = tempo
35. endfunction
```

Simulação da Busca em Profundidade

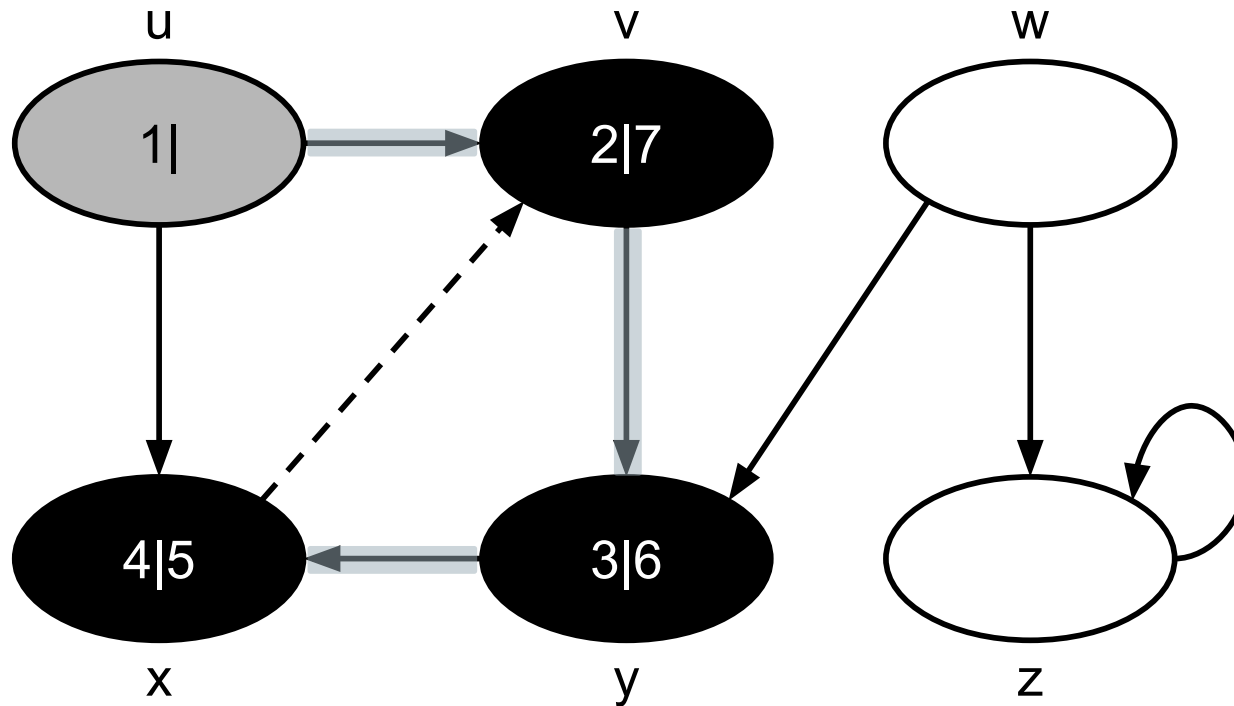


```
1. Function DFS(G)
2.   For each u in V[G]
3.     u.cor = BRANCO
4.     u.pred = NIL
5.   Endfor
6.   tempo = 0
7.   For each u in V[G]
8.     if u.cor == BRANCO
9.       DFS-Visit(G,u)
10.    Endif
11.  Endfor
12. Endfunction

15. Function DFS-Visit(G,u)
16.   tempo = tempo + 1
17.   u.d = tempo
18.   u.cor = CINZA
19.   For each v in G.Adj[u]
20.     if v.cor == BRANCO
21.       v.pred = u
22.       DFS-Visit(G,v)
23.     Endif
24.   Endfor
25.   u.cor = PRETO
26.   tempo = tempo + 1
27.   u.f = tempo
28. Endfunction

