

# Teoria dos Grafos



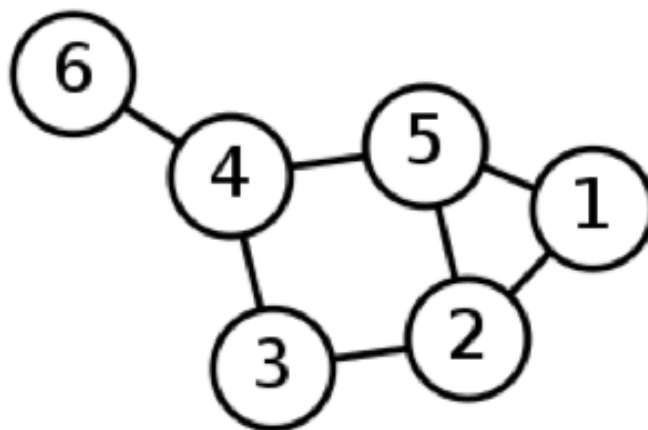
Universidade Federal do ABC

# Introdução

- Algoritmos de busca
  - Maneiras sistemáticas de **visitar** vértices
- Principais algoritmos
  - ➔ ■ BFS: **largura**
  - DFS: **profundidade**

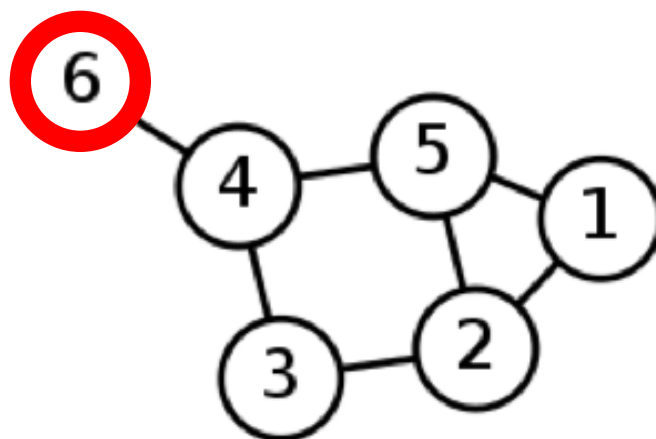
## BFS: busca em largura

- **Ideia:** Dado um **vértice inicial  $s$** , a **busca em largura** é como uma propagação de **fogo**.



