

Introdução à Teoria dos Grafos

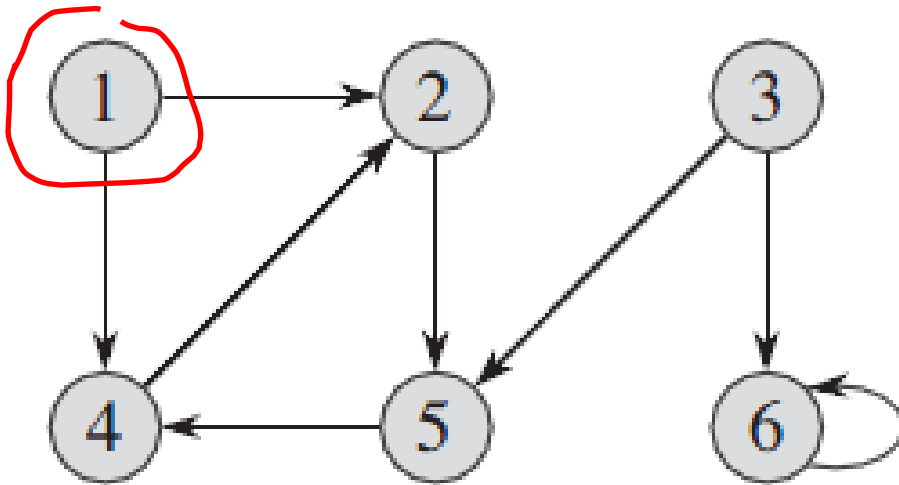
Prof. Alexandre Noma

Aula passada

- Representar um grafo no computador:
- (1) **Matriz** de adjacências
- (2) **Listas** de adjacências

Aula passada

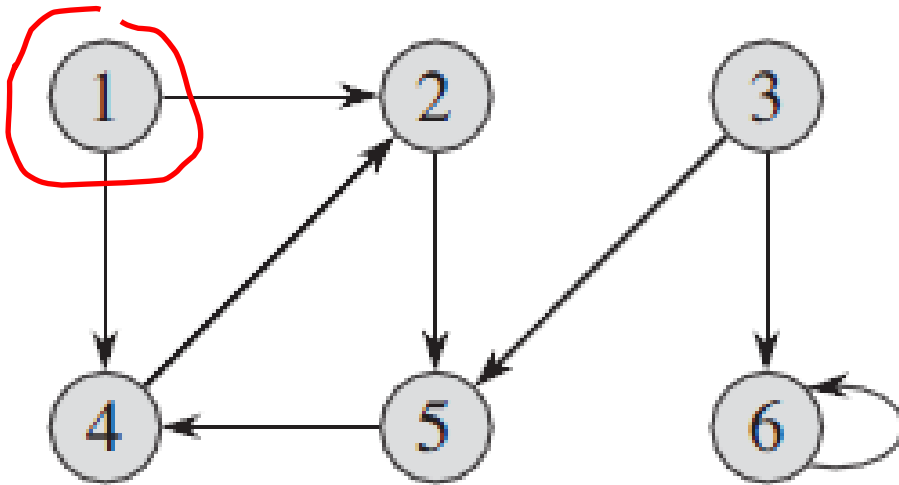
(1) Matriz de adjacências



	1	2	3	4	5	6
1	0	1	0	1	0	0
2						
3						
4						
5						
6						

Aula passada

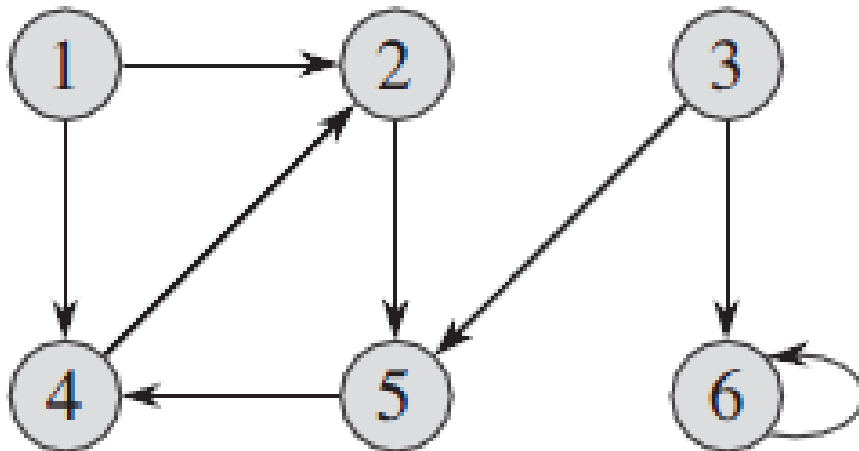
(1) Matriz de adjacências



	1	2	3	4	5	6
1	0	1	0	1	0	0
2	0	0	0	0	1	0
3	0	0	0	0	1	1
4	0	1	0	0	0	0
5	0	0	0	1	0	0
6	0	0	0	0	0	1

Aula passada

(2) Listas de adjacências



1: 2, 4

2: ?

3: ?

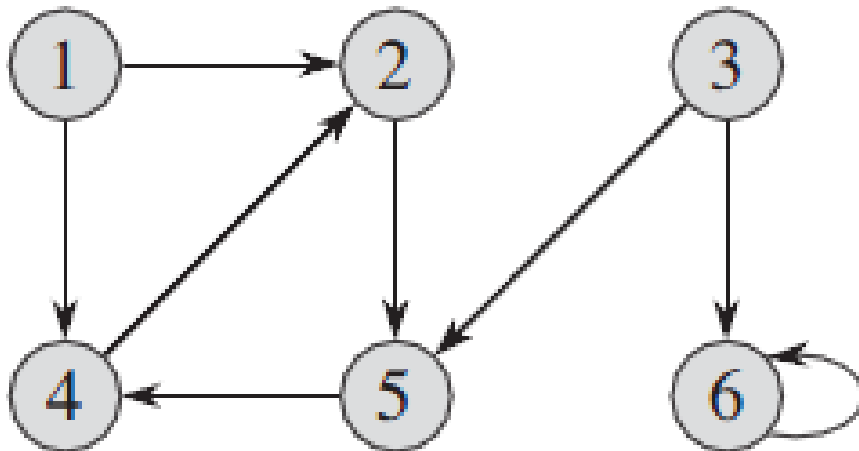
4: ?

5: ?

6: ?

Aula passada

(2) Listas de adjacências



1: 2, 4

2: 5

3: 5, 6

4: 2

5: 4

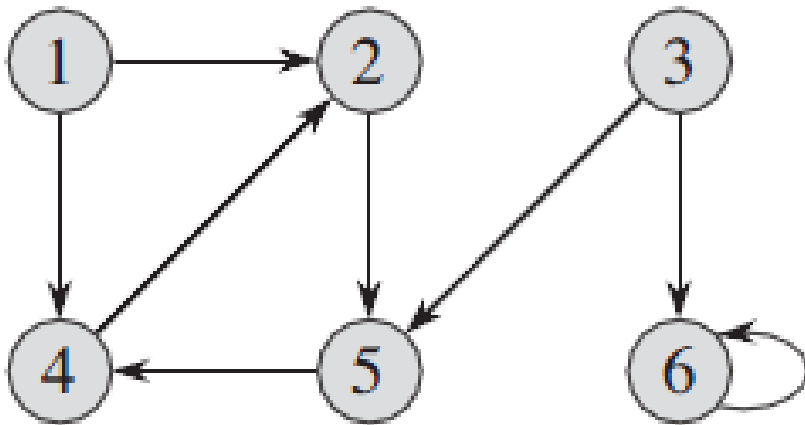
6: 6

Aula passada

- Representar um grafo no computador:
- (1) **Matriz** de adjacências
- (2) **Listas** de adjacências
- adjacência = "**vizinhança**"

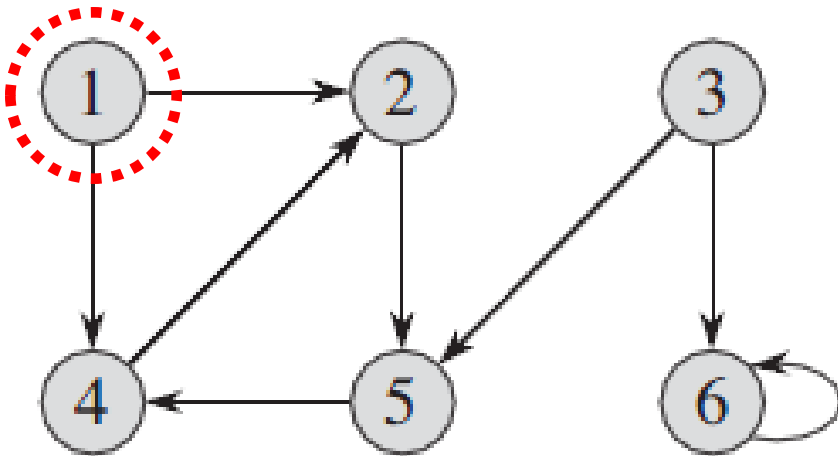
Hoje

- Como visitar TODOS os vértices de um grafo?



Hoje

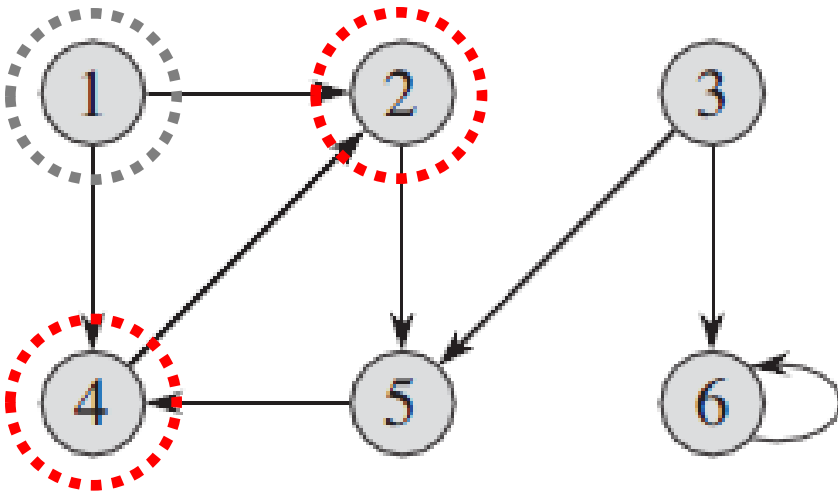
- Como visitar TODOS os vértices de um grafo?



Escolha um **vértice inicial**.

