

DISCIPLINA: ESTRUTURA DE DADOS II  
2018



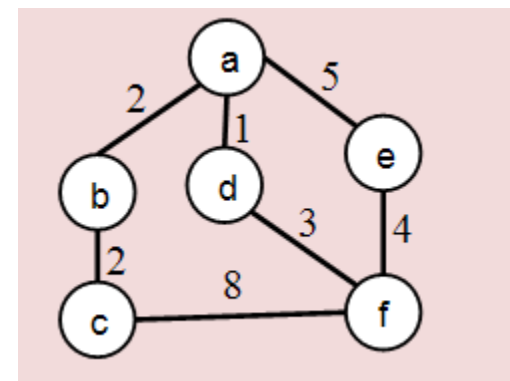
# Busca em Grafos

Prof. Luis Cuevas Rodríguez, PhD



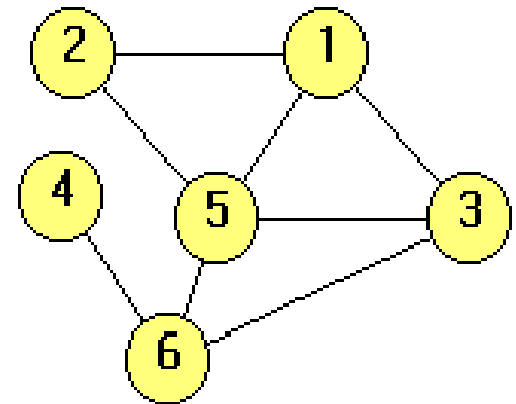
# Busca em Grafos

- Busca em grafos: seguir suas arestas sistematicamente, de modo a visitar seus vértices.
  - a busca vai parar quando encontramos o que queremos ou visitamos todos os vértices
- Algoritmos de busca:
  - Busca em profundidade.
  - Busca em largura.



# Busca em profundidade

- Busca em profundidade (Depth-first Search ou DFS): **buscar mais fundo no grafo sempre que é possível**
  1. As arestas são exploradas a partir do vértice  **$v$  mais recentemente descoberto** que ainda tem a aresta não explorada saindo dele.
    - Quando todas as arestas de  $v$  são exploradas, a busca volta ao vértice anterior a  $v$  (backtracking), para seguir arestas ainda não exploradas



| Nó | Conectado a    |
|----|----------------|
| v1 | v2, v5, v3     |
| v2 | v1, v5         |
| v3 | v1, v5, v6     |
| v4 | v6             |
| v5 | v1, v2, v3, v6 |
| v6 | v3, v4, v5     |

# Busca em profundidade

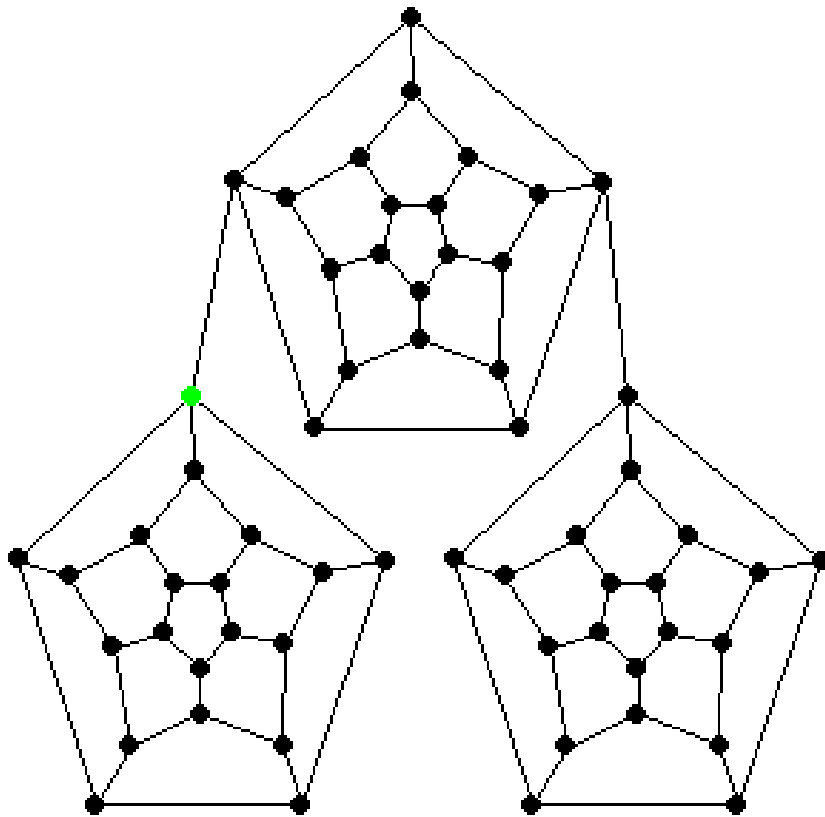
3. O processo continua até ter descoberto todos os vértices que são atingíveis a partir do vértice inicial.
  - Atingíveis: o vértice  $u$  é atingíveis desde  $v$  se houver um caminho de  $v$  até  $u$ 
    - *Grafos não conexos.*
    - *Grafos dirigidos.*
4. Se restarem ainda vértices não visitados, um é selecionado e reiniciamos a busca a partir dele

# Busca em profundidade

1. Definia um vértice inicial
2. Escolha um de seus adjacentes ainda não visitado. Visite-lho
3. Repetir o processo até atingir objetivo ou um vértice cuja adjacência já tenha sido visitada ou não tenha adjacência
4. Se atingir um nó final que não seja objetivos:
  1. Continue de um vértice irmão ainda não visitado
  2. Voltar ao pai deste
  3. Escolher outro nó inicial

# Busca em profundidade

Depth-First Search

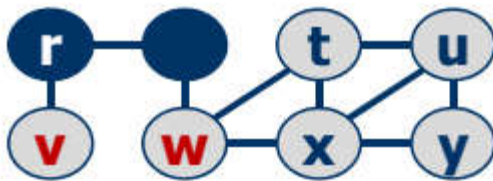
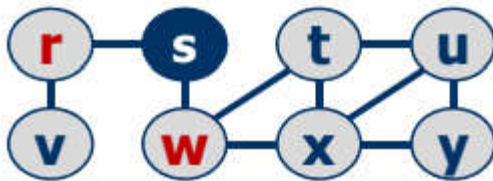
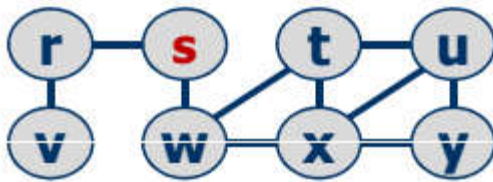


[www.combinatorica.com](http://www.combinatorica.com)

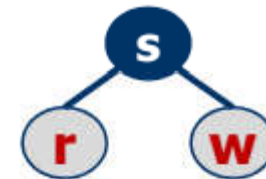
TDA necessários

- O grafo: lista de adjacência ou matriz de adjacência.
- Uma PILHA: armazenar o caminho atual.
- Um vetor: armazenar os no visitados.

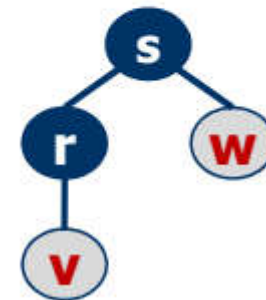
# Busca em profundidade



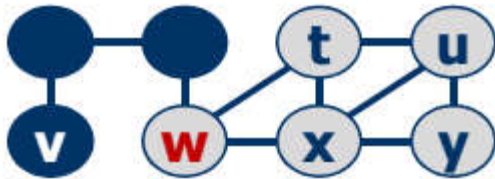
$\text{pilha} = \{s\}$   
 $\text{adj} = \{r, w\}$   
 $\text{vist} = \{s\}$



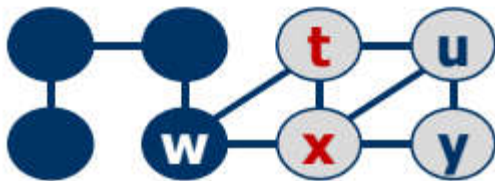
$\text{pilha} = \{r, s\}$   
 $\text{adj} = \{v\}$   
 $\text{vist} = \{r, s\}$



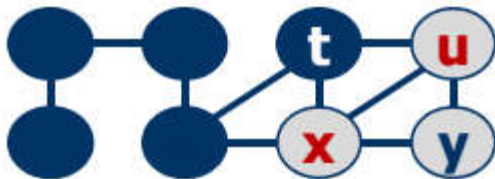
# Busca em profundidade



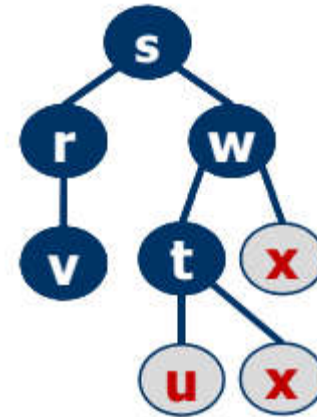
$\text{pilha} = \{v, r, s\}$   
 $\text{adj} = \{\}$   
 $\text{vist} = \{v, r, s\}$



$\text{pilha} = \{w, s\}$   
 $\text{adj} = \{t, x\}$   
 $\text{vist} = \{v, r, w, s\}$



$\text{pilha} = \{t, w, s\}$   
 $\text{adj} = \{x, u, w\}$   
 $\text{vist} = \{t, v, r, w, s\}$

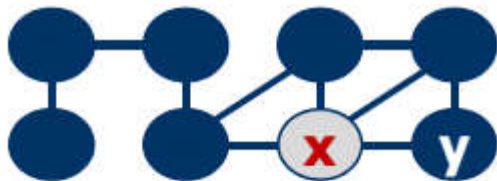




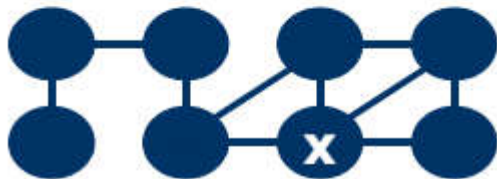
# Busca em profundidade



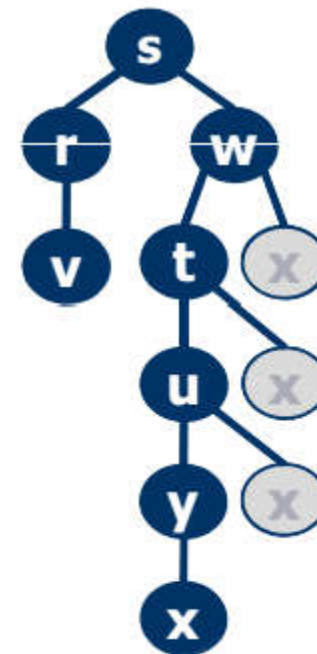
$\text{pilha} = \{u, t, w, s\}$   
 $\text{adj} = \{x, y, t\}$   
 $\text{vist} = \{u, t, v, r, w, s\}$



$\text{pilha} = \{y, u, t, w, s\}$   
 $\text{adj} = \{x, u\}$   
 $\text{vist} = \{y, u, t, v, r, w, s\}$

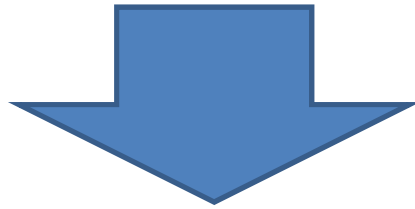


$\text{pilha} = \{x, u, y, u, t, w, s\}$   
 $\text{adj} = \{\}$   
 $\text{vist} = \{x, y, u, t, v, r, w, s\}$



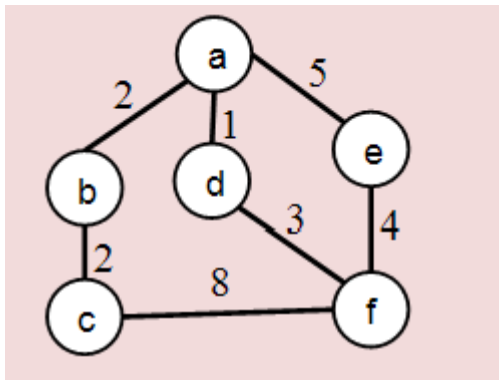
# Busca em profundidade

- A estrutura resultante é um arvore o uma floresta (vários árvores)



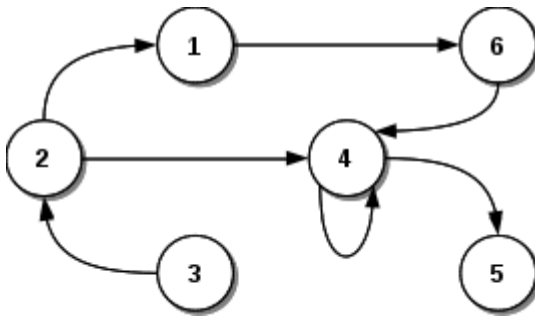
Árvores de busca em profundidade

# Busca em profundidade - Exercício



|      |                    |
|------|--------------------|
| va → | vb,2 → vd,1 → ve,5 |
| vb → | va,2 → vc,2        |
| vc → | vb,2 → vf,8        |
| vd → | va,1 → vf,3        |
| ve → | va,5 → vf,4        |
| vf → | ve,4 → vd,3 → vc,8 |

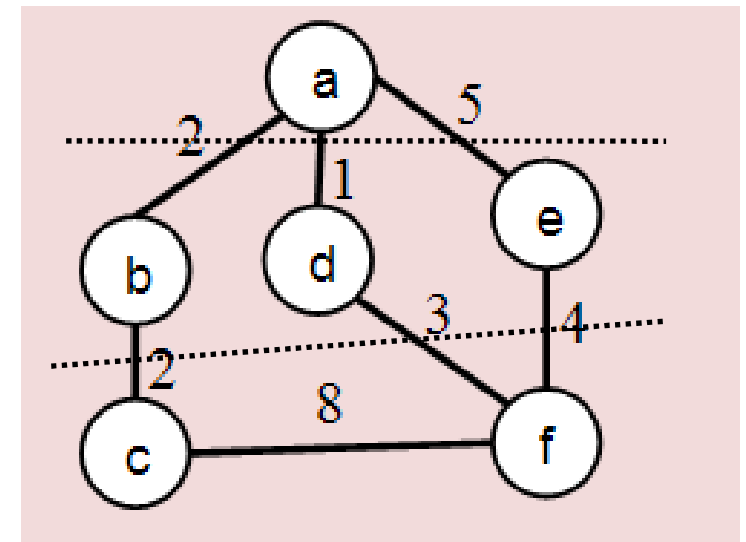
# Busca em profundidade - Exercício



|      |         |
|------|---------|
| v1 → | V6      |
| V2 → | v1, v4  |
| V3 → | v2      |
| v4 → | v4, v5, |
| v5 → |         |
| V6 → | v4      |

# Busca em largura

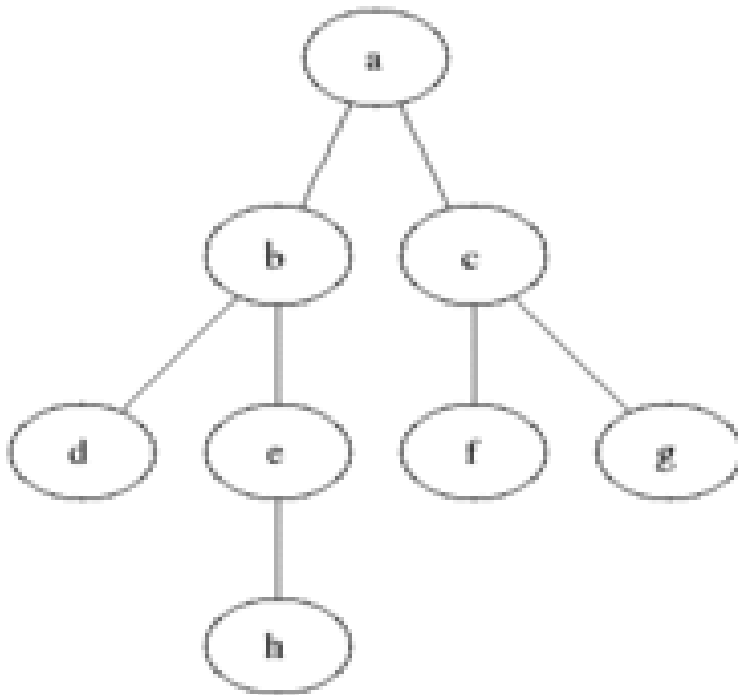
- Busca em largura (breadth-first search ou BFS): primeiro visita todos os adjacentes de um vértice para depois visitar os adjacentes dos adjacentes.
- Busca em forma de camada



# Busca em largura

- Os vértices a serem visitados são colocados em uma FILA.
- Ao visitar um vértice colocamos seus adjacentes na FILA, somente se não estiverem.
- Continuar até achar o vértices objetivo ou a FILA ficar vazia.
- Se a FILA esta vazia e ainda restarem nós não visitados, reiniciar o procedimento a partir de um destes.

# Busca em largura

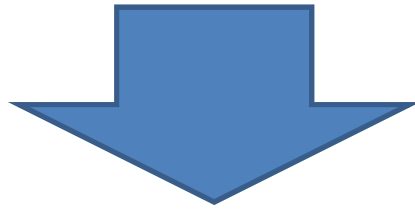


## TDA necessários

- O grafo: lista de adjacência ou matriz de adjacência.
- Uma FILA: armazenar no descobertos.
- Um vetor: armazenar os no visitados.

# Busca em largura

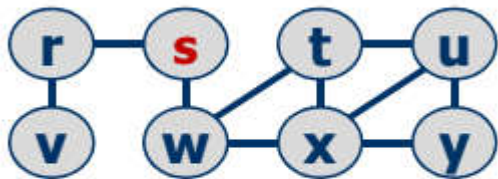
- A estrutura resultante é um arvore o uma floresta (vários árvores)



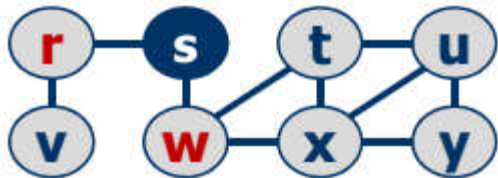
Árvores de busca em largura



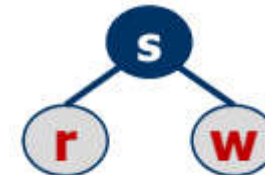
# Busca em largura



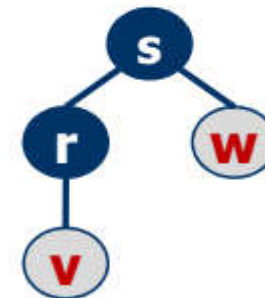
fila={s}  
adj={r,w}  
vist={}



fila={r,w}  
adj={v}  
vist={s}



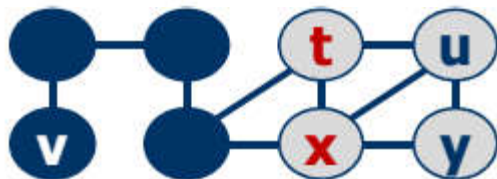
fila={w,v}  
adj={t,x}  
vist={s,r}



# Busca em largura



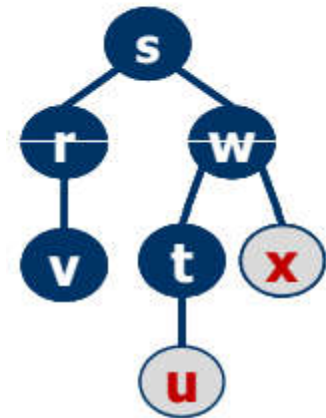
fila={v,t, x}  
adj={}  
vist={s,r,w}



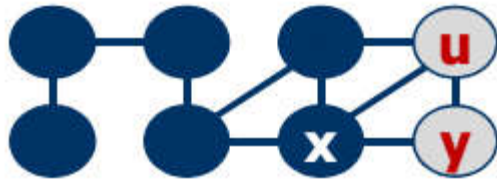
fila={t,x}  
adj={u,x}  
vist={s,r,w, v}



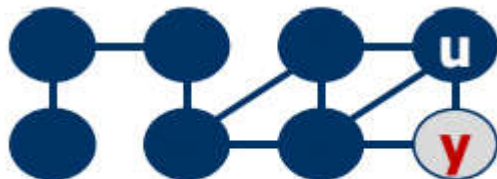
fila={x,u}  
adj={y,u}  
vist={s,r,w, v,t}



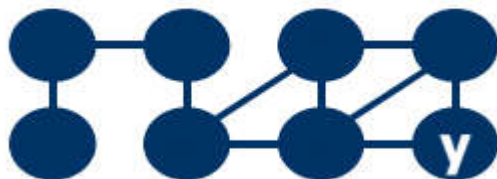
# Busca em largura



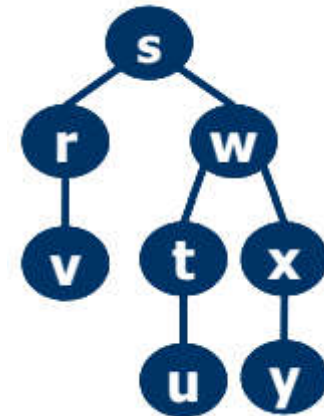
$\text{fila} = \{u, y\}$   
 $\text{adj} = \{\}$   
 $\text{vist} = \{s, r, w, t, x\}$



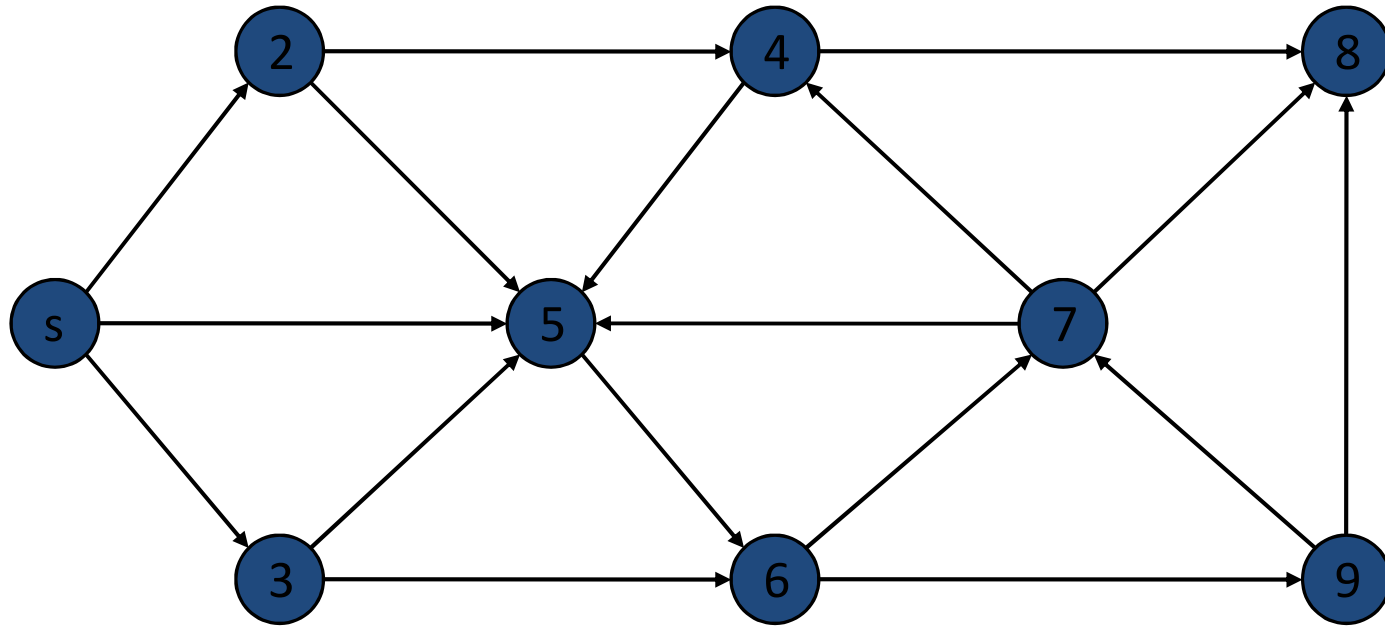
$\text{fila} = \{y\}$   
 $\text{adj} = \{\}$   
 $\text{vist} = \{s, r, w, t, x, u\}$



$\text{fila} = \{\}$   
 $\text{adj} = \{\}$   
 $\text{vist} = \{s, r, w, t, x, u, y\}$

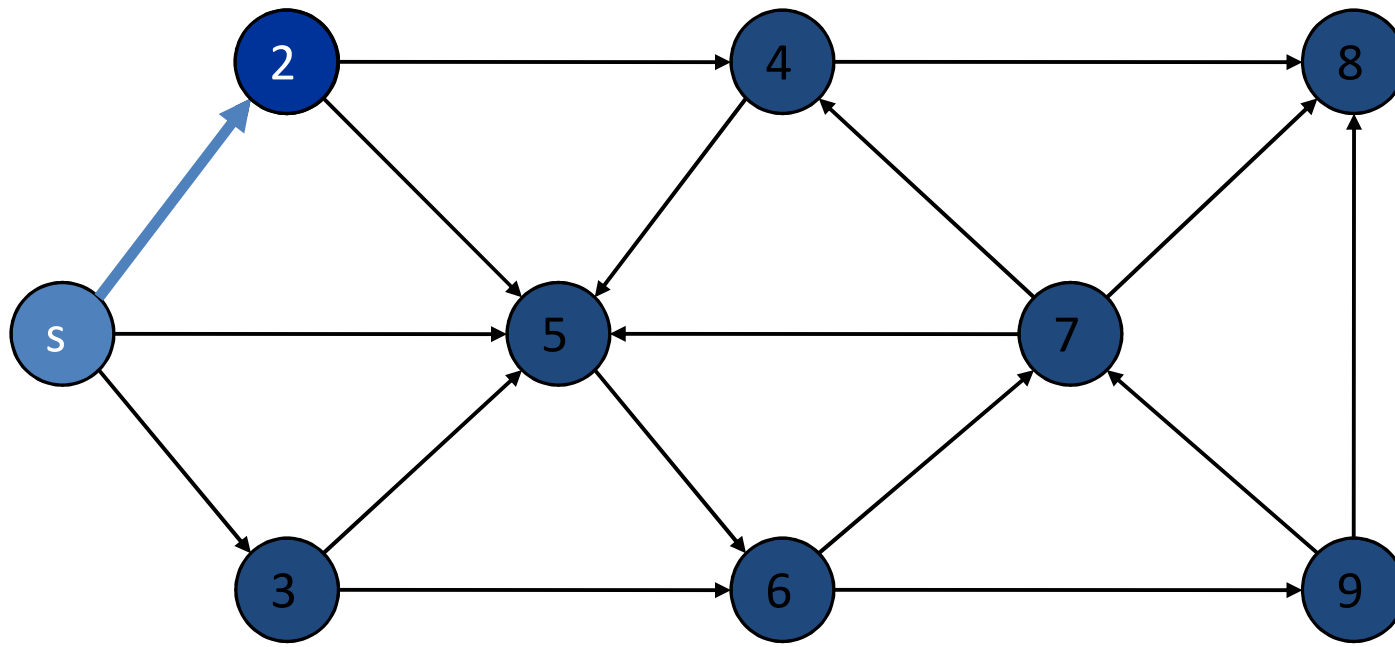


# Exercício- Breadth First Search



<http://www.cc.gatech.edu/~bader/COURSES/GATECH/CS6505-Spring2006/demo-bfs.ppt>

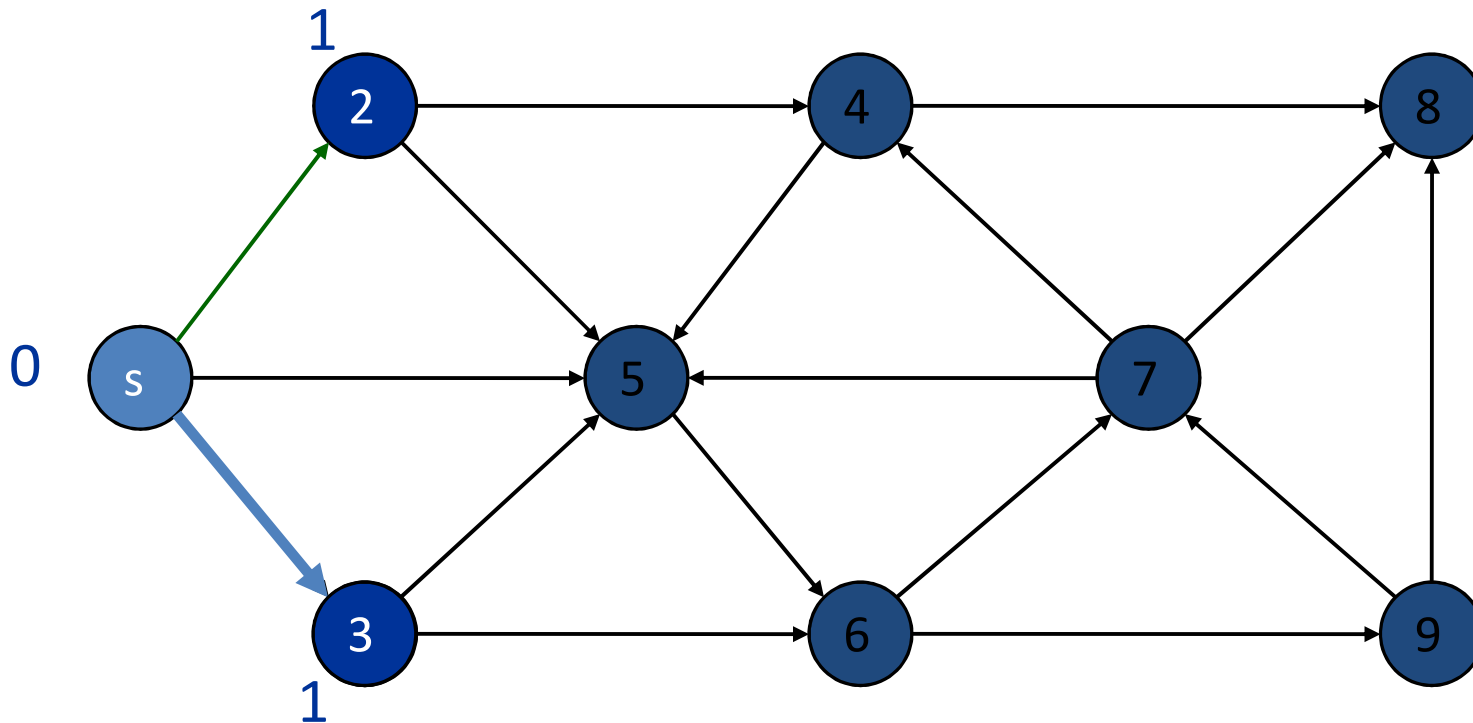
# Breadth First Search



|               |
|---------------|
| Sem descobrir |
| Descoberto    |
| Top da Pilha  |
| Fim           |

Pilha: s

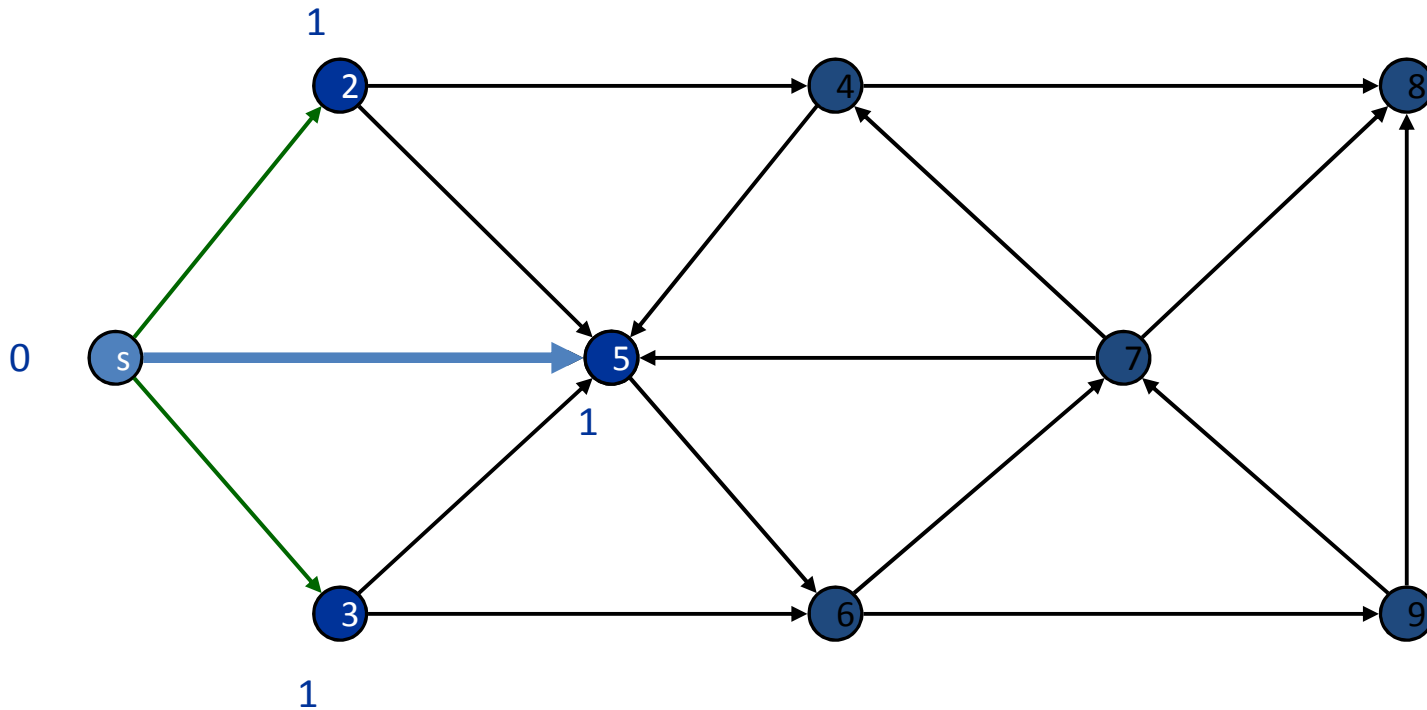
# Breadth First Search



|               |
|---------------|
| Sem descobrir |
| Descoberto    |
| Top of Pilha  |
| Fim           |

Pilha: s 2

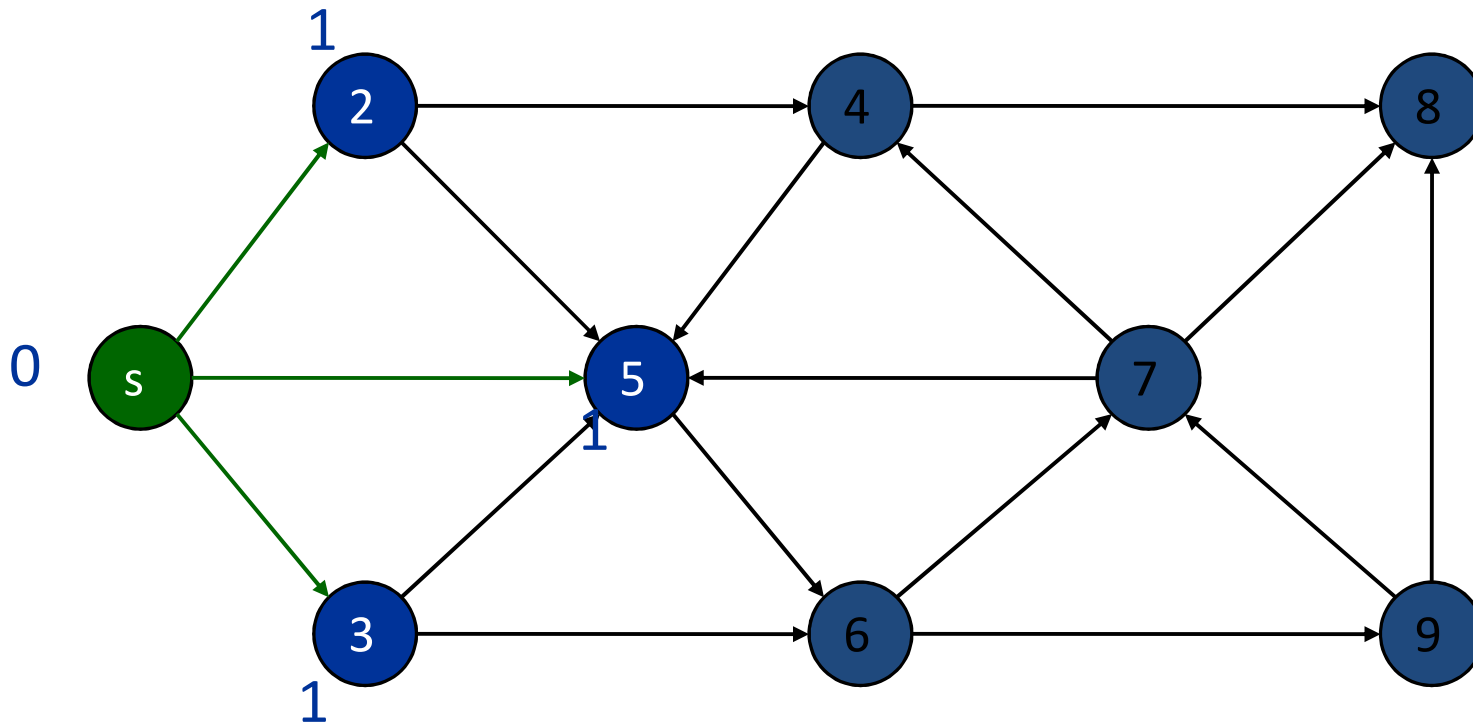
# Breadth First Search



|               |
|---------------|
| Sem descobrir |
| Descoberto    |
| Top of Pilha  |
| Fim           |