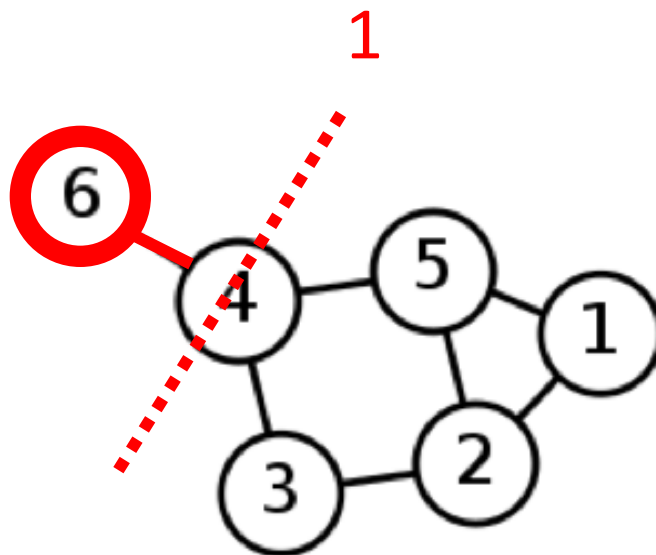
## BFS: busca em largura

$s = 6$



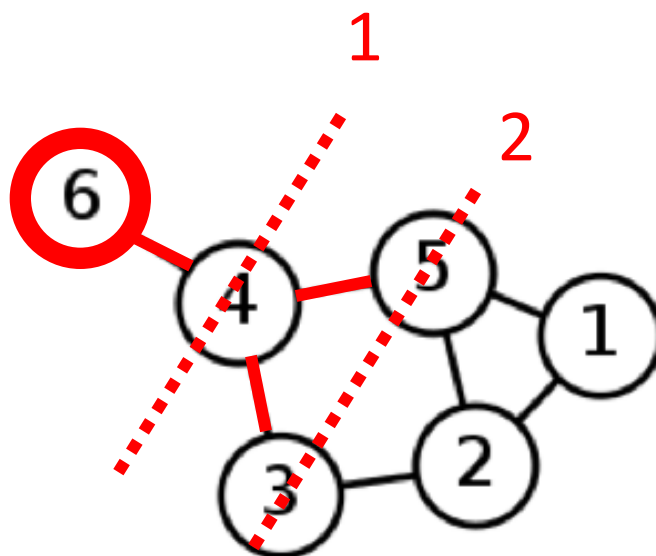
## BFS: busca em largura

$s = 6$



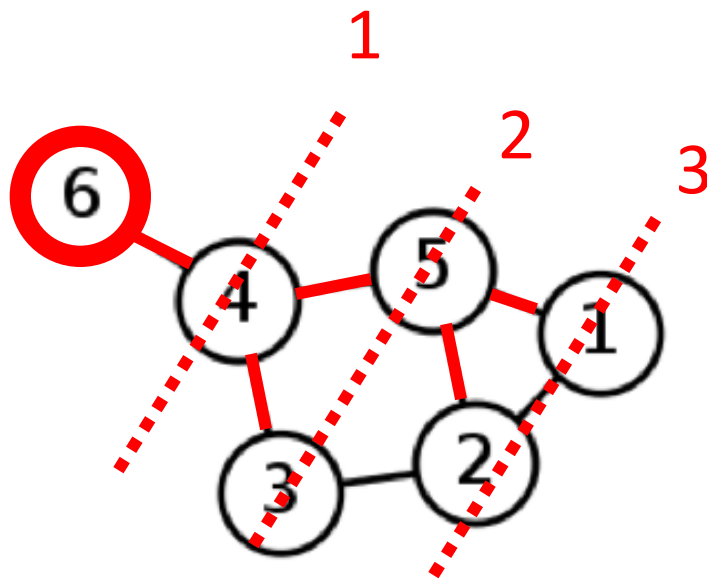
## BFS: busca em largura

$s = 6$

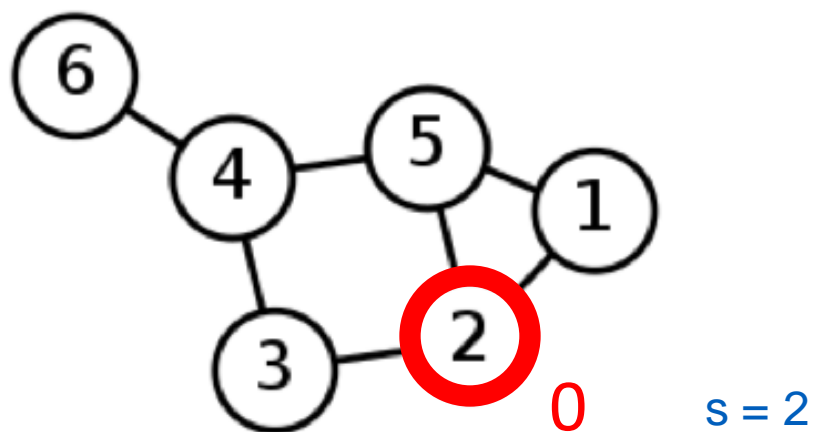


## BFS: busca em largura

$s = 6$

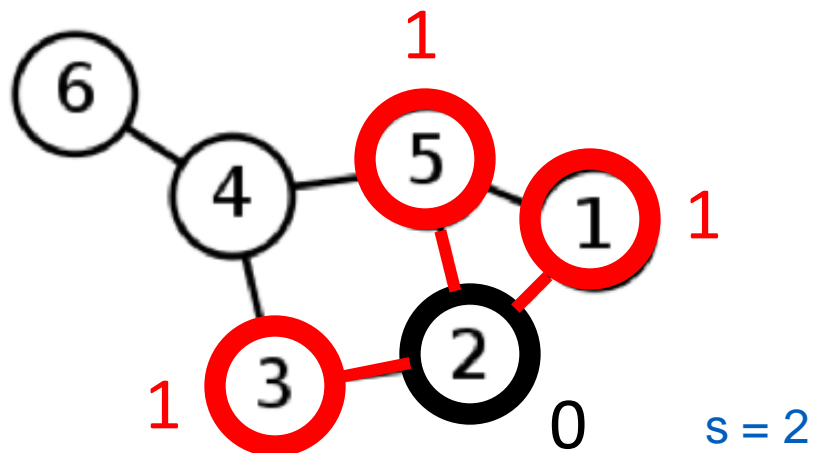