Hoje

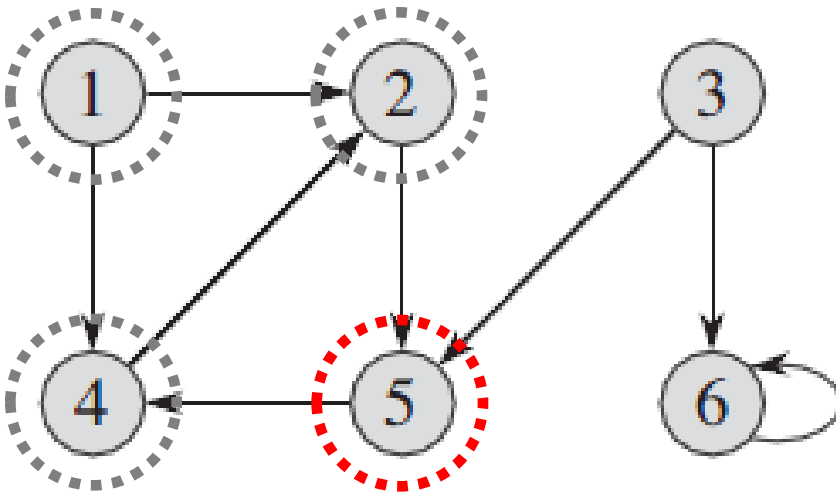
- Como visitar TODOS os vértices de um grafo?



Visite os seus **vizinhos**.

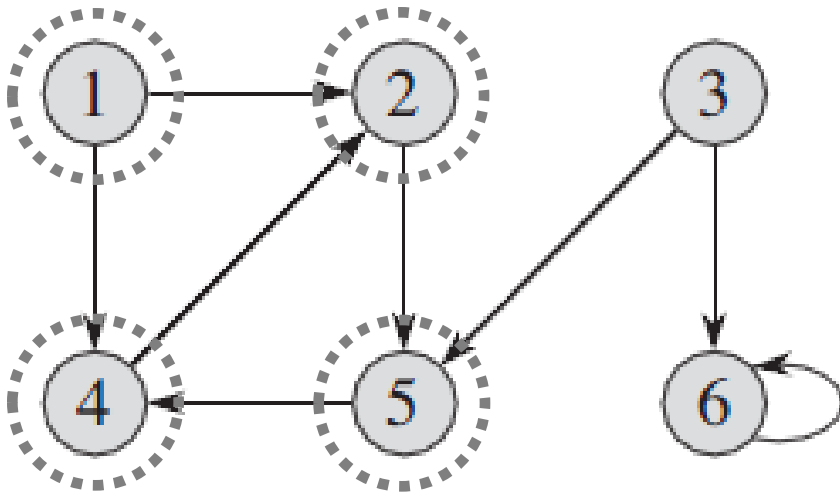
Hoje

- Como visitar TODOS os vértices de um grafo?



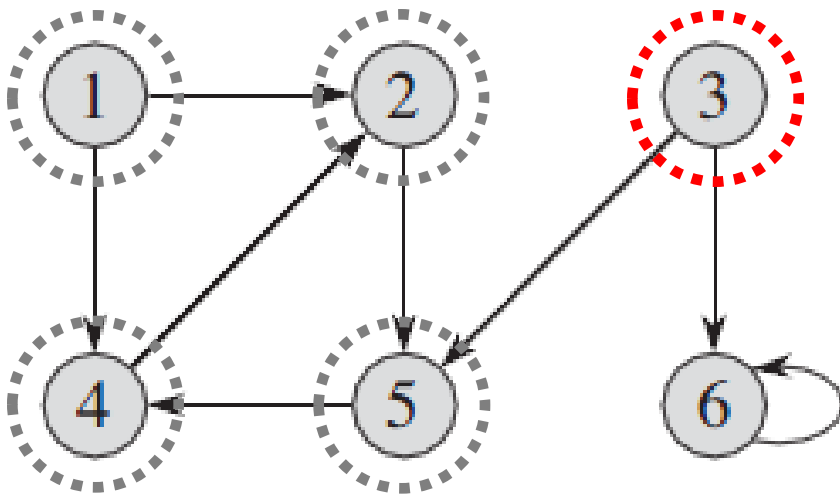
Visite os seus **vizinhos**.

- Como visitar TODOS os vértices de um grafo?



Acabou **vizinhos**???

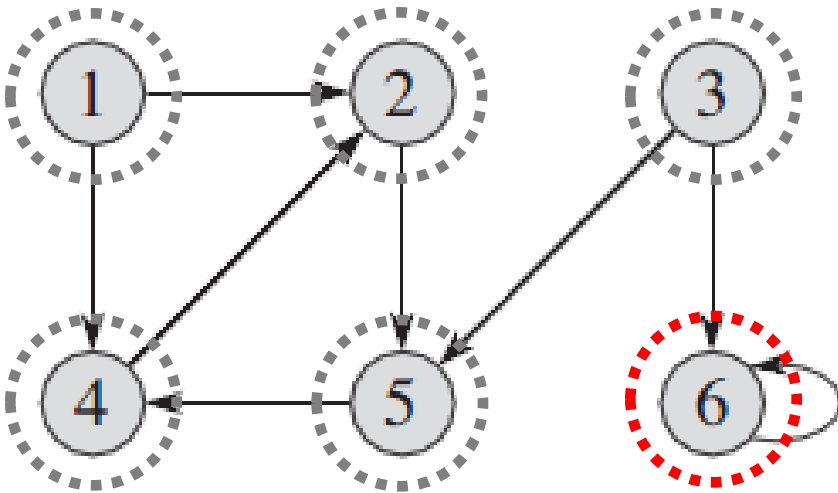
- Como visitar TODOS os vértices de um grafo?



Acabou vizinhos???

Escolha “vértice inicial”.

- Como visitar TODOS os vértices de um grafo?



Visite os seus **vizinhos**.

Nosso primeiro algoritmo de busca...

- **BFS**: breadth first search

"Busca em largura"

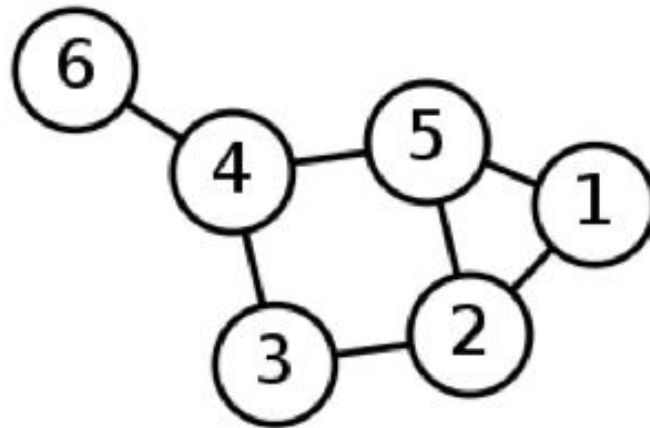
Nosso primeiro algoritmo de busca...

- **BFS**: breadth first search

"Busca em largura"

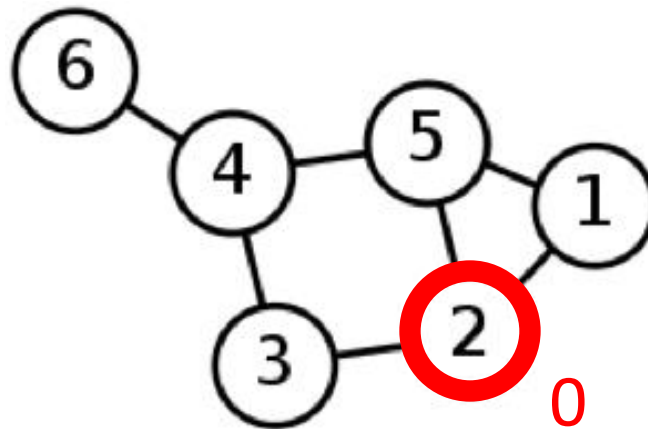
- Por que este nome???

Propagação de “fogo”



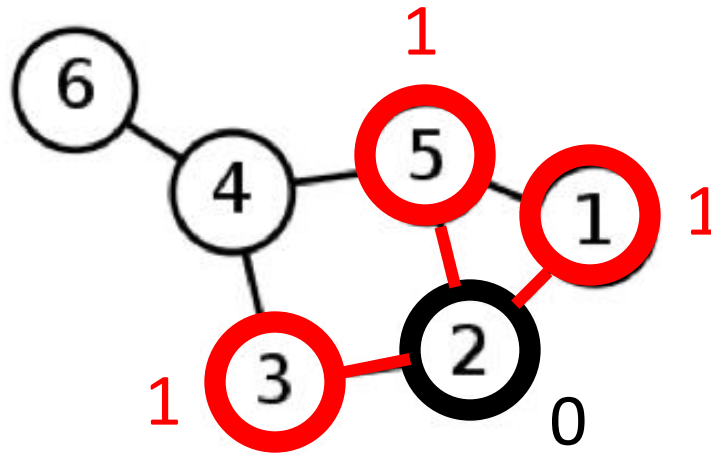
Propagação de “fogo”

Vértice inicial.



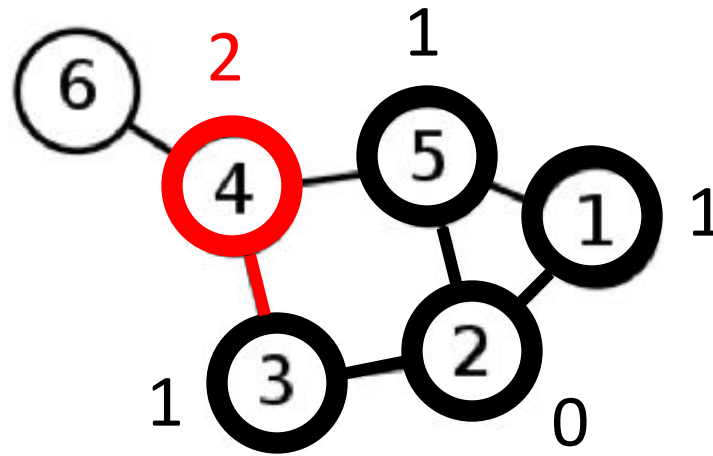
Propagação de “fogo”

Fogo espalha para vizinhos.