Pilha: s 2 3

# Breadth First Search

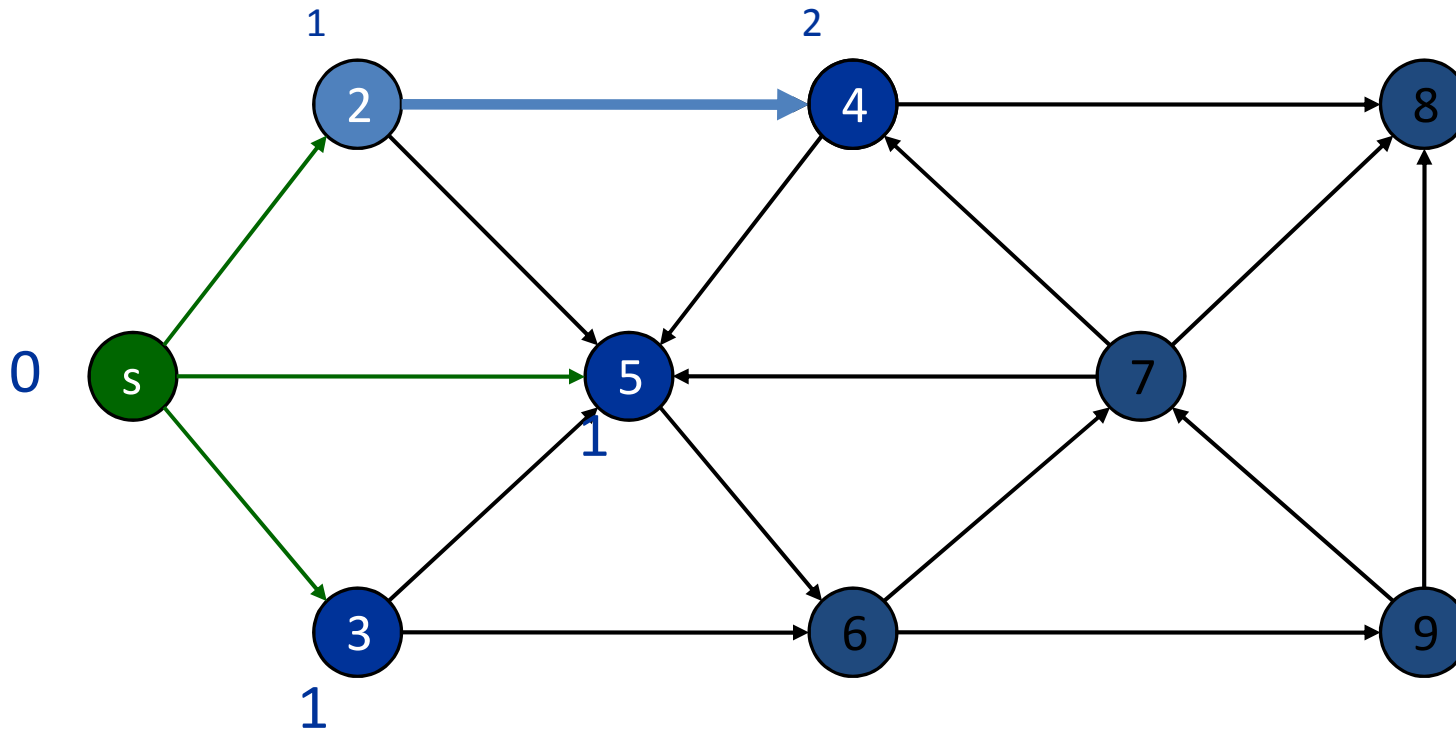


|               |
|---------------|
| Sem descobrir |
| Descoberto    |
| Top of Pilha  |
| Fim           |

Pilha: 2 3 5



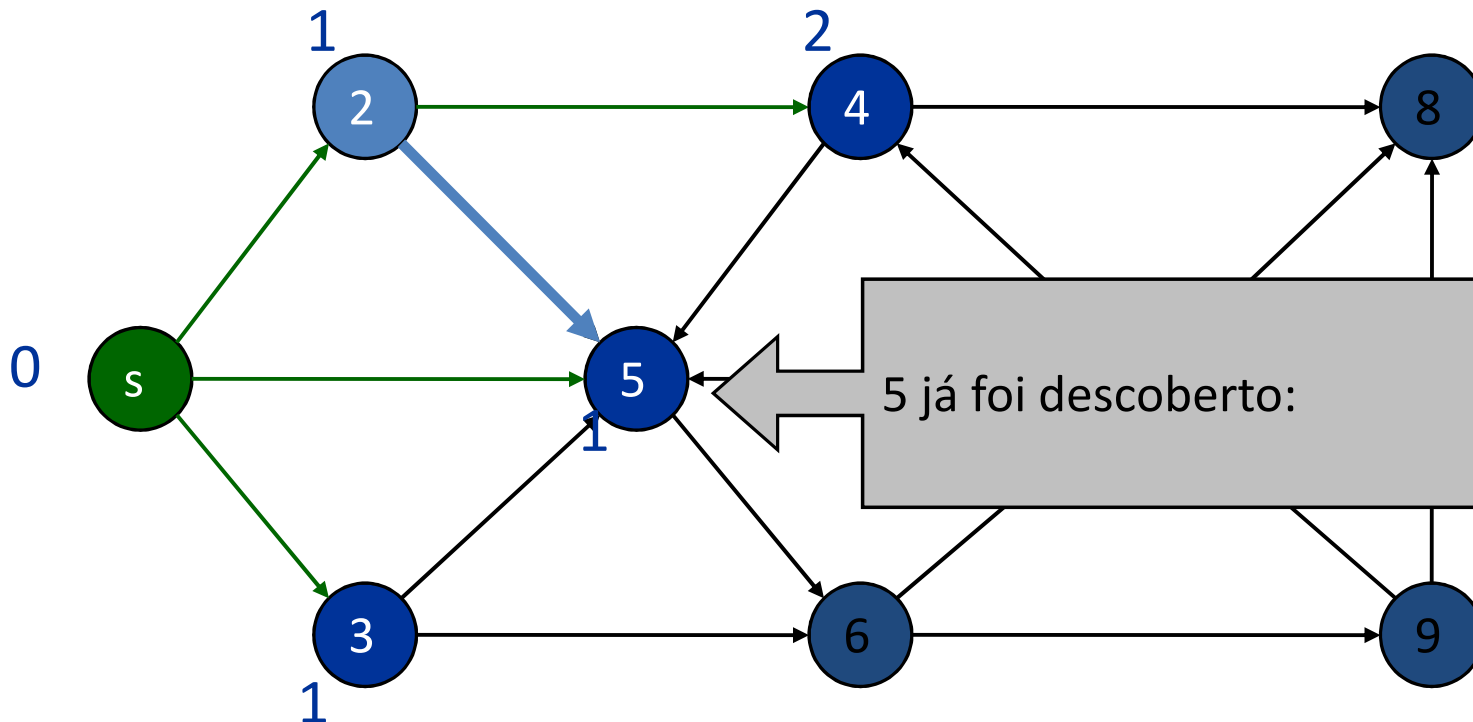
# Breadth First Search



|               |
|---------------|
| Sem descobrir |
| Descoberto    |
| Top of Pilha  |
| Fim           |

Pilha: 2 3 5

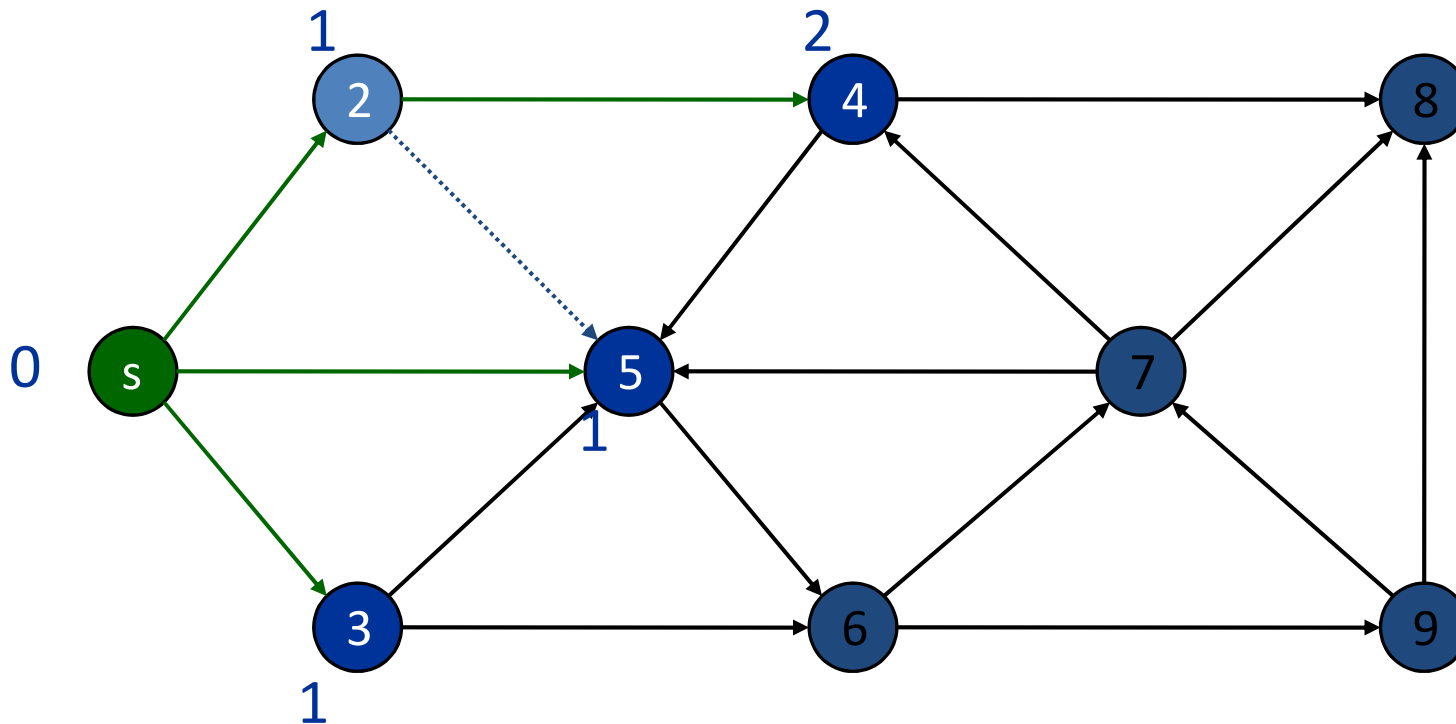
# Breadth First Search



|               |
|---------------|
| Sem descobrir |
| Descoberto    |
| Top of Pilha  |
| Fim           |

Pilha: 2 3 5 4

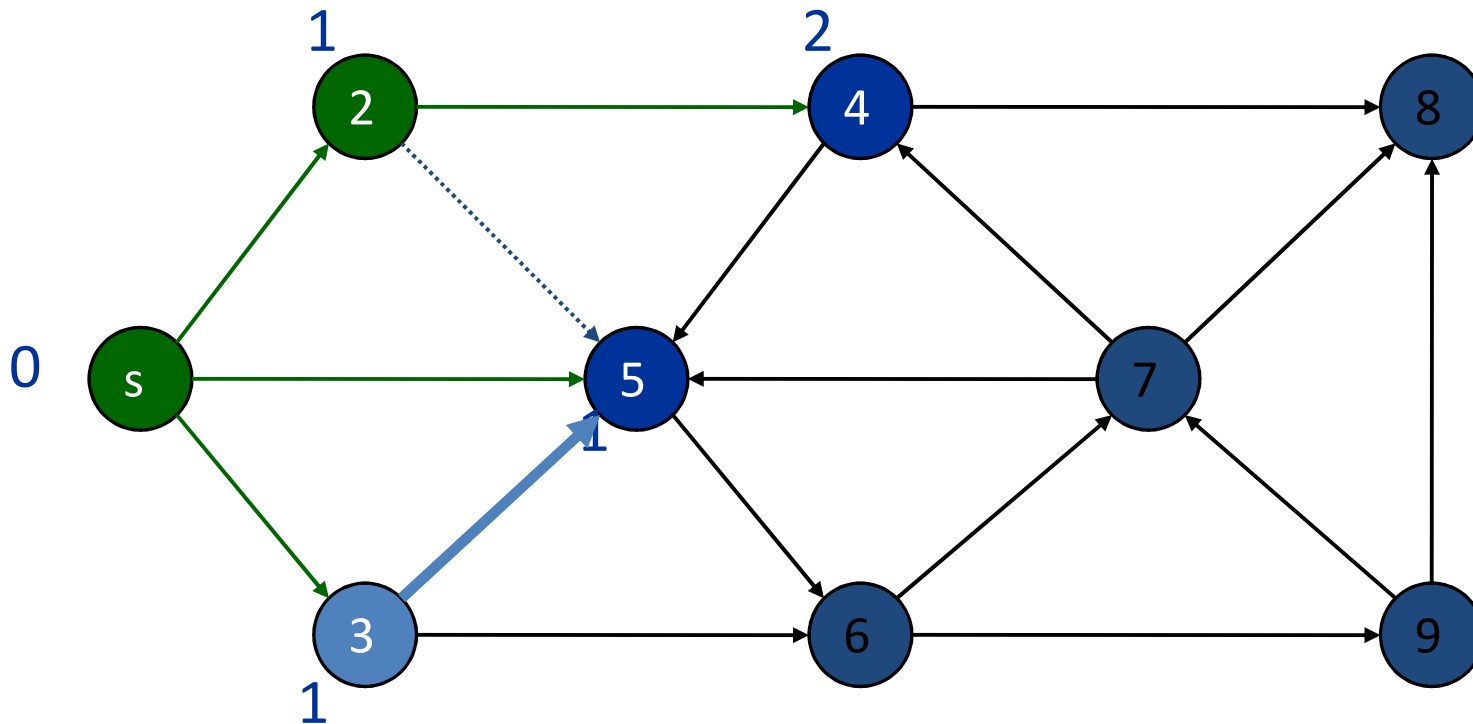
# Breadth First Search



|               |
|---------------|
| Sem descobrir |
| Descoberto    |
| Top of Pilha  |
| Fim           |

Pilha: 2 3 5 4

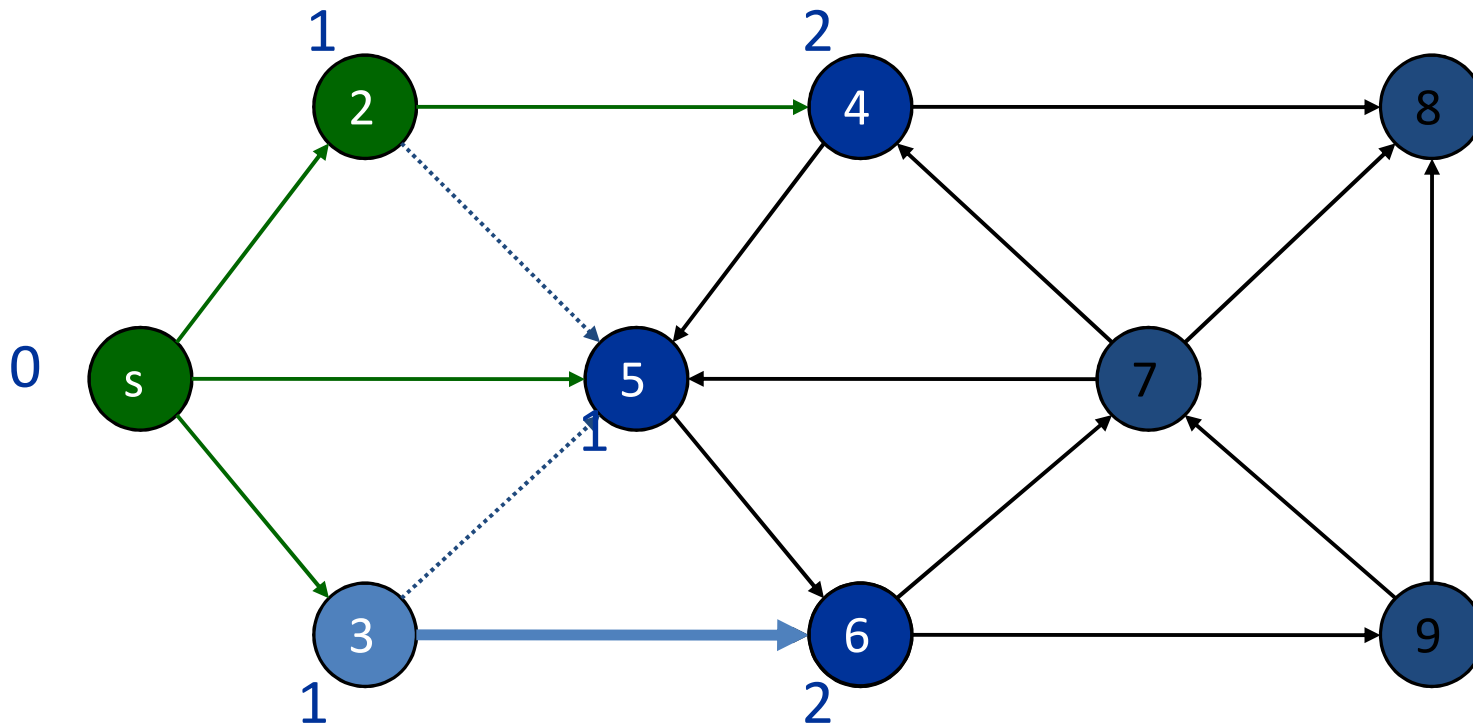
# Breadth First Search



|               |
|---------------|
| Sem descobrir |
| Descoberto    |
| Top of Pilha  |
| Fim           |

Pilha: 3 5 4

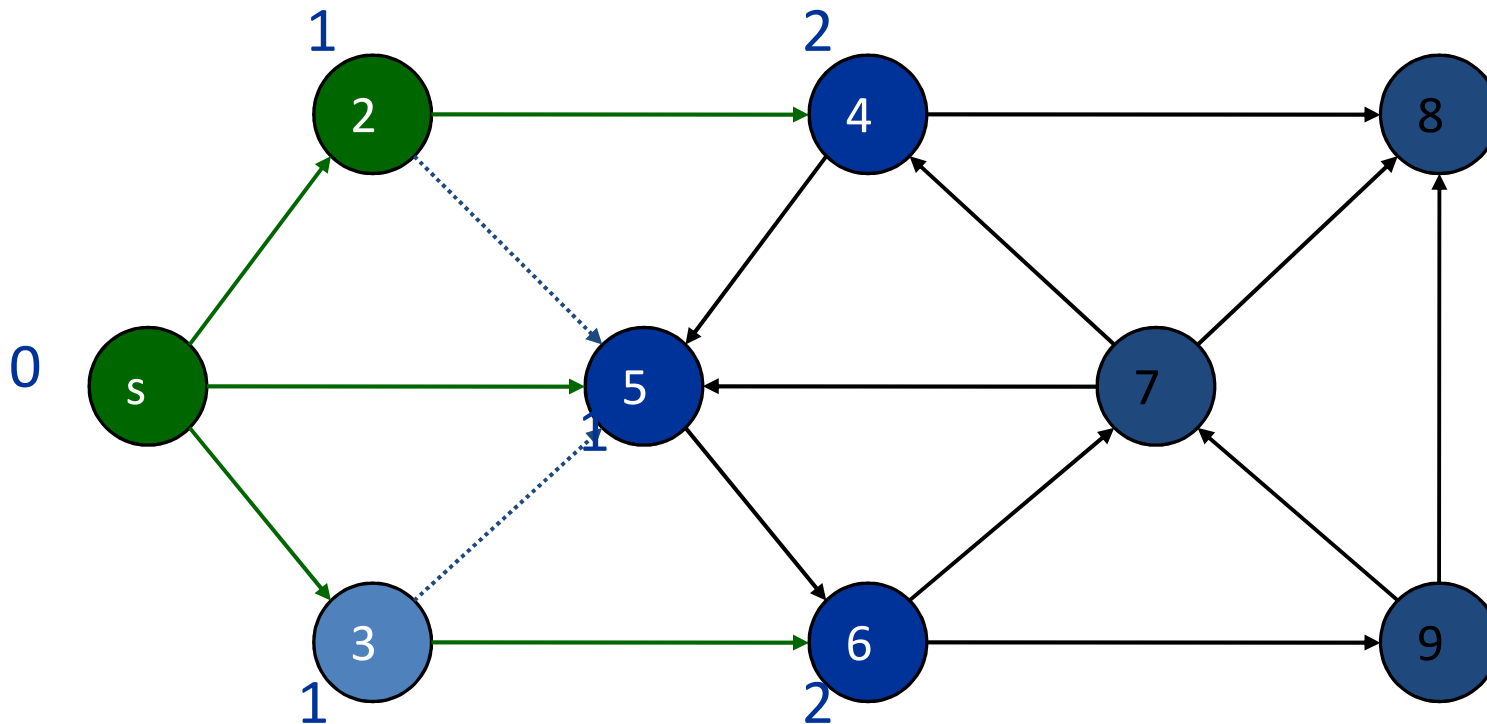
# Breadth First Search



|               |
|---------------|
| Sem descobrir |
| Descoberto    |
| Top of Pilha  |
| Fim           |

Pilha: 3 5 4

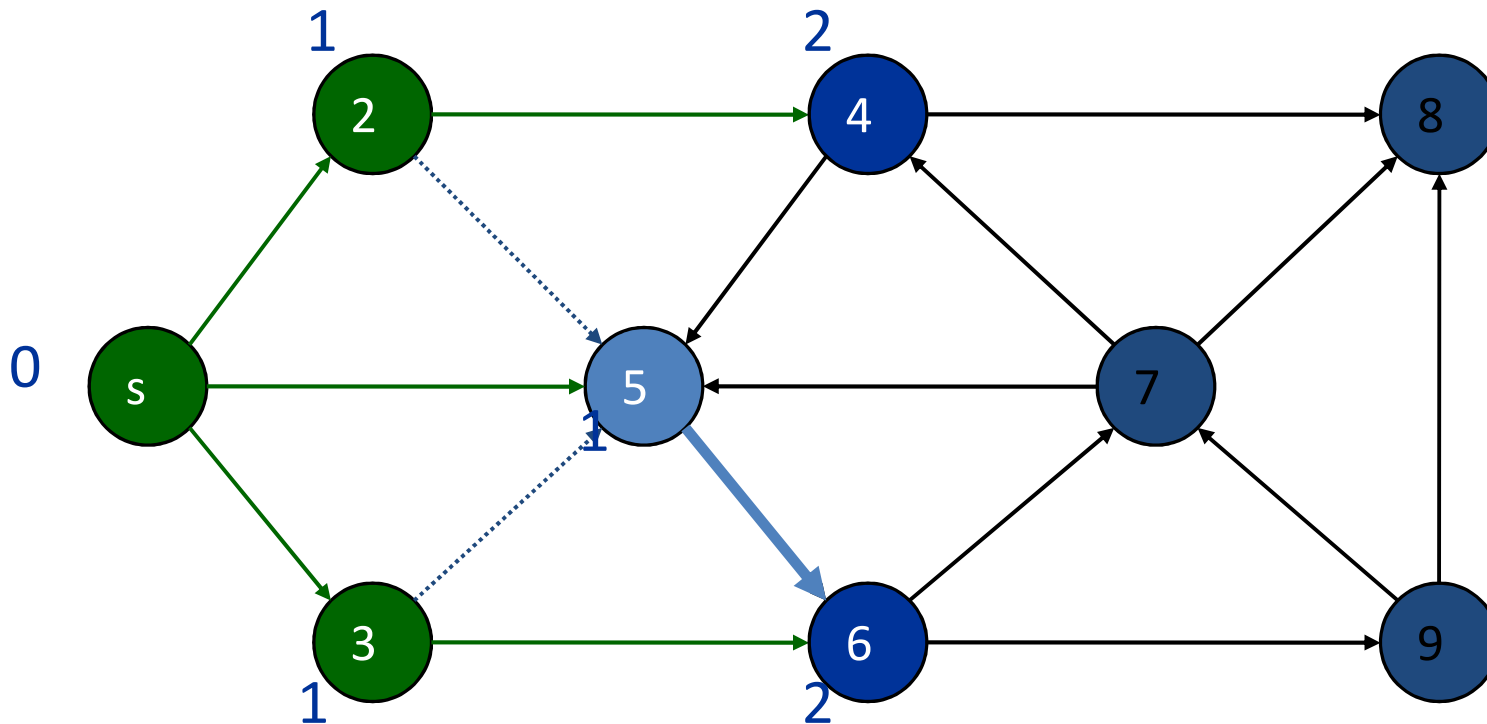
# Breadth First Search



|               |
|---------------|
| Sem descobrir |
| Descoberto    |
| Top of Pilha  |
| Fim           |

Pilha: 3 5 4 6

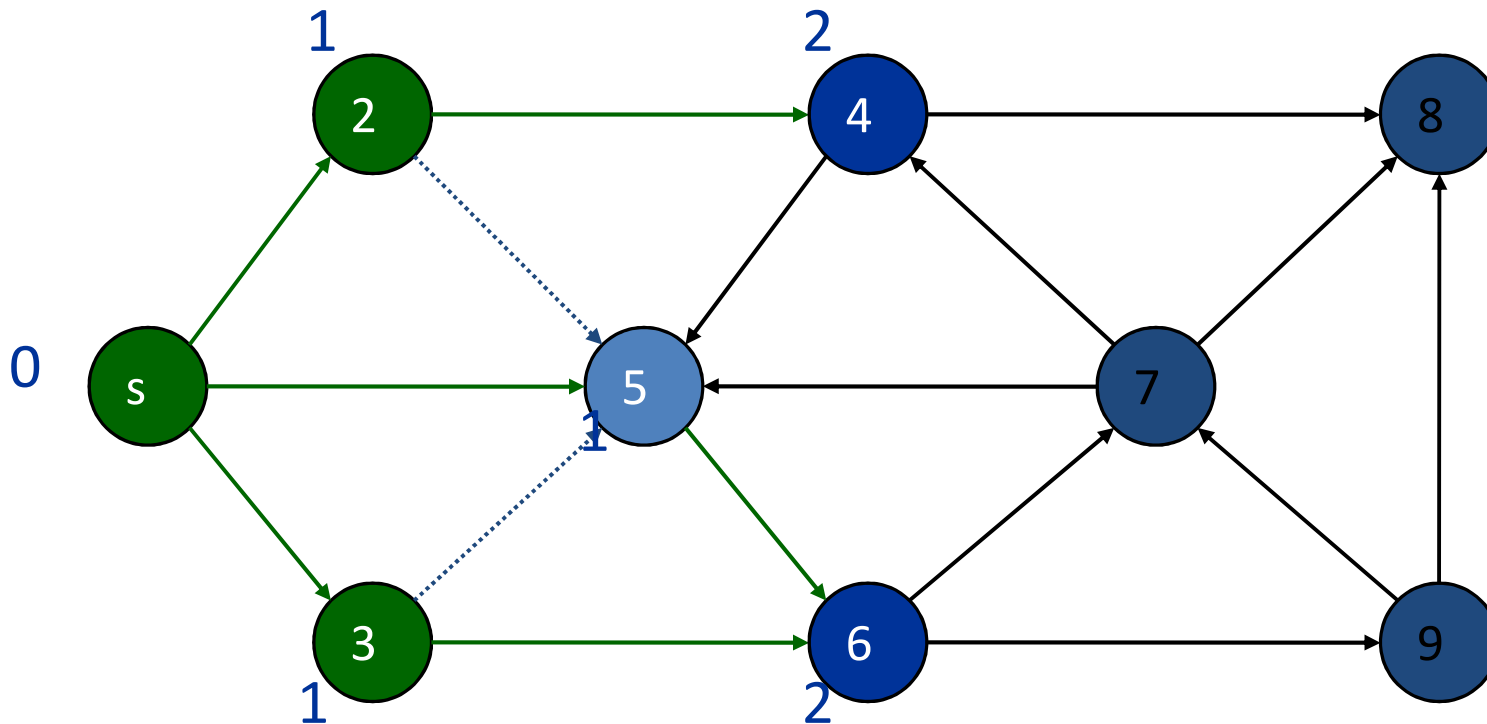
# Breadth First Search



|               |
|---------------|
| Sem descobrir |
| Descoberto    |
| Top of Pilha  |
| Fim           |

Pilha: 5 4 6

# Breadth First Search

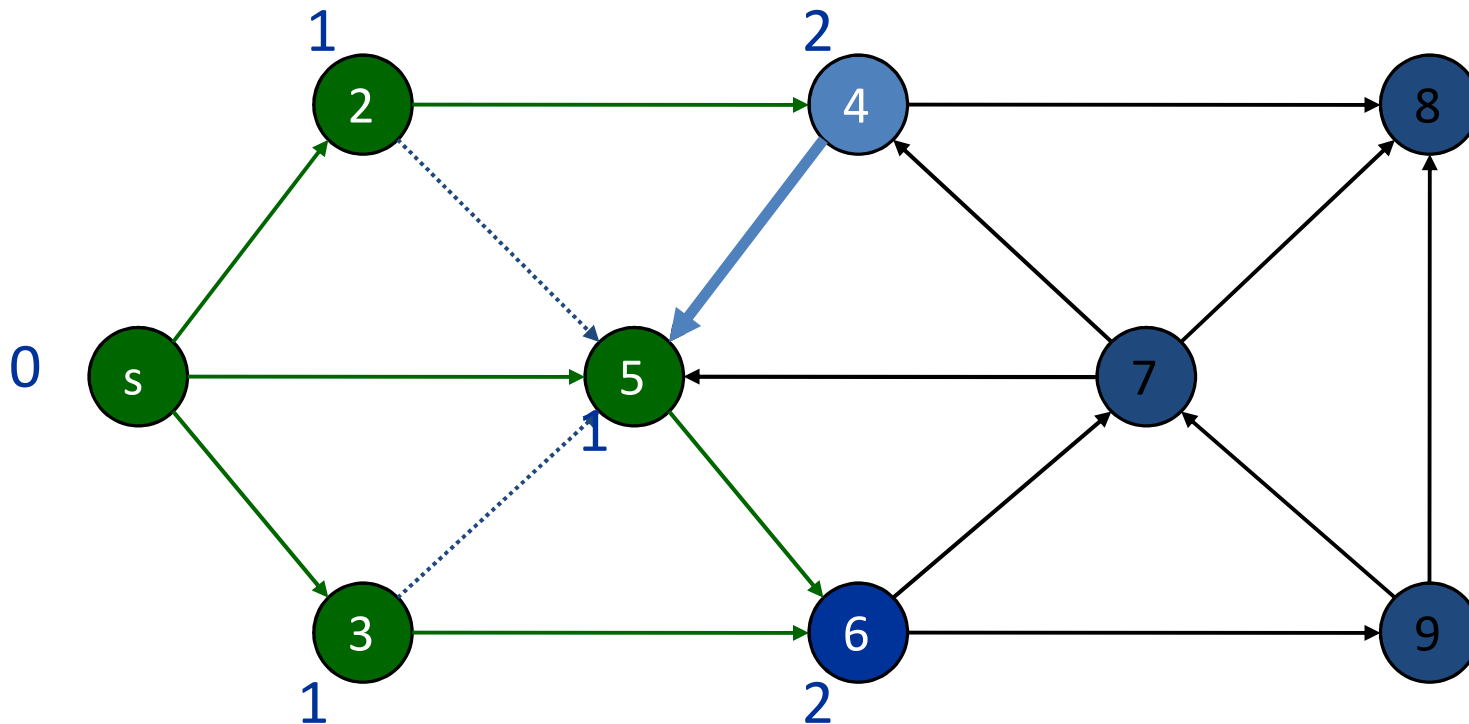


|               |
|---------------|
| Sem descobrir |
| Descoberto    |
| Top of Pilha  |
| Fim           |

Pilha: 5 4 6



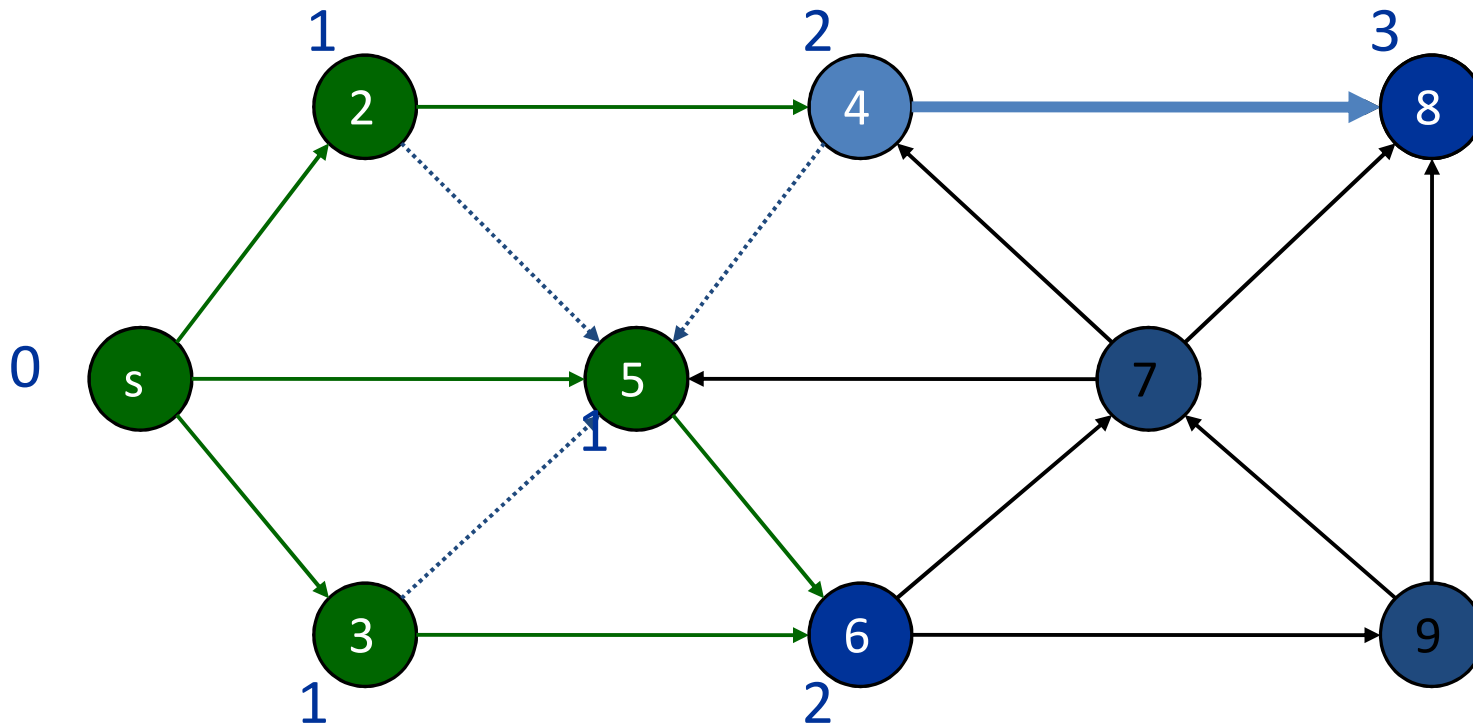
# Breadth First Search



|               |
|---------------|
| Sem descobrir |
| Descoberto    |
| Top of Pilha  |
| Fim           |

Pilha: 4 6

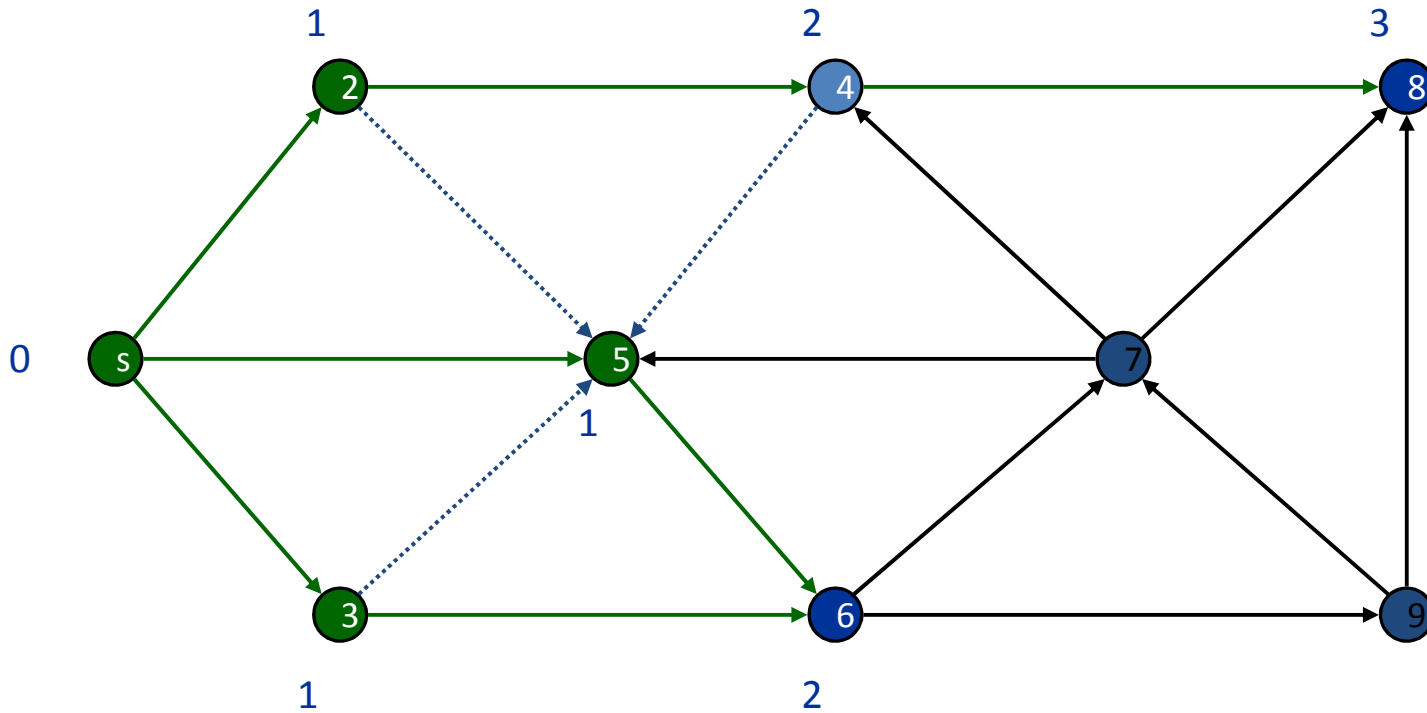
# Breadth First Search



|               |
|---------------|
| Sem descobrir |
| Descoberto    |
| Top of Pilha  |
| Fim           |