## BFS: busca em largura

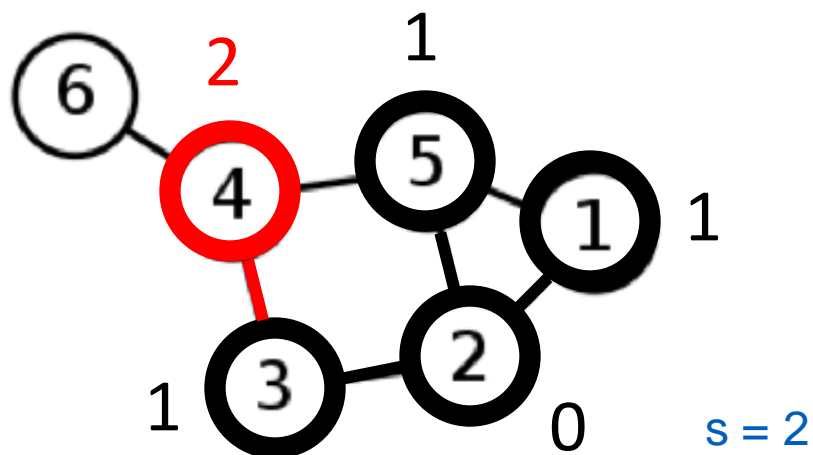




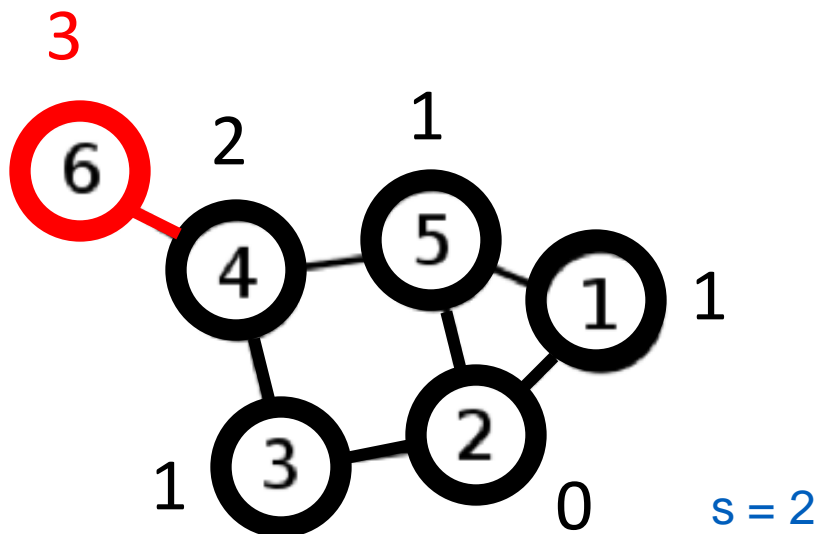
## BFS: busca em largura



## BFS: busca em largura

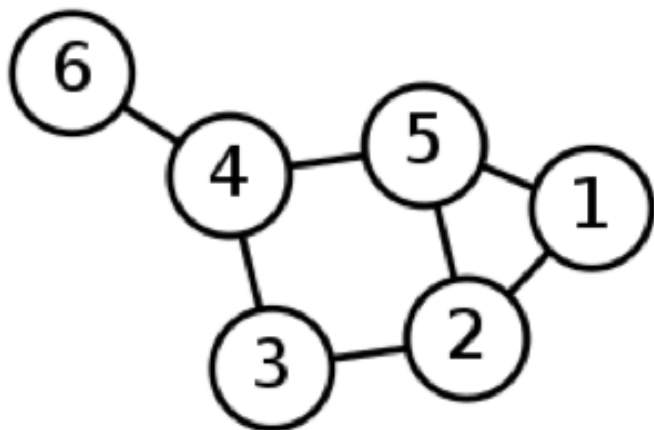


## BFS: busca em largura



### Exemplo

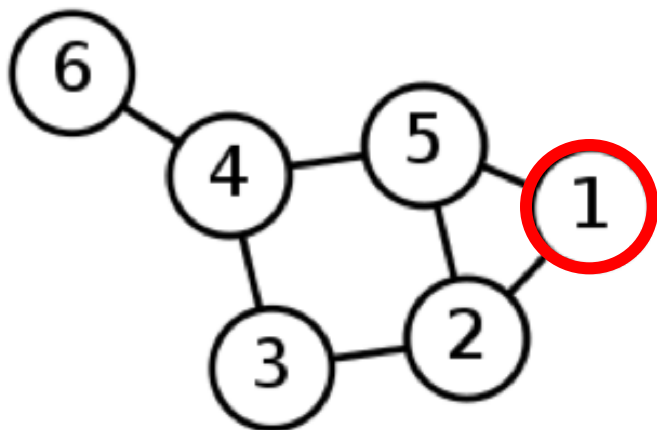
- Para o grafo abaixo, calcular a **matriz de distâncias**.