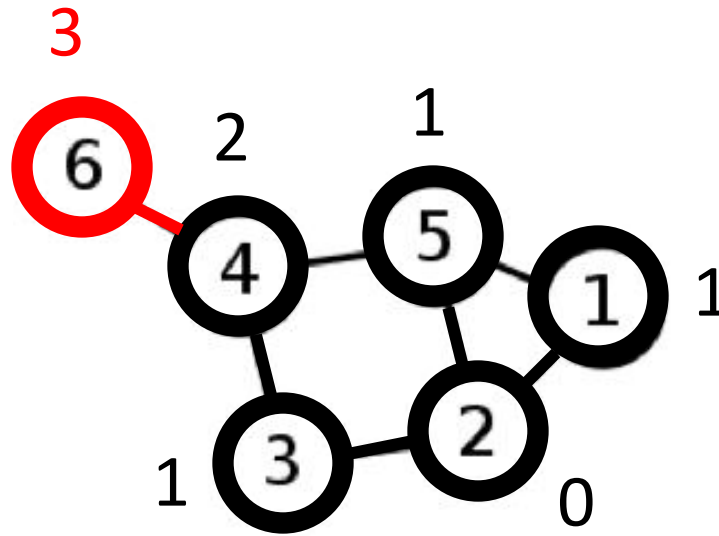
Propagação de “fogo”

Fogo espalha para vizinhos.



Propagação de “fogo”

Fogo espalha para vizinhos.



Implementação

- Lembra da estrutura de dados **Fila** ?



Implementação

- Lembra da estrutura de dados **Fila** ?



INÍCIO

FIM

Algumas operações com Fila:

- **Inserir(Q, x)**
 - insere elemento **x** no **final** da fila **Q**
- **Remove(Q)**
 - remove e devolve o elemento do **início** da fila **Q**
- **FilaVazia(Q)**
 - devolve verdadeiro se a fila estiver vazia, falso caso contrário.

INÍCIO



FIM

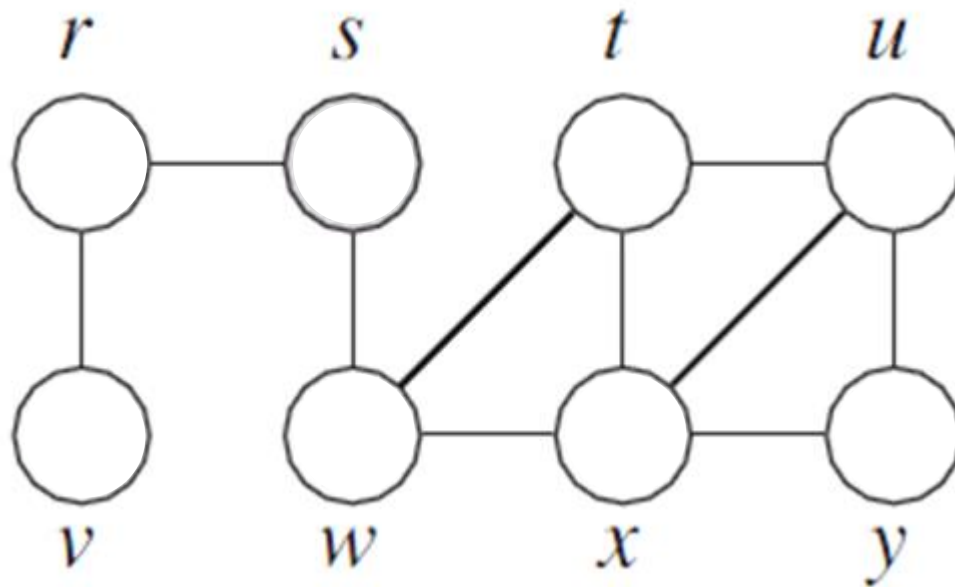
Busca em largura

- **BFS**(G, s)
 - **Entrada**: um grafo G e um vértice inicial s
 - **Saída**: “**distâncias**” em relação ao vértice s

Busca em largura

- **BFS**(G, s)
 - **Entrada**: um grafo G e um vértice inicial s
 - **Saída**: “**distâncias**” em relação ao vértice s
- Atributos
 - $v.d$: distância
 - $v.cor$: BRANCO, CINZA, PRETO
(inicialmente não visitado
visitado,
finalizado)

Exemplo



BFS (**G**, **s**) :

```
1 para cada vértice  $v_i$  em  $G.V$  faça
2      $v_i.d = \text{INFINITO}$ 
3      $v_i.cor = \text{BRANCO}$ 
4  $s.d = 0$ 
5  $s.cor = \text{CINZA}$ 
6 Q = VAZIO // Inicialização
7 Insere(Q,  $s$ )
.....
8 enquanto  $Q \neq \text{VAZIO}$  faça
9      $u_i = \text{Remove}(Q)$ 
10    para cada  $v_i$  em  $G.Adj[u_i]$  faça
11        se  $v_i.cor == \text{BRANCO}$ 
12            então  $v_i.d = u_i.d + 1$ 
13                 $v_i.cor = \text{CINZA}$ 
14                Insere(Q,  $v_i$ )
15     $u_i.cor = \text{PRETO}$ 
```

BFS (**G**, **s**) :

```
1 para cada vértice  $v_i$  em  $G.V$  faça
2      $v_i.d = \text{INFINITO}$ 
3      $v_i.cor = \text{BRANCO}$ 
4  $s.d = 0$ 
5  $s.cor = \text{CINZA}$ 
6 Q = VAZIO
7 Insere(Q, s)
```

// Inicialização

BFS (**G**, **s**) :

```
8 enquanto Q != VAZIO faça
9     ui = Remove(Q)
10    para cada vi em G.Adj[ui] faça
11        se vi.cor == BRANCO
12            entao vi.d = ui.d + 1
13                vi.cor = CINZA
14                Insere(Q, vi)
15    ui.cor = PRETO
```

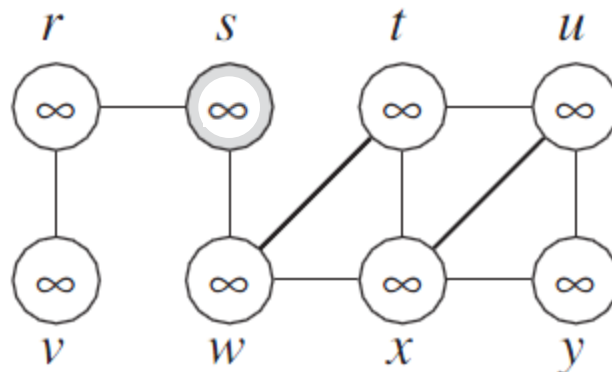
BFS (**G**, **s**) :

```
8 enquanto Q != VAZIO faça
9     ui = Remove(Q)
10    para cada vi em G.Adj[ui] faça
11        se vi.cor == BRANCO
12            entao vi.d = ui.d + 1
13                vi.cor = CINZA
14                Insere(Q, vi)
15    ui.cor = PRETO
```

BFS (**G**, **s**) :

```
1 para cada vértice  $v_i$  em  $G.V$  faça
2    $v_i.d = \text{INFINITO}$ 
3    $v_i.cor = \text{BRANCO}$ 
4  $s.d = 0$ 
5  $s.cor = \text{CINZA}$ 
6 Q = VAZIO
7 Insere(Q,  $s$ )
```

(Inicialização)



BFS (**G**, **s**) :

1 para cada vértice v_i em $G.V$ faça

2 $v_i.d = \text{INFINITO}$

3 $v_i.cor = \text{BRANCO}$

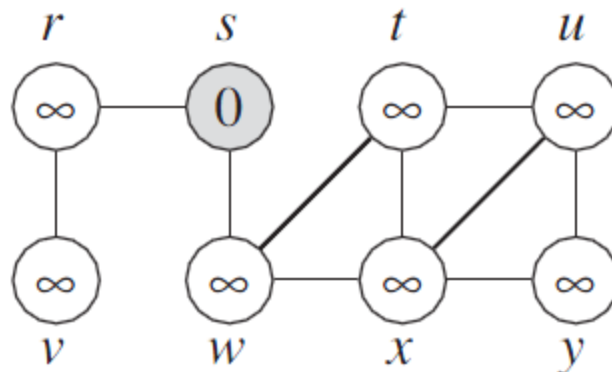
4 $s.d = 0$

5 $s.cor = \text{CINZA}$

6 **Q** = VAZIO

7 Insere(**Q**, s)

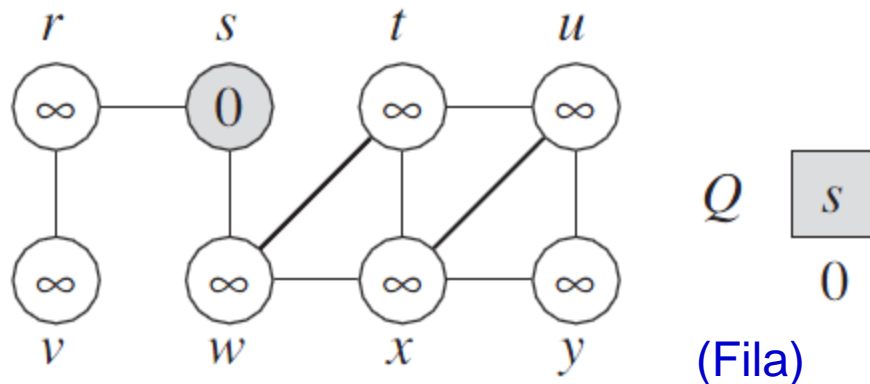
(Inicialização)



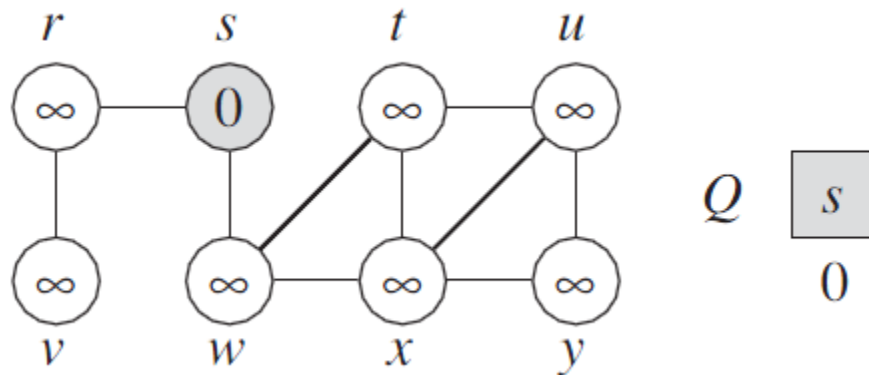
BFS (**G**, **s**) :

```
1 para cada vértice  $v_i$  em  $G.V$  faça
2    $v_i.d = \text{INFINITO}$ 
3    $v_i.\text{cor} = \text{BRANCO}$ 
4  $s.d = 0$ 
5  $s.cor = \text{CINZA}$ 
6 Q = VAZIO
7 Insere(Q,  $s$ )
```