34.   u.f = tempo
35. endfunction
```

Simulação da Busca em Profundidade

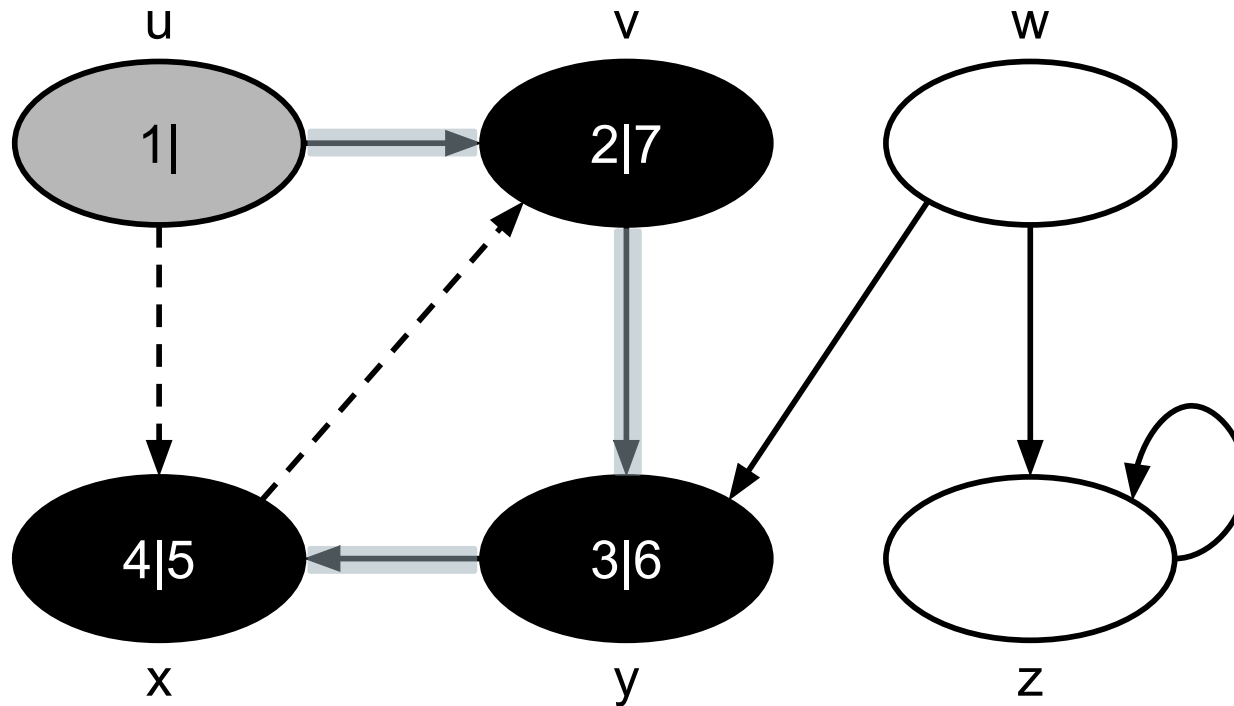


```
1. Function DFS(G)
2.   For each u in V[G]
3.     u.cor = BRANCO
4.     u.pred = NIL
5.   Endfor
6.   tempo = 0
7.   For each u in V[G]
8.     if u.cor == BRANCO
9.       DFS-Visit(G,u)
10.    Endif
11.  Endfor
12. Endfunction

15. Function DFS-Visit(G,u)
16.   tempo = tempo + 1
17.   u.d = tempo
18.   u.cor = CINZA
19.   For each v in G.Adj[u]
20.     if v.cor == BRANCO
21.       v.pred = u
22.       DFS-Visit(G,v)
23.     Endif
24.   Endfor
25.   u.cor = PRETO
26.   tempo = tempo + 1
27.   u.f = tempo
28. Endfunction

34.   u.f = tempo
35. endfunction
```

Simulação da Busca em Profundidade

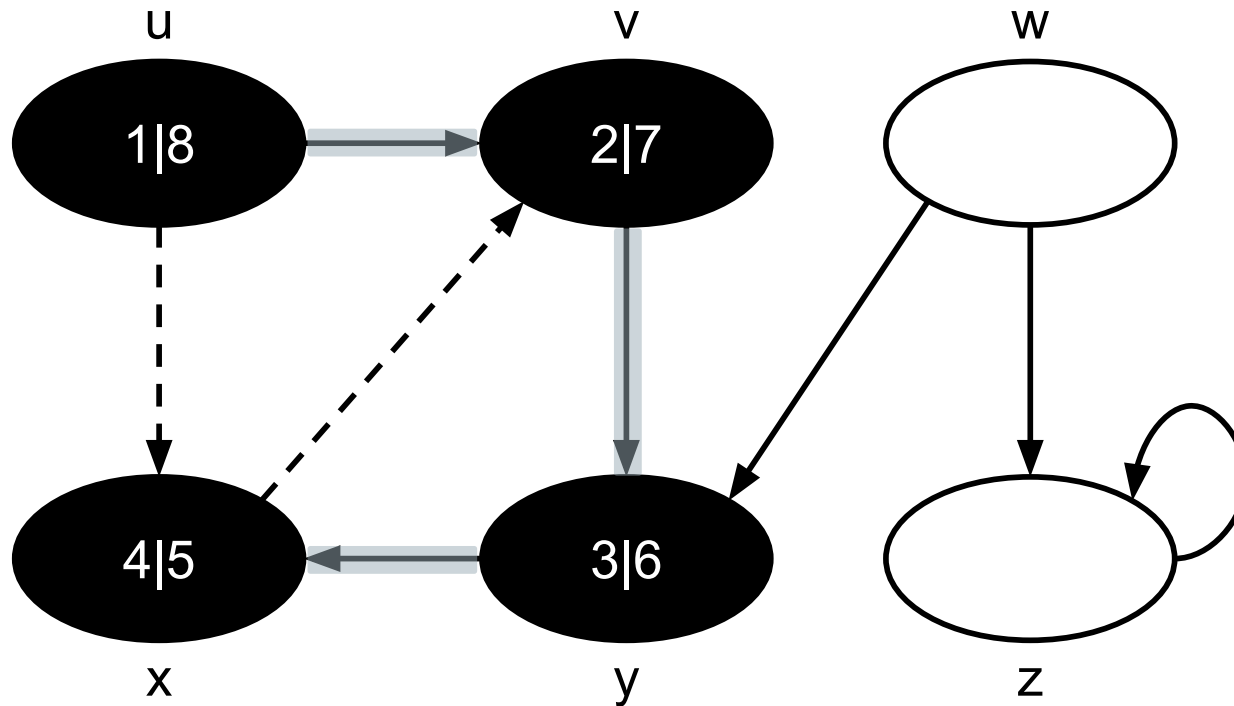