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

### Exemplo

- Para o grafo abaixo, calcular a **matriz de distâncias**.

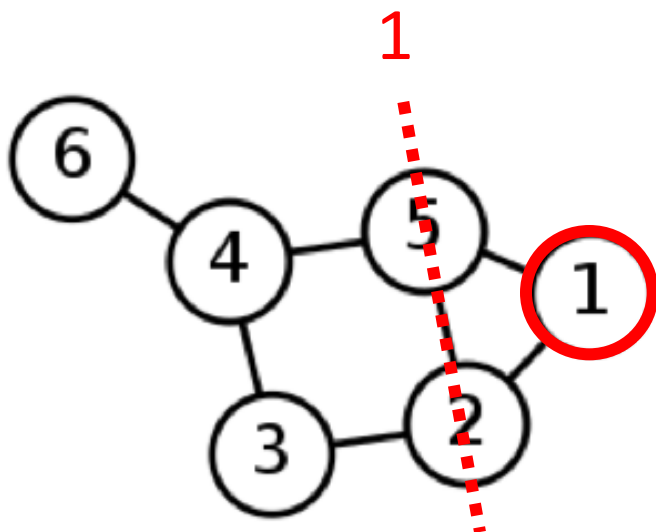


➔

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

### Exemplo

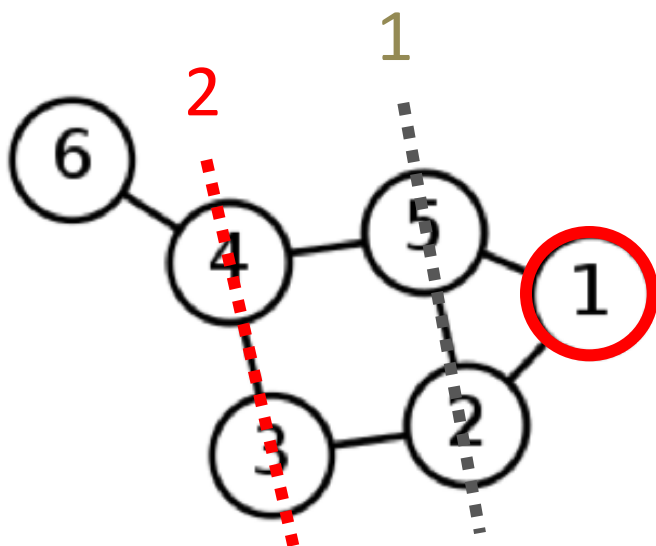
- Para o grafo abaixo, calcular a **matriz de distâncias**.