Pilha: 4 6

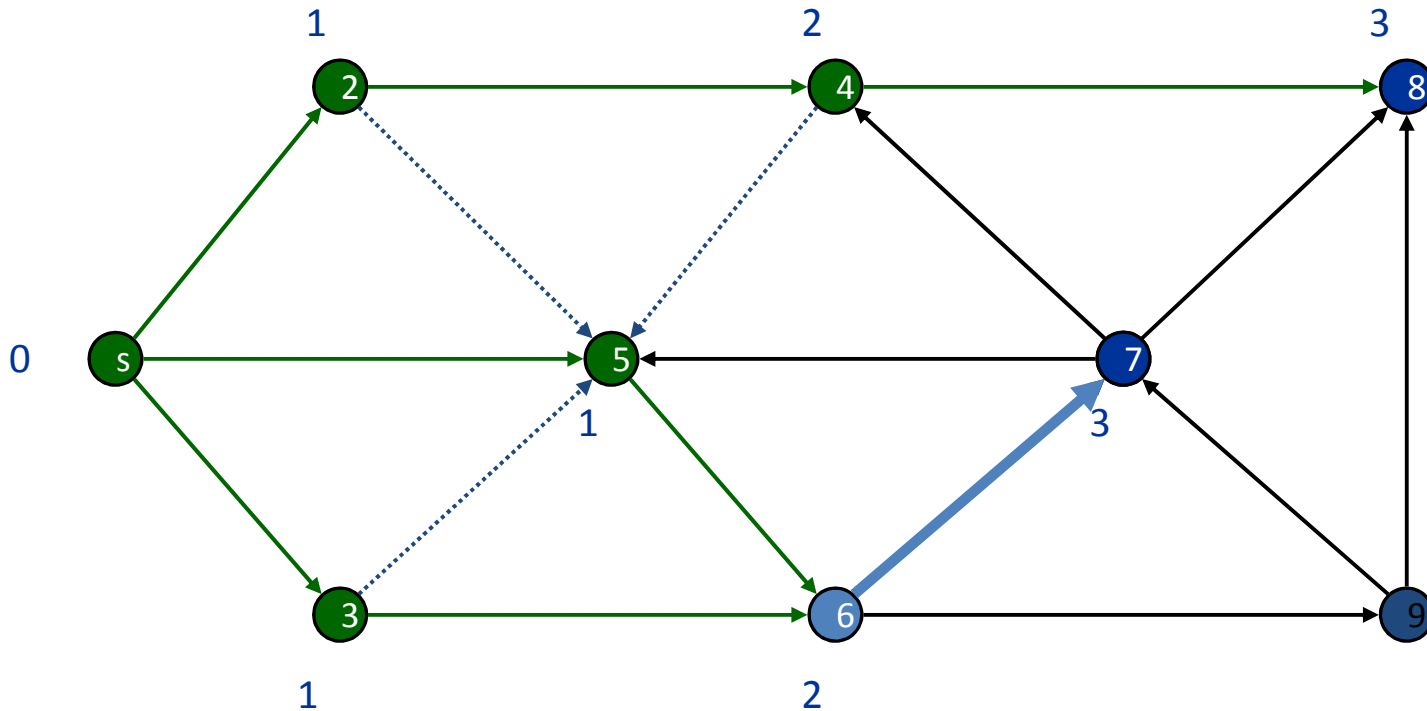
# Breadth First Search



|               |
|---------------|
| Sem descobrir |
| Descoberto    |
| Top of Pilha  |
| Fim           |

Pilha: 4 6 8

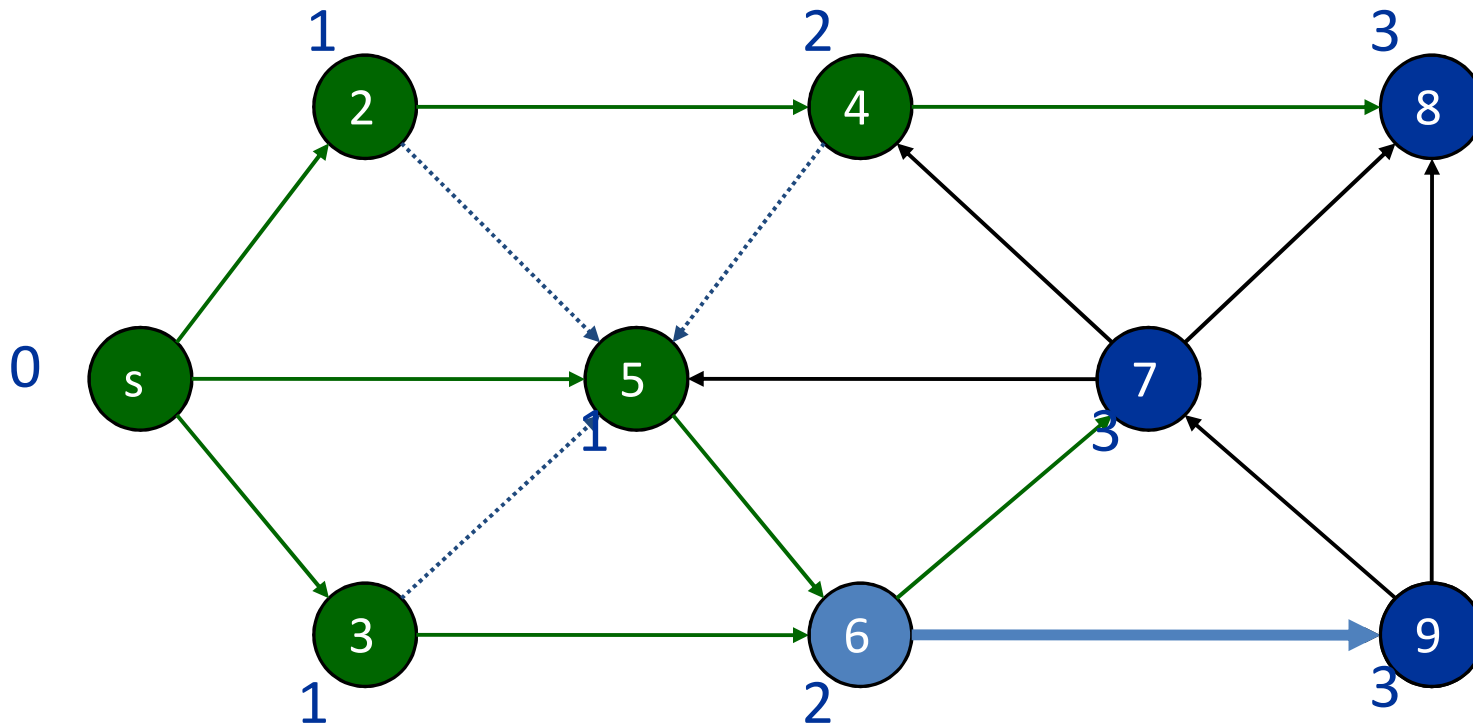
# Breadth First Search



|               |
|---------------|
| Sem descobrir |
| Descoberto    |
| Top of Pilha  |
| Fim           |

Pilha: 6 8

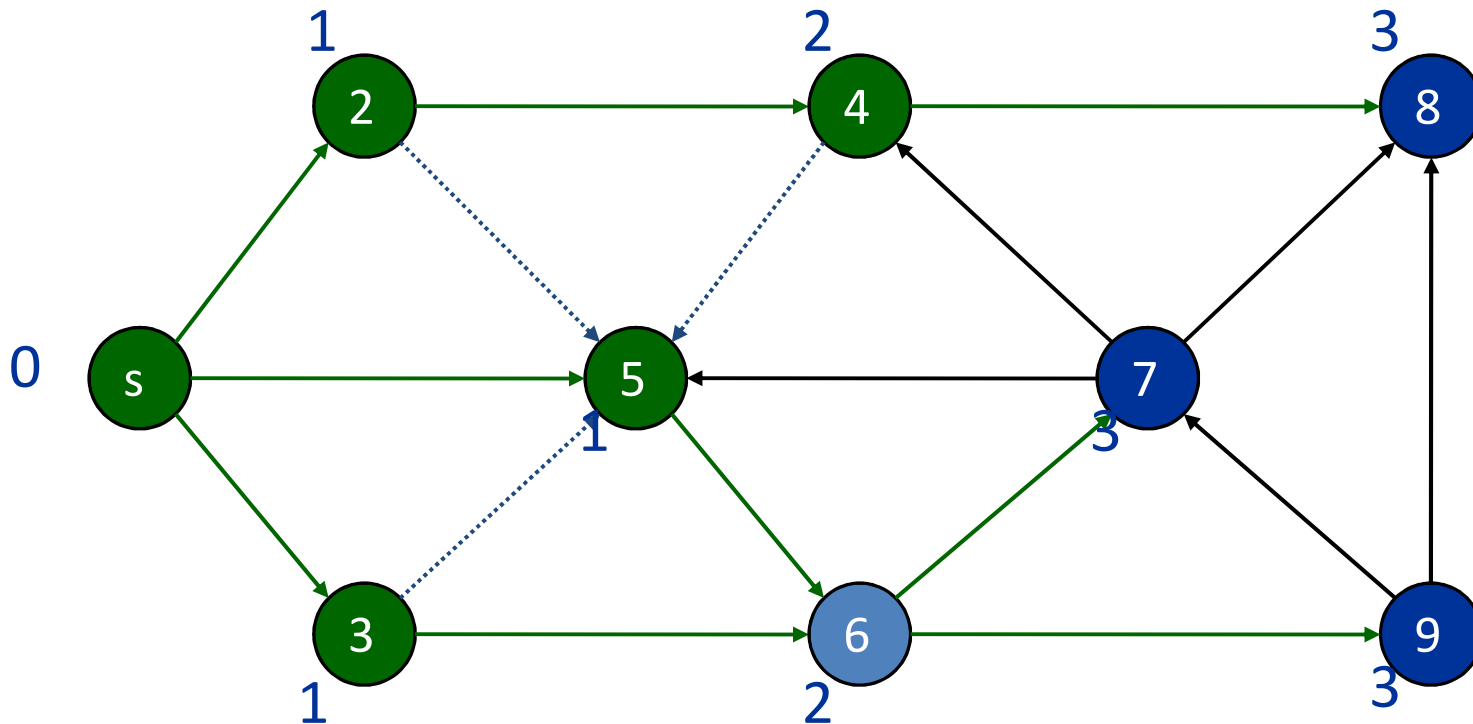
# Breadth First Search



|               |
|---------------|
| Sem descobrir |
| Descoberto    |
| Top of Pilha  |
| Fim           |

Pilha: 6 8 7

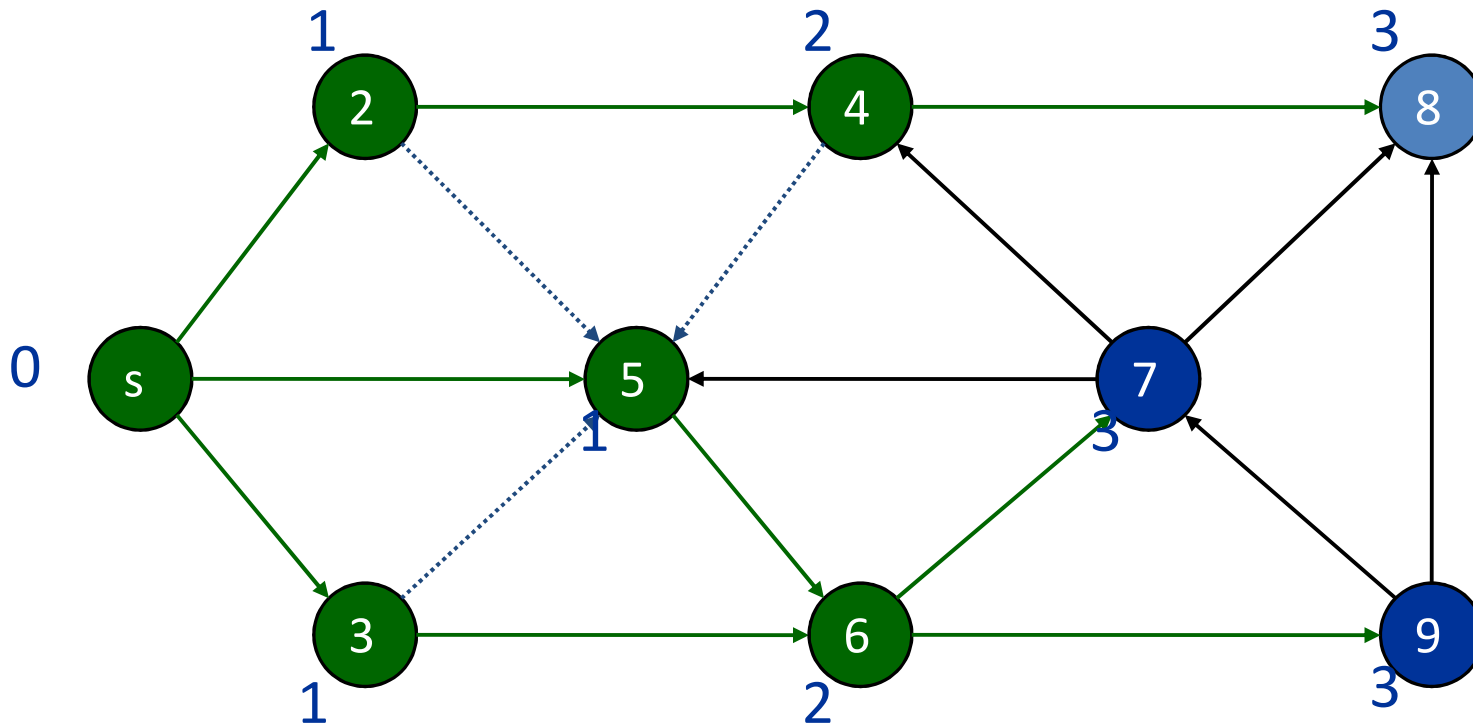
# Breadth First Search



|               |
|---------------|
| Sem descobrir |
| Descoberto    |
| Top of Pilha  |
| Fim           |

Pilha: 6 8 7 9

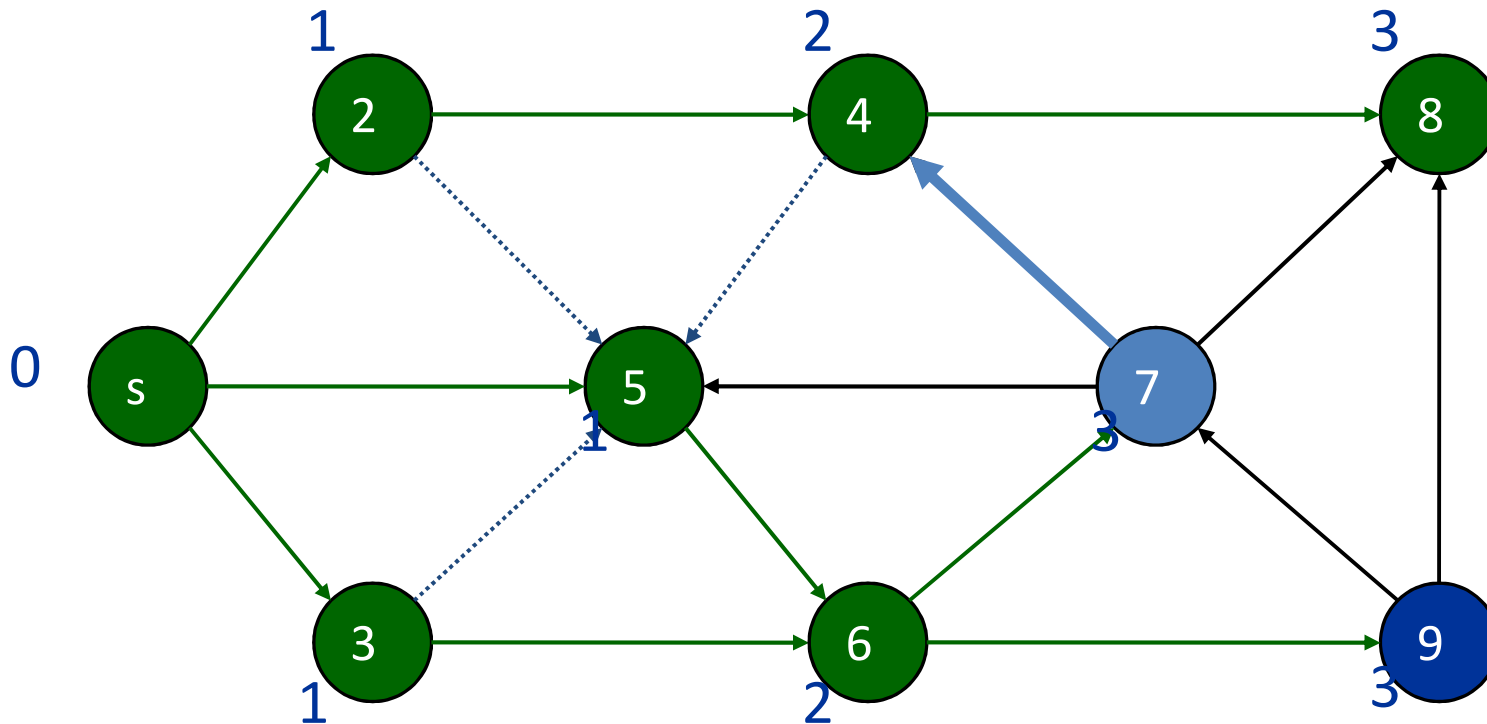
# Breadth First Search



|               |
|---------------|
| Sem descobrir |
| Descoberto    |
| Top of Pilha  |
| Fim           |

Pilha: 8 7 9

# Breadth First Search

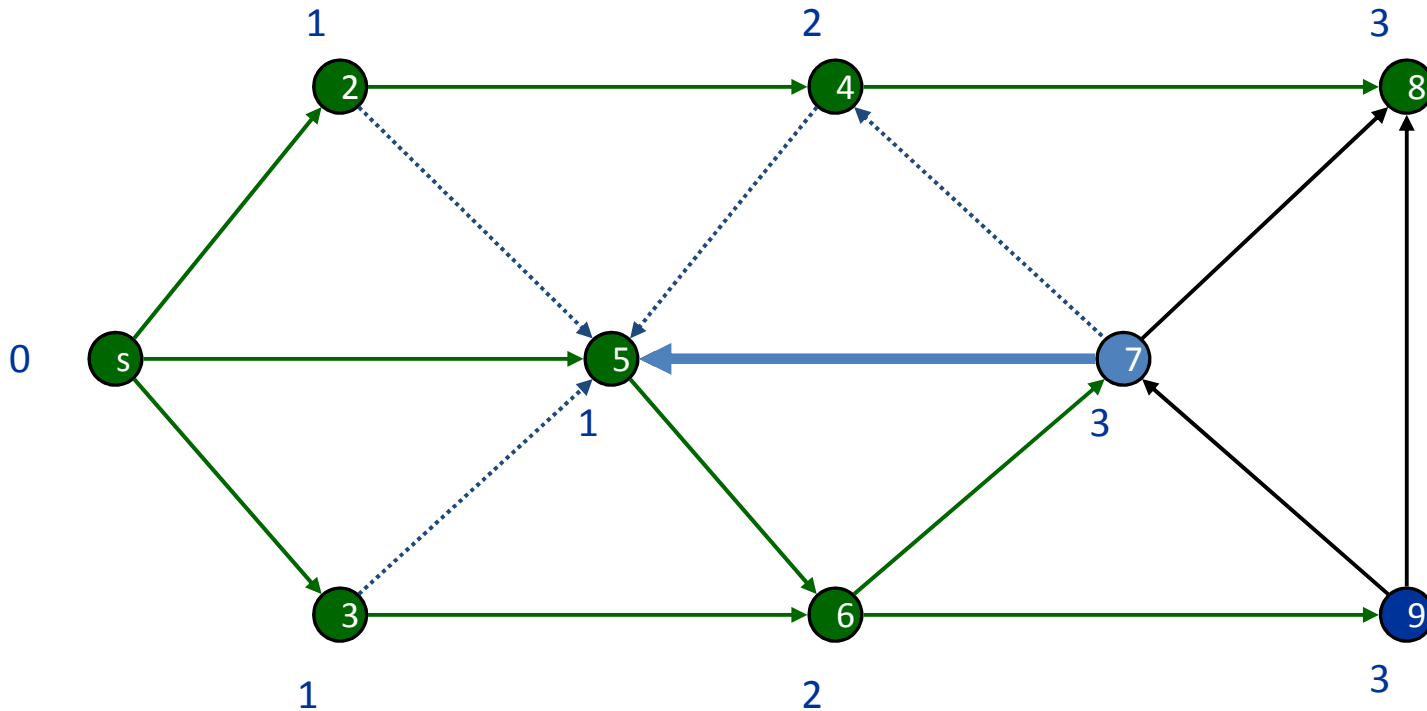


|               |
|---------------|
| Sem descobrir |
| Descoberto    |
| Top of Pilha  |
| Fim           |

Pilha: 7 9



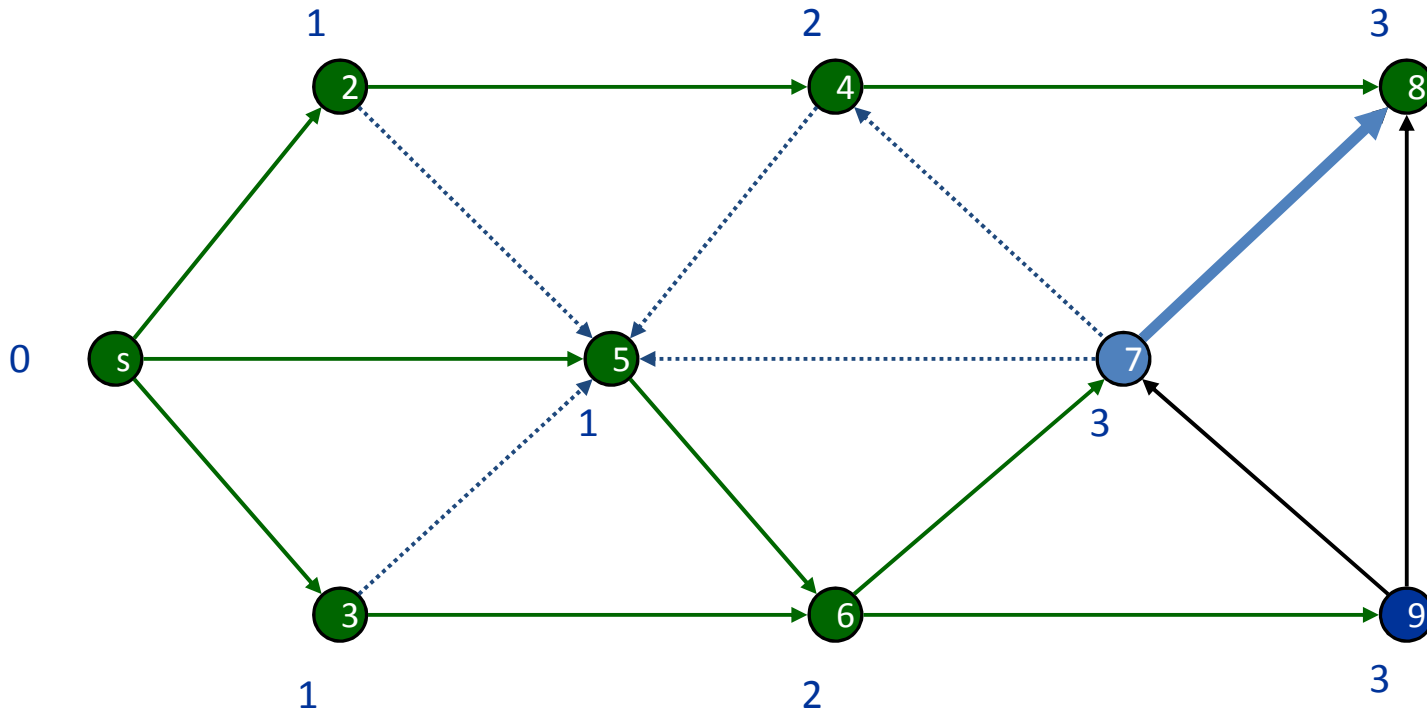
# Breadth First Search



|               |
|---------------|
| Sem descobrir |
| Descoberto    |
| Top of Pilha  |
| Fim           |

Pilha: 7 9

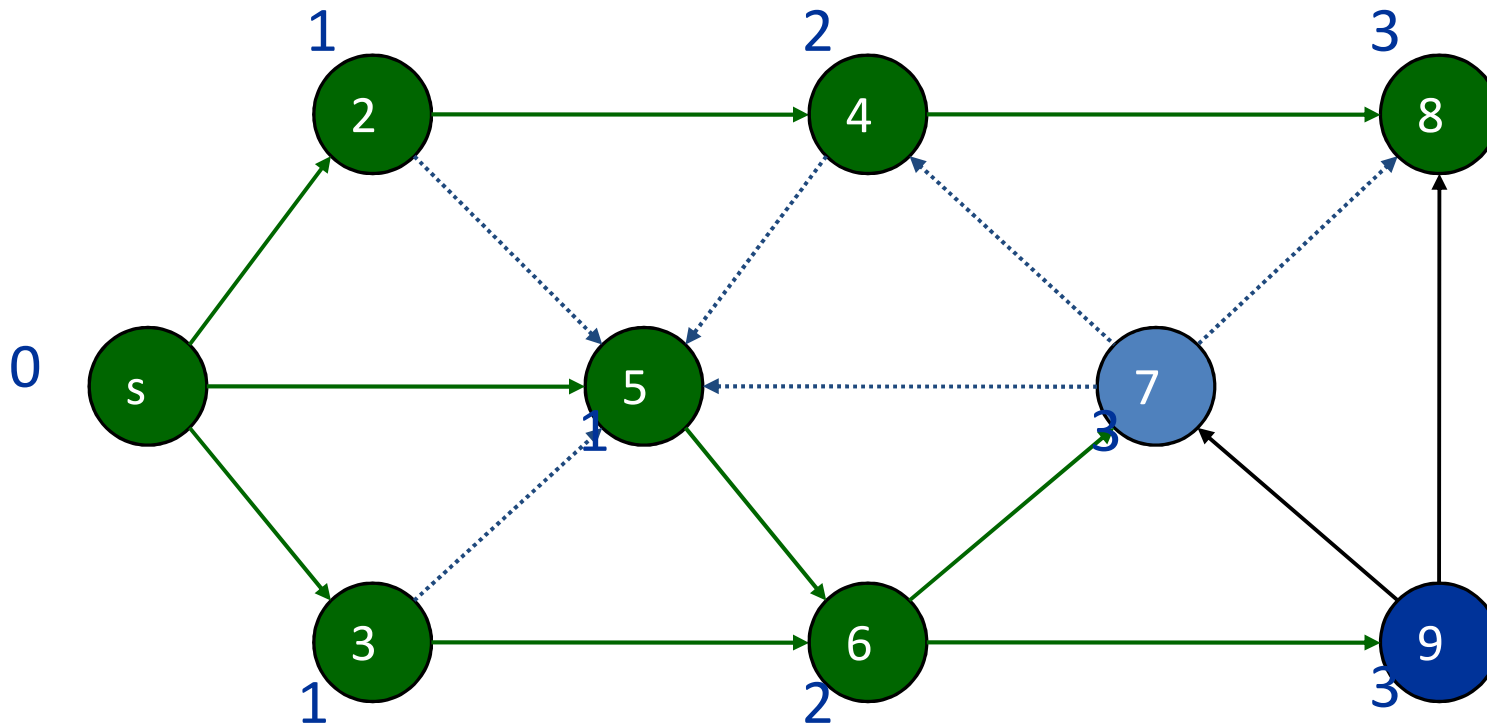
# Breadth First Search



|               |
|---------------|
| Sem descobrir |
| Descoberto    |
| Top of Pilha  |
| Fim           |

Pilha: 7 9

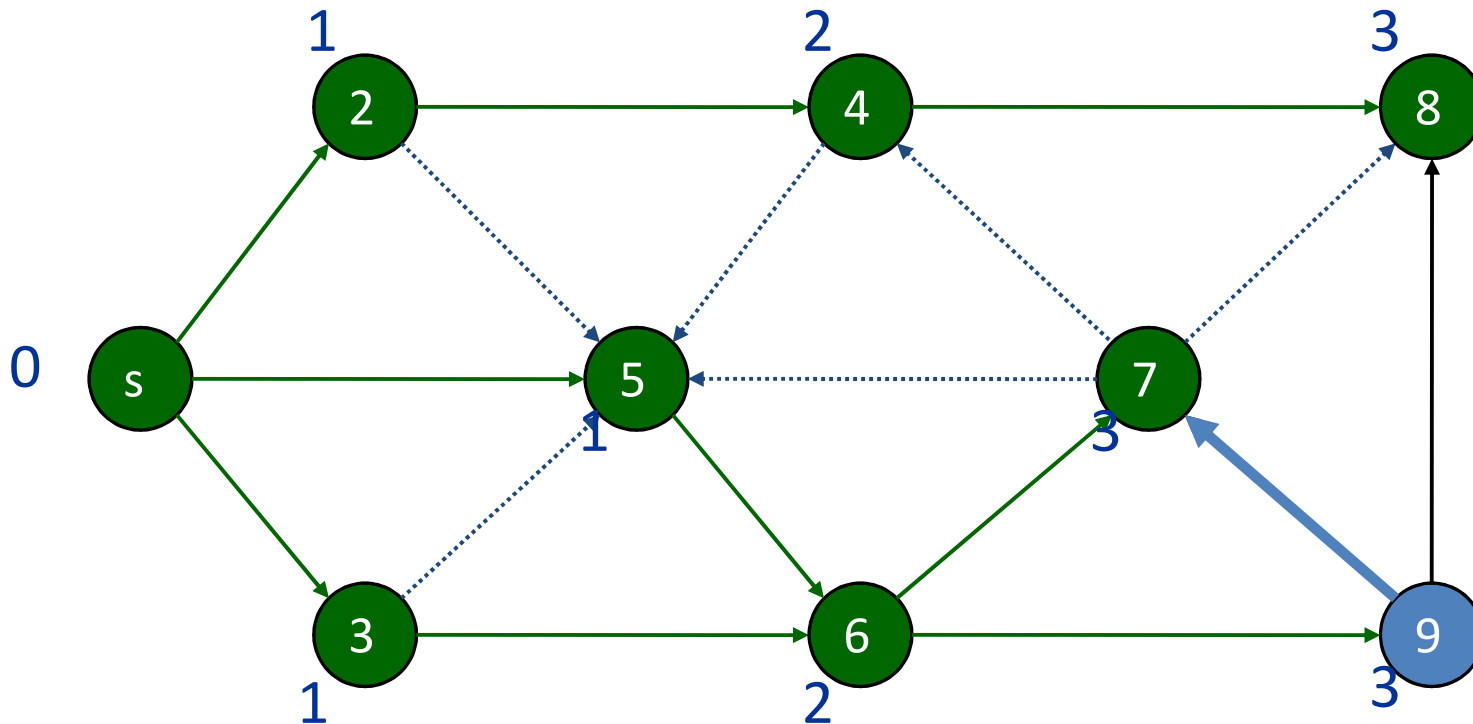
# Breadth First Search



|               |
|---------------|
| Sem descobrir |
| Descoberto    |
| Top of Pilha  |
| Fim           |

Pilha: 7 9

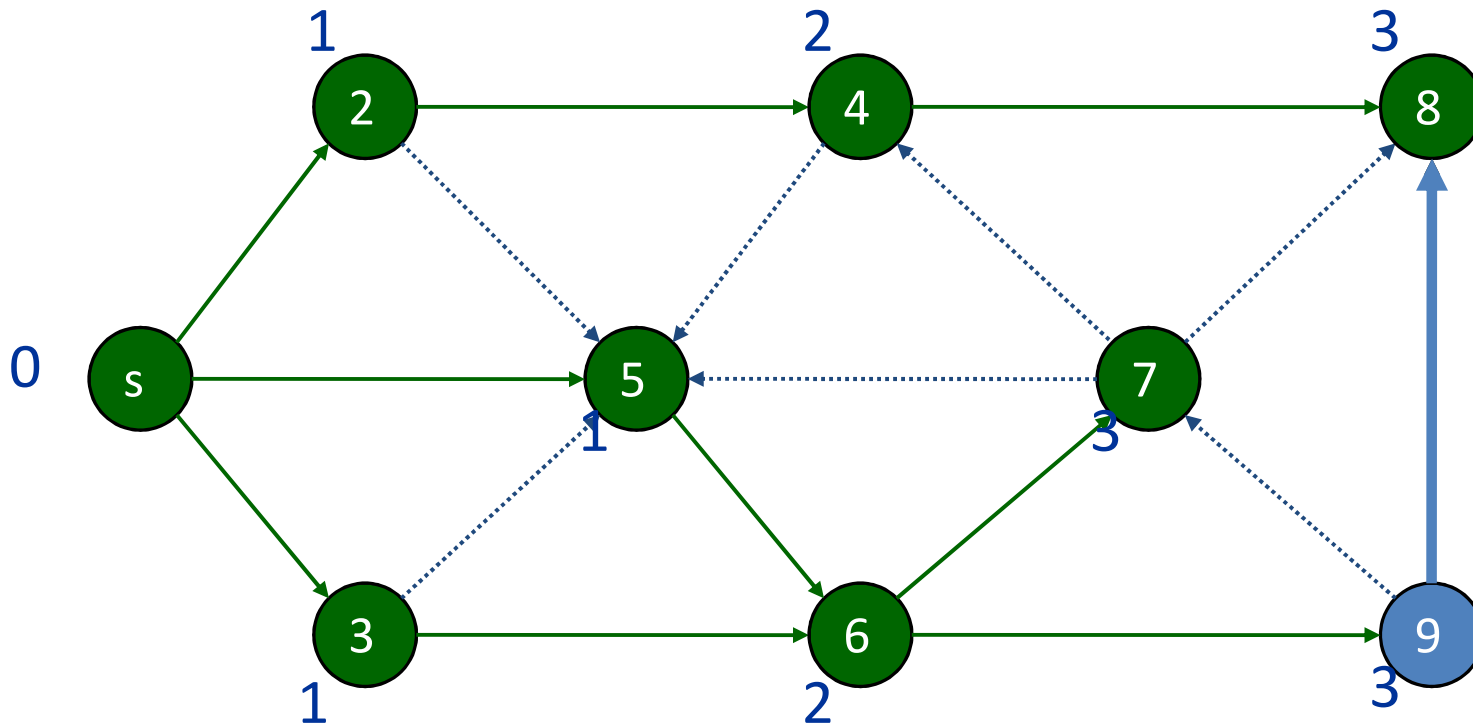
# Breadth First Search



|               |
|---------------|
| Sem descobrir |
| Descoberto    |
| Top of Pilha  |
| Fim           |