```
1. Function DFS(G)
2.   For each u in V[G]
3.     u.cor = BRANCO
4.     u.pred = NIL
5.   Endfor
6.   tempo = 0
7.   For each u in V[G]
8.     if u.cor == BRANCO
9.       DFS-Visit(G,u)
10.    Endif
11.  Endfor
12. Endfunction

15. Function DFS-Visit(G,u)
16.   tempo = tempo + 1
17.   u.d = tempo
18.   u.cor = CINZA
19.
20.   For each v in G.Adj[u]
21.     if v.cor == BRANCO
22.       v.pred = u
23.       DFS-Visit(G,v)
24.     Endif
25.   Endfor
26.
27.   u.cor = PRETO
28.   tempo = tempo + 1
29.   u.f = tempo
30. Endfunction

34.   u.f = tempo
35. endfunction
```


Simulação da Busca em Profundidade

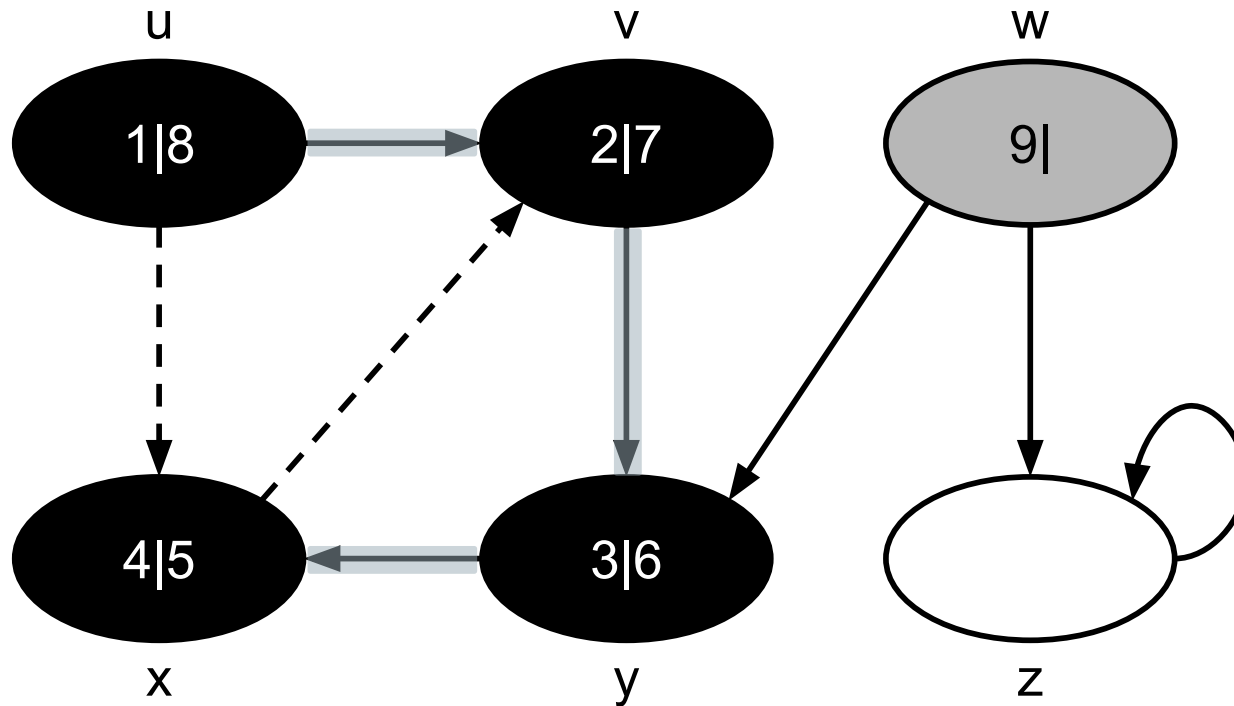