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

### Exemplo

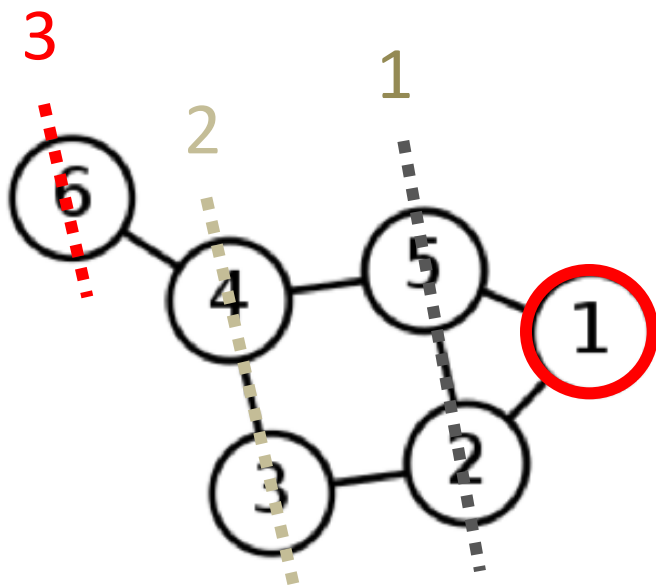
- Para o grafo abaixo, calcular a **matriz de distâncias**.



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

### Exemplo

- Para o grafo abaixo, calcular a **matriz de distâncias**.

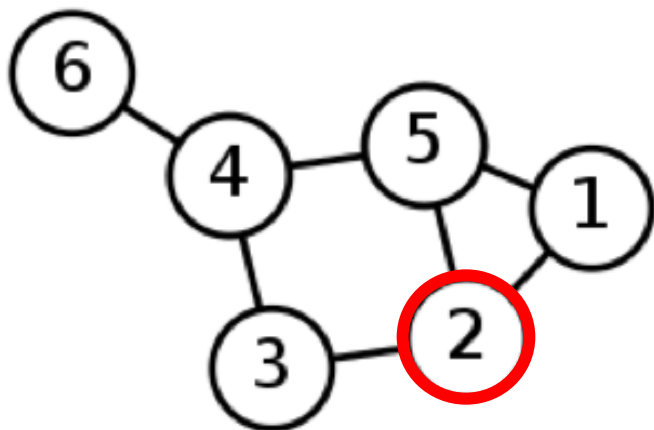


	1	2	3	4	5	6
1	0	1	2	2	1	3
2						
3						
4						
5						
6						



### Exemplo

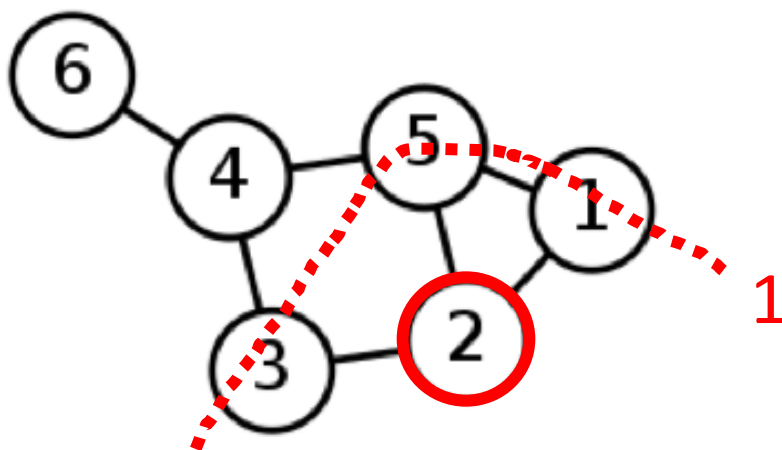
- Para o grafo abaixo, calcular a **matriz de distâncias**.



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

### Exemplo

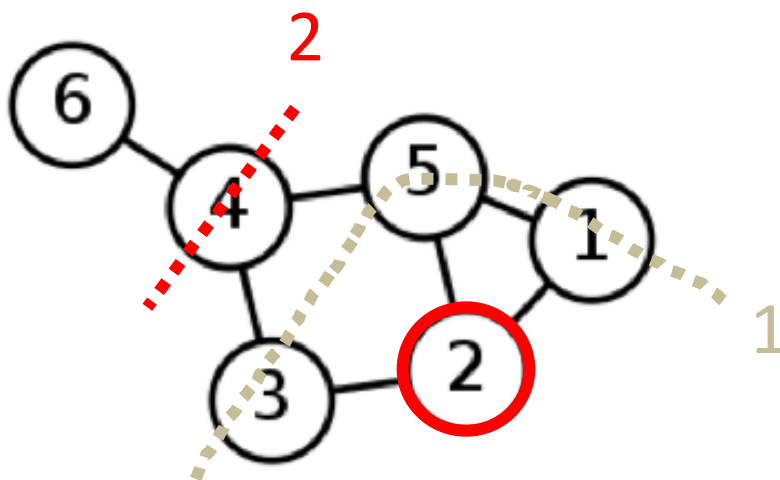
- Para o grafo abaixo, calcular a **matriz de distâncias**.