(Inicialização)

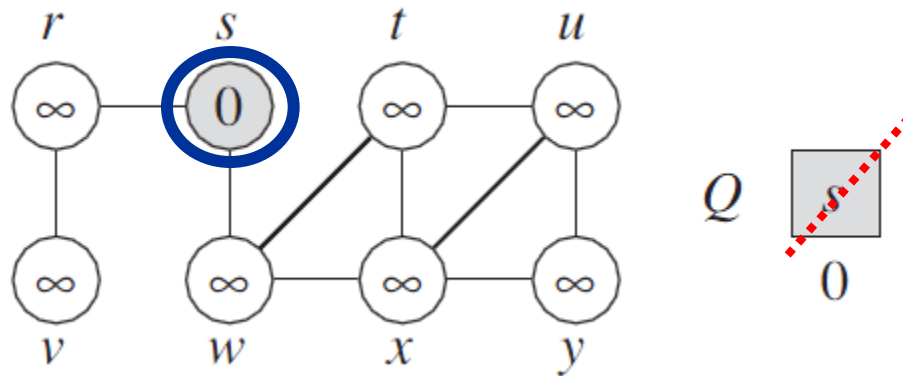


BFS (G, s) :




```
8 enquanto  $Q \neq \text{VAZIO}$  faça
9      $u_i = \text{Remove}(Q)$ 
10    para cada  $v_i$  em  $G.\text{Adj}[u_i]$  faça
11        se  $v_i.\text{cor} == \text{BRANCO}$ 
12            entao  $v_i.d = u_i.d + 1$ 
13                 $v_i.\text{cor} = \text{CINZA}$ 
14                 $\text{Insere}(Q, v_i)$ 
15     $u_i.\text{cor} = \text{PRETO}$ 
```

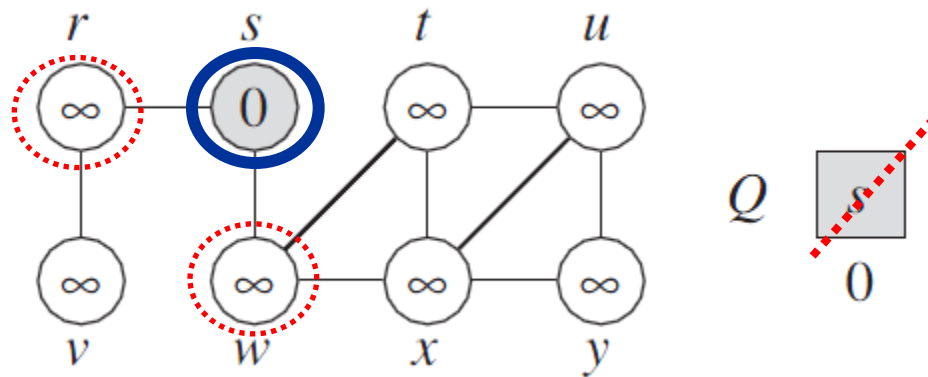
BFS (G, s) :



```

8 enquanto Q != VAZIO faça
9      ui = Remove(Q)
10    para cada vi em G.Adj[ui] faça
11        se vi.cor == BRANCO
12            entao vi.d = ui.d + 1
13            vi.cor = CINZA
14            Insere(Q, vi)
15    ui.cor = PRETO
    
```

BFS (G, s) :

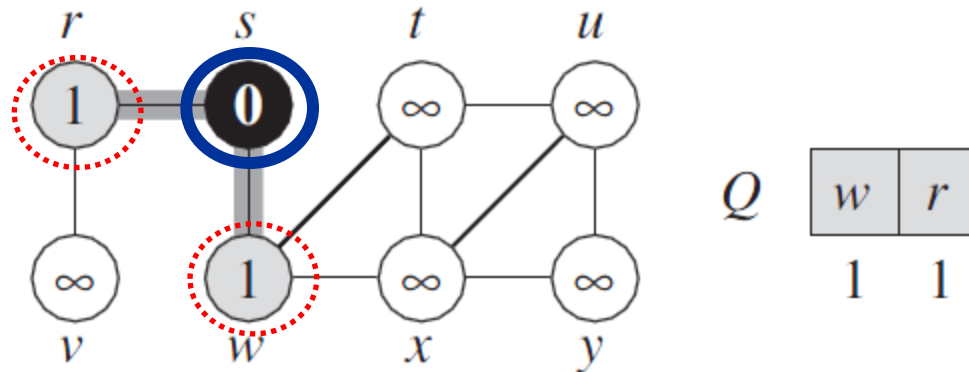


```

8 enquanto  $Q \neq \text{VAZIO}$  faça
9    $u_i = \text{Remove}(Q)$ 
10  para cada  $v_i$  em  $G.\text{Adj}[u_i]$  faça
11    se  $v_i.\text{cor} == \text{BRANCO}$ 
12      entao  $v_i.d = u_i.d + 1$ 
13       $v_i.\text{cor} = \text{CINZA}$ 
14       $\text{Insere}(Q, v_i)$ 
15   $u_i.\text{cor} = \text{PRETO}$ 

```

BFS (G, s) :



8 enquanto $Q \neq \text{VAZIO}$ faça

9 $u_i = \text{Remove}(Q)$

10 para cada v_i em $G.\text{Adj}[u_i]$ faça

11 se $v_i.\text{cor} == \text{BRANCO}$

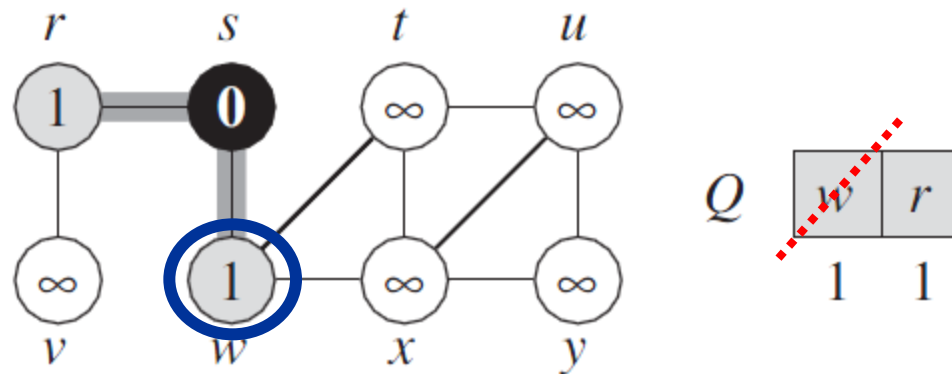
12 entao $v_i.d = u_i.d + 1$

13 $v_i.\text{cor} = \text{CINZA}$

14 $\text{Insere}(Q, v_i)$

➔ 15 $u_i.\text{cor} = \text{PRETO}$

BFS (G, s) :

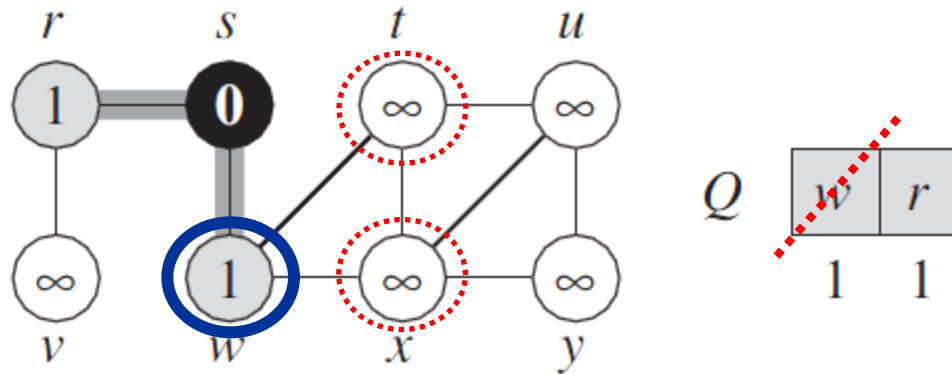


```

8 enquanto  $Q \neq \text{VAZIO}$  faça
9      $u_i = \text{Remove}(Q)$ 
10    para cada  $v_i$  em  $G.\text{Adj}[u_i]$  faça
11        se  $v_i.\text{cor} == \text{BRANCO}$ 
12            entao  $v_i.d = u_i.d + 1$ 
13             $v_i.\text{cor} = \text{CINZA}$ 
14             $\text{Insere}(Q, v_i)$ 
15     $u_i.\text{cor} = \text{PRETO}$ 

```

BFS (G, s) :

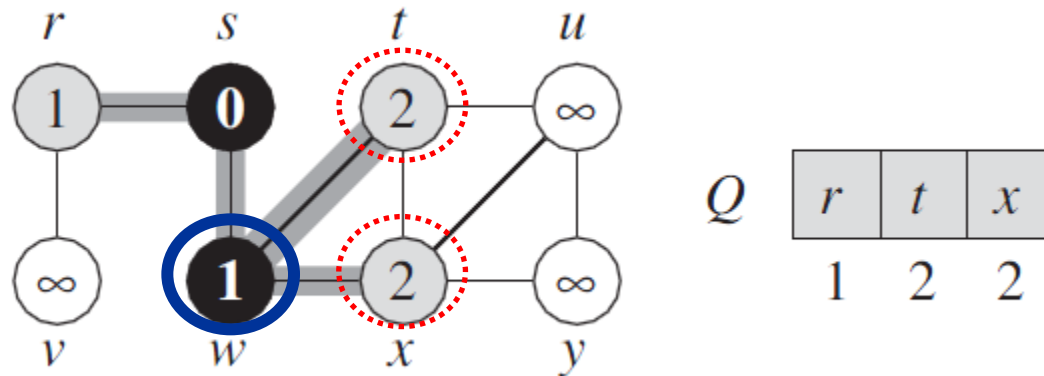


```

8 enquanto  $Q \neq \text{VAZIO}$  faça
9    $u_i = \text{Remove}(Q)$ 
10  para cada  $v_i$  em  $G.\text{Adj}[u_i]$  faça
11    se  $v_i.\text{cor} == \text{BRANCO}$ 
12      entao  $v_i.d = u_i.d + 1$ 
13       $v_i.\text{cor} = \text{CINZA}$ 
14       $\text{Insere}(Q, v_i)$ 
15   $u_i.\text{cor} = \text{PRETO}$ 

```


BFS (G, s) :