```
1. Function DFS(G)
2.   For each u in V[G]
3.     u.cor = BRANCO
4.     u.pred = NIL
5.   Endfor
6.   tempo = 0
7.   For each u in V[G]
8.     if u.cor == BRANCO
9.       DFS-Visit(G,u)
10.    Endif
11.  Endfor
12. Endfunction

15. Function DFS-Visit(G,u)
16.   tempo = tempo + 1
17.   u.d = tempo
18.   u.cor = CINZA
19.   For each v in G.Adj[u]
20.     if v.cor == BRANCO
21.       v.pred = u
22.       DFS-Visit(G,v)
23.     Endif
24.   Endfor
25.   u.cor = PRETO
26.   tempo = tempo + 1
27.   u.f = tempo
28. Endfunction

34.   u.f = tempo
35. endfunction
```

Simulação da Busca em Profundidade

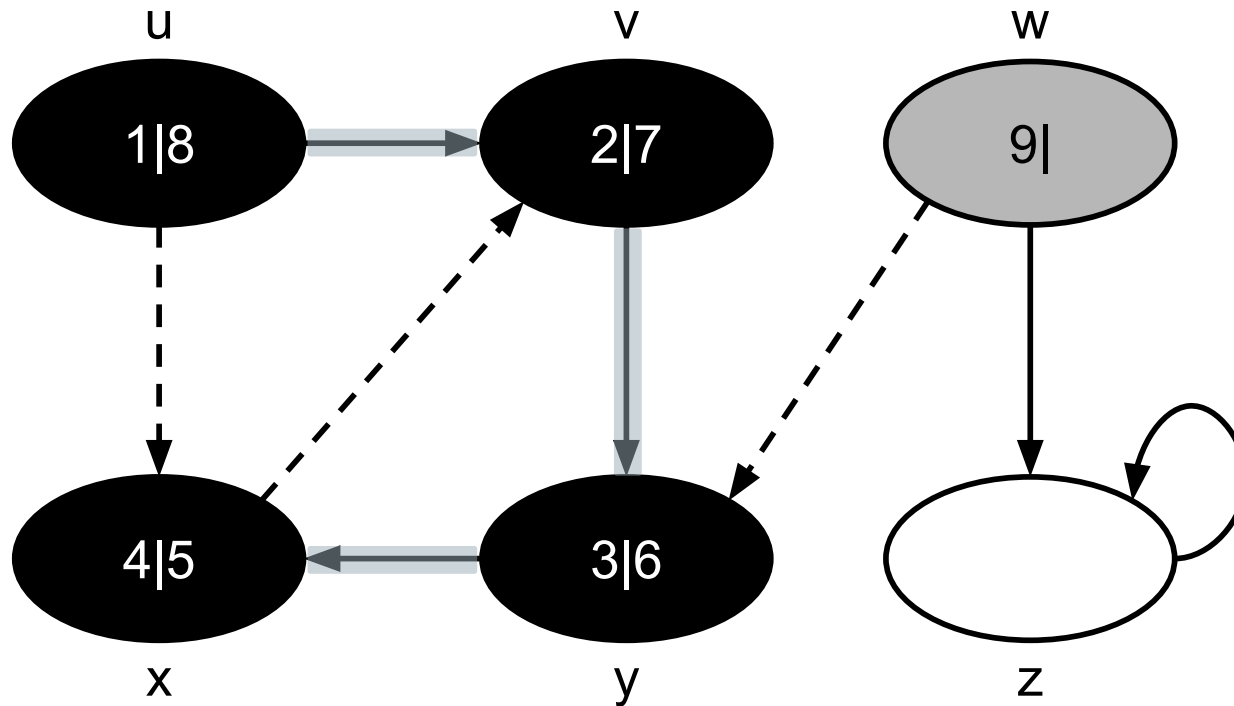


```
1. Function DFS(G)
2.   For each u in V[G]
3.     u.cor = BRANCO
4.     u.pred = NIL
5.   Endfor
6.   tempo = 0
7.   For each u in V[G]
8.     if u.cor == BRANCO
9.       DFS-Visit(G,u)
10.    Endif
11.  Endfor
12. Endfunction

15. Function DFS-Visit(G,u)
16.   tempo = tempo + 1
17.   u.d = tempo
18.   u.cor = CINZA
19.   For each v in G.Adj[u]
20.     if v.cor == BRANCO
21.       v.pred = u
22.       DFS-Visit(G,v)
23.     Endif
24.   Endfor
25.   u.cor = PRETO
26.   tempo = tempo + 1
27.   u.f = tempo
28. Endfunction