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

### Exemplo

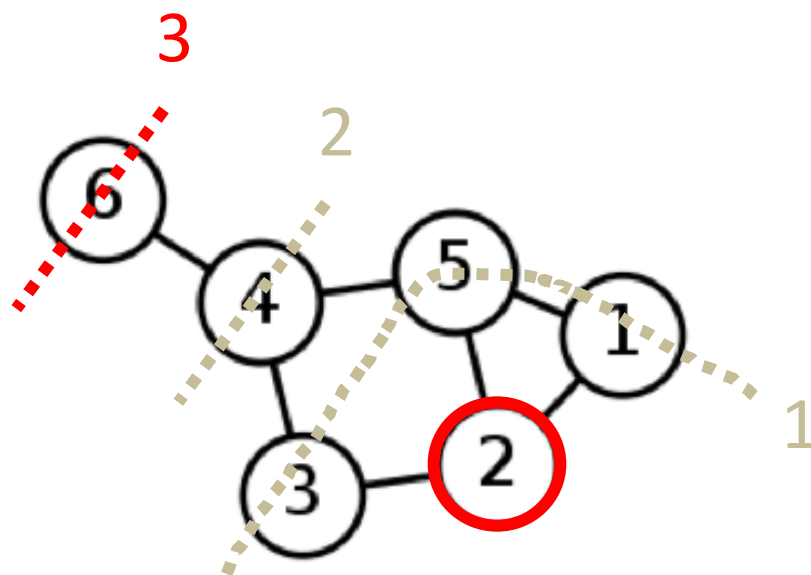
- Para o grafo abaixo, calcular a **matriz de distâncias**.



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

### Exemplo

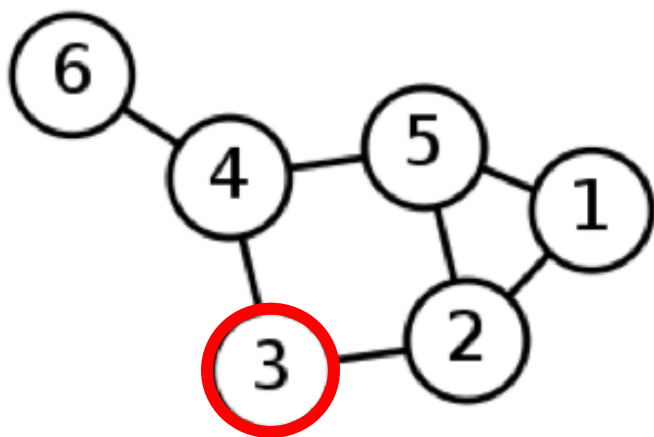
- Para o grafo abaixo, calcular a **matriz de distâncias**.



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

### Exemplo

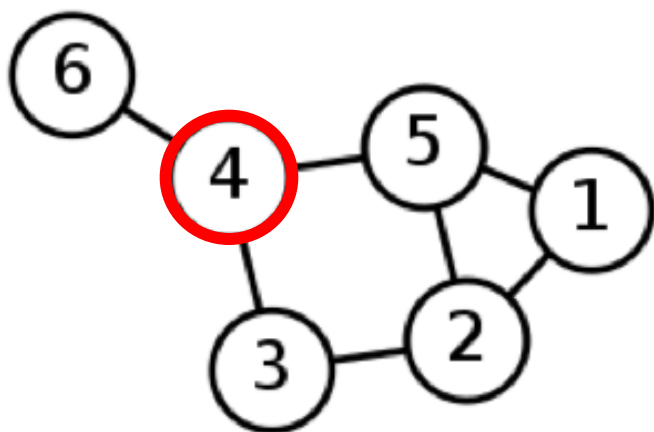
- Para o grafo abaixo, calcular a **matriz de distâncias**.