8 enquanto $Q \neq \text{VAZIO}$ faça

9 $u_i = \text{Remove}(Q)$

10 para cada v_i em $G.\text{Adj}[u_i]$ faça

11 se $v_i.\text{cor} == \text{BRANCO}$

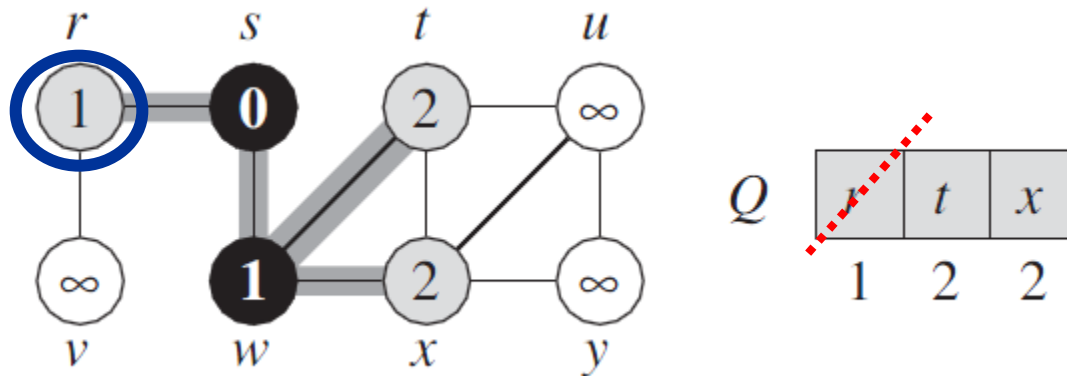
12 entao $v_i.d = u_i.d + 1$

13 $v_i.\text{cor} = \text{CINZA}$

14 $\text{Insere}(Q, v_i)$

➔ 15 $u_i.\text{cor} = \text{PRETO}$

BFS (G, s) :

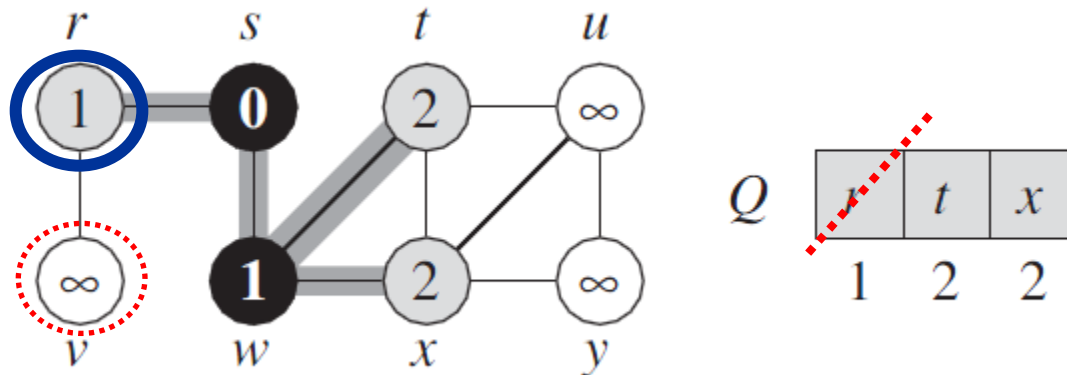


```

8 enquanto  $Q \neq \text{VAZIO}$  faça
9      $u_i = \text{Remove}(Q)$ 
10    para cada  $v_i$  em  $G.\text{Adj}[u_i]$  faça
11        se  $v_i.\text{cor} == \text{BRANCO}$ 
12            entao  $v_i.d = u_i.d + 1$ 
13             $v_i.\text{cor} = \text{CINZA}$ 
14             $\text{Insere}(Q, v_i)$ 
15     $u_i.\text{cor} = \text{PRETO}$ 

```

BFS (G, s) :

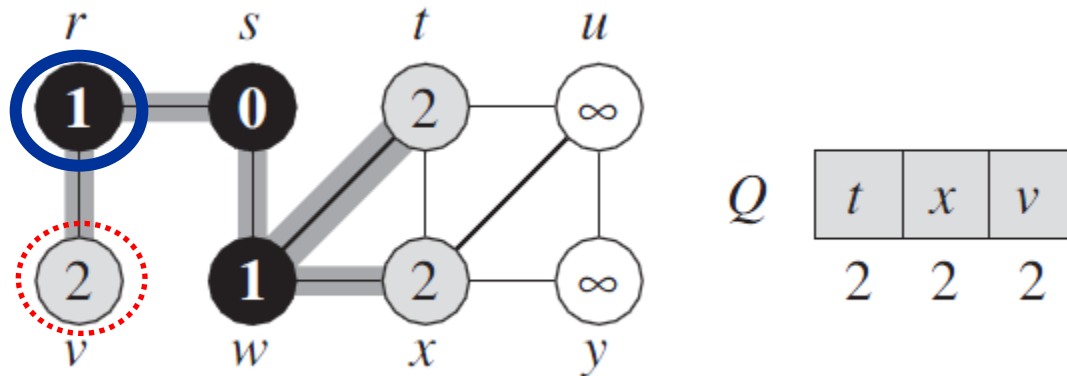


```

8 enquanto  $Q \neq \text{VAZIO}$  faça
9    $u_i = \text{Remove}(Q)$ 
10  para cada  $v_i$  em  $G.\text{Adj}[u_i]$  faça
11    se  $v_i.\text{cor} == \text{BRANCO}$ 
12      entao  $v_i.d = u_i.d + 1$ 
13       $v_i.\text{cor} = \text{CINZA}$ 
14       $\text{Insere}(Q, v_i)$ 
15   $u_i.\text{cor} = \text{PRETO}$ 

```

BFS (G, s) :



8 enquanto $Q \neq \text{VAZIO}$ faça

9 $u_i = \text{Remove}(Q)$

10 para cada v_i em $G.\text{Adj}[u_i]$ faça

11 se $v_i.\text{cor} == \text{BRANCO}$

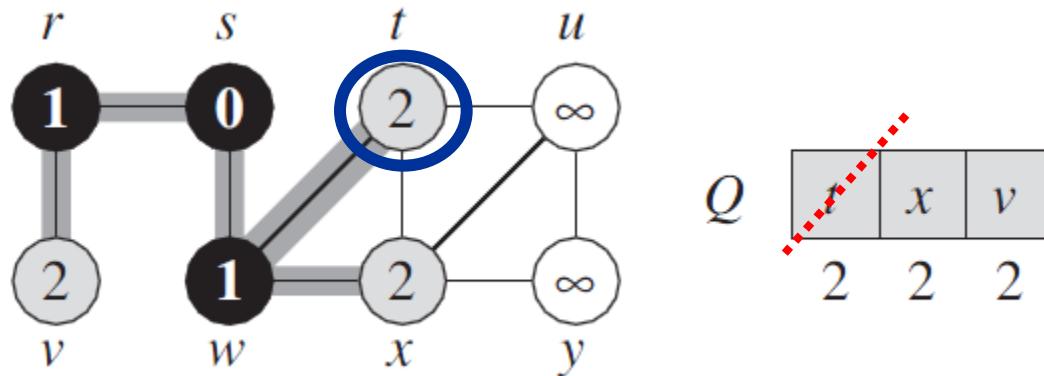
12 entao $v_i.d = u_i.d + 1$

13 $v_i.\text{cor} = \text{CINZA}$

14 $\text{Insere}(Q, v_i)$

15 $u_i.\text{cor} = \text{PRETO}$

BFS (G, s) :

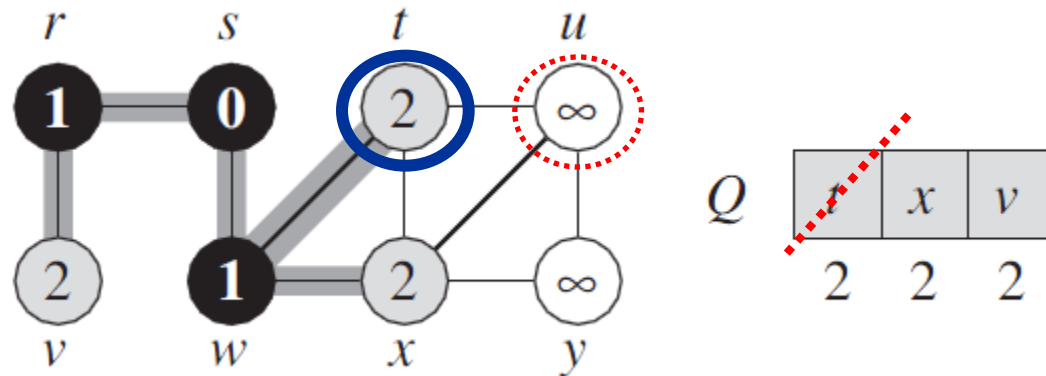


```

8 enquanto Q != VAZIO faça
9      $u_i = \text{Remove}(Q)$ 
10    para cada  $v_i$  em  $G.\text{Adj}[u_i]$  faça
11        se  $v_i.\text{cor} == \text{BRANCO}$ 
12            entao  $v_i.d = u_i.d + 1$ 
13             $v_i.\text{cor} = \text{CINZA}$ 
14             $\text{Insere}(Q, v_i)$ 
15     $u_i.\text{cor} = \text{PRETO}$ 

```

BFS (G, s) :

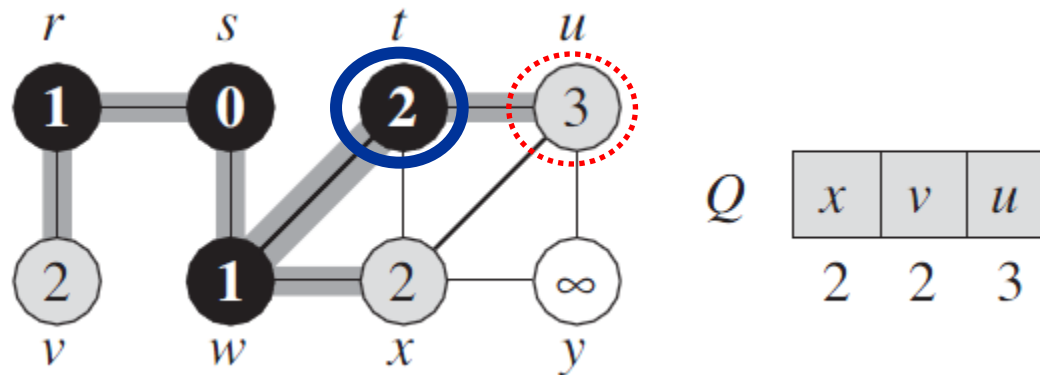


```

8 enquanto  $Q \neq \text{VAZIO}$  faça
9    $u_i = \text{Remove}(Q)$ 
10  para cada  $v_i$  em  $G.\text{Adj}[u_i]$  faça
11    se  $v_i.\text{cor} == \text{BRANCO}$ 
12      entao  $v_i.d = u_i.d + 1$ 
13       $v_i.\text{cor} = \text{CINZA}$ 
14       $\text{Insere}(Q, v_i)$ 
15   $u_i.\text{cor} = \text{PRETO}$ 