34.   u.f = tempo
35. endfunction
```

Simulação da Busca em Profundidade

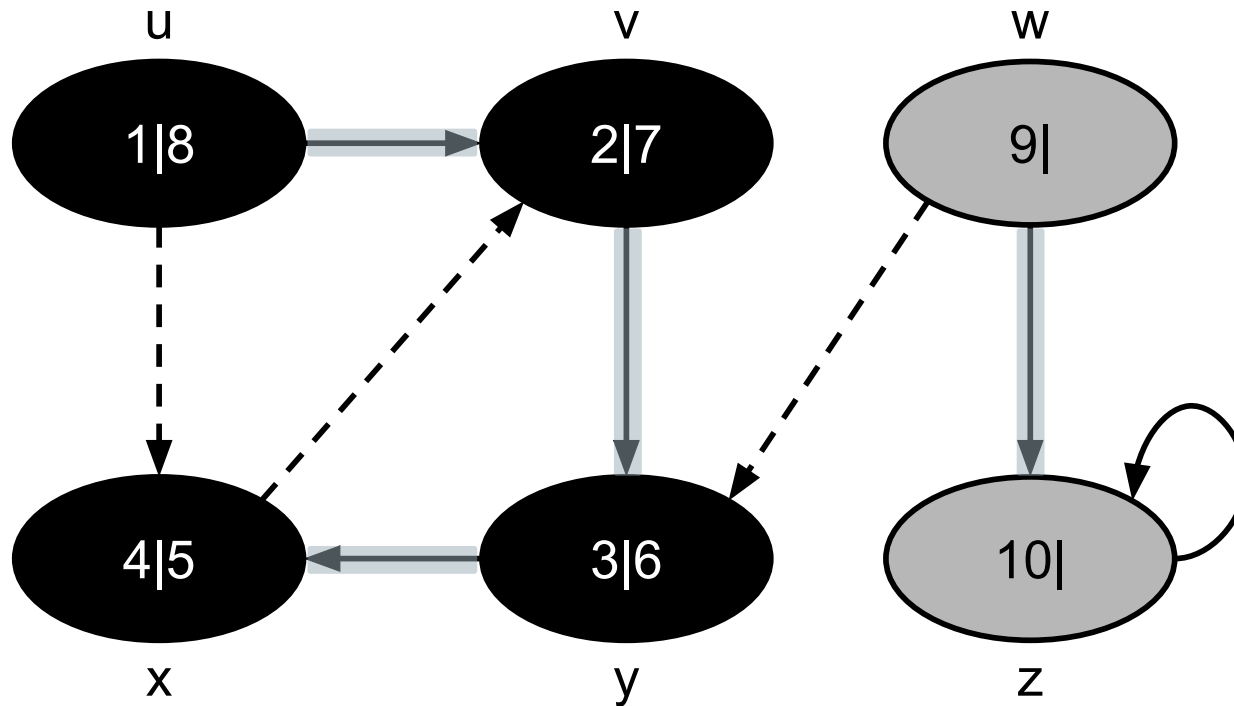


```
1. Function DFS(G)
2.   For each u in V[G]
3.     u.cor = BRANCO
4.     u.pred = NIL
5.   Endfor
6.   tempo = 0
7.   For each u in V[G]
8.     if u.cor == BRANCO
9.       DFS-Visit(G,u)
10.    Endif
11.  Endfor
12. Endfunction

15. Function DFS-Visit(G,u)
16.   tempo = tempo + 1
17.   u.d = tempo
18.   u.cor = CINZA
19.
20.   For each v in G.Adj[u]
21.     if v.cor == BRANCO
22.       v.pred = u
23.       DFS-Visit(G,v)
24.     Endif
25.   Endfor
26.
27.   u.cor = PRETO
28.   tempo = tempo + 1
29.   u.f = tempo
30. Endfunction