Pilha: 9

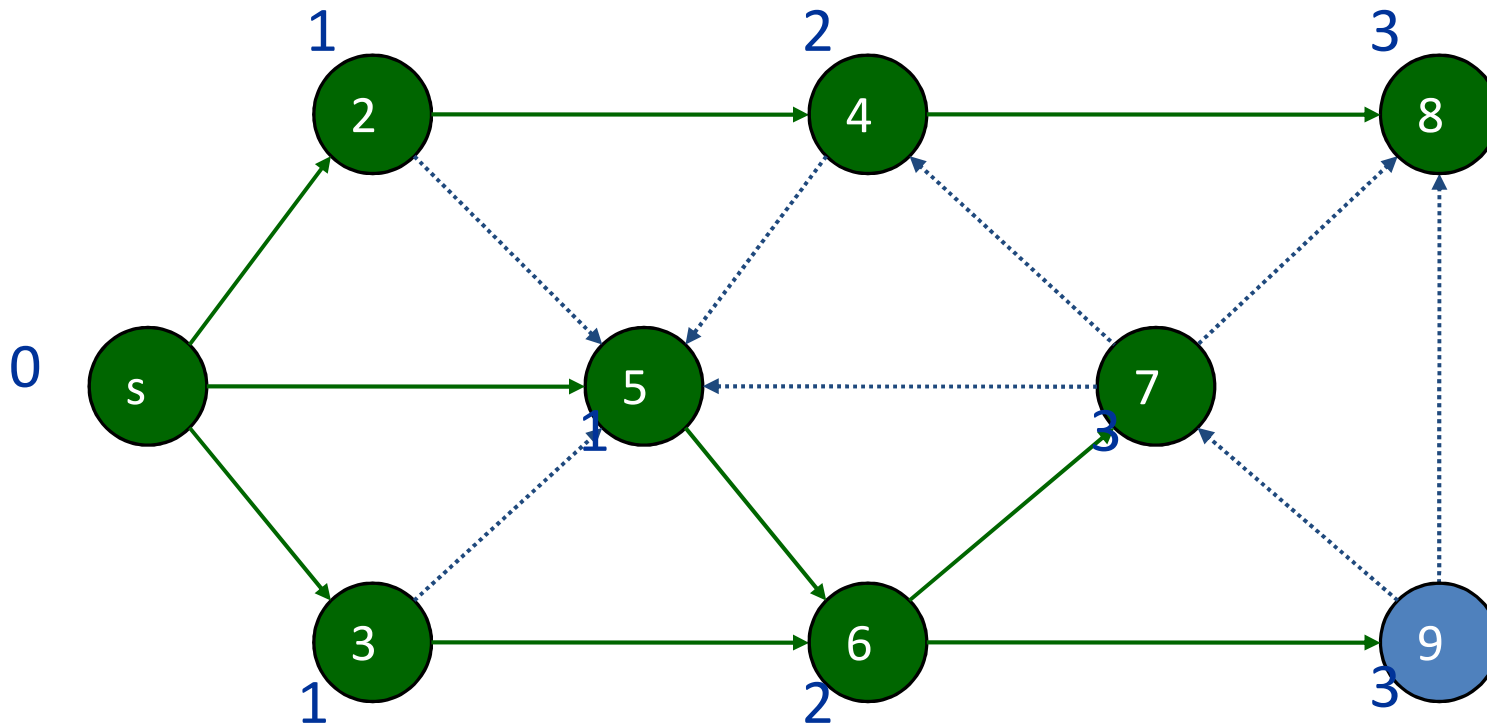
# Breadth First Search



|               |
|---------------|
| Sem descobrir |
| Descoberto    |
| Top of Pilha  |
| Fim           |

Pilha: 9

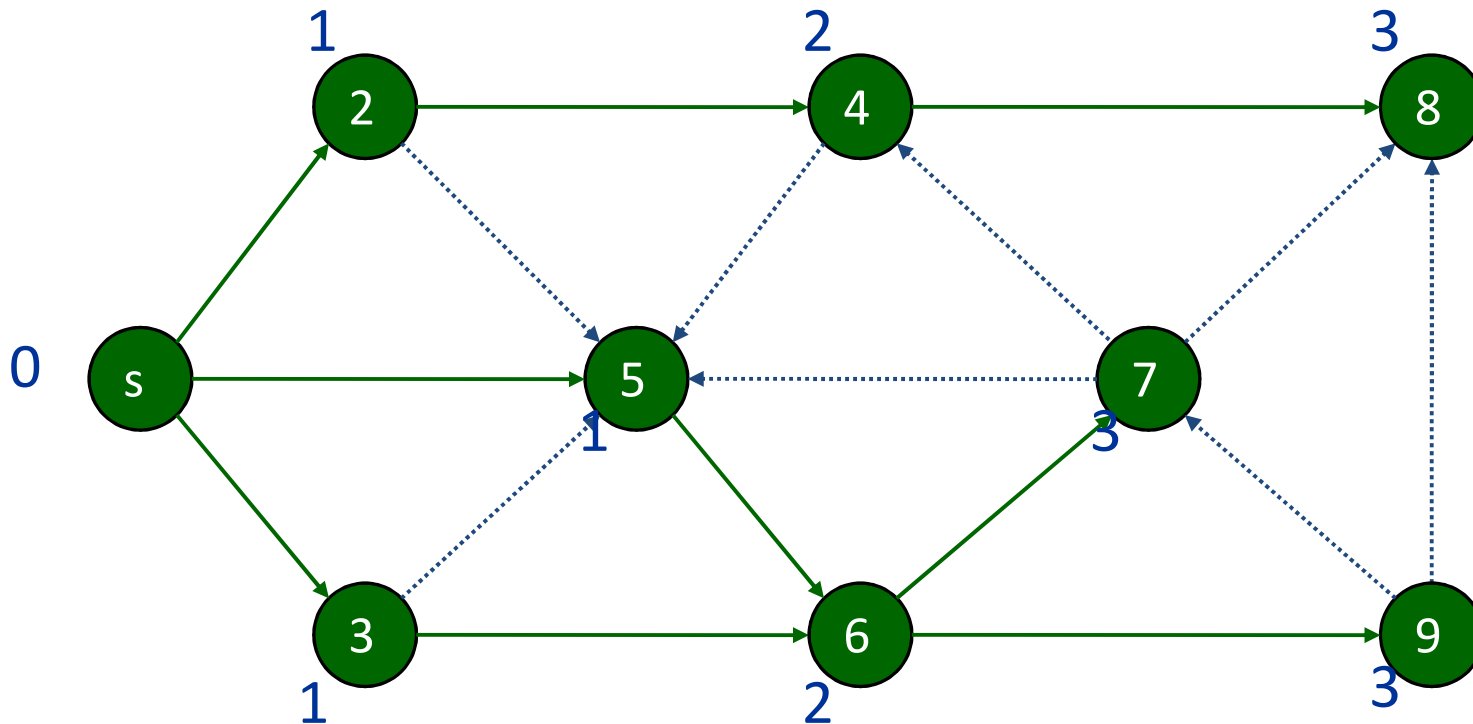
# Breadth First Search



|               |
|---------------|
| Sem descobrir |
| Descoberto    |
| Top of Pilha  |
| Fim           |

Pilha: 9

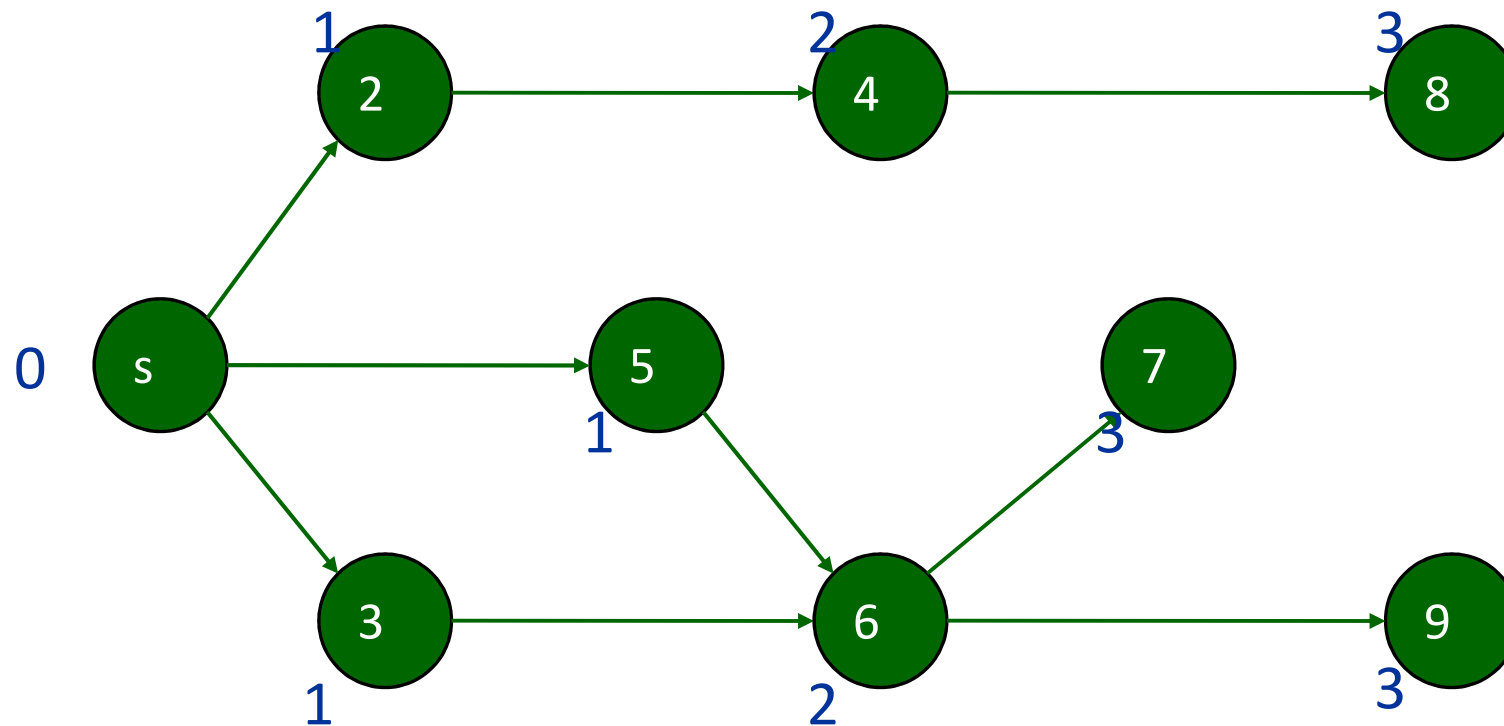
# Breadth First Search



|               |
|---------------|
| Sem descobrir |
| Descoberto    |
| Top of Pilha  |
| Fim           |

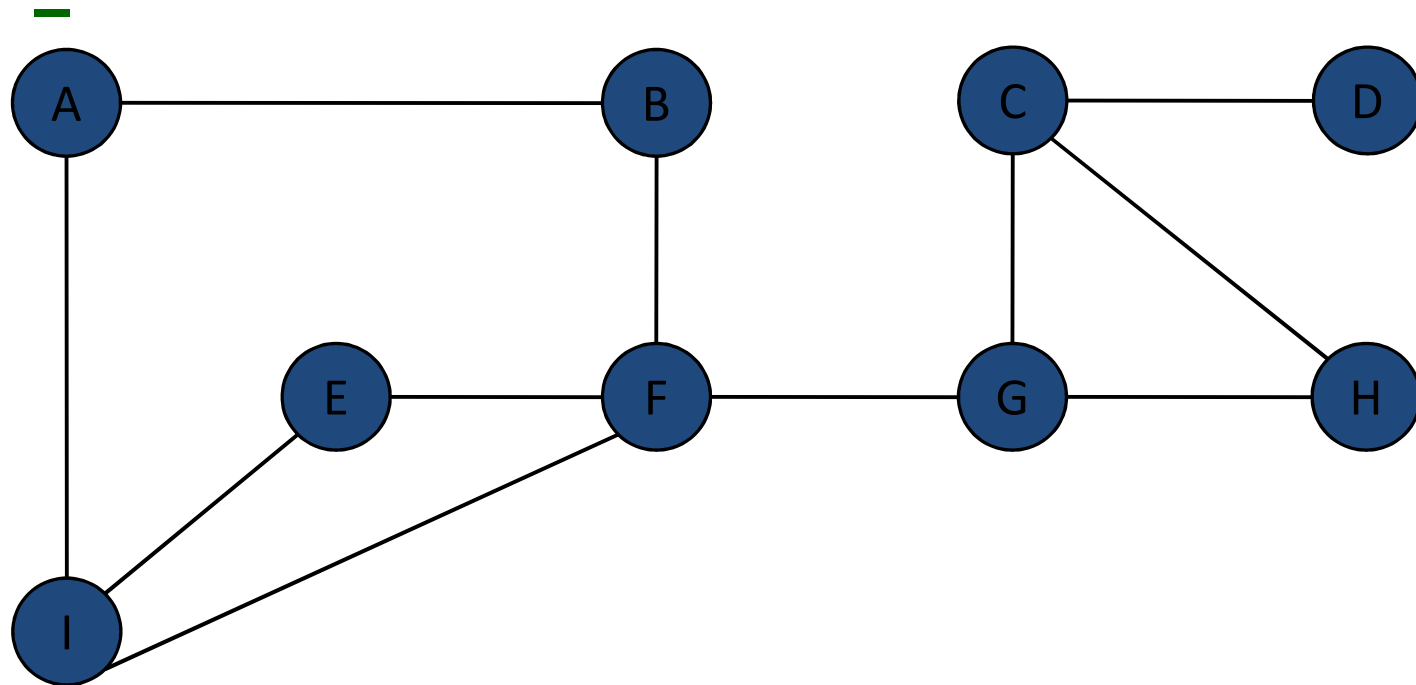
Pilha:

# Breadth First Search





# Exercício 2 - Breadth First Search



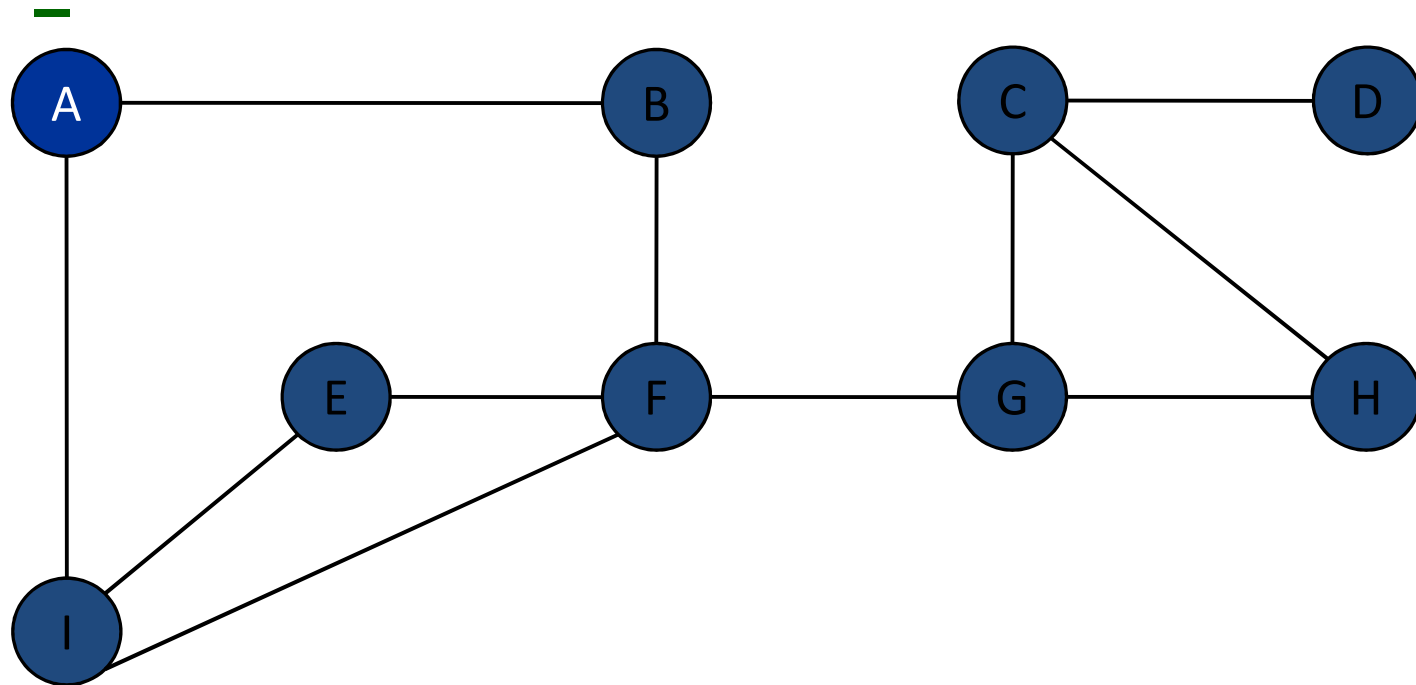
front



FIFO

<http://www.cs.princeton.edu/courses/archive/spr10/cos226/demo/51demo-bfs.ppt>

# Breadth First Search



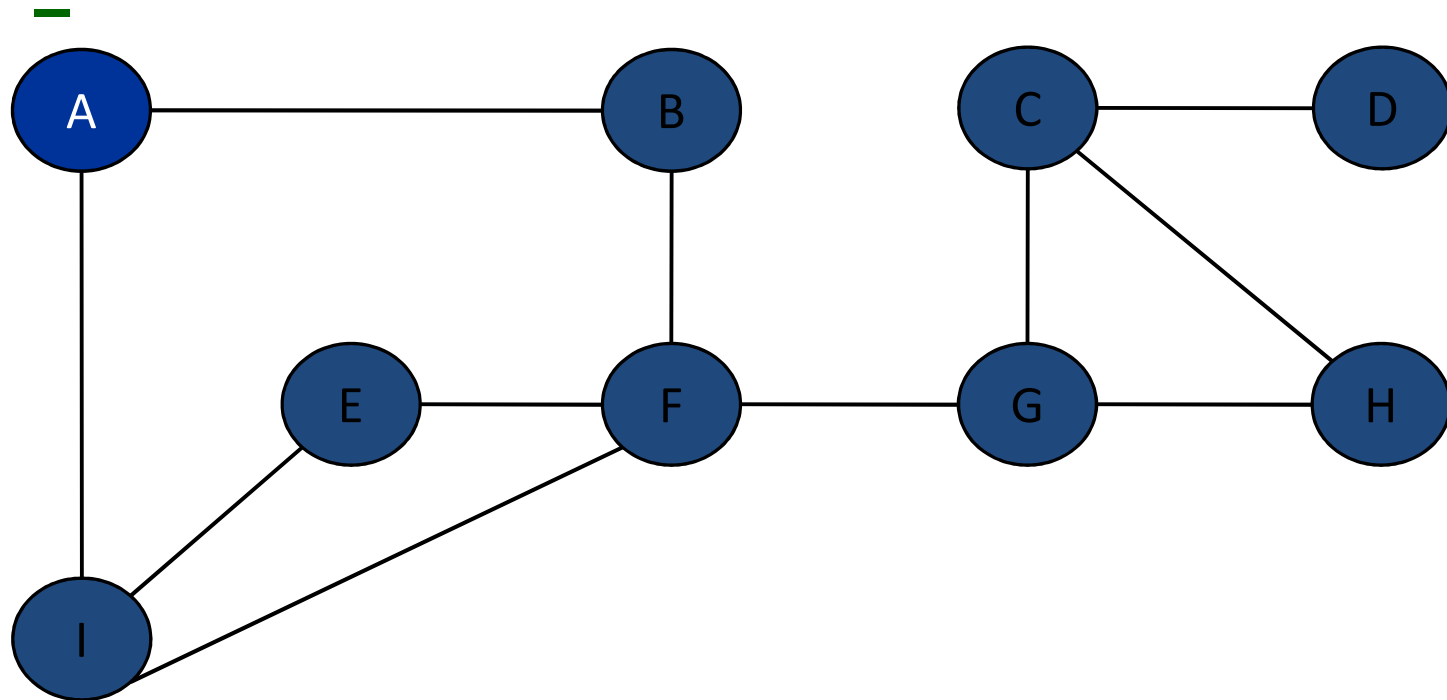
No inicial para a Fila

front

**A**

FIFO

# Breadth First Search



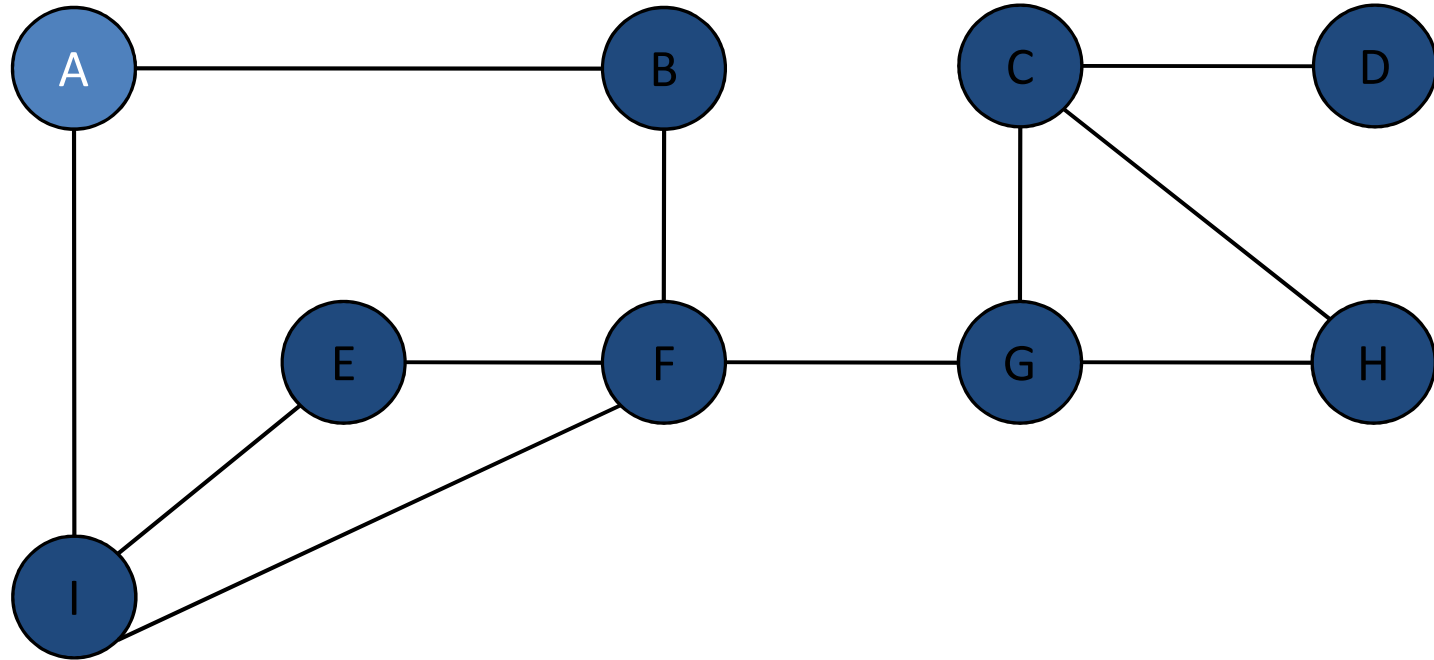
Próximo no na filha

front

**A**

FIFO

# Breadth First Search

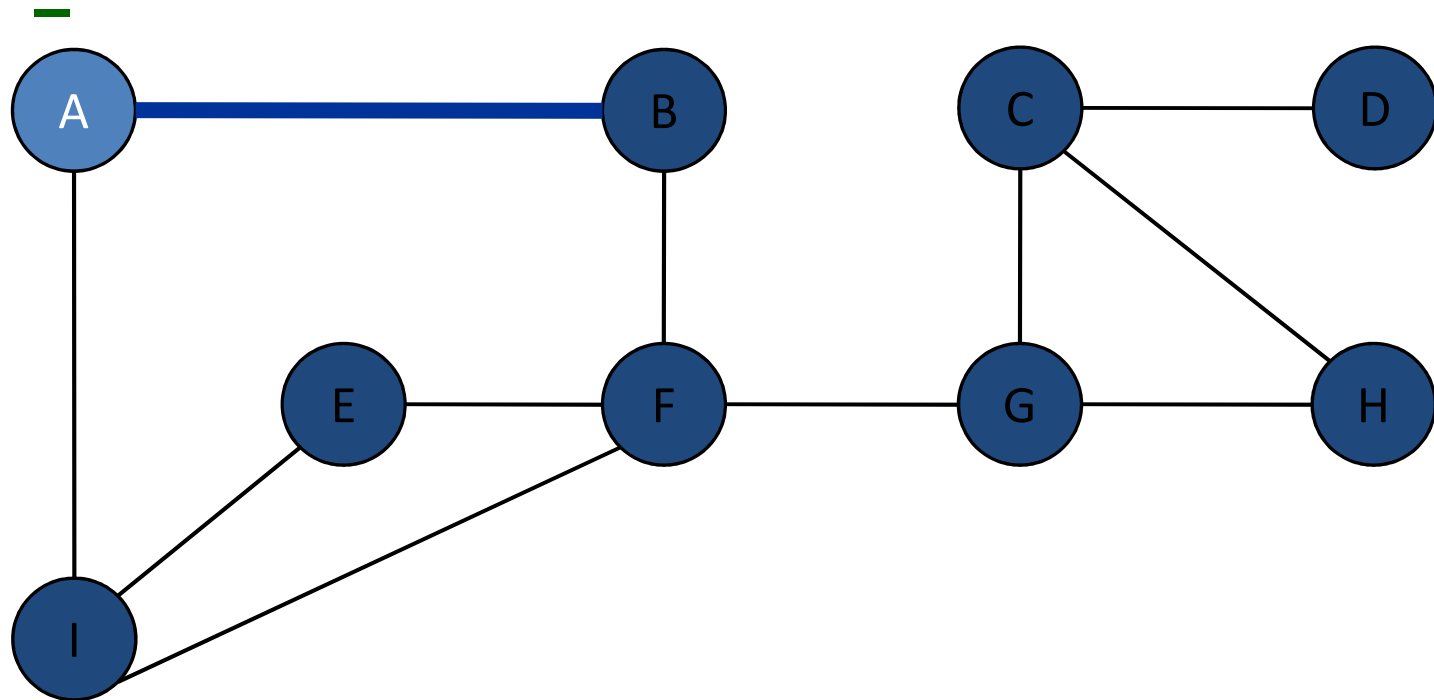


Visita vizinhos de A

front

FIFO

# Breadth First Search

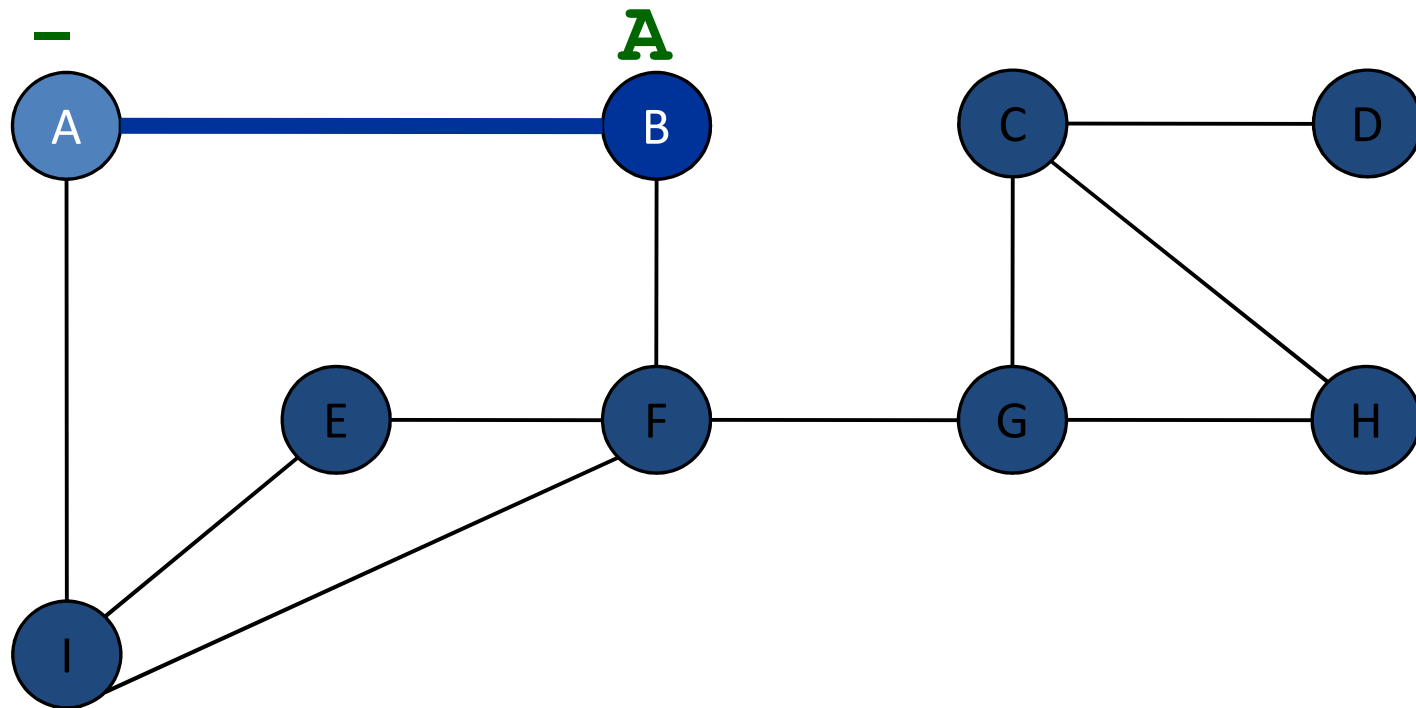


Visita vizinhos de A

front

FIFO

# Breadth First Search



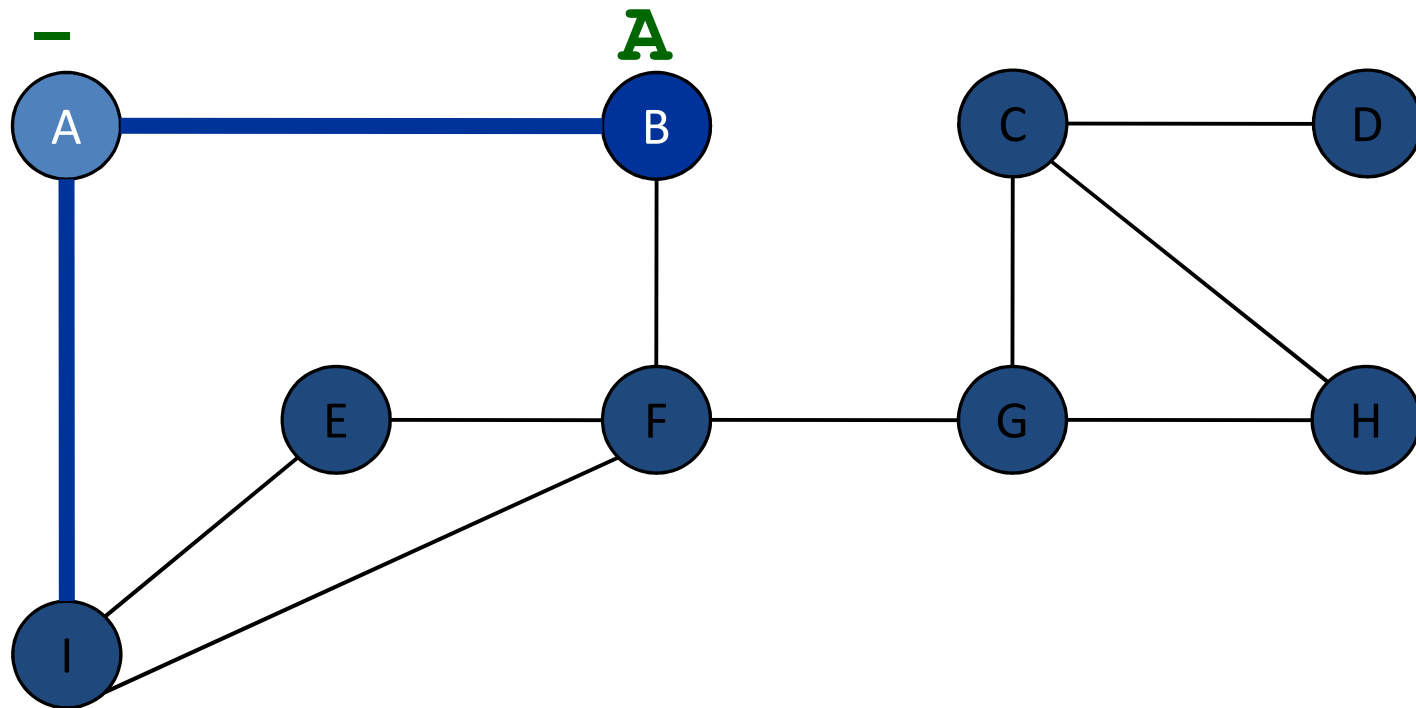
B Descoberto

front

B

FIFO

# Breadth First Search



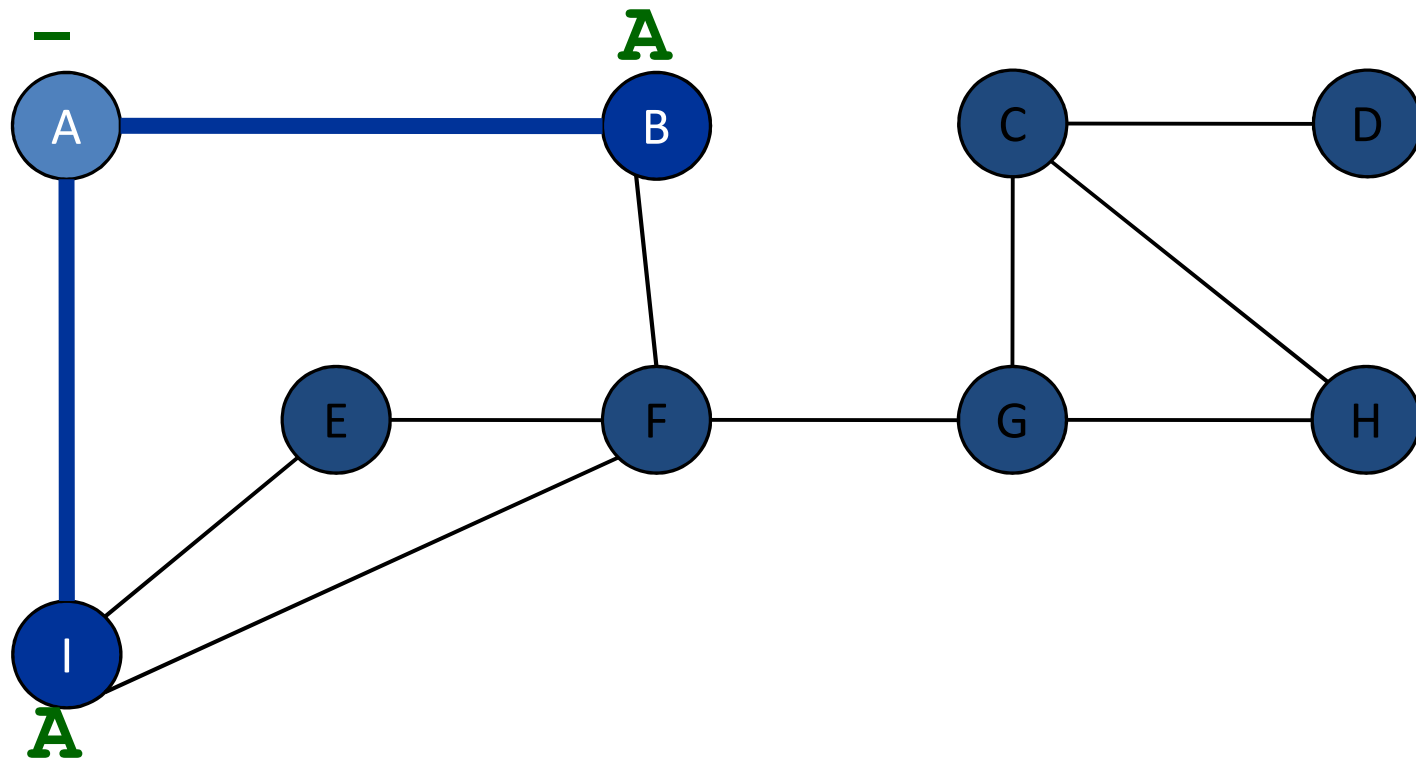
Visita vizinhos de A

front

B

FIFO

# Breadth First Search



I Descoberto

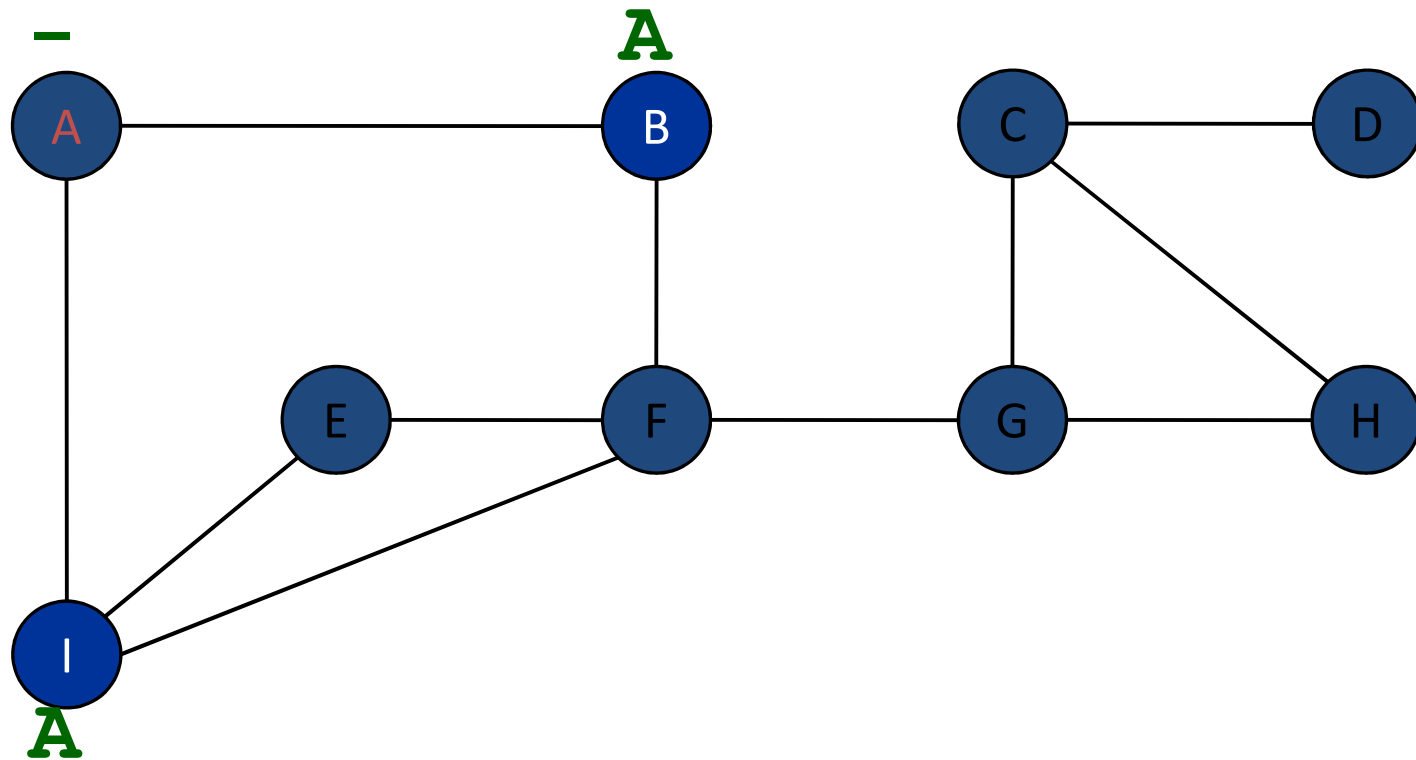
front

B I

FIFO



# Breadth First Search



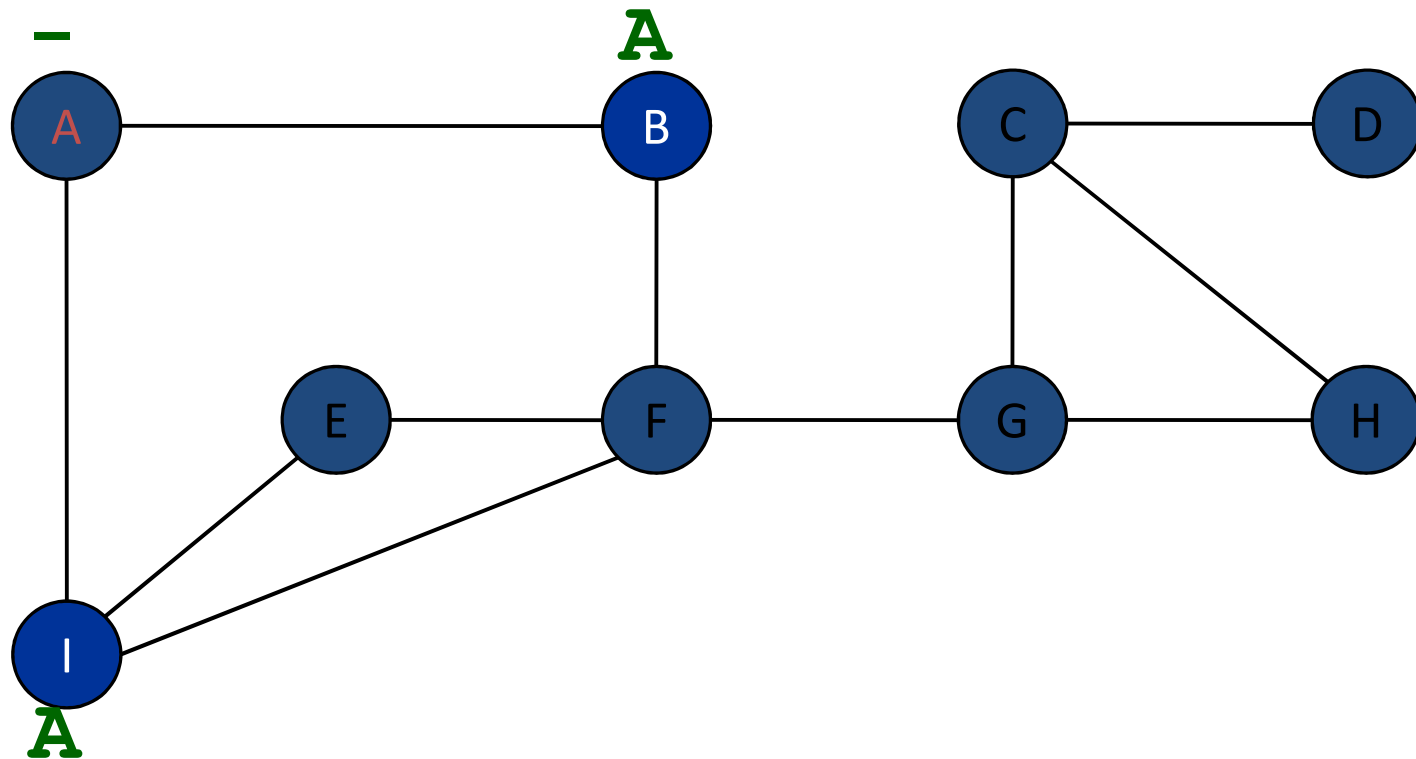
Fim com A

front

B I

FIFO

# Breadth First Search



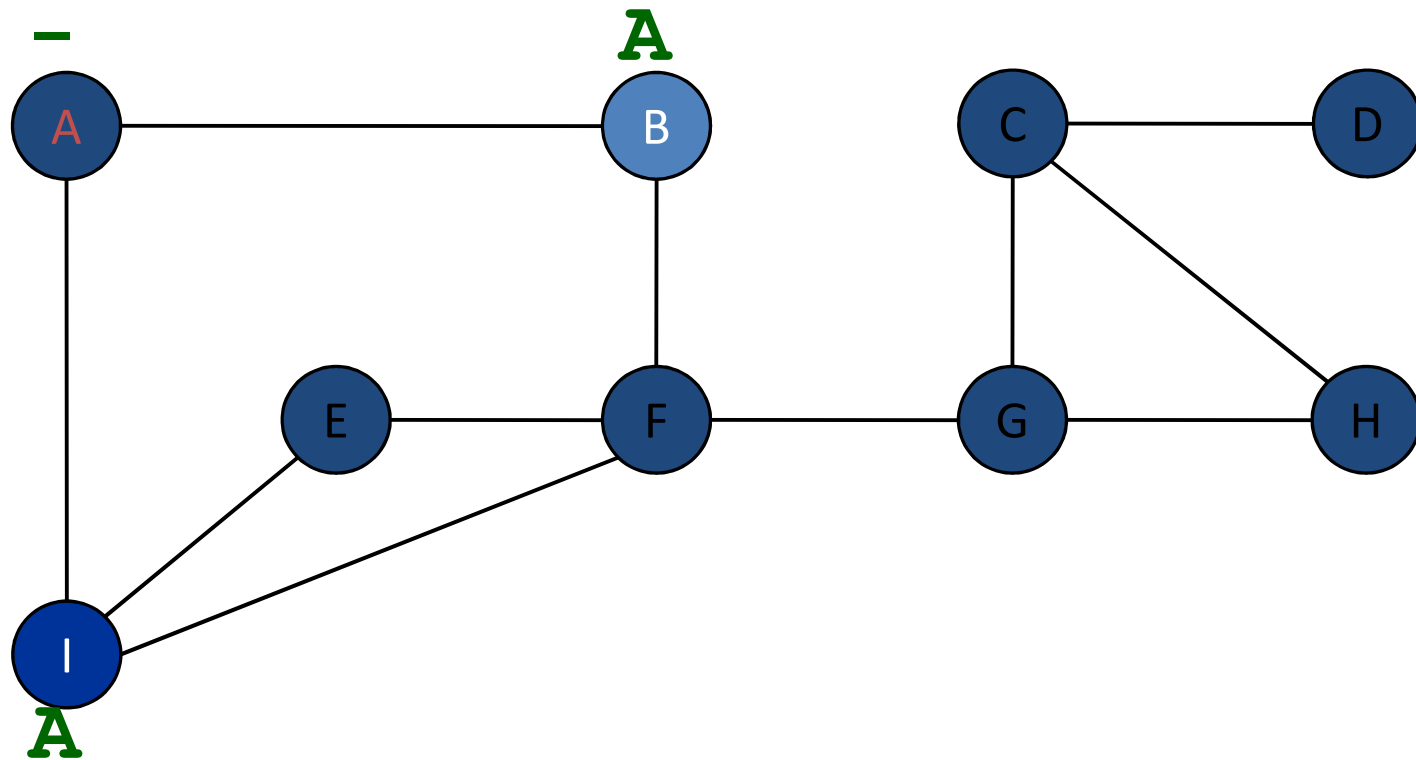
Próximo no na filha

front

B I

FIFO

# Breadth First Search



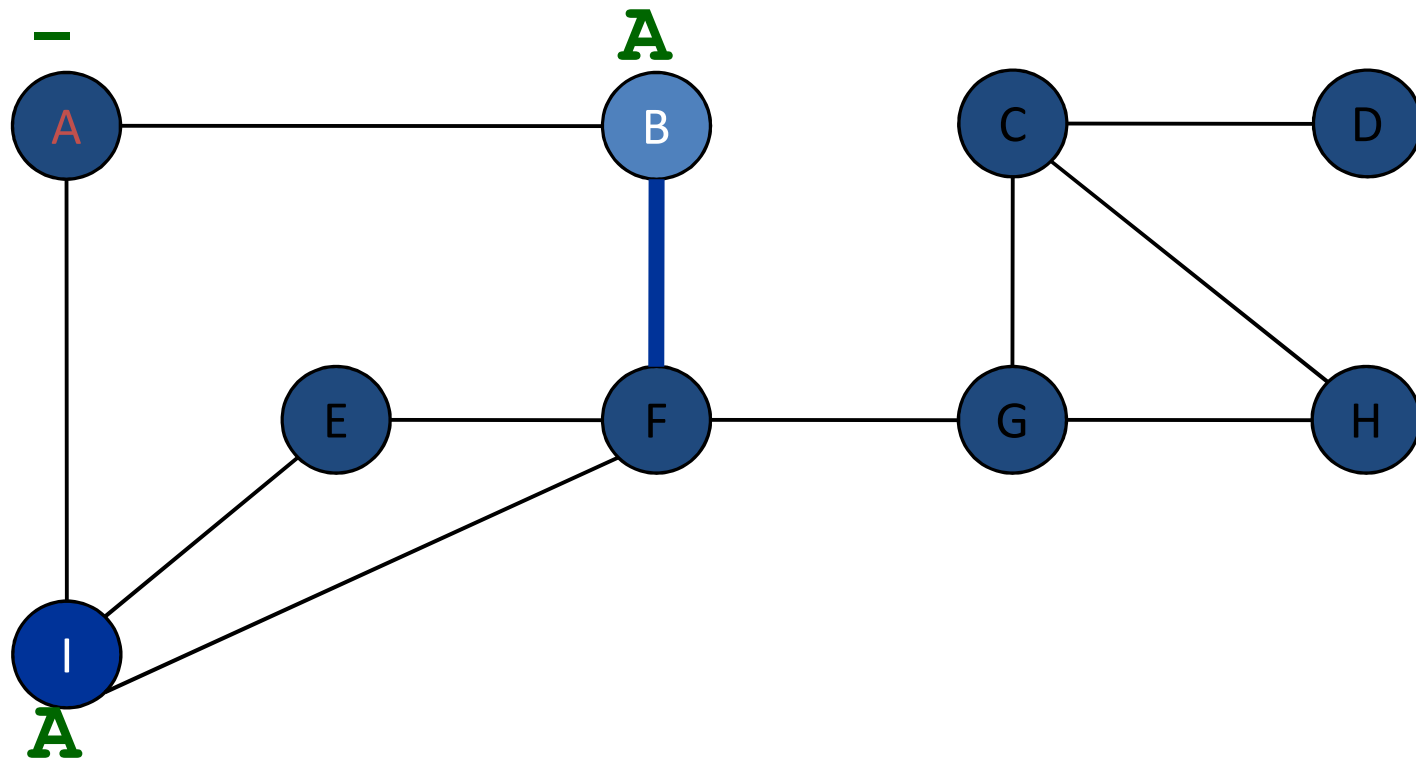
Visita vizinhos de B

front

I

FIFO

# Breadth First Search



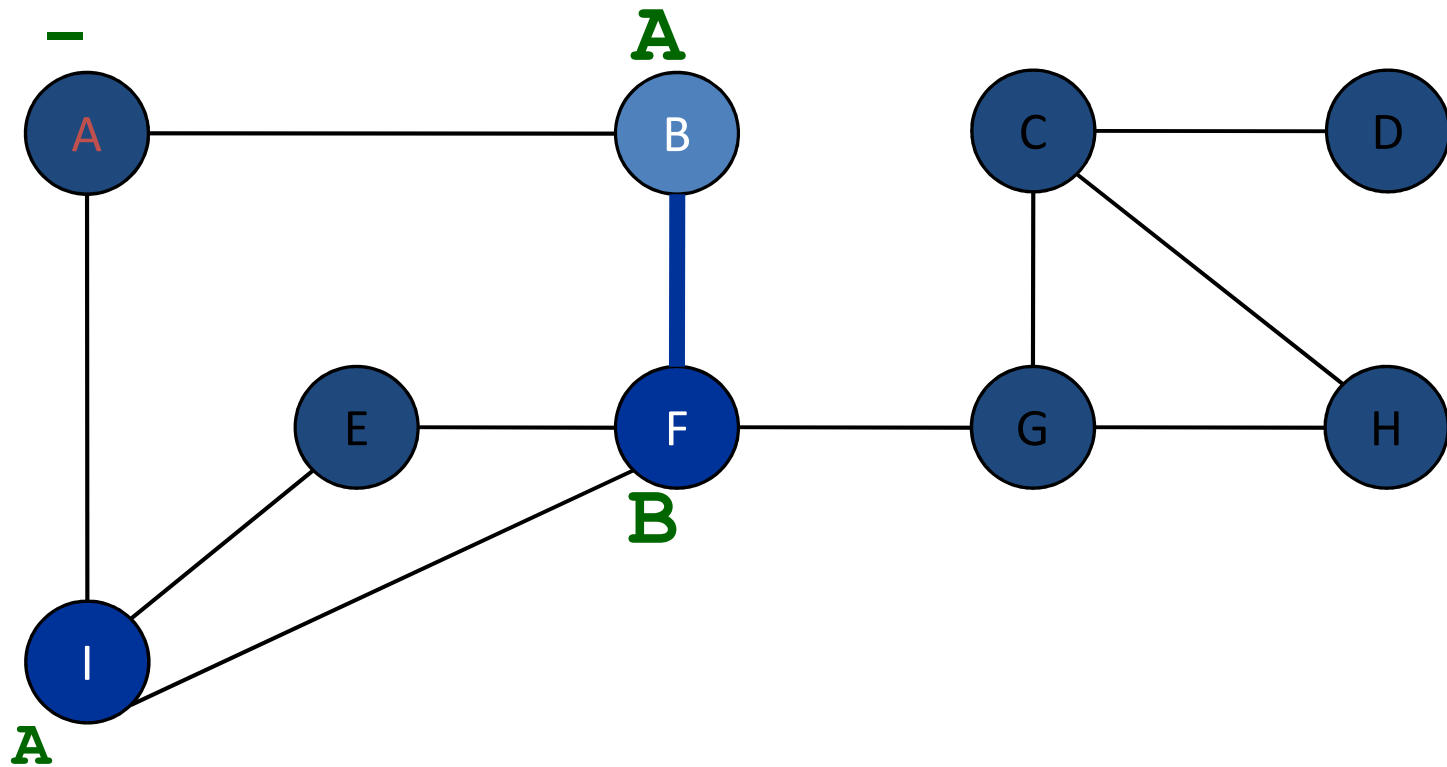
Visita vizinhos de B

front

I

FIFO

# Breadth First Search



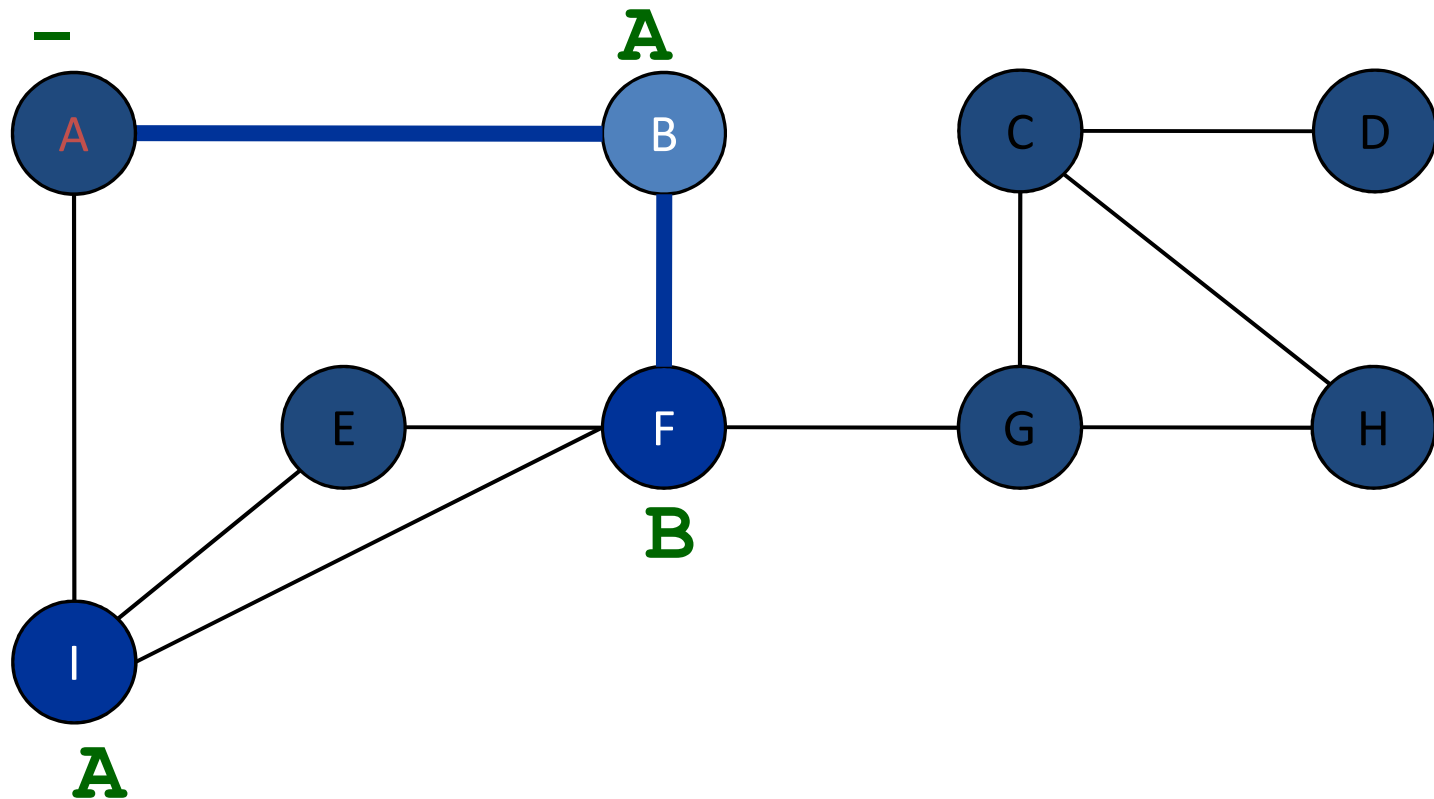
F Descoberto

front

I F

FIFO

# Breadth First Search



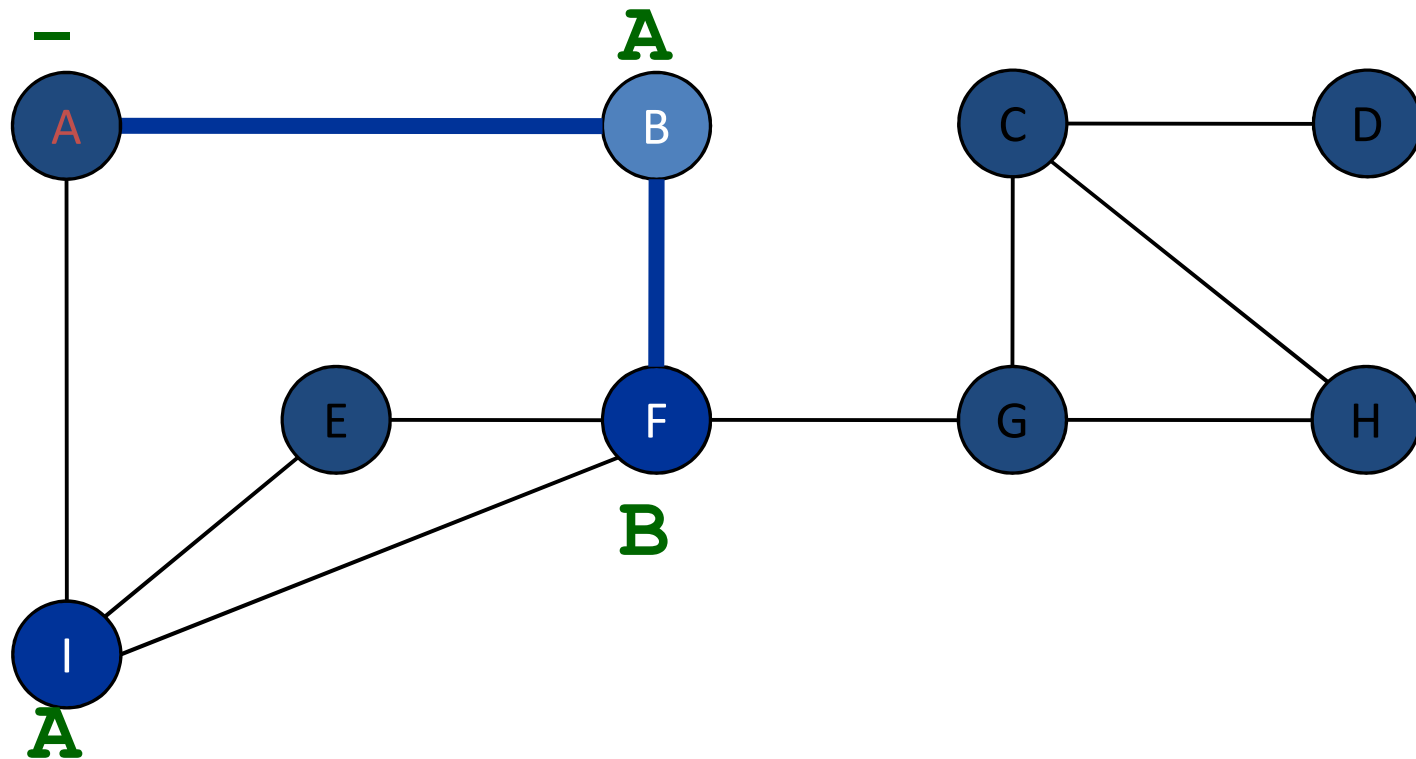
Visita vizinhos de B

front

I F

FIFO

# Breadth First Search



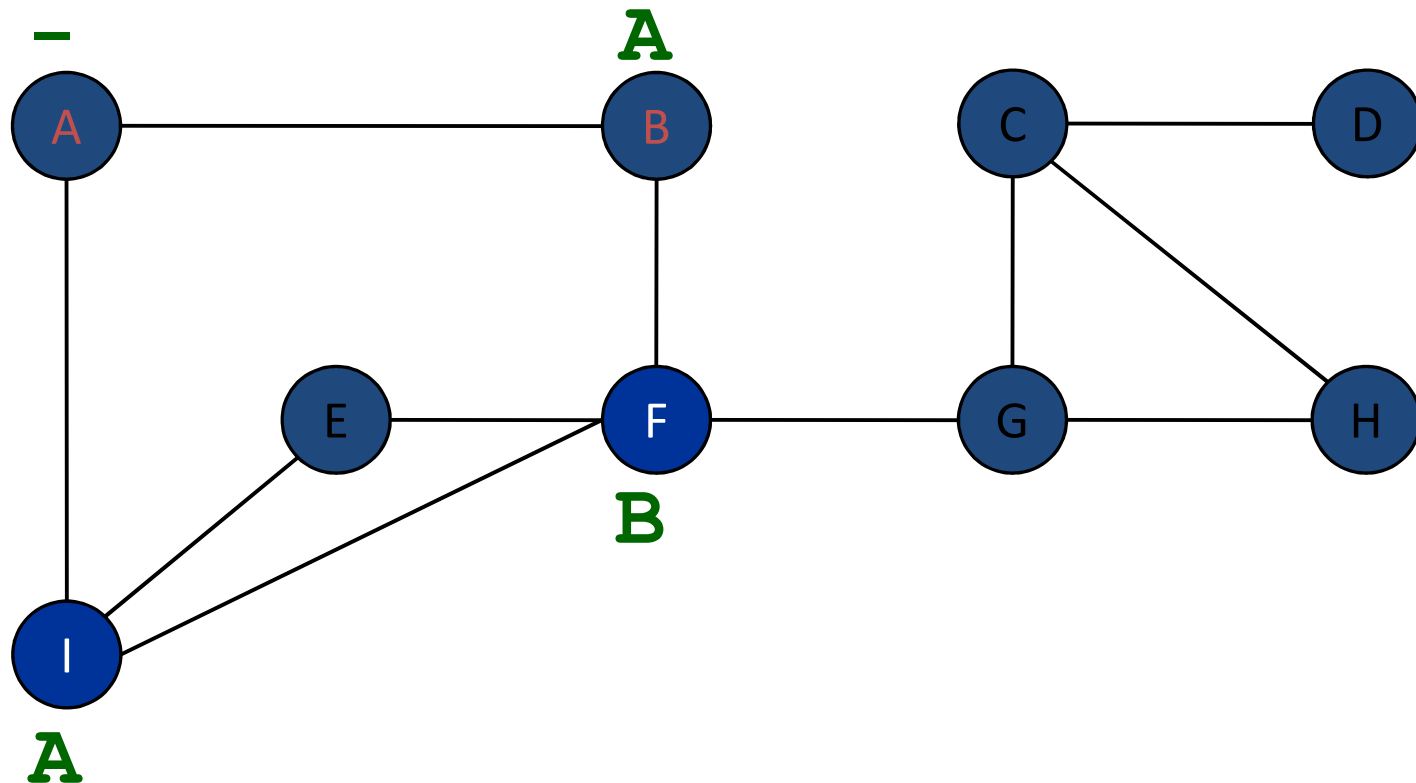
A já foi descoberto

front

I F

FIFO