```

BFS (G, s) :



8 enquanto $Q \neq \text{VAZIO}$ faça

9 $u_i = \text{Remove}(Q)$

10 para cada v_i em $G.\text{Adj}[u_i]$ faça

11 se $v_i.\text{cor} == \text{BRANCO}$

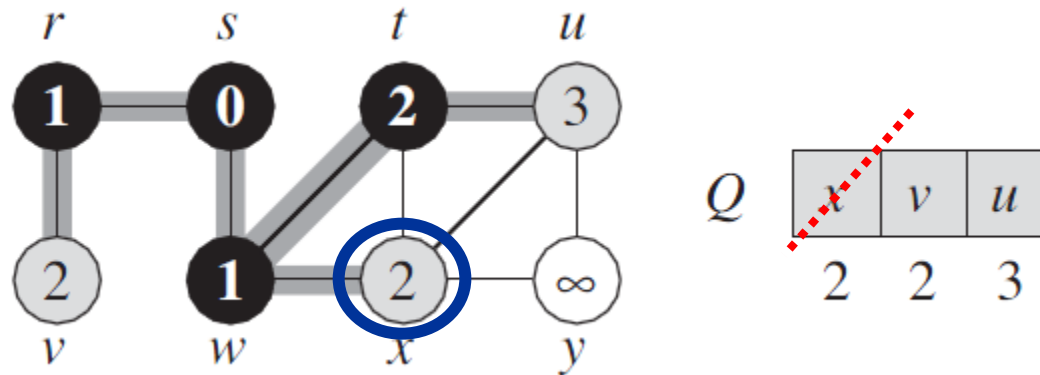
12 entao $v_i.d = u_i.d + 1$

13 $v_i.\text{cor} = \text{CINZA}$


14 $\text{Insere}(Q, v_i)$

➔ 15 $u_i.\text{cor} = \text{PRETO}$

BFS (G, s) :

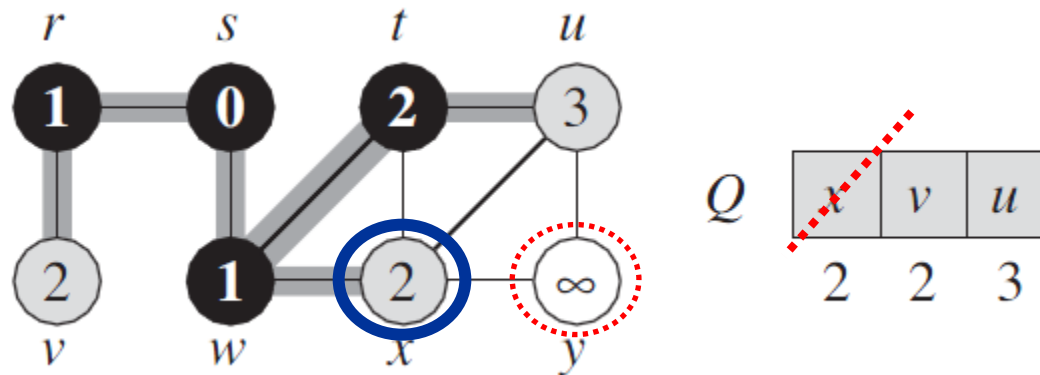


```

8 enquanto Q != VAZIO faça
9       $u_i = \text{Remove}(Q)$ 
10     para cada  $v_i$  em  $G.\text{Adj}[u_i]$  faça
11         se  $v_i.\text{cor} == \text{BRANCO}$ 
12             entao  $v_i.d = u_i.d + 1$ 
13              $v_i.\text{cor} = \text{CINZA}$ 
14              $\text{Insere}(Q, v_i)$ 
15      $u_i.\text{cor} = \text{PRETO}$ 

```


BFS (G, s) :

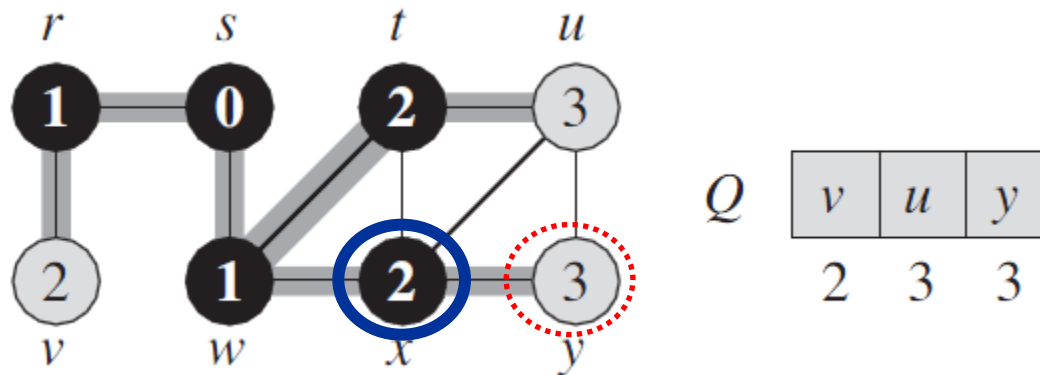


```

8 enquanto Q != VAZIO faça
9    $u_i = \text{Remove}(Q)$ 
10  para cada  $v_i$  em  $G.\text{Adj}[u_i]$  faça
11    se  $v_i.\text{cor} == \text{BRANCO}$ 
12      entao  $v_i.d = u_i.d + 1$ 
13       $v_i.\text{cor} = \text{CINZA}$ 
14       $\text{Insere}(Q, v_i)$ 
15   $u_i.\text{cor} = \text{PRETO}$ 

```

BFS (G, s) :



8 enquanto *Q* != VAZIO faça

9 *u_i* = Remove(*Q*)

10 para cada *v_i* em *G*.Adj[*u_i*] faça

11 se *v_i*.cor == BRANCO

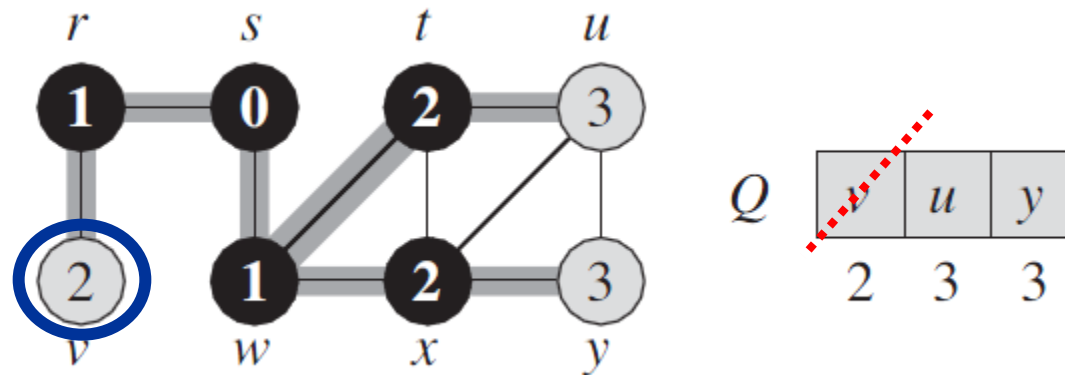
12 entao *v_i*.d = *u_i*.d + 1

13 *v_i*.cor = CINZA

14 Insere(*Q*, *v_i*)

→ 15 *u_i*.cor = PRETO

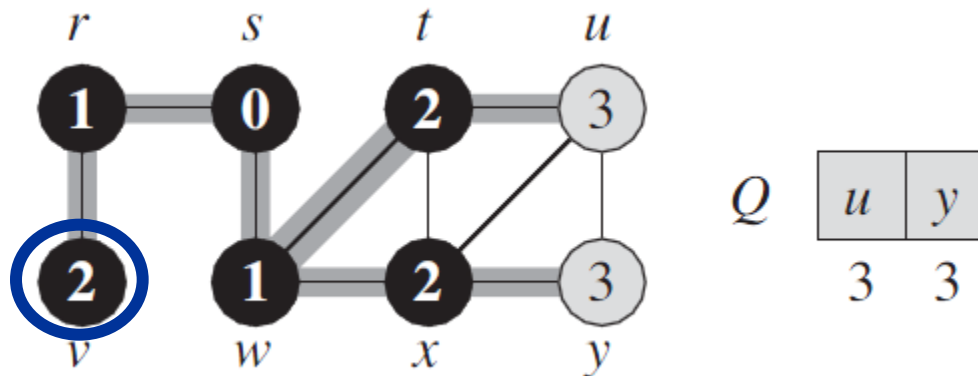
BFS (G, s) :



```

8 enquanto Q != VAZIO faça
9     ui = Remove(Q)
10    para cada vi em G.Adj[ui] faça
11        se vi.cor == BRANCO
12            entao vi.d = ui.d + 1
13            vi.cor = CINZA
14            Insere(Q, vi)
15    ui.cor = PRETO
  
```

BFS (G, s) :



8 enquanto *Q* != VAZIO faça

9 *u_i* = Remove(*Q*)

10 para cada *v_i* em *G*.Adj[*u_i*] faça

11 se *v_i*.cor == BRANCO

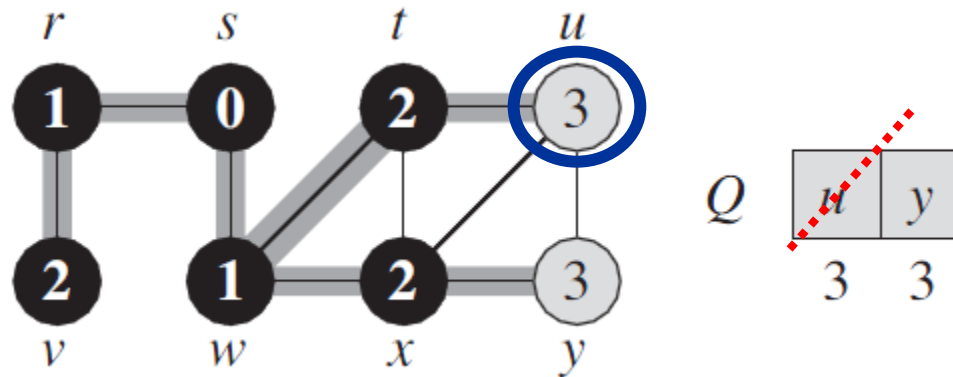
12 entao *v_i*.d = *u_i*.d + 1

13 *v_i*.cor = CINZA

14 Insere(*Q*, *v_i*)

→ 15 *u_i*.cor = PRETO

BFS (G, s) :