34.   u.f = tempo
35. endfunction
```

Simulação da Busca em Profundidade

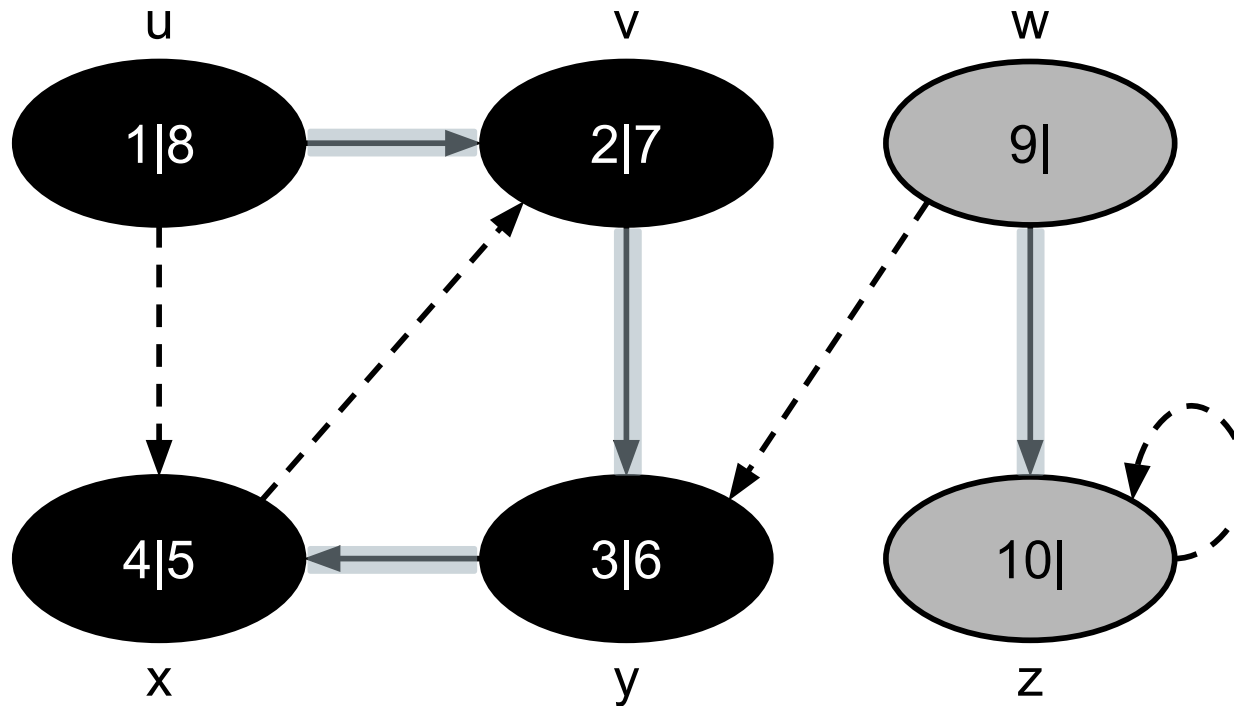


```
1. Function DFS(G)
2.   For each u in V[G]
3.     u.cor = BRANCO
4.     u.pred = NIL
5.   Endfor
6.   tempo = 0
7.   For each u in V[G]
8.     if u.cor == BRANCO
9.       DFS-Visit(G,u)
10.    Endif
11.  Endfor
12. Endfunction

15. Function DFS-Visit(G,u)
16.   tempo = tempo + 1
17.   u.d = tempo
18.   u.cor = CINZA
19.   For each v in G.Adj[u]
20.     if v.cor == BRANCO
21.       v.pred = u
22.       DFS-Visit(G,v)
23.     Endif
24.   Endfor
25.   u.cor = PRETO
26.   tempo = tempo + 1
27.   u.f = tempo
28. Endfunction

34.   u.f = tempo
35. endfunction
```

Simulação da Busca em Profundidade

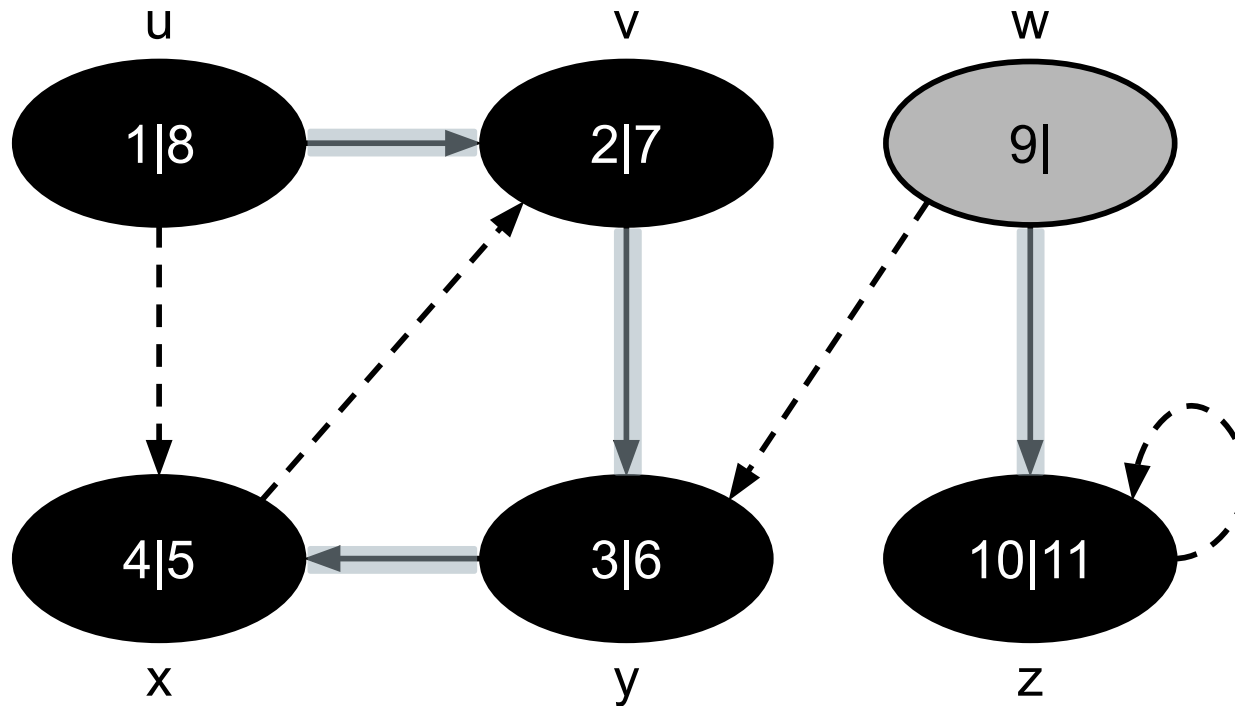