# Breadth First Search



Fim com B

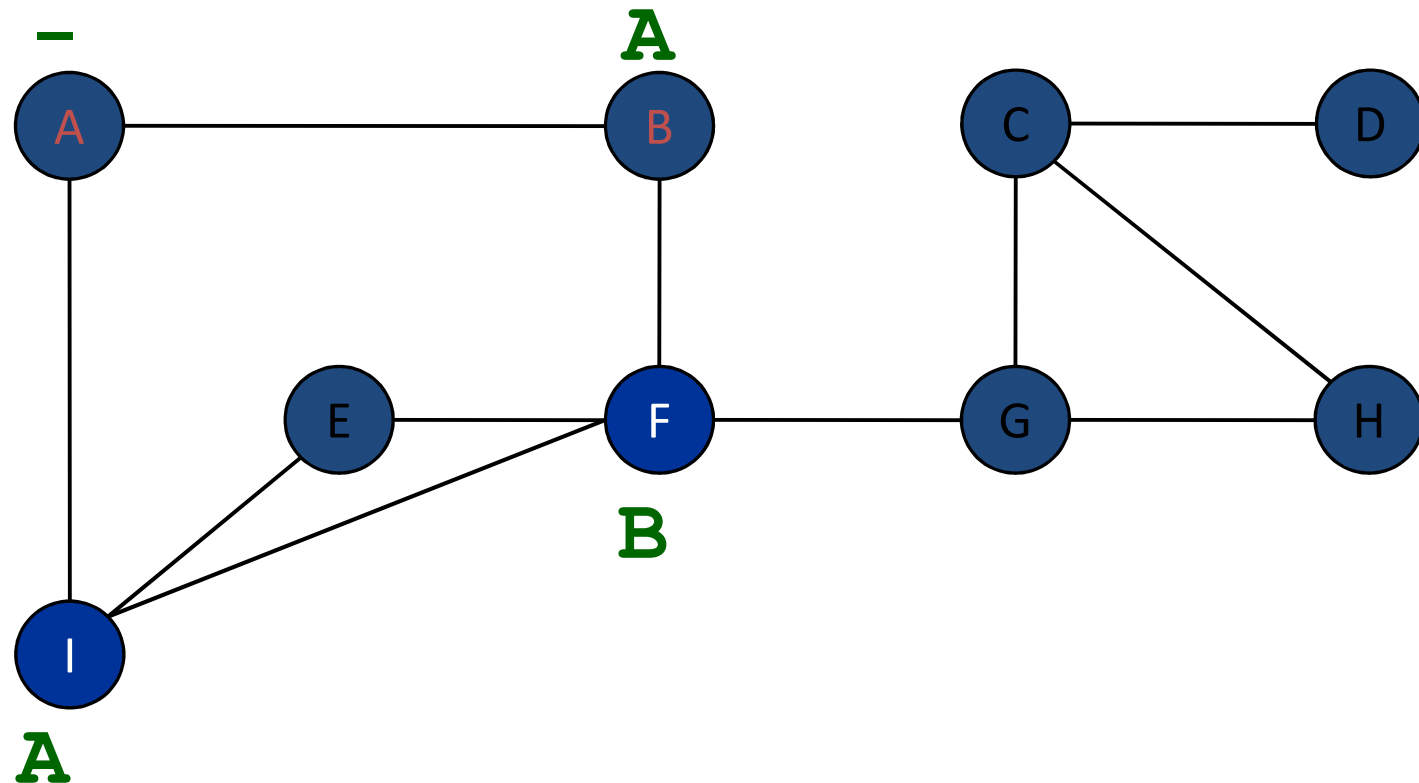
front

I F

FIFO



# Breadth First Search



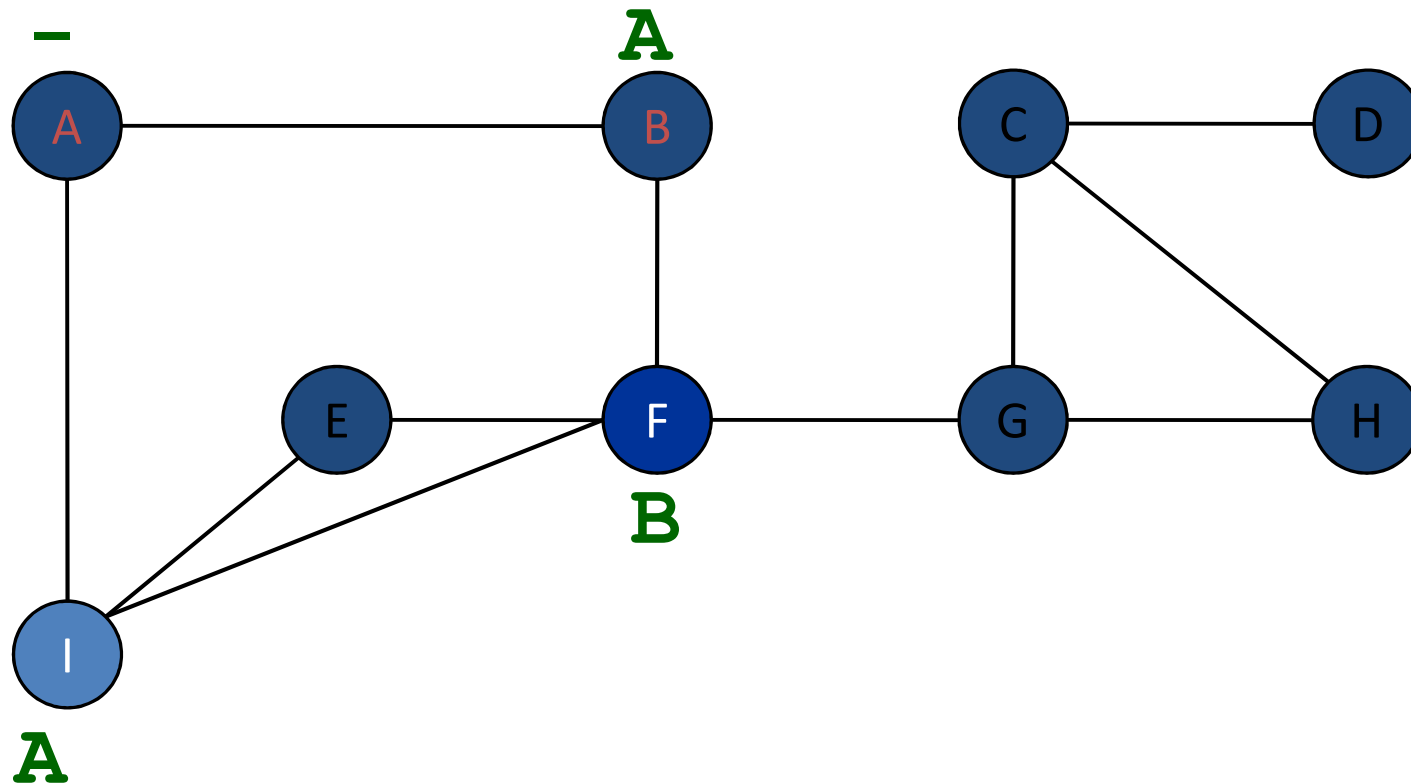
Próximo no na filha

front

I F

FIFO

# Breadth First Search



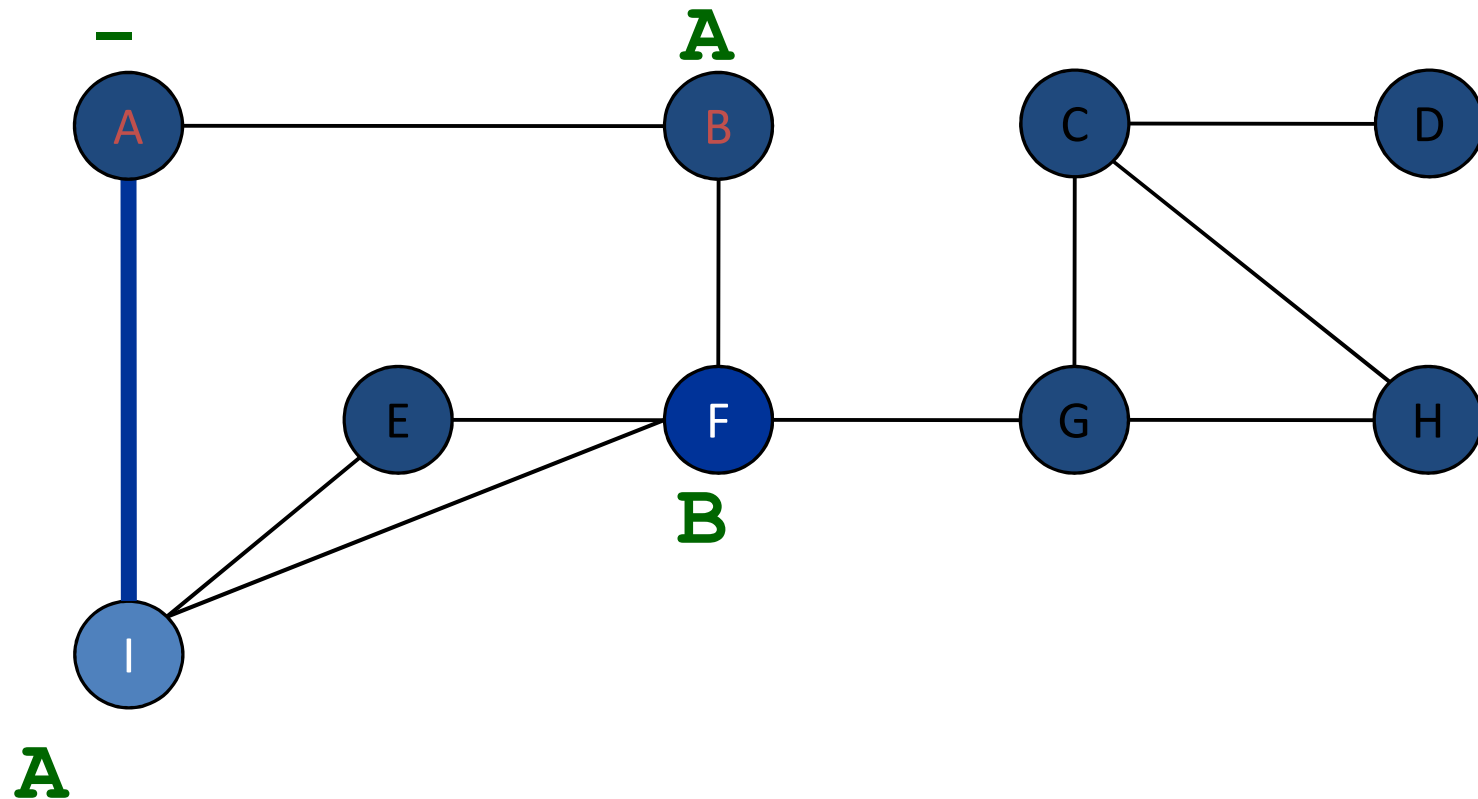
Visita vizinhos de I

front

**F**

FIFO

# Breadth First Search



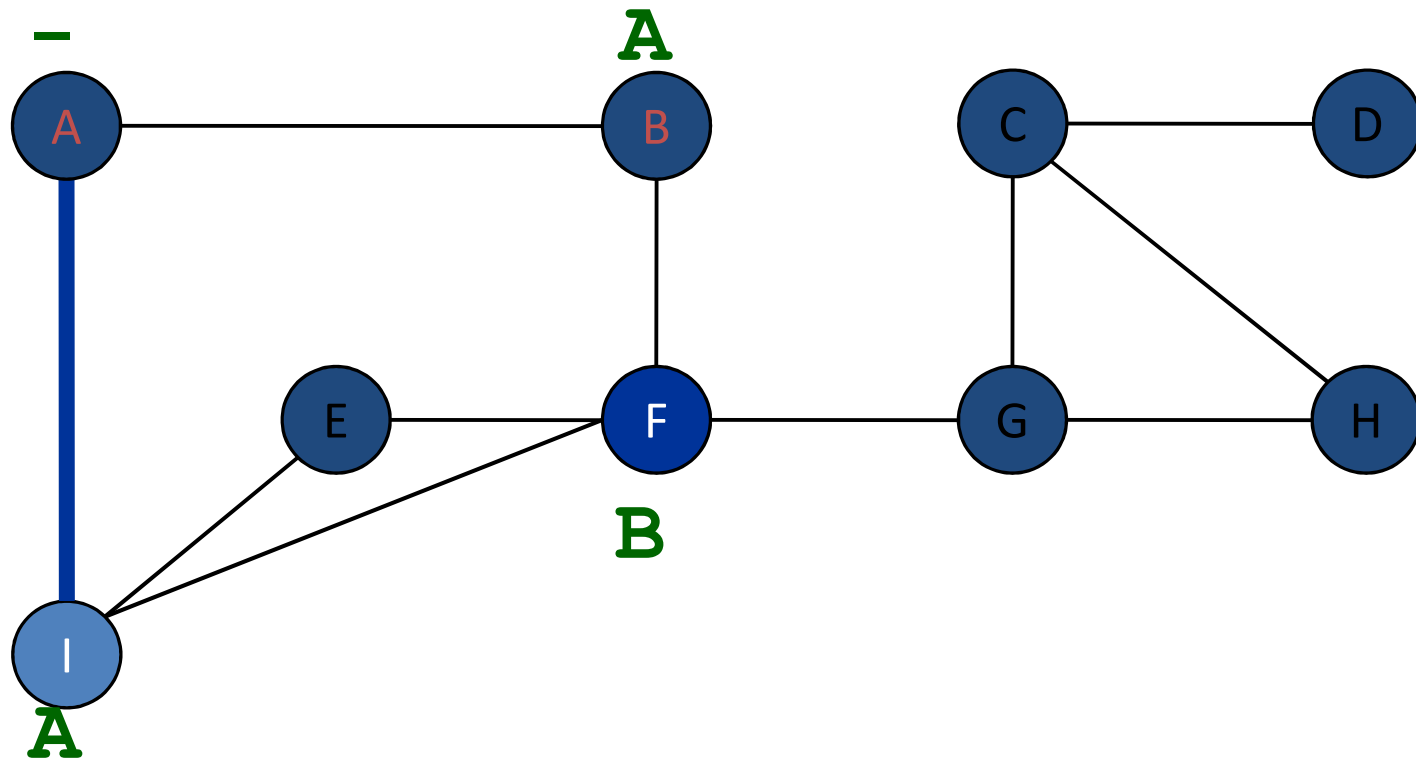
Visita vizinhos de I

front

F

FIFO

# Breadth First Search



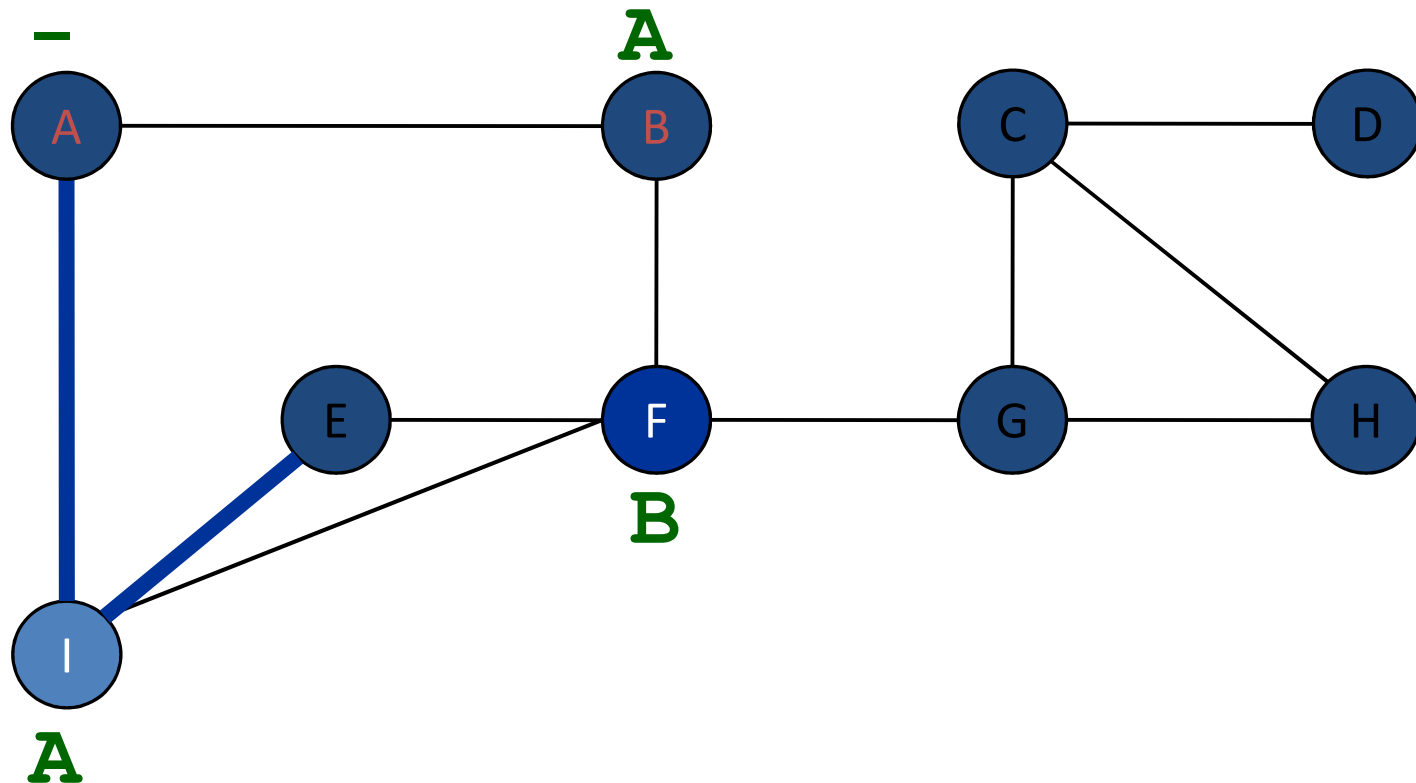
A já foi descoberto

front

**F**

FIFO

# Breadth First Search



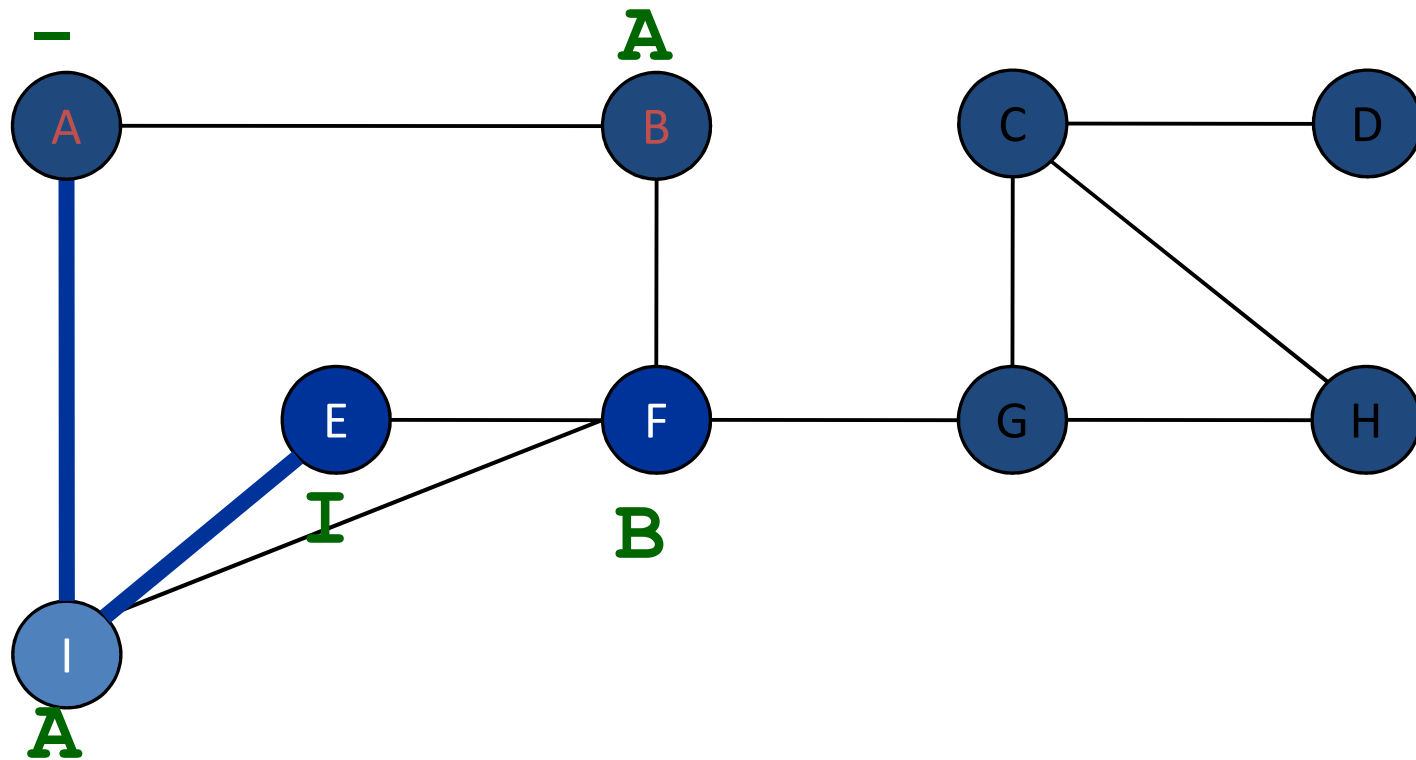
Visita vizinhos de I

front

F

FIFO

# Breadth First Search



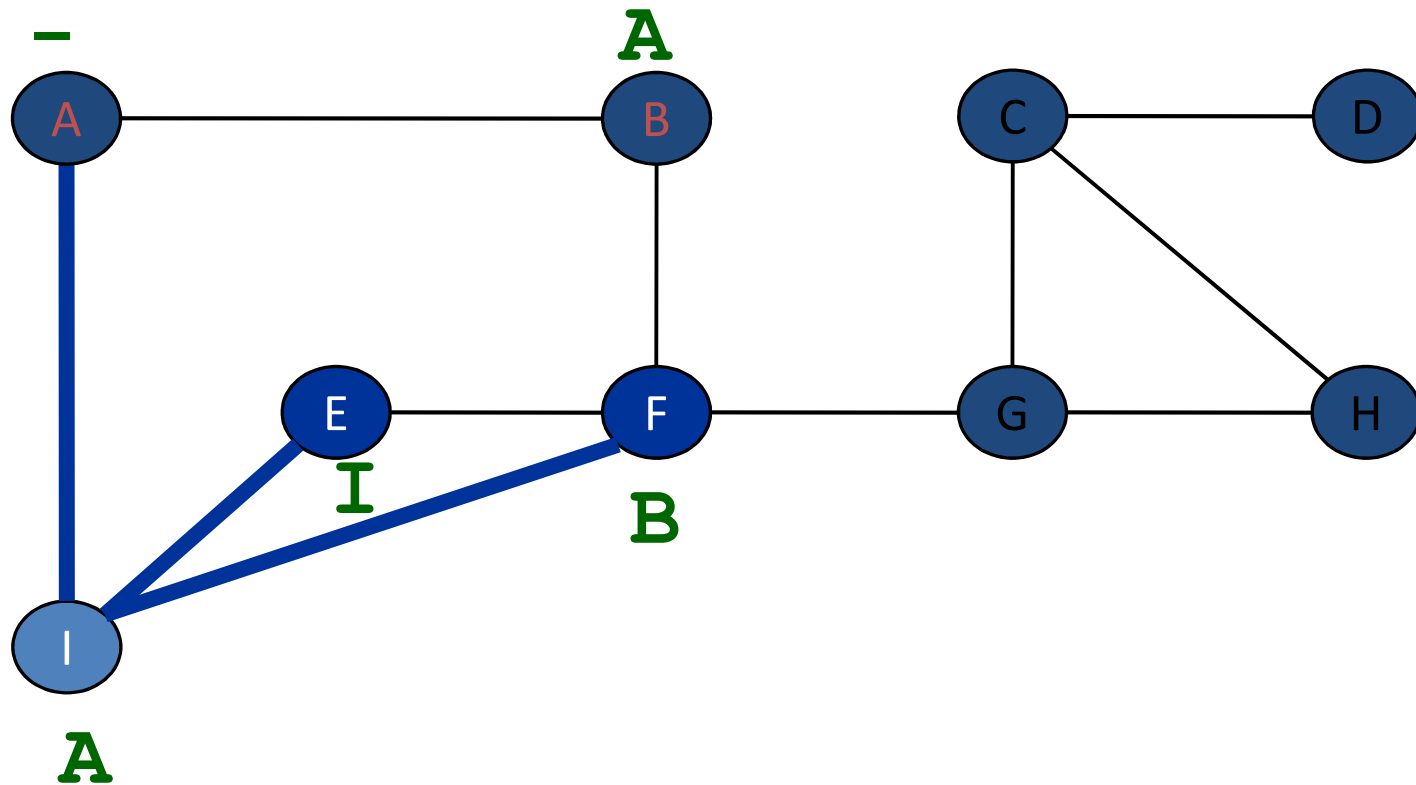
E Descoberto

front

F E

FIFO

# Breadth First Search



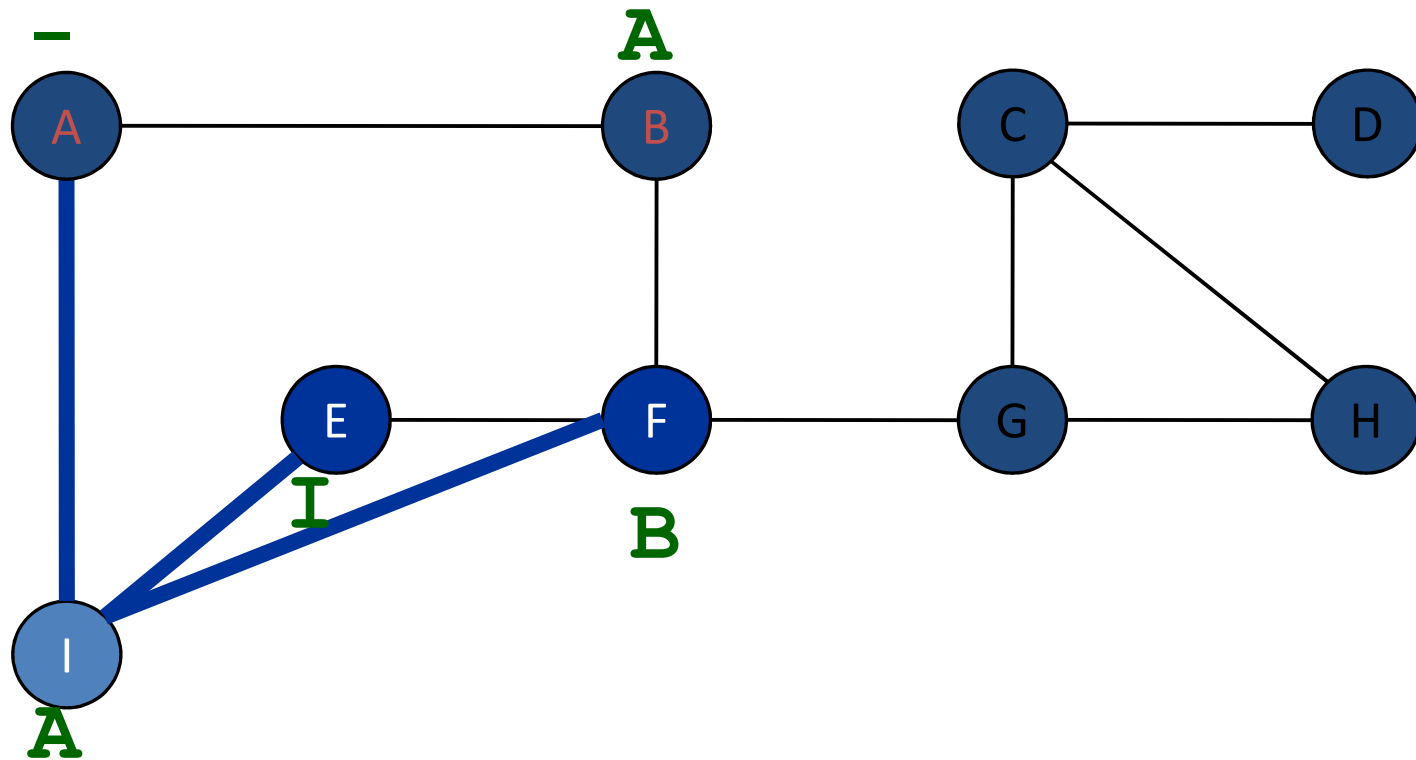
## Visita vizinhos de I

front

F E

## FIFO

# Breadth First Search



F já foi descoberto

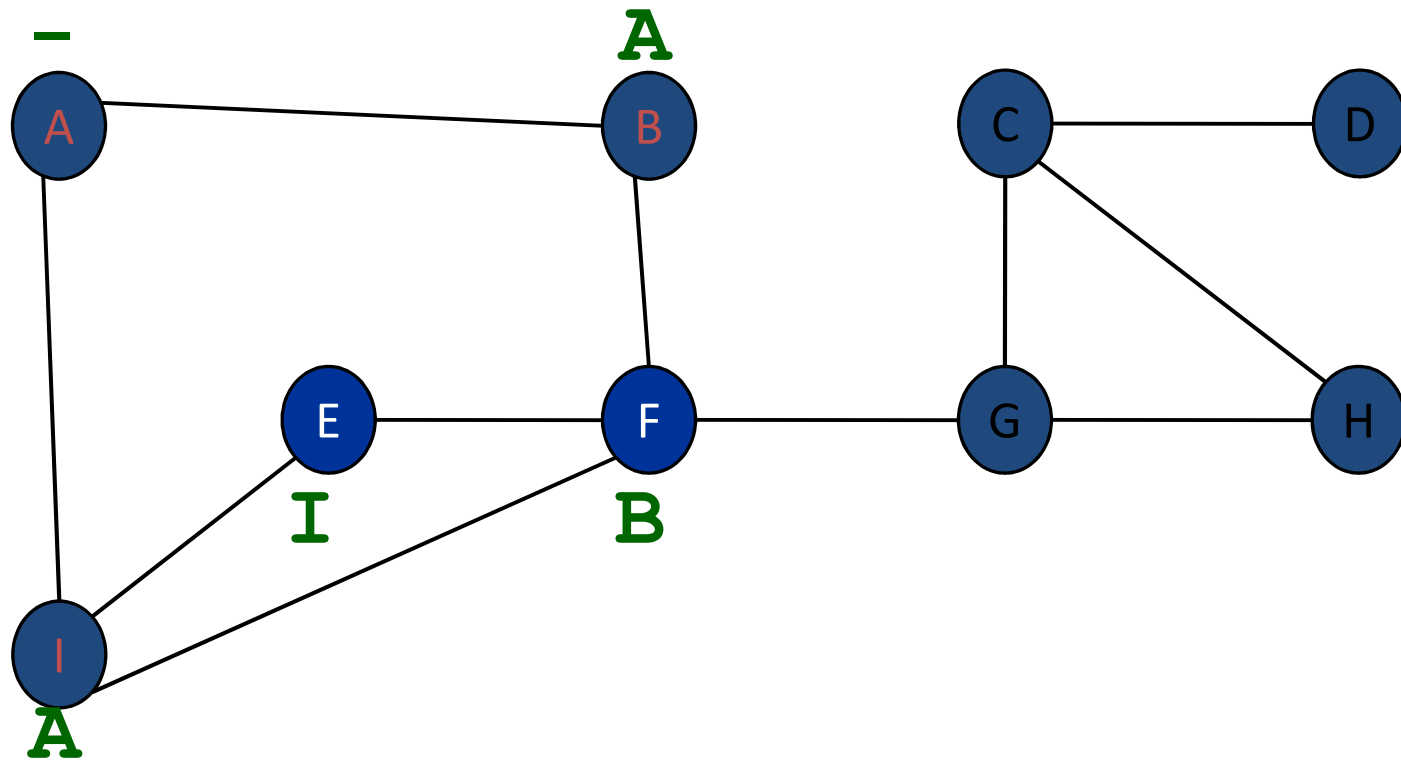
front

F E

## FIFO



# Breadth First Search



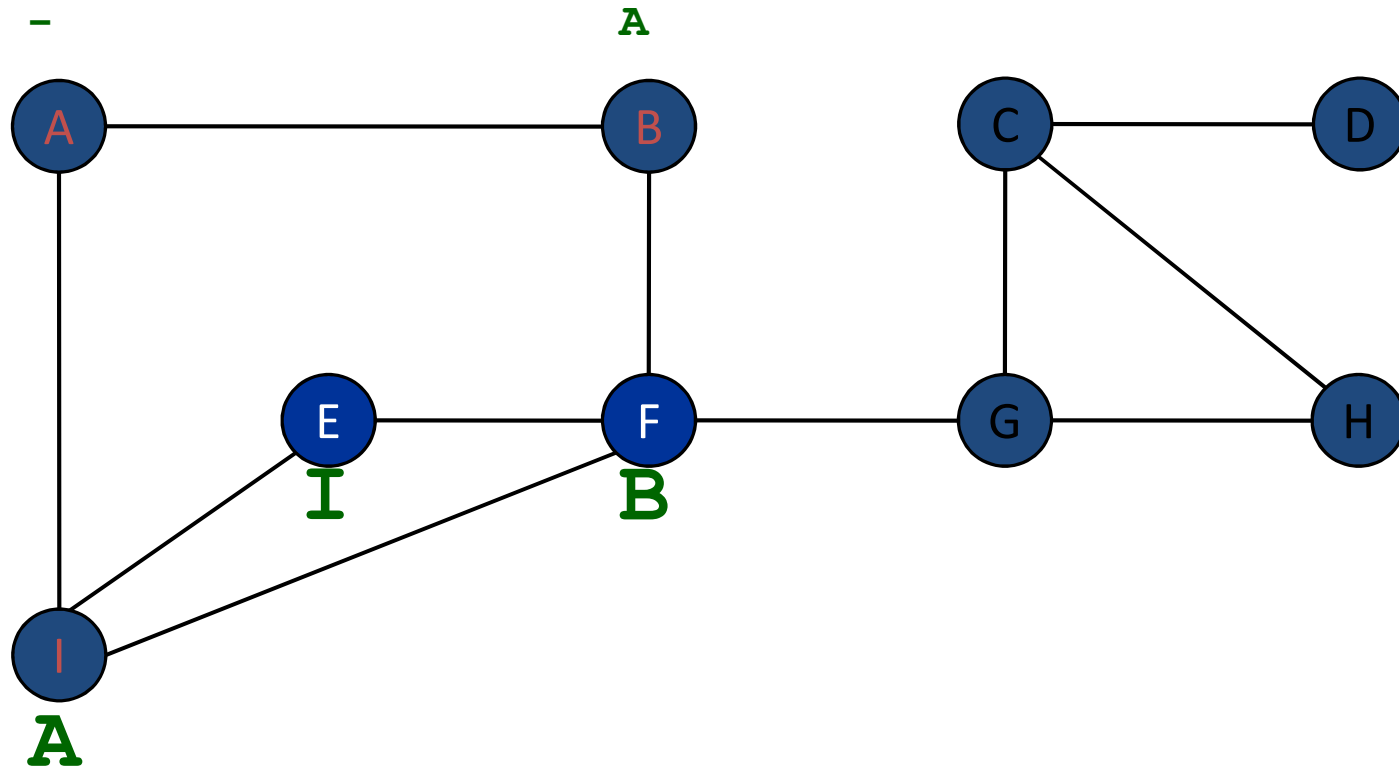
I Fim

front

F E

FIFO

# Breadth First Search



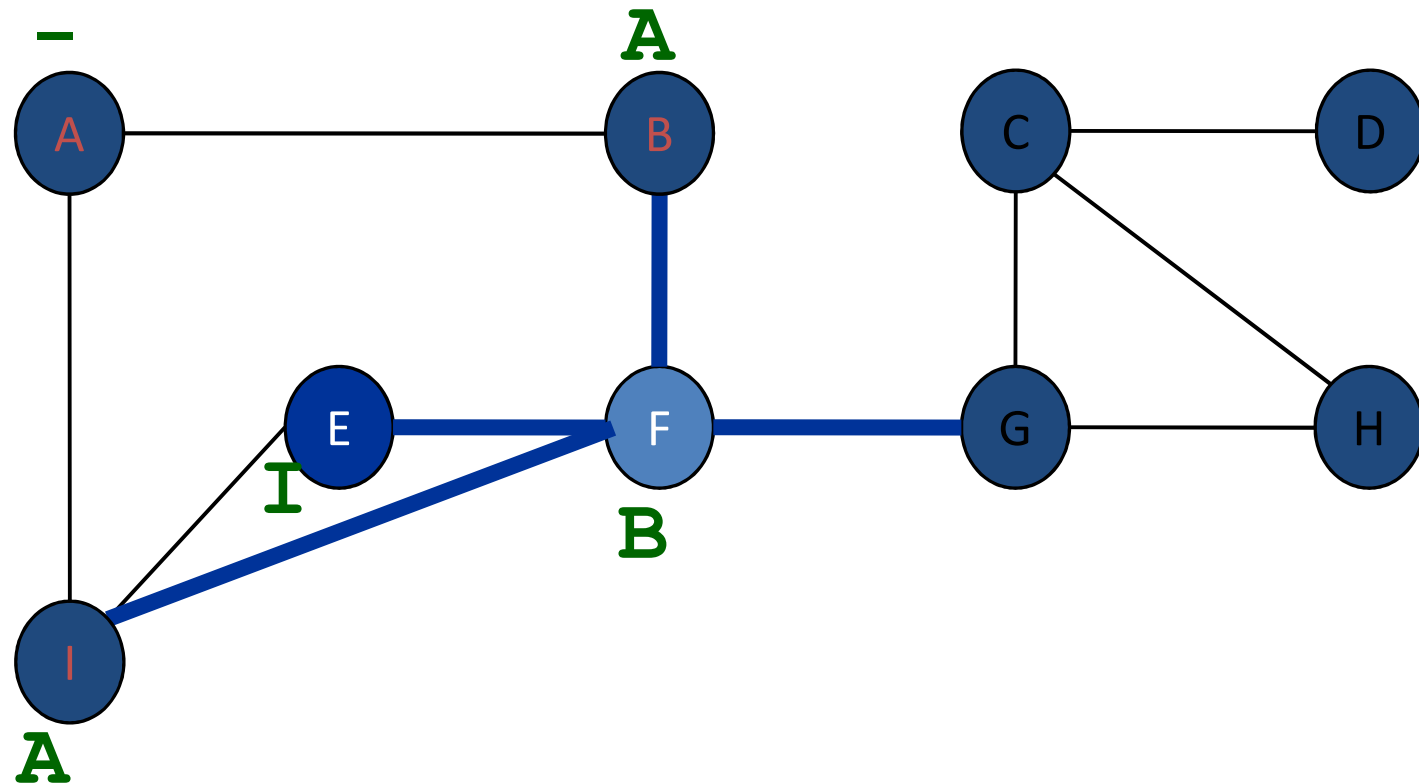
Próximo no na filha

front

**F E**

FIFO

# Breadth First Search



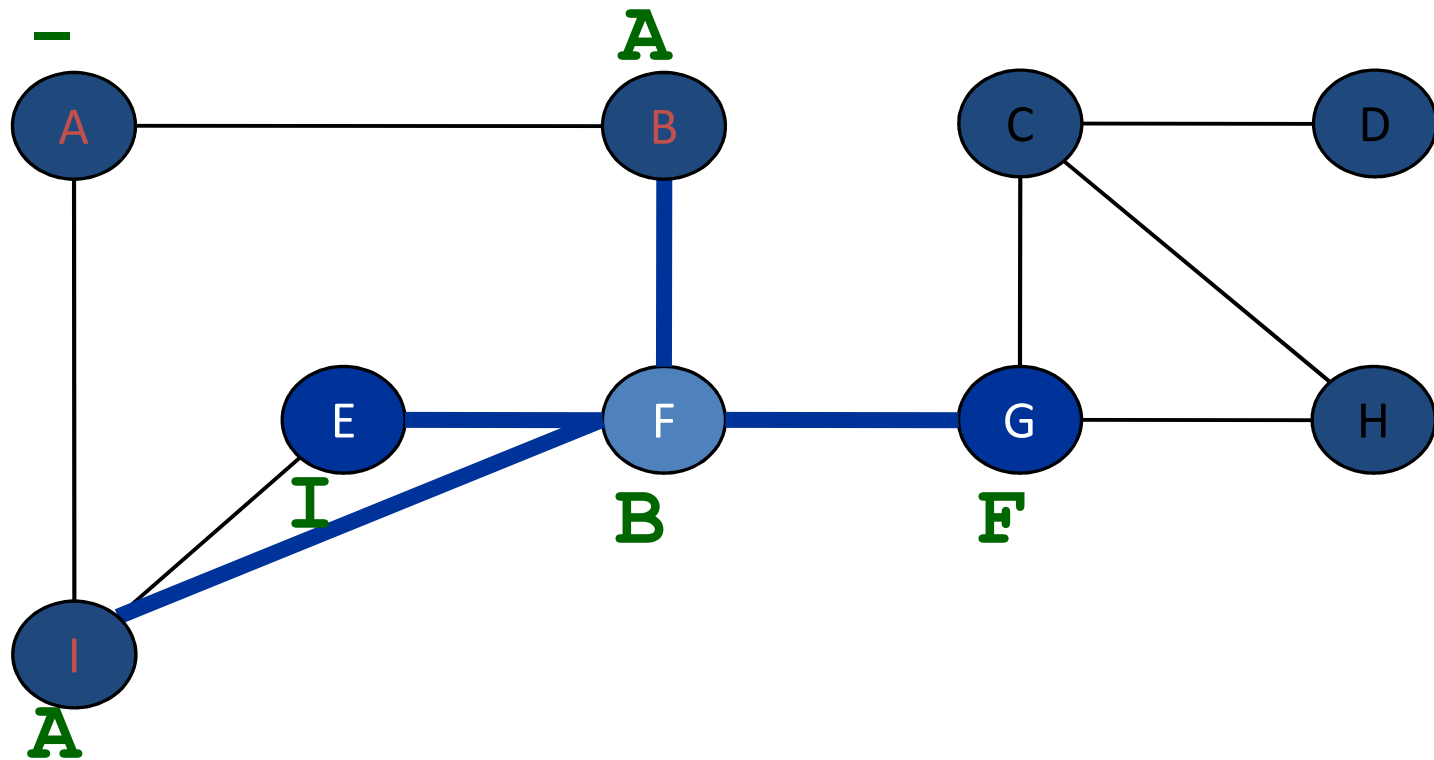
Visita vizinhos de F

front

E

FIFO

# Breadth First Search



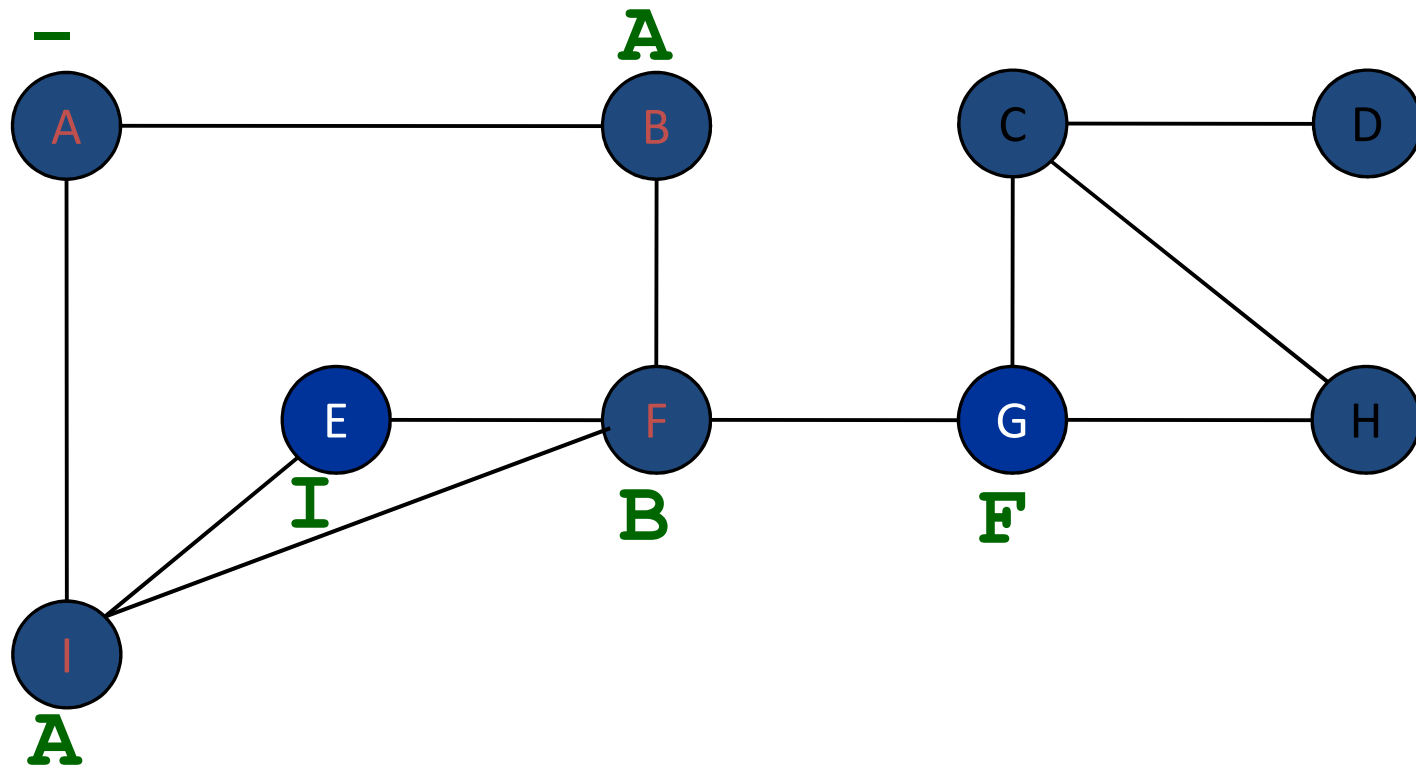
G Descoberto

front

E G

FIFO

# Breadth First Search



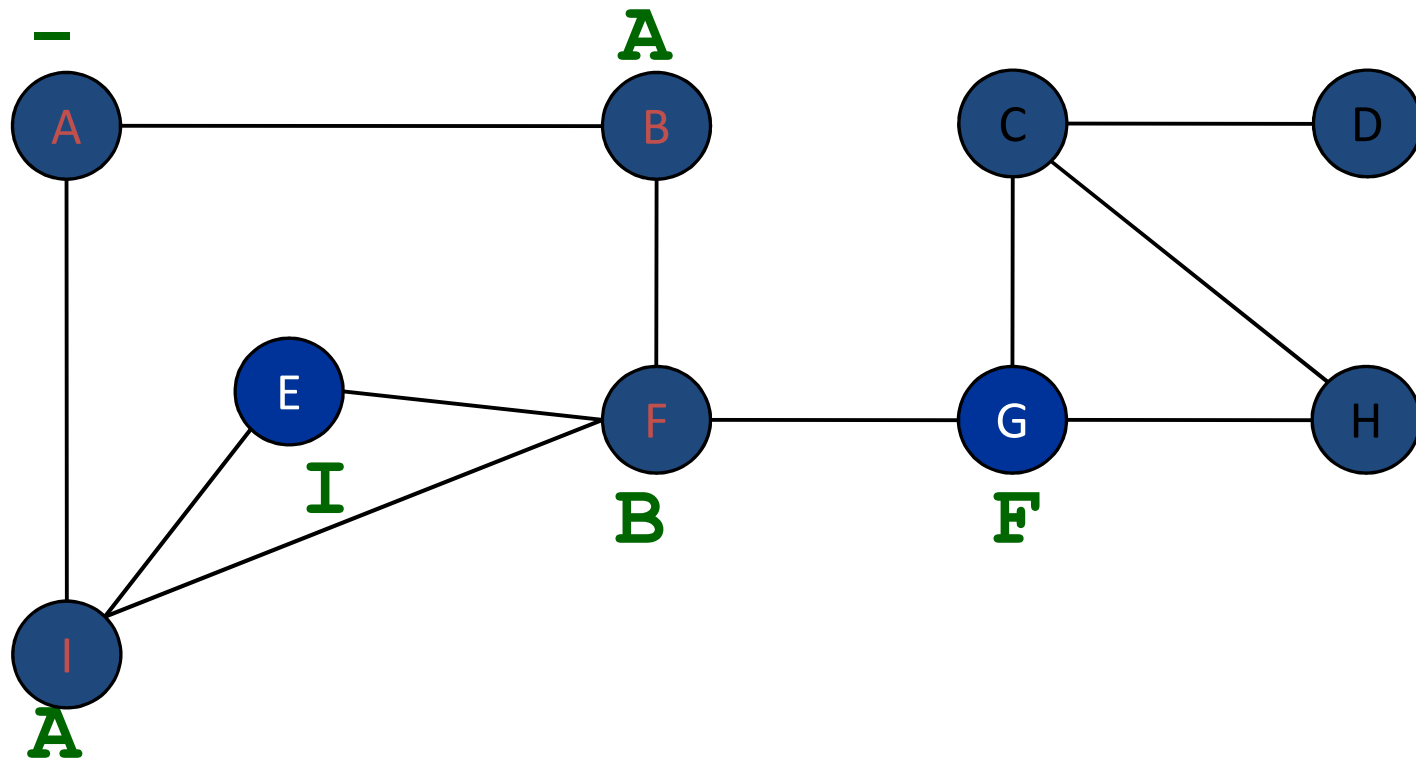
F Fim

front

E G

FIFO

# Breadth First Search



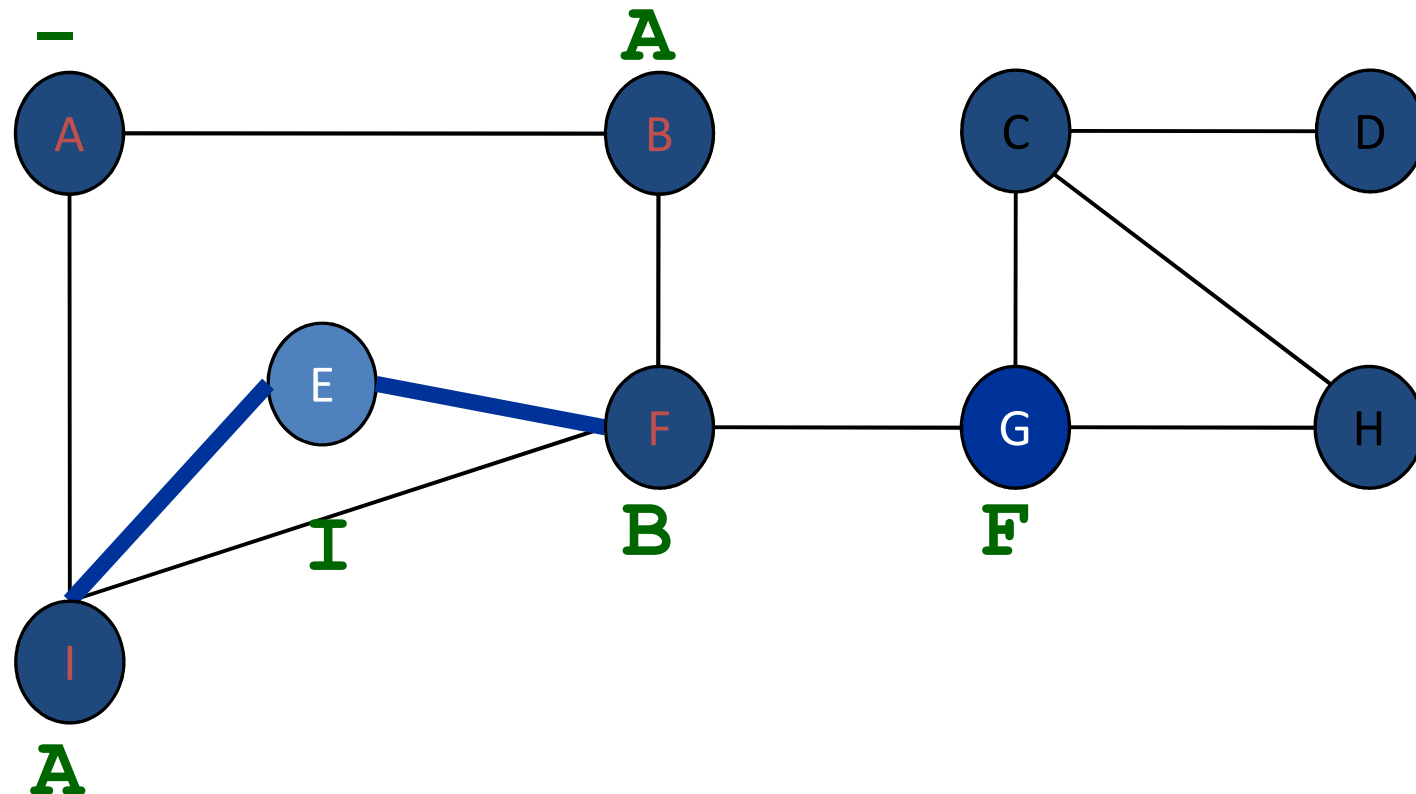
Próximo no na filha

front

E G

FIFO

# Breadth First Search



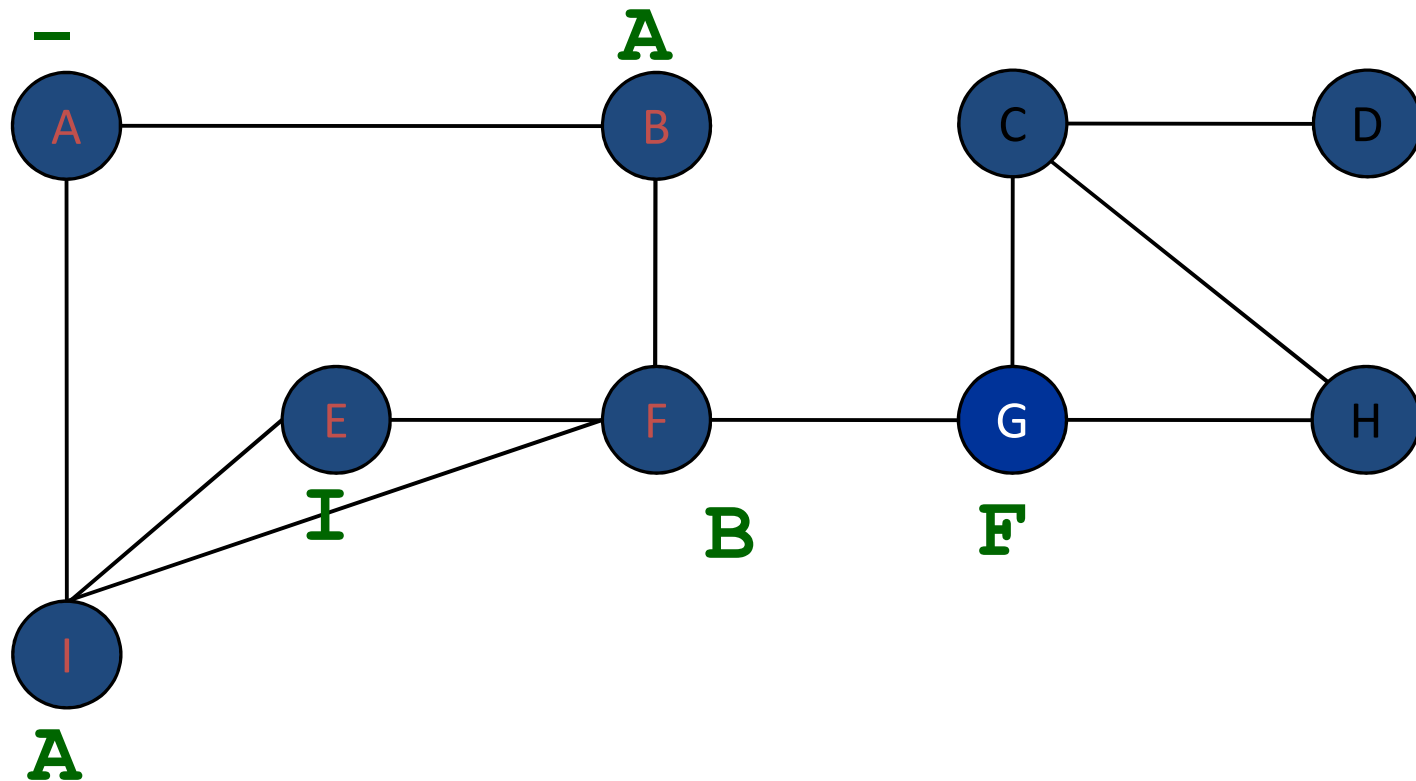
Visita vizinhos de E

front

G

FIFO