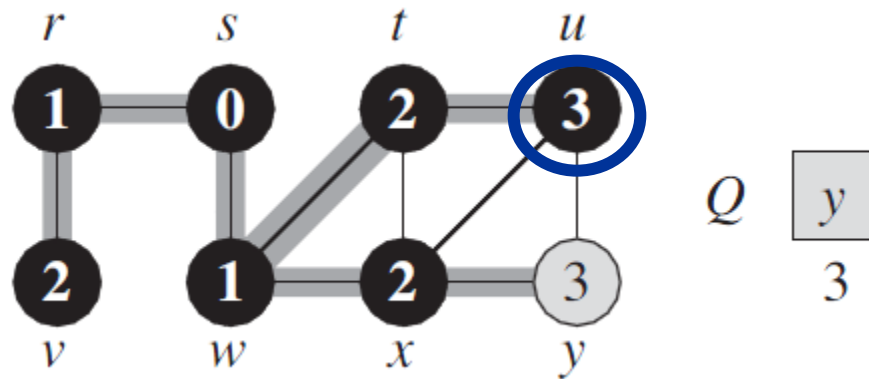


```

8 enquanto Q != VAZIO faça
9     ui = Remove(Q)
10    para cada vi em G.Adj[ui] faça
11        se vi.cor == BRANCO
12            entao vi.d = ui.d + 1
13            vi.cor = CINZA
14            Insere(Q, vi)
15    ui.cor = PRETO

```

BFS (G, s) :



8 enquanto $Q \neq \text{VAZIO}$ faça

9 $u_i = \text{Remove}(Q)$

10 para cada v_i em $G.\text{Adj}[u_i]$ faça

11 se $v_i.\text{cor} == \text{BRANCO}$

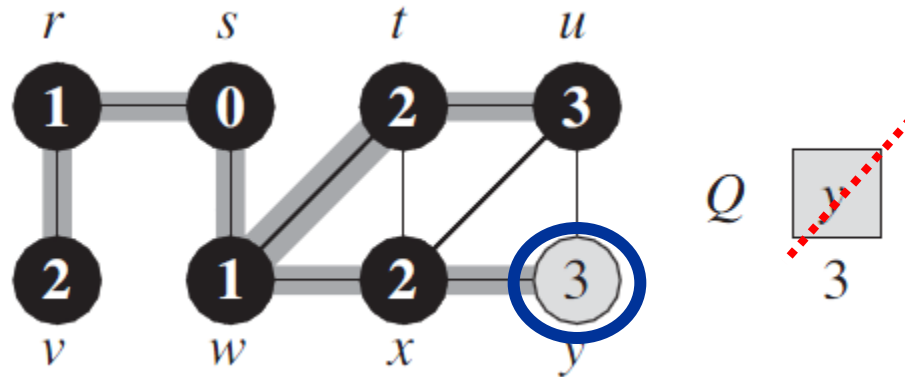
12 entao $v_i.d = u_i.d + 1$

13 $v_i.\text{cor} = \text{CINZA}$

14 $\text{Insere}(Q, v_i)$

→ 15 $u_i.\text{cor} = \text{PRETO}$

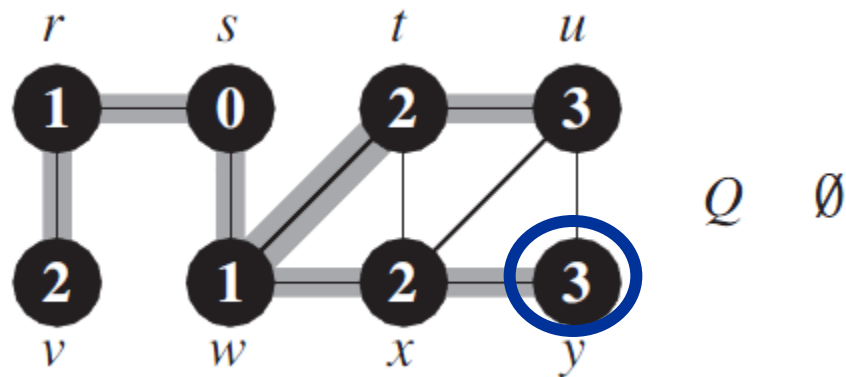
BFS (G, s) :



```

8 enquanto Q != VAZIO faça
9     ui = Remove(Q)
10    para cada vi em G.Adj[ui] faça
11        se vi.cor == BRANCO
12            entao vi.d = ui.d + 1
13            vi.cor = CINZA
14            Insere(Q, vi)
15    ui.cor = PRETO
  
```

BFS (G, s) :



```
8 enquanto Q != VAZIO faça
```

```
9    $u_i = \text{Remove}(Q)$ 
```

```
10  para cada  $v_i$  em  $G.\text{Adj}[u_i]$  faça
```

```
11    se  $v_i.\text{cor} == \text{BRANCO}$ 
```

```
12      entao  $v_i.d = u_i.d + 1$ 
```

```
13         $v_i.\text{cor} = \text{CINZA}$ 
```

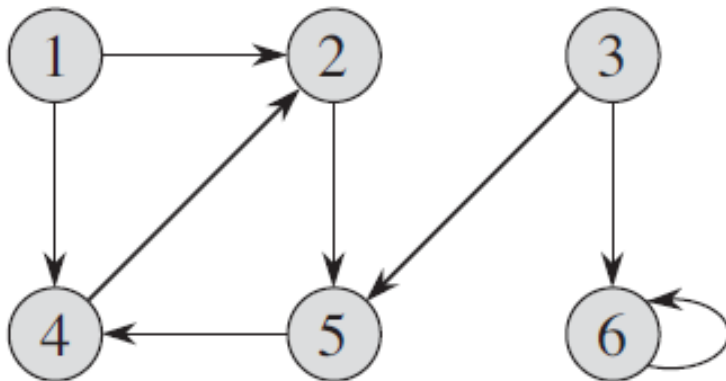
```
14         $\text{Insere}(Q, v_i)$ 
```

```
15   $u_i.\text{cor} = \text{PRETO}$ 
```

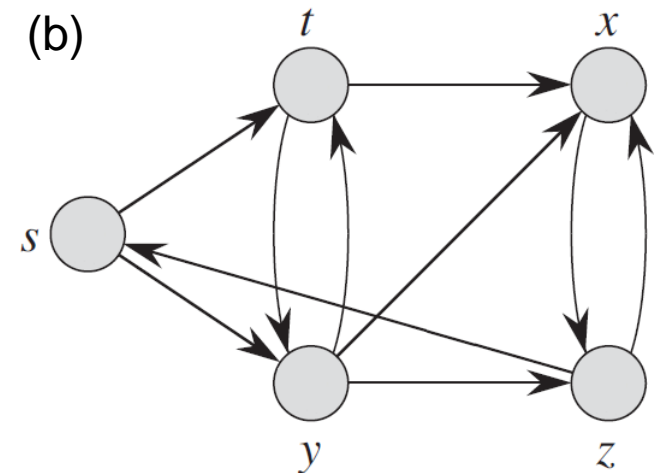

Exercícios

- Para cada grafo, escolha um **vértice inicial** e execute uma **busca em largura**.

(a)



(b)



Consumo de tempo

- BFS?

Notação O (“ó”)

- $O(1)$: constante
- $O(\log n)$: log de n
- $O(n)$: linear
- $O(n \log n)$: n log de n
- $O(n^2)$: quadrático
- $O(n^3)$: cúbico
- $O(n^k)$: polinomial
- $O(2^n)$: exponencial
- $O(k^n)$, $O(n!)$, $O(n^n)$: exponencial

BFS (**G**, **s**) :

Consumo
de tempo:

1	para cada vértice v_i em $G.V$ faça	???
2	$v_i.d = \text{INFINITO}$???
3	$v_i.cor = \text{BRANCO}$???
4	$s.d = 0$	$O(1)$
5	$s.cor = \text{CINZA}$	$O(1)$
6	$Q = \text{VAZIO}$	$O(1)$
7	Inserir(Q, s)	$O(1)$
8	enquanto $Q \neq \text{VAZIO}$ faça	???
9	$u_i = \text{Remove}(Q)$???
10	para cada v_i em $G.Adj[u_i]$ faça	???
11	se $v_i.cor == \text{BRANCO}$???
12	então $v_i.d = u_i.d + 1$???
13	$v_i.cor = \text{CINZA}$???
14	Inserir(Q, v_i)	???
15	$u_i.cor = \text{PRETO}$???

Algumas operações com Fila:

- $\text{Insere}(Q, x)$ consome $O(1)$ unidades de tempo.
 - insere elemento x no **final** da fila Q
- $\text{Remove}(Q)$ consome $O(1)$ unidades de tempo.
 - remove e devolve o elemento do **início** da fila Q
- $\text{FilaVazia}(Q)$ consome $O(1)$ unidades de tempo.
 - devolve verdadeiro se a fila estiver vazia, falso caso contrário.
- $\text{InicializarFilaVazia}(Q)$
consome $O(1)$ unidades de tempo.



BFS (**G**, **s**) :

Consumo
de tempo:

1	para cada vértice v_i em $G.V$ faça	$O(n)$
2	$v_i.d = \text{INFINITO}$	$O(n) * O(1)$
3	$v_i.cor = \text{BRANCO}$	$O(n) * O(1)$
4	$s.d = 0$	$O(1)$
5	$s.cor = \text{CINZA}$	$O(1)$
6	Q = VAZIO	$O(1)$
7	Inserere(Q , s)	$O(1)$
<hr/>		
8	enquanto $Q \neq \text{VAZIO}$ faça	???
9	$u_i = \text{Remove}(Q)$???
10	para cada v_i em $G.Adj[u_i]$ faça	???
11	se $v_i.cor == \text{BRANCO}$???
12	entao $v_i.d = u_i.d + 1$???
13	$v_i.cor = \text{CINZA}$???
14	Inserere(Q , v_i)	???
15	$u_i.cor = \text{PRETO}$???