```
1. Function DFS(G)
2.   For each u in V[G]
3.     u.cor = BRANCO
4.     u.pred = NIL
5.   Endfor
6.   tempo = 0
7.   For each u in V[G]
8.     if u.cor == BRANCO
9.       DFS-Visit(G,u)
10.    Endif
11.  Endfor
12. Endfunction

15. Function DFS-Visit(G,u)
16.   tempo = tempo + 1
17.   u.d = tempo
18.   u.cor = CINZA
19.
20.   For each v in G.Adj[u]
21.     if v.cor == BRANCO
22.       v.pred = u
23.       DFS-Visit(G,v)
24.     Endif
25.   Endfor
26.
27.   u.cor = PRETO
28.   tempo = tempo + 1
29.   u.f = tempo
30. Endfunction

34.   u.f = tempo
35. endfunction
```

Simulação da Busca em Profundidade

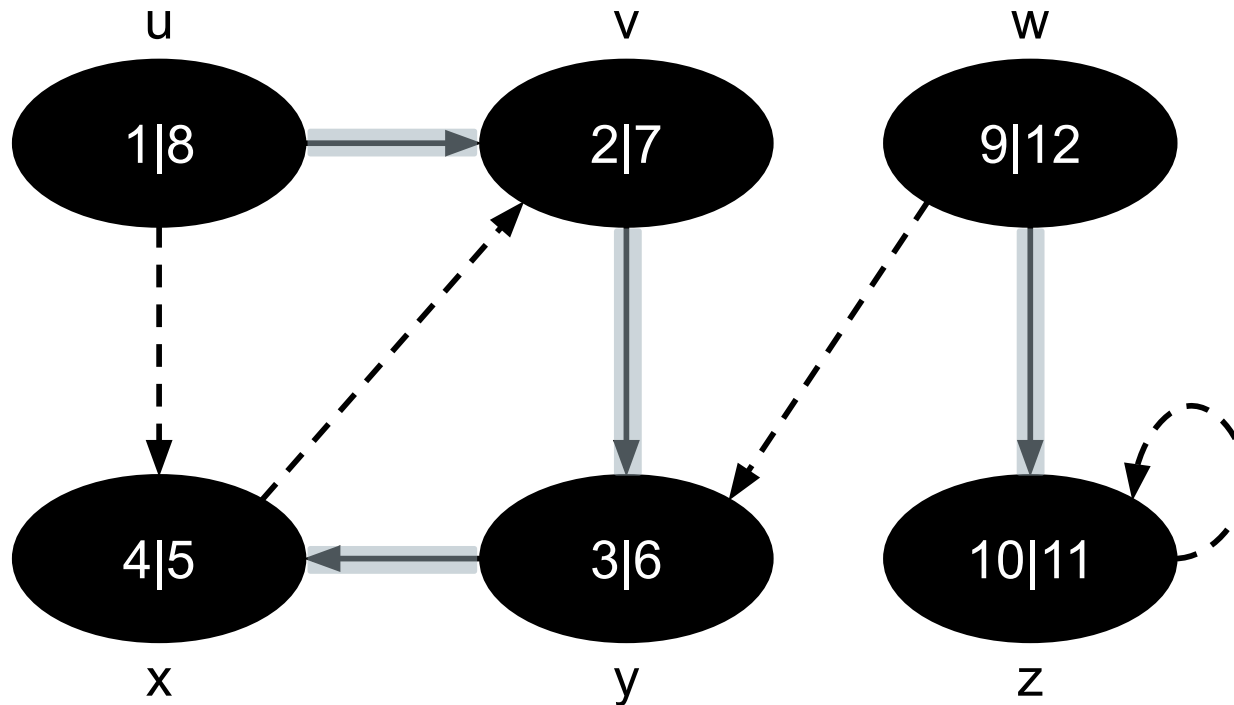