# Breadth First Search



E Fim

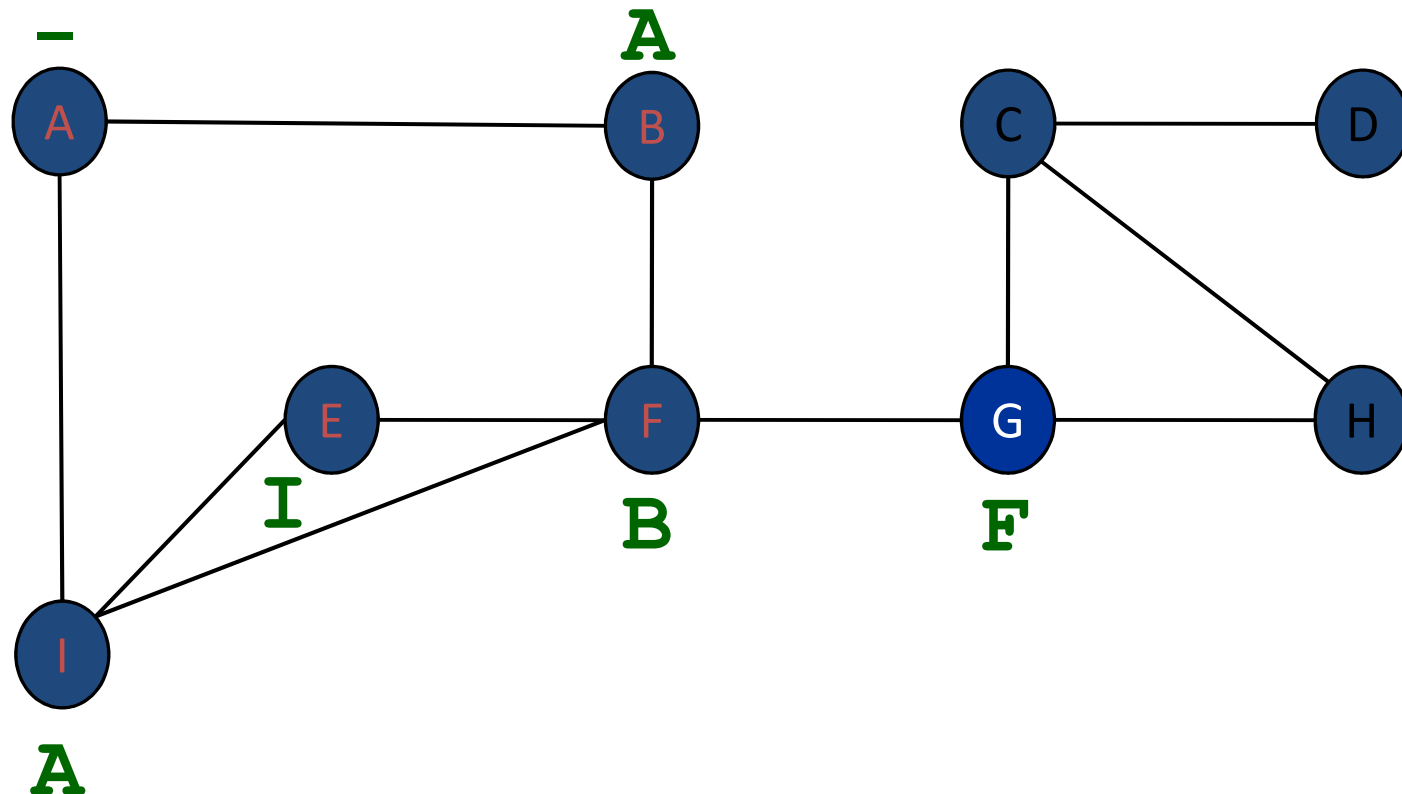
front

G

FIFO



# Breadth First Search



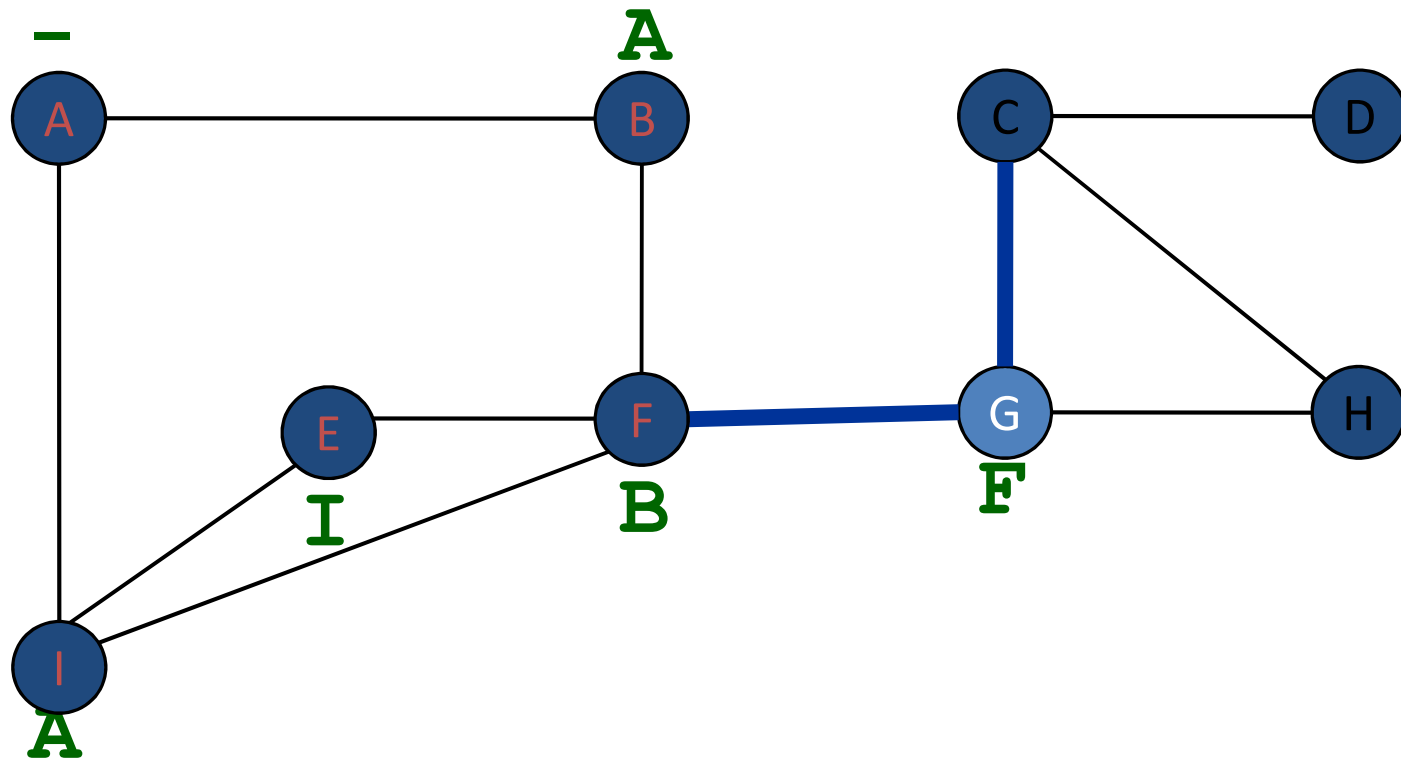
Próximo no na filha

front

G

FIFO

# Breadth First Search

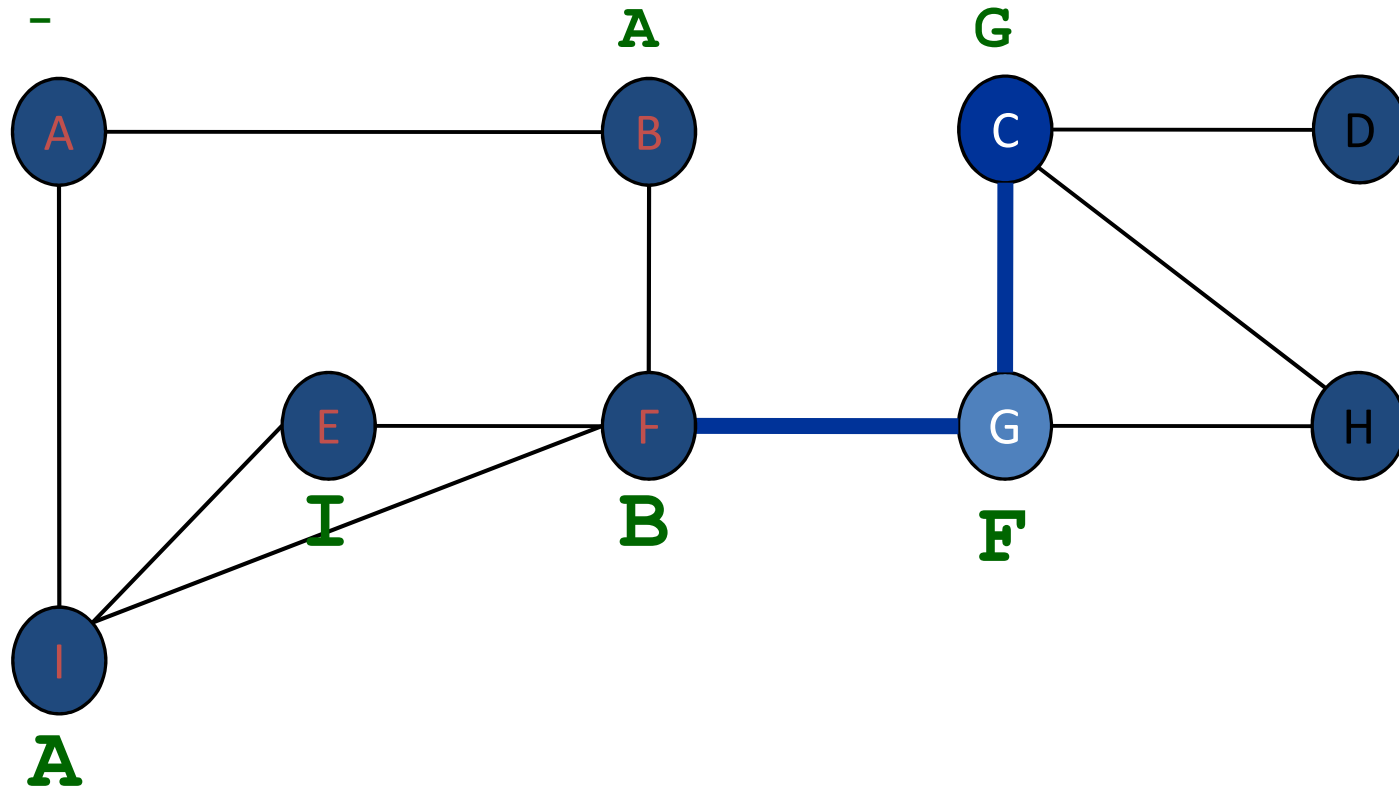


Visita vizinhos de G

front

FIFO

# Breadth First Search



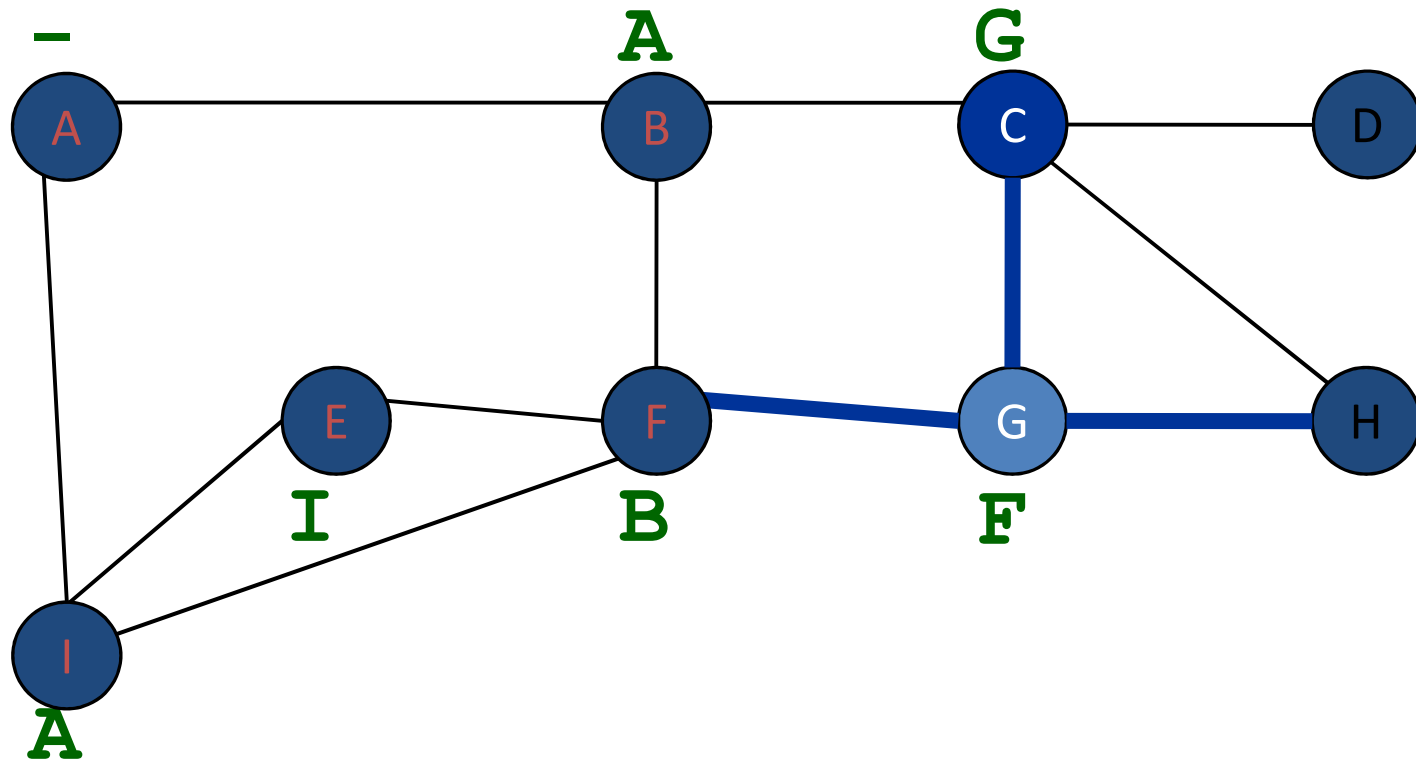
C Descoberto

front

C

FIFO

# Breadth First Search



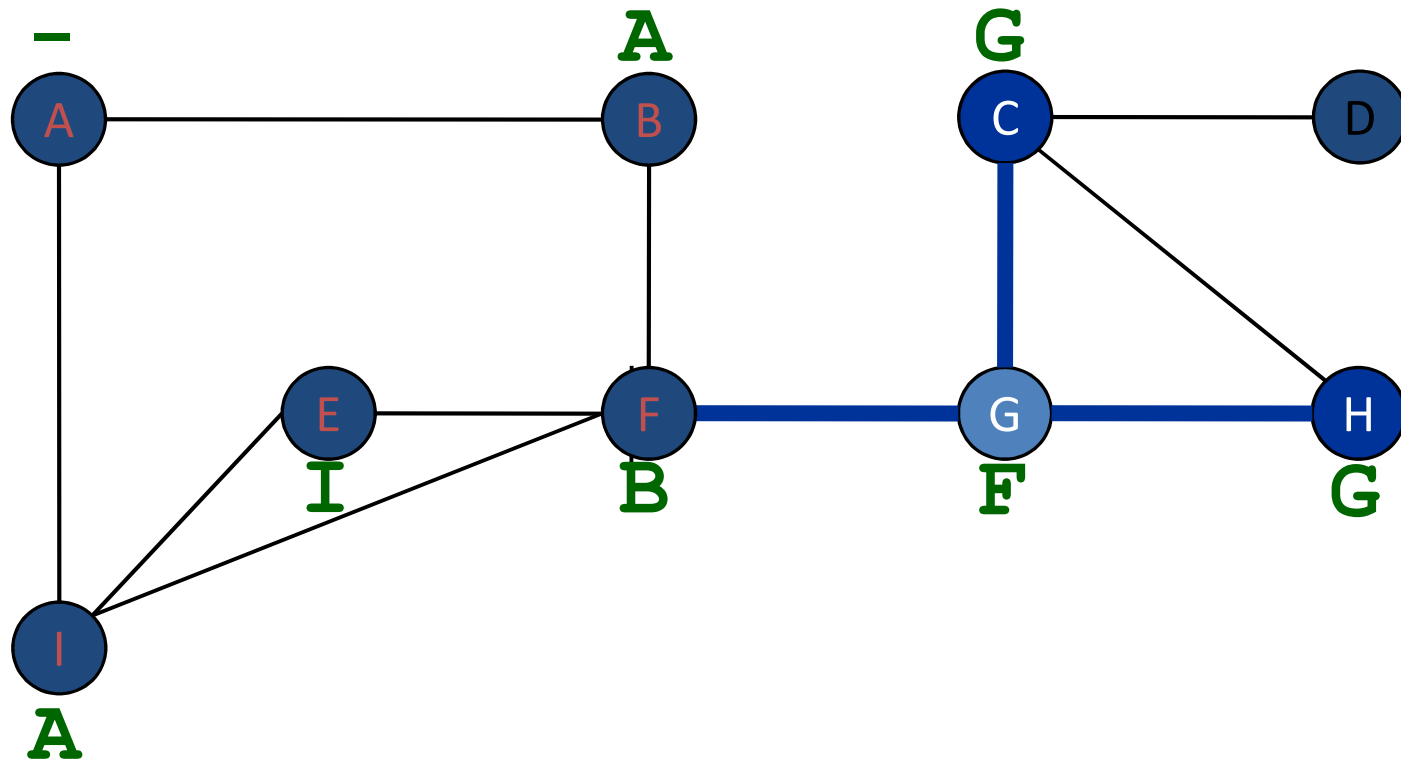
Visita vizinhos de G

front

C

FIFO

# Breadth First Search



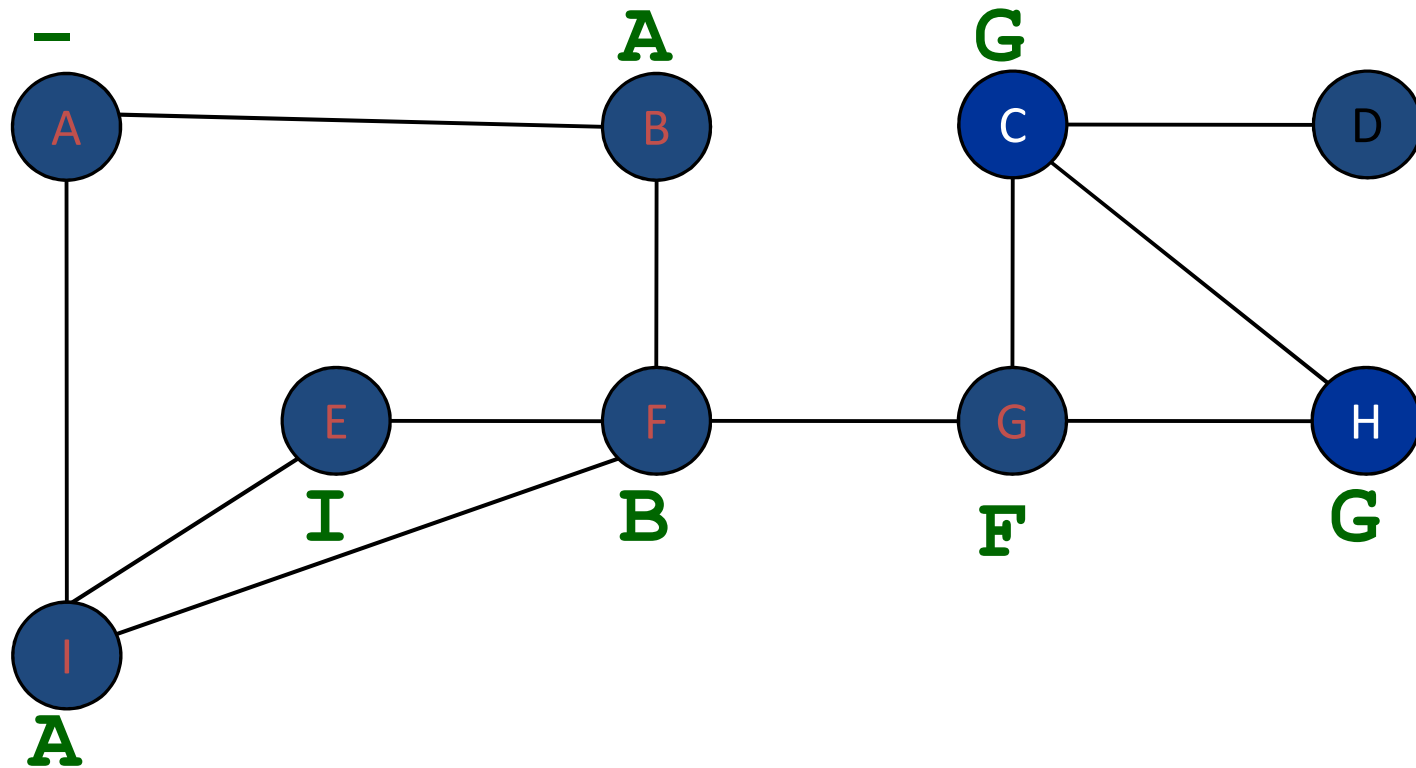
H Descoberto

front

C H

FIFO

# Breadth First Search



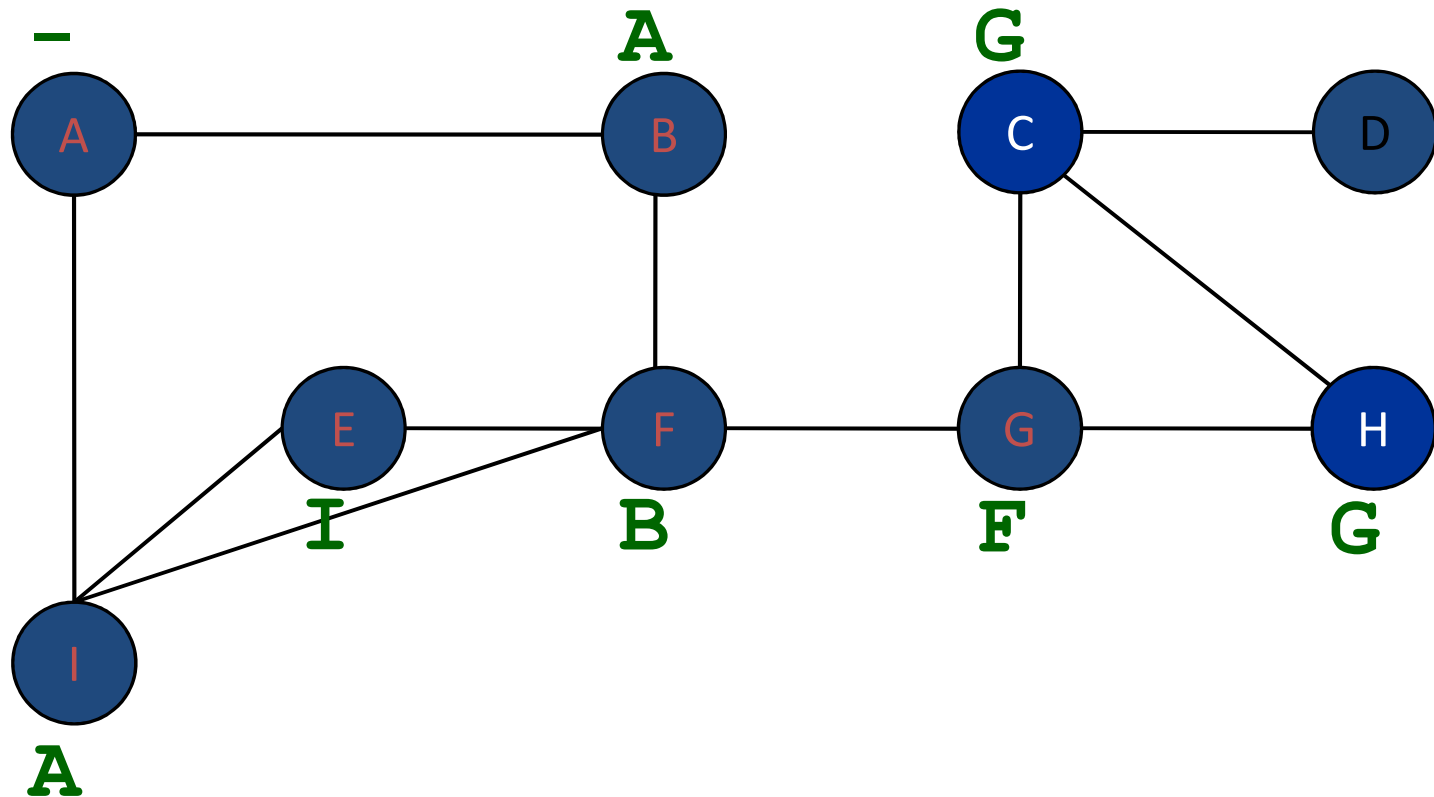
G Fim

front

C H

FIFO

# Breadth First Search



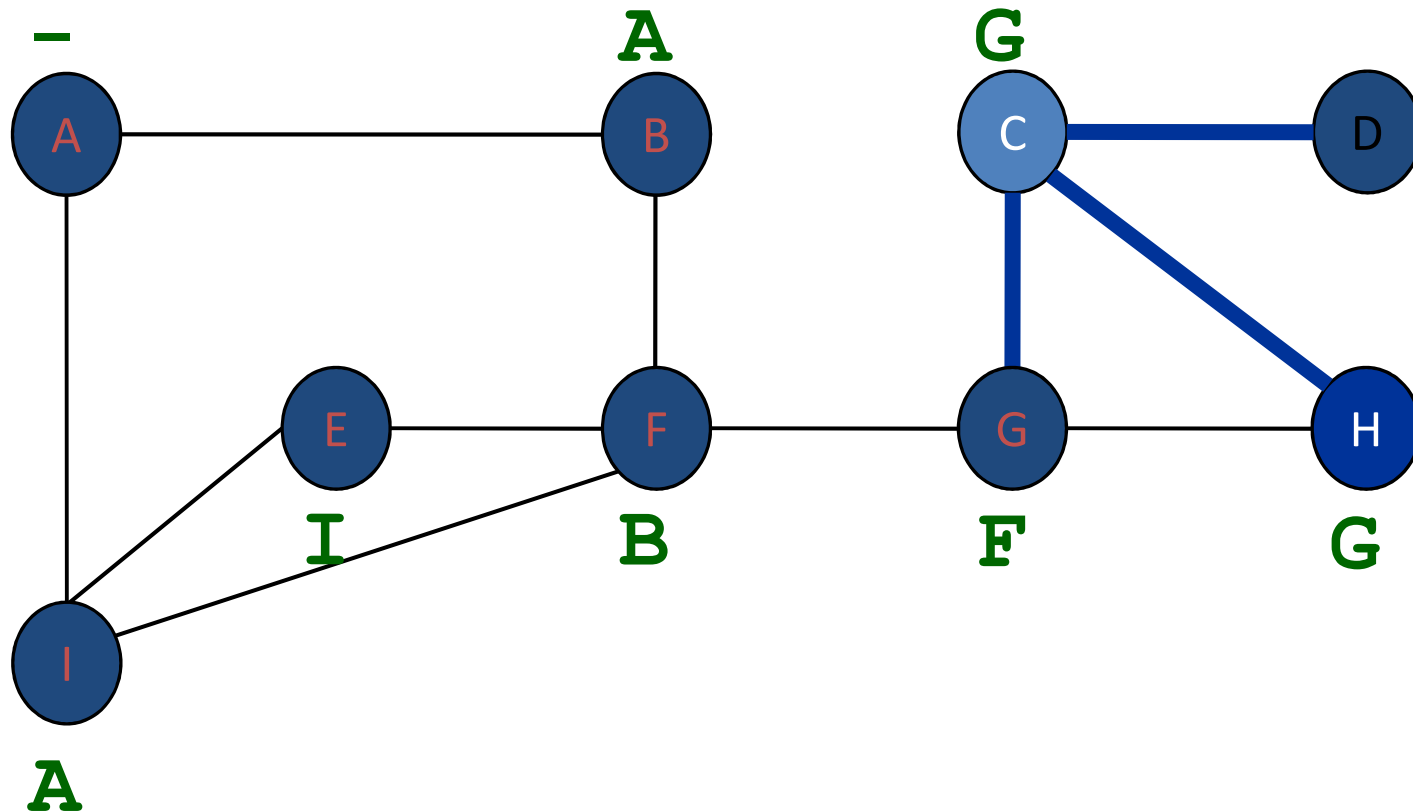
Próximo no na filha

front

C H

FIFO

# Breadth First Search



Visita vizinhos de C

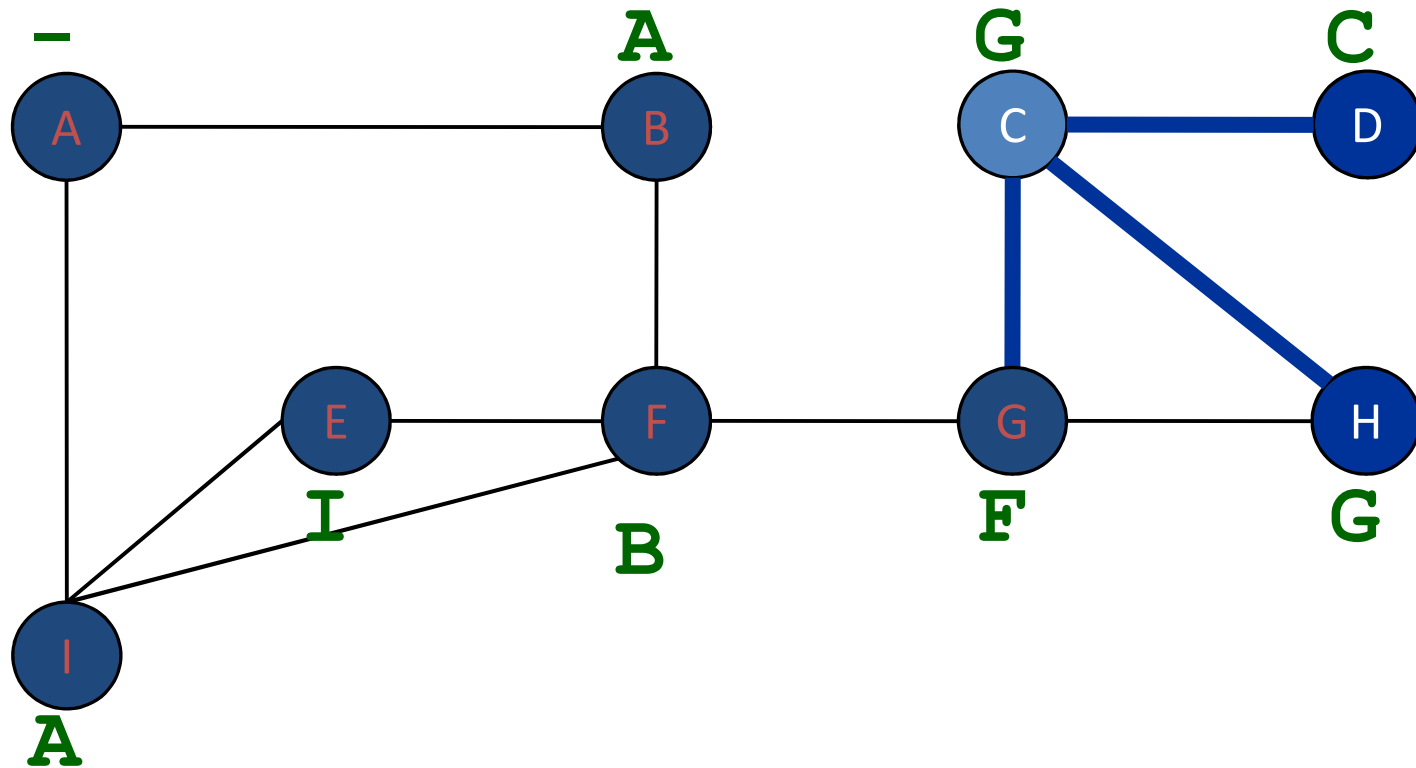
front

H

FIFO



# Breadth First Search



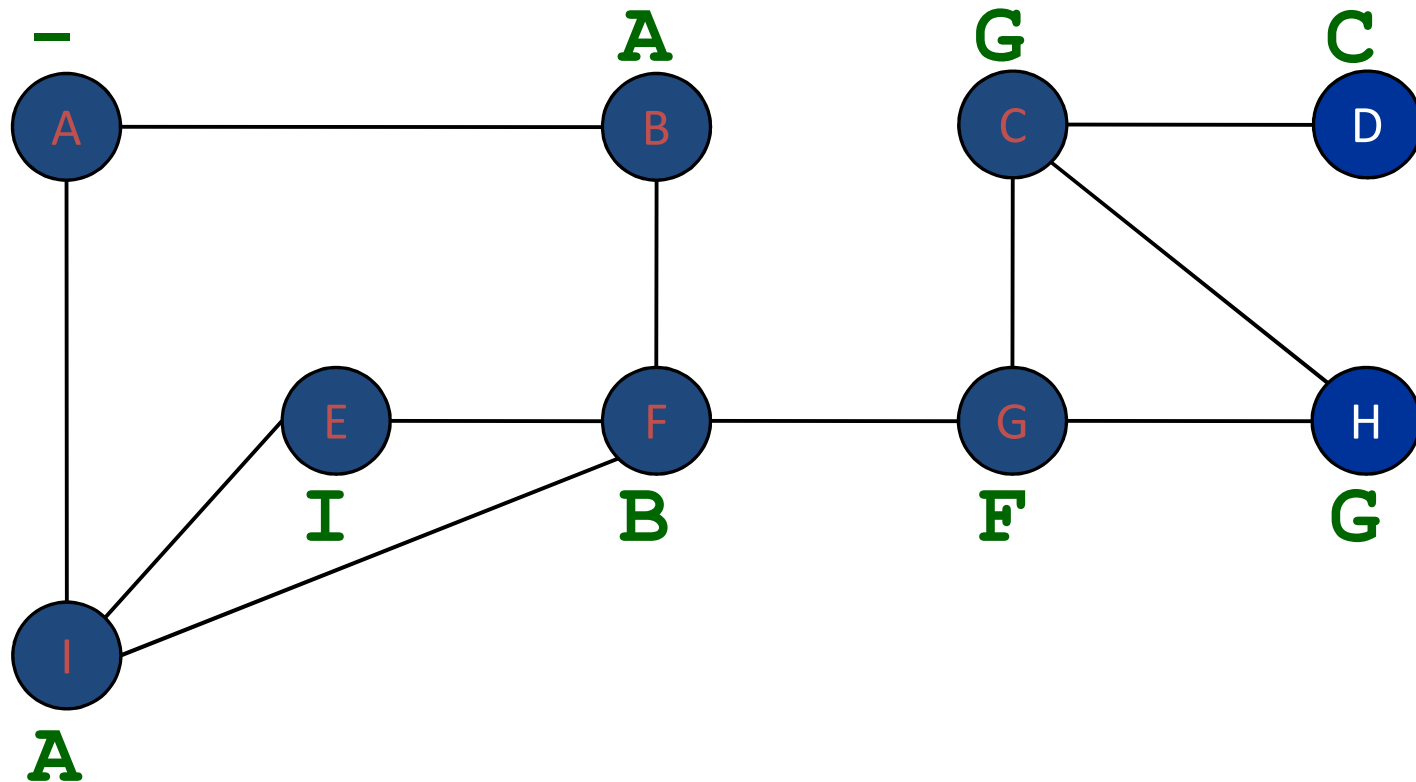
D Descoberto

front

H D

FIFO

# Breadth First Search



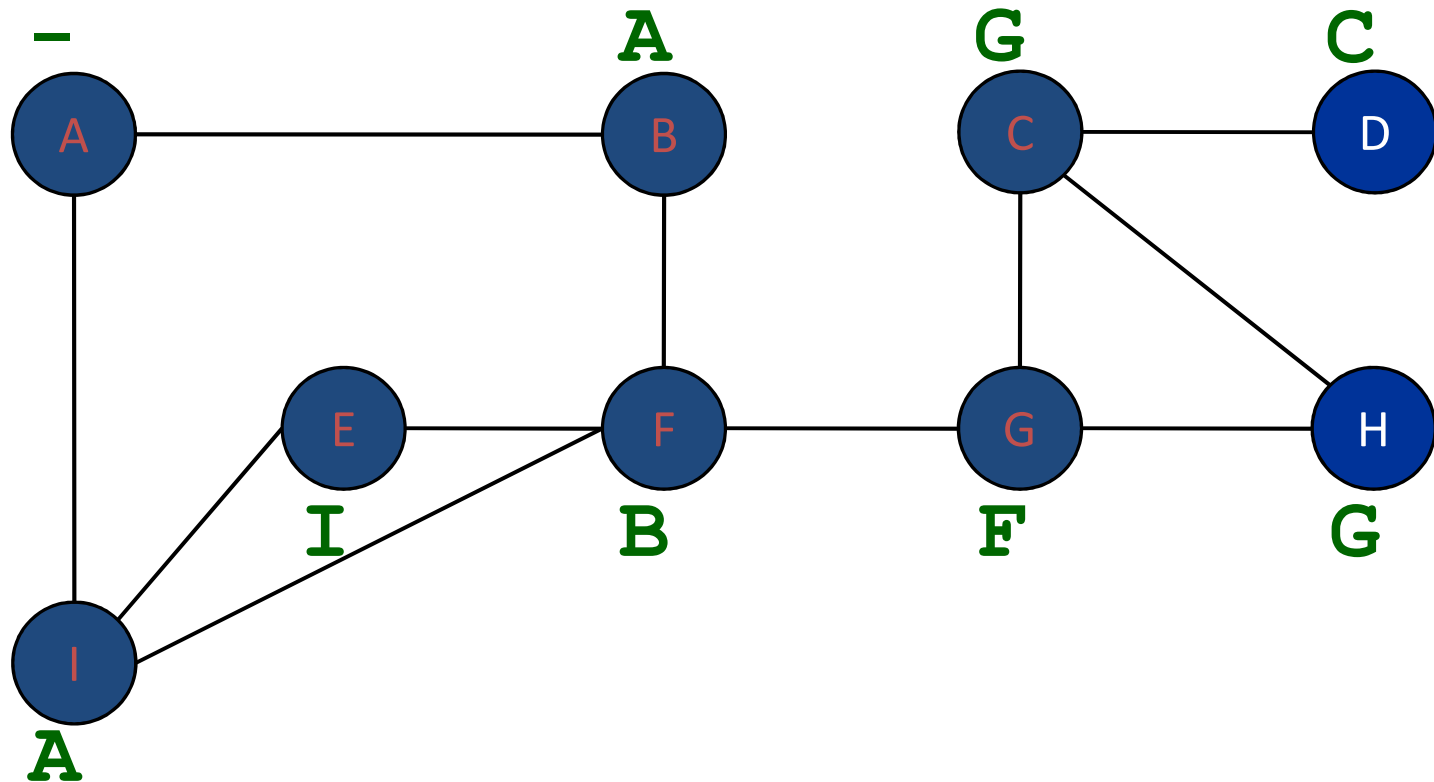
C Fim

front

H D

FIFO

# Breadth First Search



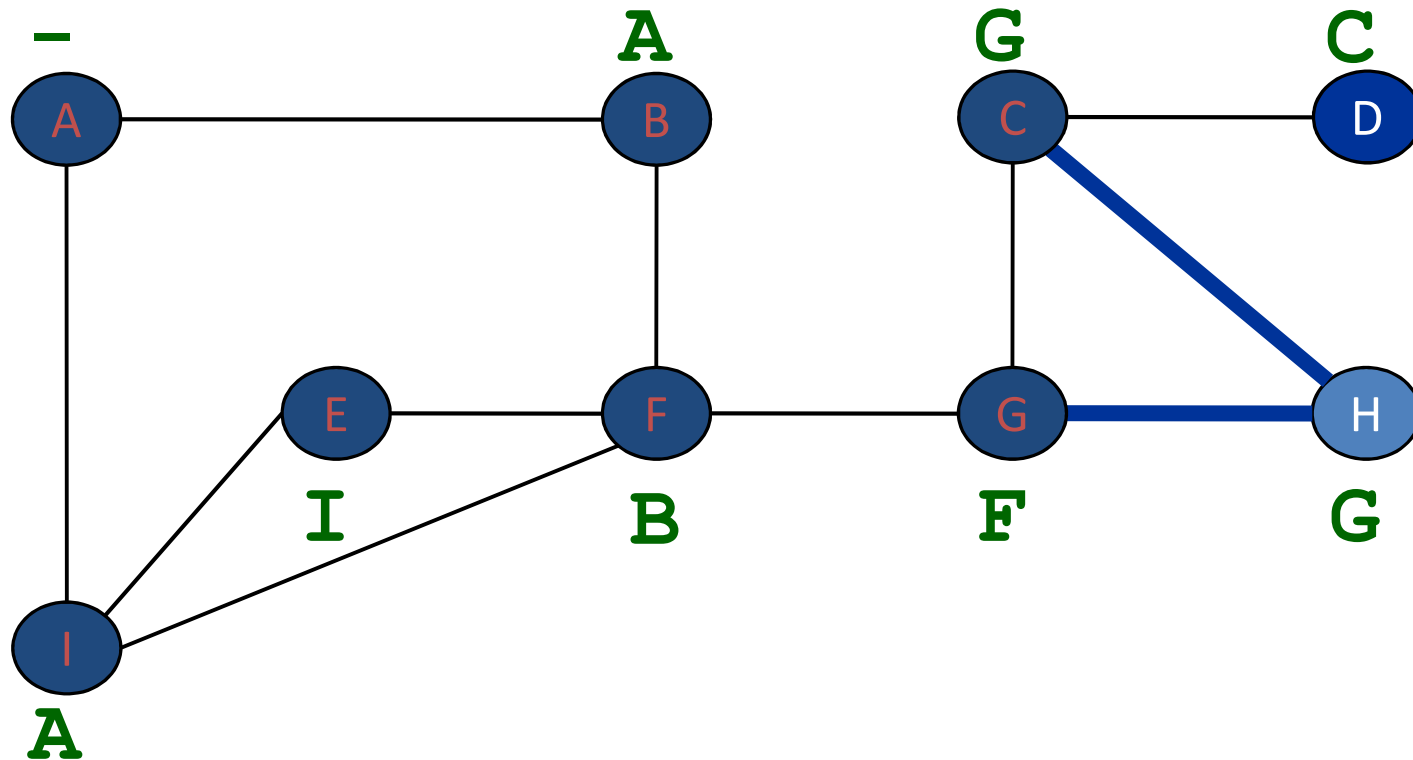
get next vertex

front

H D

FIFO

# Breadth First Search



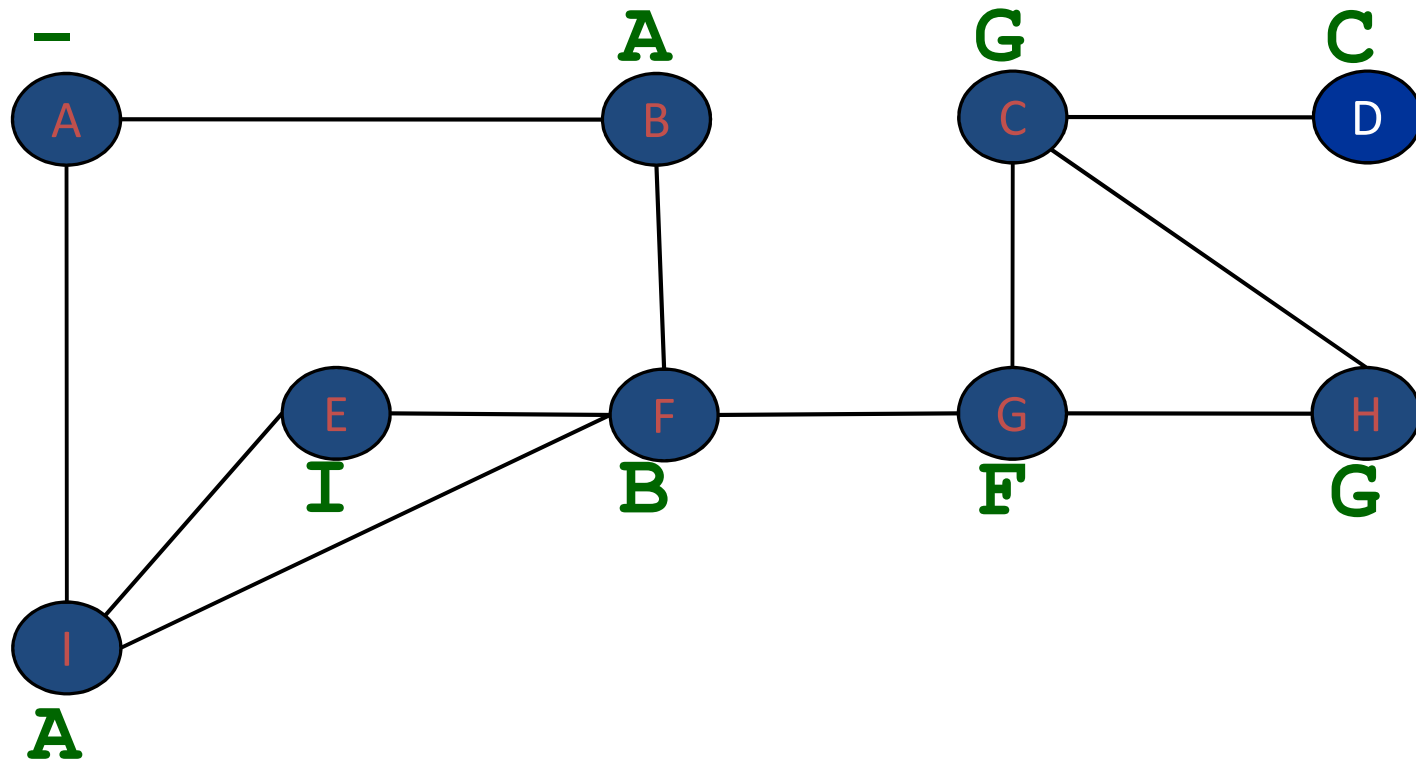
Visita vizinhos de H

front

D

FIFO

# Breadth First Search



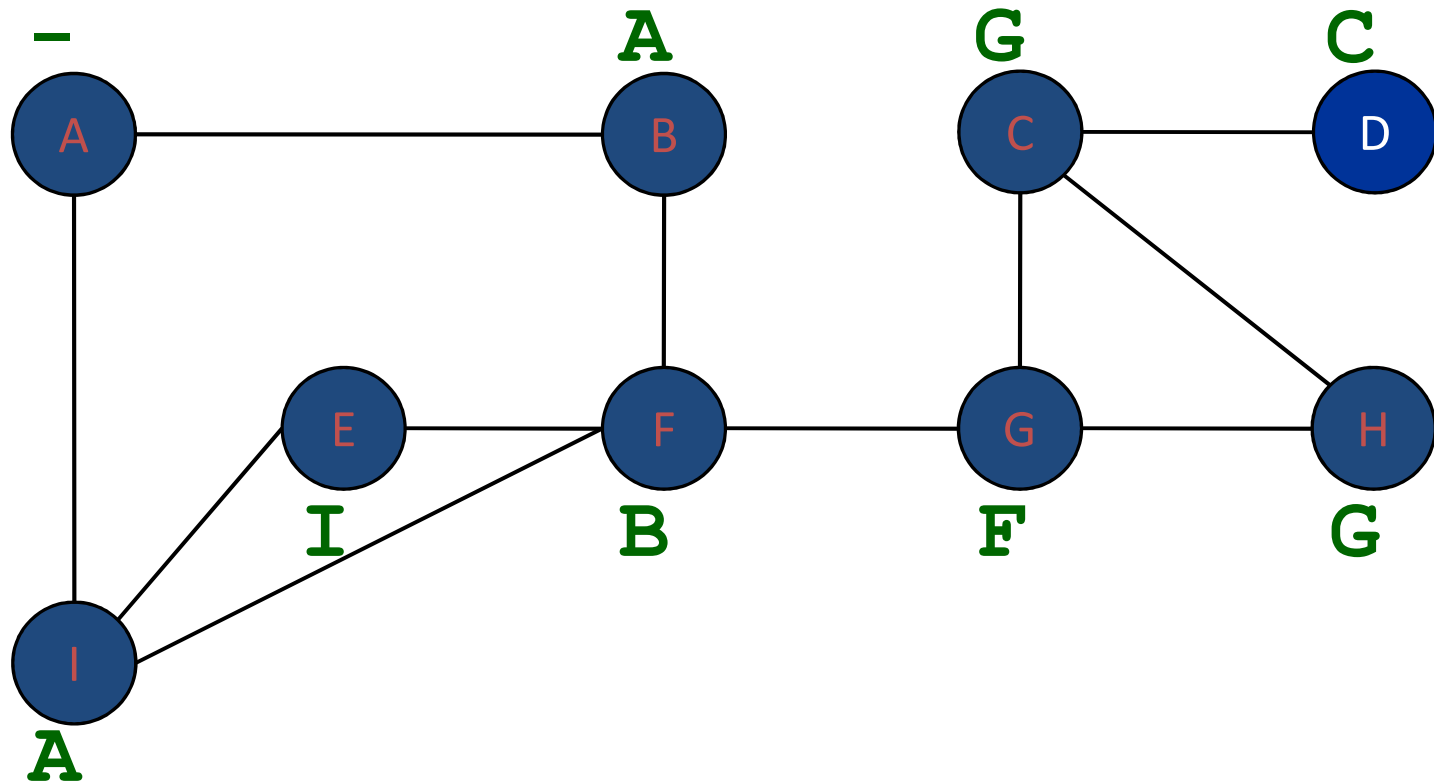
Fim H

front

D

FIFO

# Breadth First Search



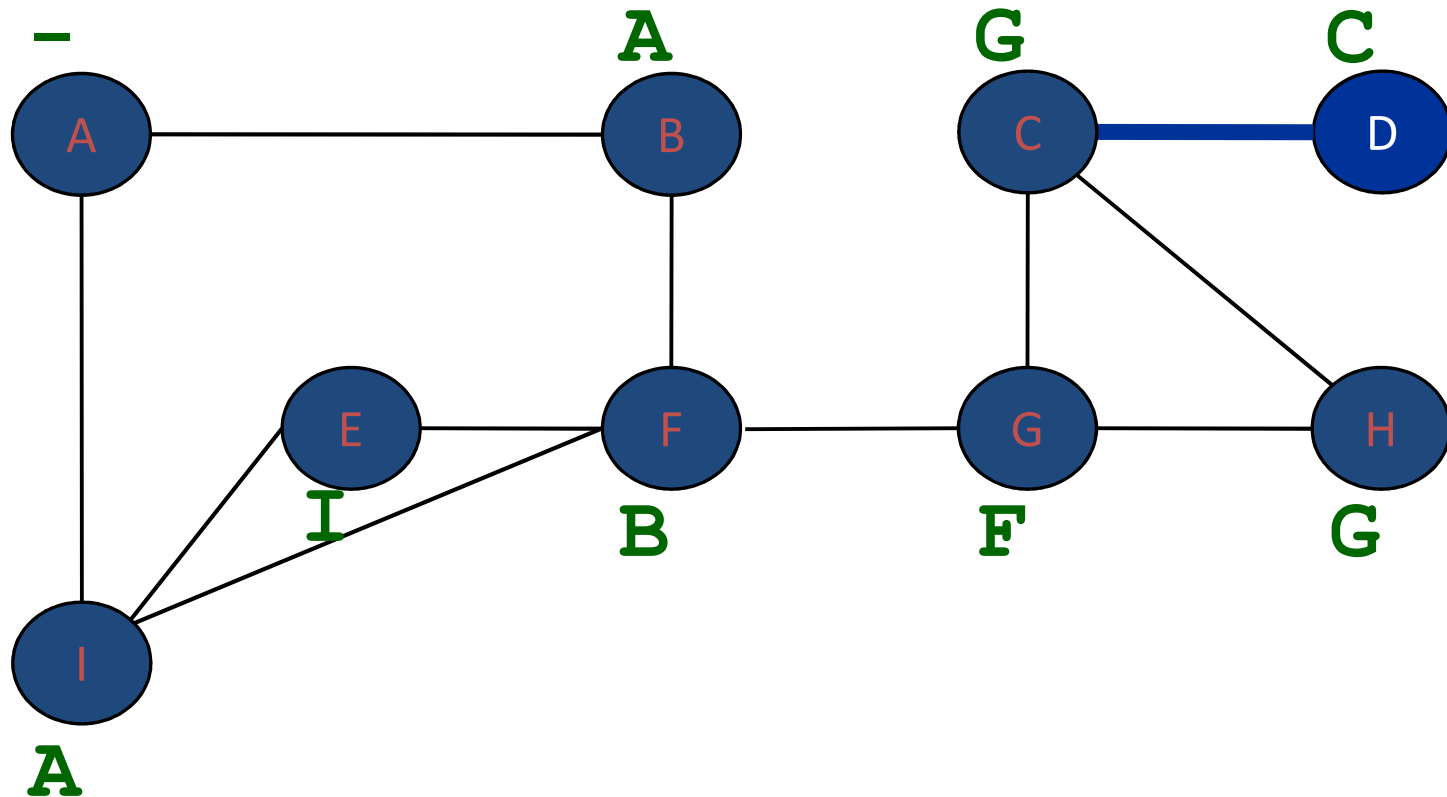
Próximo no na filha

front

D

FIFO

# Breadth First Search

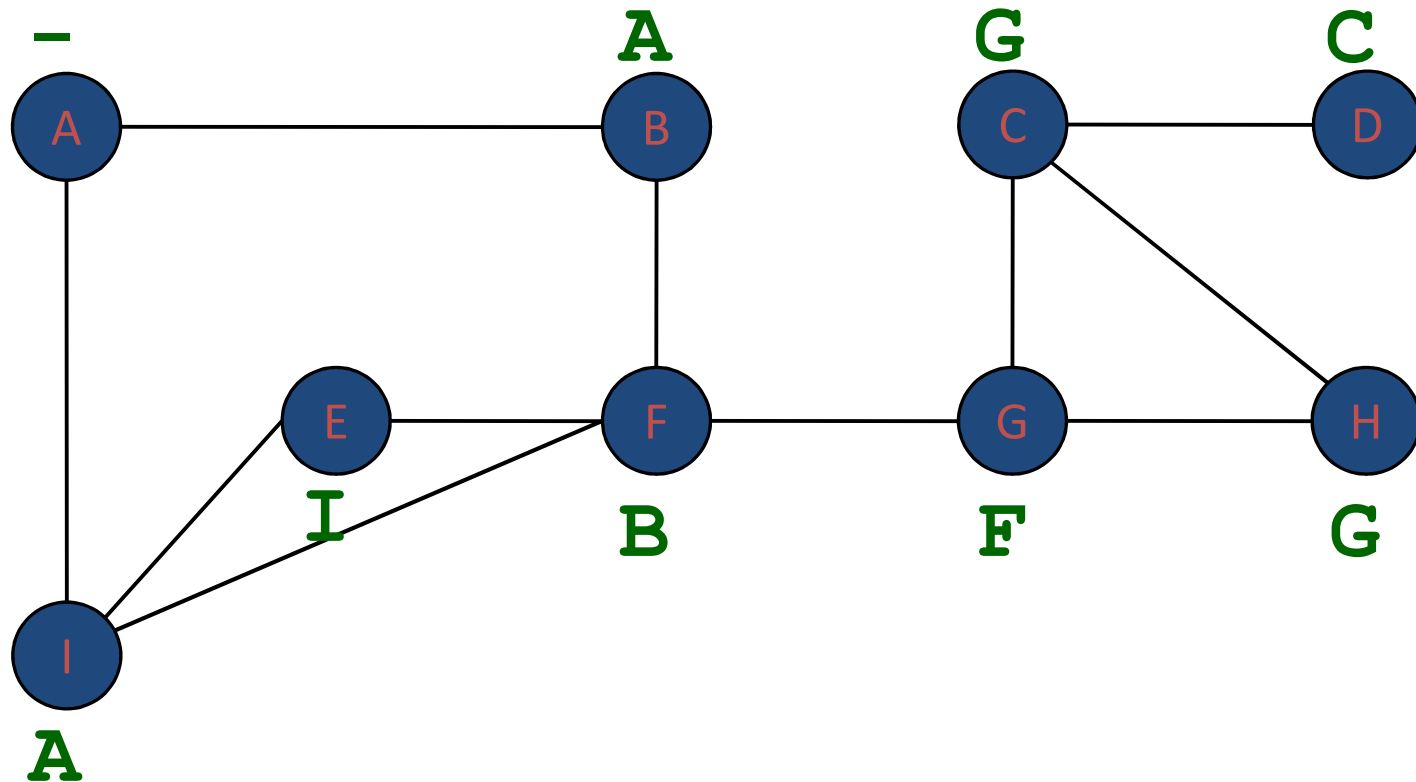


Visita vizinhos de D

front

FIFO

# Breadth First Search



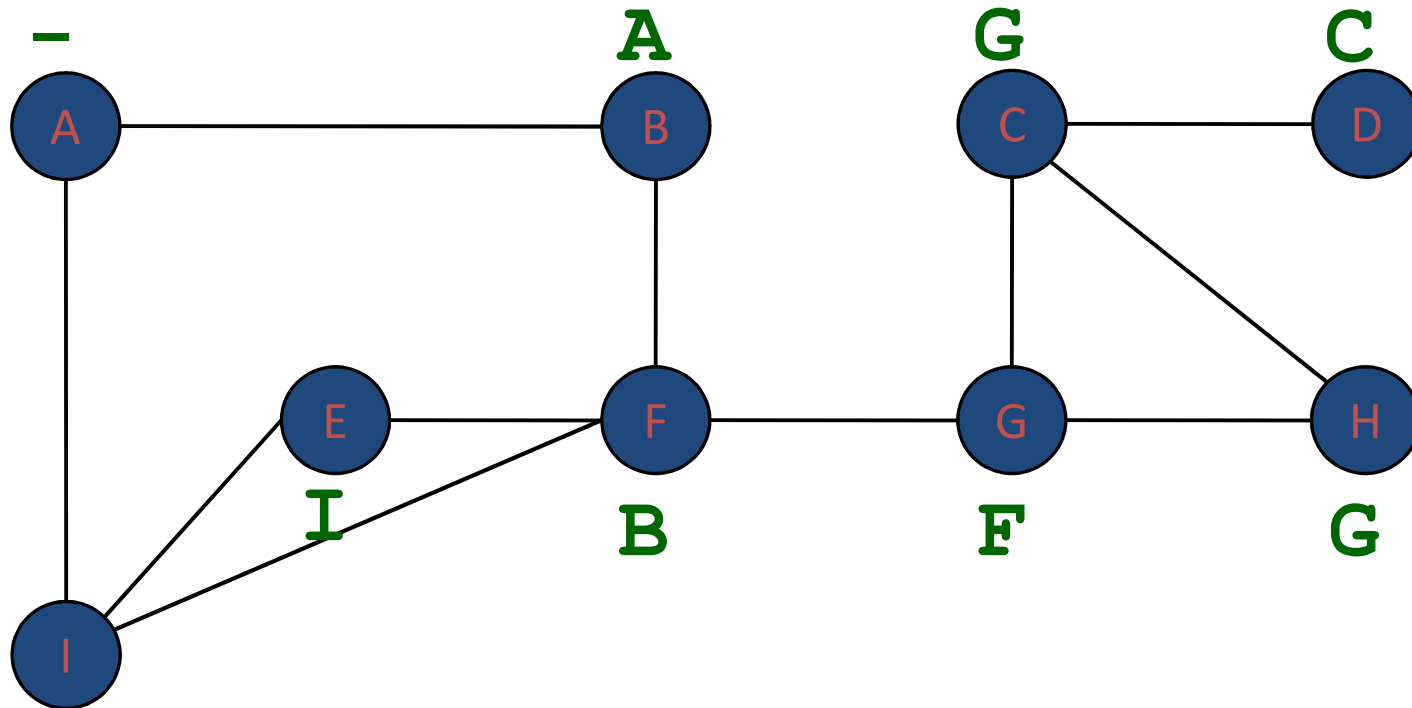