BFS (**G**, **s**) :

Consumo
de tempo:

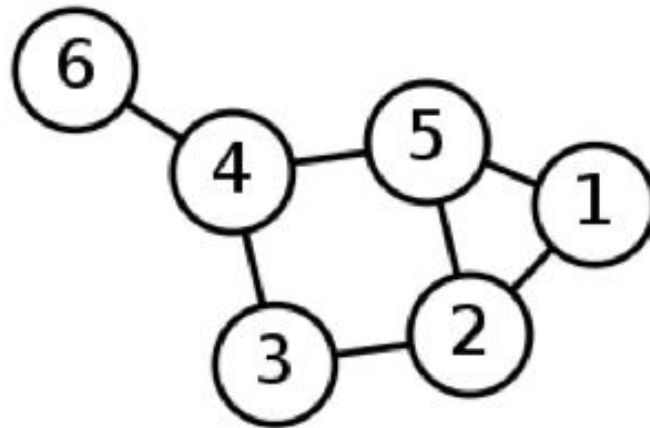
1	para cada vértice v_i em $G.V$ faça	$O(n)$
2	$v_i.d = \text{INFINITO}$	$O(n)$
3	$v_i.cor = \text{BRANCO}$	$O(n)$
4	$s.d = 0$	$O(1)$
5	$s.cor = \text{CINZA}$	$O(1)$
6	$Q = \text{VAZIO}$	$O(1)$
7	Inserere(Q, s)	$O(1)$
<hr/>		
8	enquanto $Q \neq \text{VAZIO}$ faça	$O(n)$
9	$u_i = \text{Remove}(Q)$	$O(n) * O(1)$
10	para cada v_i em $G.Adj[u_i]$ faça	???
11	se $v_i.cor == \text{BRANCO}$???
12	entao $v_i.d = u_i.d + 1$???
13	$v_i.cor = \text{CINZA}$???
14	Inserere(Q, v_i)	???
15	$u_i.cor = \text{PRETO}$	$O(n) * O(1)$

BFS (**G**, **s**) :

Consumo
de tempo:

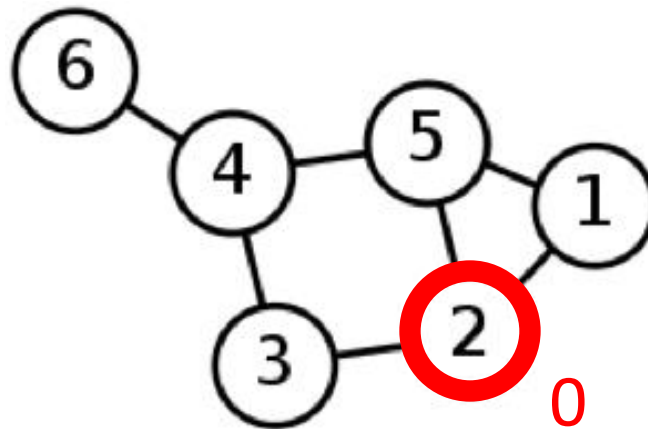
1	para cada vértice v_i em $G.V$ faça	$O(n)$
2	$v_i.d = \text{INFINITO}$	$O(n)$
3	$v_i.cor = \text{BRANCO}$	$O(n)$
4	$s.d = 0$	$O(1)$
5	$s.cor = \text{CINZA}$	$O(1)$
6	$Q = \text{VAZIO}$	$O(1)$
7	Inserere(Q, s)	$O(1)$
8	enquanto $Q \neq \text{VAZIO}$ faça	$O(n)$
9	$u_i = \text{Remove}(Q)$	$O(n)$
10	para cada v_i em $G.Adj[u_i]$ faça	$O(m)$
11	se $v_i.cor == \text{BRANCO}$???
12	entao $v_i.d = u_i.d + 1$???
13	$v_i.cor = \text{CINZA}$???
14	Inserere(Q, v_i)	???
15	$u_i.cor = \text{PRETO}$	$O(n)$

Propagação de “fogo”



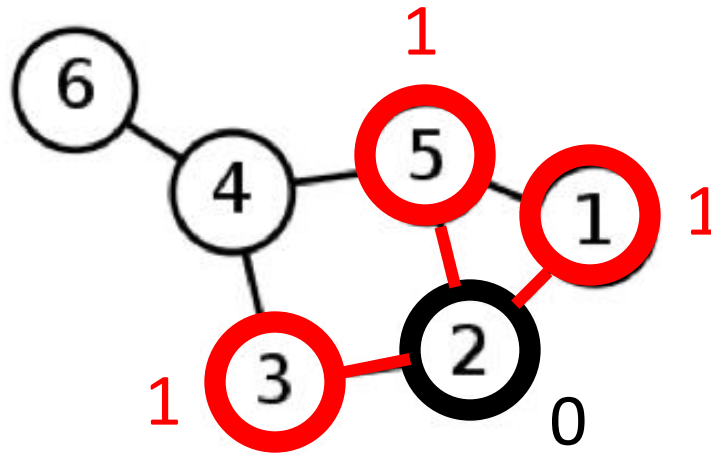
Propagação de “fogo”

Vértice inicial.



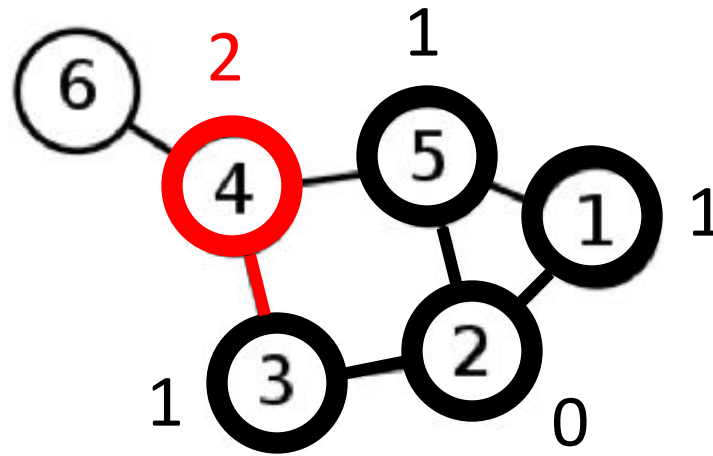
Propagação de “fogo”

Fogo espalha para vizinhos.



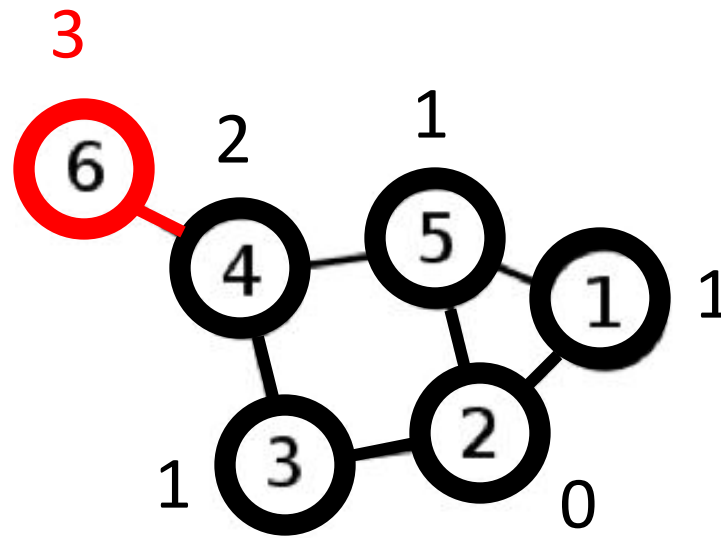
Propagação de “fogo”

Fogo espalha para vizinhos.



Propagação de “fogo”

Fogo espalha para vizinhos.



BFS (**G**, **s**) :

Consumo
de tempo:

1	para cada vértice v_i em $G.V$ faça	$O(n)$
2	$v_i.d = \text{INFINITO}$	$O(n)$
3	$v_i.cor = \text{BRANCO}$	$O(n)$
4	$s.d = 0$	$O(1)$
5	$s.cor = \text{CINZA}$	$O(1)$
6	$Q = \text{VAZIO}$	$O(1)$
7	Inserere(Q, s)	$O(1)$
8	enquanto $Q \neq \text{VAZIO}$ faça	$O(n)$
9	$u_i = \text{Remove}(Q)$	$O(n)$
10	para cada v_i em $G.Adj[u_i]$ faça	$O(m)$
11	se $v_i.cor == \text{BRANCO}$	$O(m) * O(1)$
12	entao $v_i.d = u_i.d + 1$	$O(m) * O(1)$
13	$v_i.cor = \text{CINZA}$	$O(m) * O(1)$
14	Inserere(Q, v_i)	$O(m) * O(1)$
15	$u_i.cor = \text{PRETO}$	$O(n)$

BFS (**G**, **s**) :

Consumo
de tempo:

1	para cada vértice v_i em $G.V$ faça	$O(n)$
2	$v_i.d = \text{INFINITO}$	$O(n)$
3	$v_i.cor = \text{BRANCO}$	$O(n)$
4	$s.d = 0$	$O(1)$
5	$s.cor = \text{CINZA}$	$O(1)$
6	$Q = \text{VAZIO}$	$O(1)$
7	Inserir(Q, s)	$O(1)$
<hr/>		
8	enquanto $Q \neq \text{VAZIO}$ faça	$O(n)$
9	$u_i = \text{Remove}(Q)$	$O(n)$
10	para cada v_i em $G.Adj[u_i]$ faça	$O(m)$
11	se $v_i.cor == \text{BRANCO}$	$O(m)$
12	então $v_i.d = u_i.d + 1$	$O(m)$
13	$v_i.cor = \text{CINZA}$	$O(m)$
14	Inserir(Q, v_i)	$O(m)$
15	$u_i.cor = \text{PRETO}$	$O(n)$

Total: $T(n,m) = 6*O(n) + 5*O(m) + 4*O(1)$

BFS (**G**, **s**) :

Consumo
de tempo:

1	para cada vértice v_i em $G.V$ faça	$O(n)$
2	$v_i.d = \text{INFINITO}$	$O(n)$
3	$v_i.cor = \text{BRANCO}$	$O(n)$
4	$s.d = 0$	$O(1)$
5	$s.cor = \text{CINZA}$	$O(1)$
6	$Q = \text{VAZIO}$	$O(1)$
7	Inserere(Q, s)	$O(1)$
<hr/>		
8	enquanto $Q \neq \text{VAZIO}$ faça	$O(n)$
9	$u_i = \text{Remove}(Q)$	$O(n)$
10	para cada v_i em $G.Adj[u_i]$ faça	$O(m)$
11	se $v_i.cor == \text{BRANCO}$	$O(m)$
12	entao $v_i.d = u_i.d + 1$	$O(m)$
13	$v_i.cor = \text{CINZA}$	$O(m)$
14	Inserere(Q, v_i)	$O(m)$
15	$u_i.cor = \text{PRETO}$	$O(n)$

Total: $T(n, m) = 6 * O(n) + 5 * O(m) + 4 * O(1)$
 $= O(n + m)$

Exercício Programa

- 02-bfs.py