```
1. Function DFS(G)
2.   For each u in V[G]
3.     u.cor = BRANCO
4.     u.pred = NIL
5.   Endfor
6.   tempo = 0
7.   For each u in V[G]
8.     if u.cor == BRANCO
9.       DFS-Visit(G,u)
10.    Endif
11.  Endfor
12. Endfunction

15. Function DFS-Visit(G,u)
16.   tempo = tempo + 1
17.   u.d = tempo
18.   u.cor = CINZA
19.
20.   For each v in G.Adj[u]
21.     if v.cor == BRANCO
22.       v.pred = u
23.       DFS-Visit(G,v)
24.     Endif
25.   Endfor
26.
27.   u.cor = PRETO
28.   tempo = tempo + 1
29.   u.f = tempo
30. Endfunction

34.   u.f = tempo
35. endfunction
```

Simulação da Busca em Profundidade



```
1. Function DFS(G)
2.   For each u in V[G]
3.     u.cor = BRANCO
4.     u.pred = NIL
5.   Endfor
6.   tempo = 0
7.   For each u in V[G]
8.     if u.cor == BRANCO
9.       DFS-Visit(G,u)
10.    Endif
11.  Endfor
12. Endfunction

15. Function DFS-Visit(G,u)
16.   tempo = tempo + 1
17.   u.d = tempo
18.   u.cor = CINZA
19.   For each v in G.Adj[u]
20.     if v.cor == BRANCO
21.       v.pred = u
22.       DFS-Visit(G,v)
23.     Endif
24.   Endfor
25.   u.cor = PRETO
26.   tempo = tempo + 1
27.   u.f = tempo
30. Endfunction

34.   u.f = tempo
35. endfunction
```

Tempo gasto pela Busca em Profundidade

```
1. Function DFS(G)
2.   For each u in V[G]
3.     u.cor = BRANCO
4.     u.pred = NIL
5.   Endfor
6.   tempo = 0
7.   For each u in V[G]
8.     if u.cor == BRANCO
9.       DFS-Visit(G,u)
10.    Endif
11.  Endfor
12.Endfunction
```

```
15.Function DFS-Visit(G,u)
16.  tempo = tempo + 1
17.  u.d = tempo
18.  u.cor = CINZA
19.
20.  For each v in G.Adj[u]
21.    if v.cor == BRANCO
22.      v.pred = u
23.      DFS-Visit(G,v)
24.    Endif
25.  Endfor
26.
27.  u.cor = PRETO
28.  tempo = tempo + 1
29.  u.f = tempo
30.Endfunction
```


Tempo gasto pela Busca em Profundidade

```
1. Function DFS(G)
2.   For each u in V[G]
3.     u.cor = BRANCO
4.     u.pred = NIL
5.   Endfor
6.   tempo = 0
7.   For each u in V[G]
8.     if u.cor == BRANCO
9.       DFS-Visit(G,u)
10.    Endif
11.  Endfor
12 Endfunction
```

$O(|V|)$

```
15 Function DFS-Visit(G,u)
16   tempo = tempo + 1
17   u.d = tempo
18   u.cor = CINZA
19
20   For each v in G.Adj[u]
21     if v.cor == BRANCO
22       v.pred = u
23       DFS-Visit(G,v)
24     Endif
25   Endfor
26
27   u.cor = PRETO
28   tempo = tempo + 1
29   u.f = tempo
30 Endfunction
```

$O(|E|)$

Tempo gasto pela Busca em Profundidade

```
1. Function DFS(G)
2.   For each u in V[G]
3.     u.cor = BRANCO
4.     u.pred = NIL
5.   Endfor
6.   tempo = 0
7.   For each u in V[G]
8.     if u.cor == BRANCO
9.       DFS-Visit(G,u)
10.    Endif
11.  Endfor
12 Endfunction

15 Function DFS-Visit(G,u)
16   tempo = tempo + 1
17   u.d = tempo
18   u.cor = CINZA
19
20   For each v in G.Adj[u]
21     if v.cor == BRANCO
22       v.pred = u
23       DFS-Visit(G,v)
24     Endif
25   Endfor
26
27   u.cor = PRETO
28   tempo = tempo + 1
29   u.f = tempo
30 Endfunction
```

$O(|V| + |E|)$

- Simulação da Busca em Profundidade

Aplicações práticas da Busca em Largura

- [GeeksfoGeeks](#)

- CORMEN, Thomas H et al. Algoritmos: teoria e prática. Rio de Janeiro: Campus, 2002. ISBN 978-85-352-0926-6.
tradução de ""Introduction to algorithms"" 2.ed.
- ZIVIANI, N. Projeto de algoritmos: com implementações em PASCAL e C. 2 ed. rev. e ampl. São Paulo: Thomson, 2004. ISBN 978-85-221-0390-4. (Também disponível em e-book)