D Fim

front

FIFO



# Breadth First Search

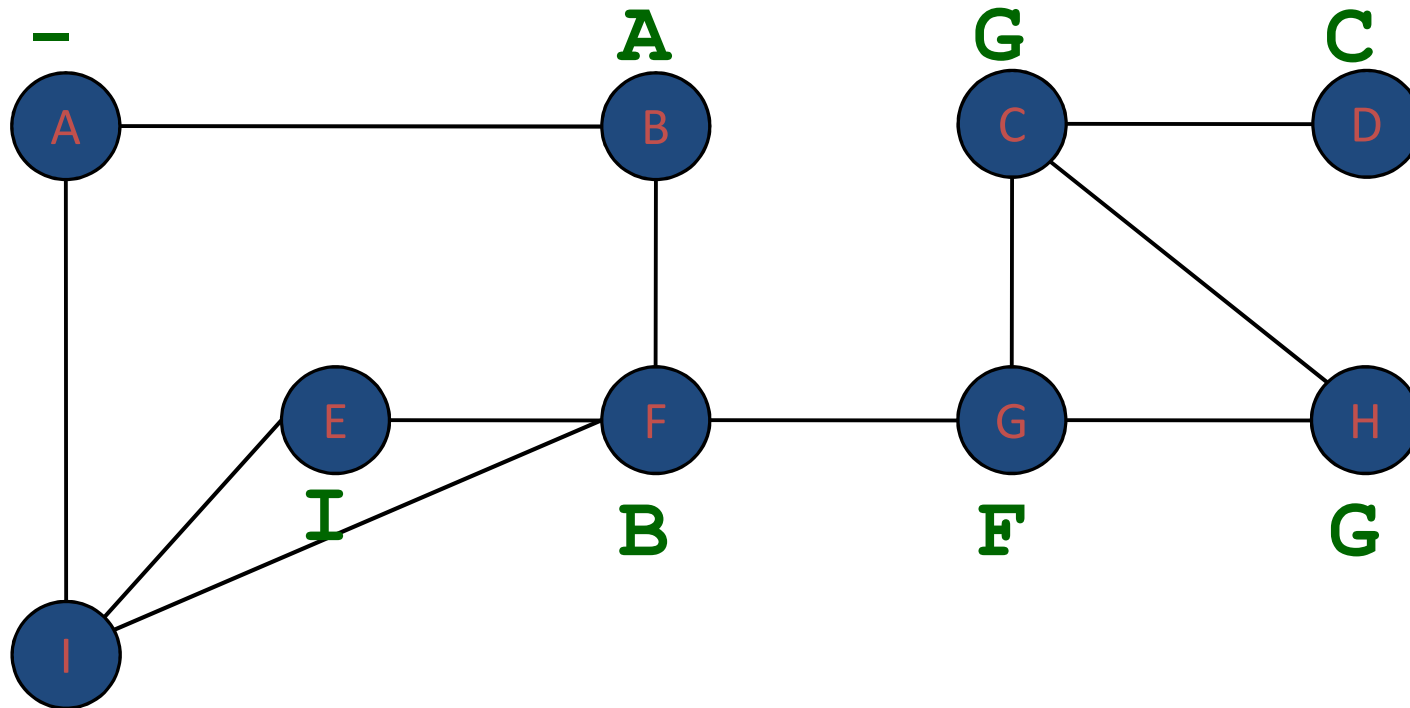


Próximo no na filha

front

FIFO

# Breadth First Search



FIM

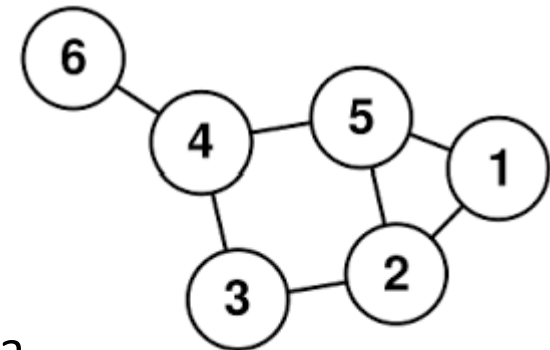
front

FIFO

# Exercícios

Do seguinte grafo diga:

- a) O grafo é dirigido o no dirigido?
- b) Identifique um vértice adjacente do vértice 4
- c) Qual é o grau do vértice 1
- d) Apresente o caminho desde o vértice 4 até o vértice 1
- e) Qual é comprimento do caminho (5,2,3,4,6)
- f) O grafo é acíclico o cíclico.
- g) É um grafo ponderado o não.
- h) Represente as arvores de busca a para uma busca em largura no grafo. Qual é a estrutura de dados utilizada para percorrer o grafo?. Mostre para cada passo como fica essa estrutura.
- i) Represente as arvores de busca a para uma busca em profundidade no grafo . Qual é a estrutura de dados utilizada para percorrer o grafo?. Mostre para cada passo como fica essa estrutura.



# Exercícios

Do seguinte grafo diga:

- a) O grafo é dirigido o no dirigido?
- b) Identifique um vértice adjacente do vértice 3
- c) Qual é o grau de entrada do vértice 4
- d) Apresente o caminho desde o vértice 5 até o vértice 3
- e) Qual é comprimento do caminho (1,3,4,2)
- f) O grafo é acíclico o cíclico.
- g) É um grafo ponderado o não.
- h) Represente as arvores de busca a para uma busca em largura no grafo. Qual é a estrutura de dados utilizada para percorrer o grafo?. Mostre para cada passo como fica essa estrutura.
- i) Represente as arvores de busca a para uma busca em profundidade no grafo . Qual é a estrutura de dados utilizada para percorrer o grafo?. Mostre para cada passo como fica essa estrutura.