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

### Exemplo

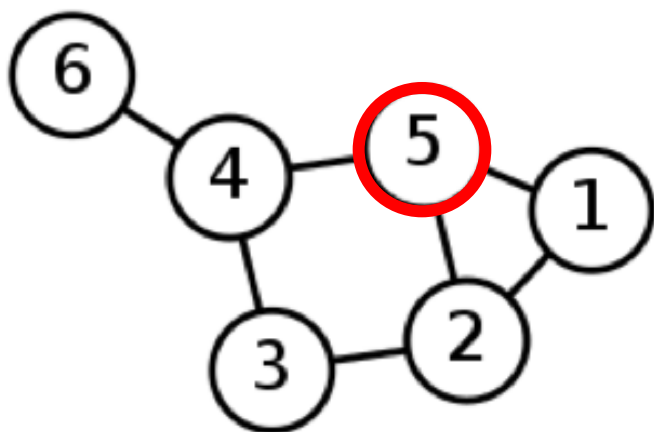
- Para o grafo abaixo, calcular a **matriz de distâncias**.



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

### Exemplo

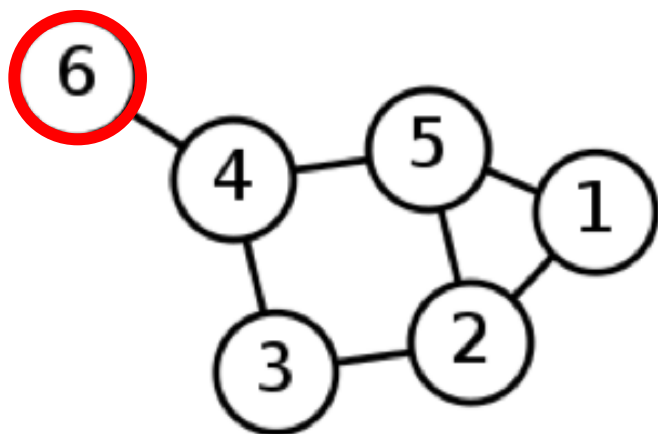
- Para o grafo abaixo, calcular a **matriz de distâncias**.



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

### Exemplo

- Para o grafo abaixo, calcular a **matriz de distâncias**.



→

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



## BFS: busca em largura

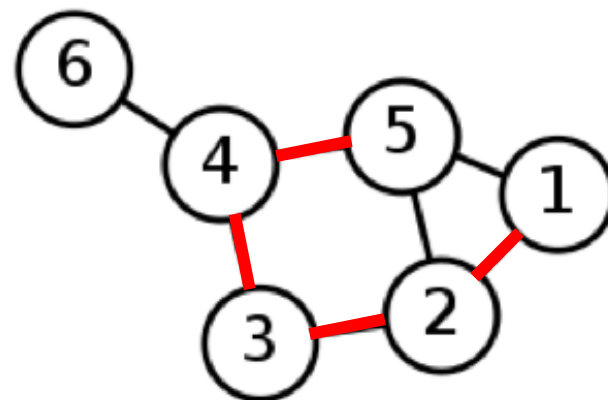
### Tarefa

- Exercícios:
  - Lista 1



## BFS: busca em largura

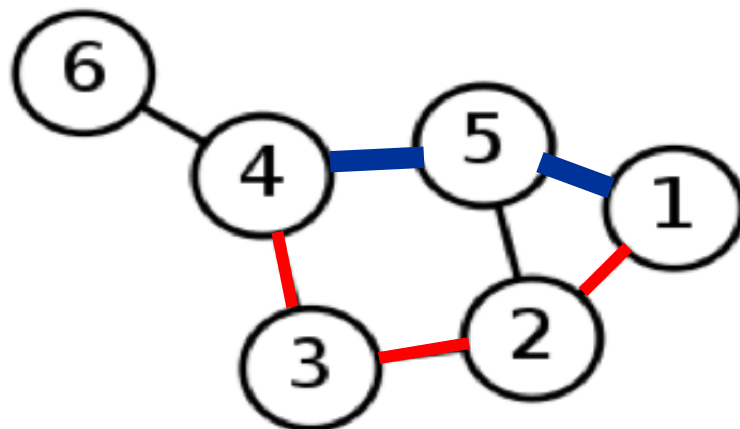
- Um **caminho** de **s** a **t** é uma sequência de vértices  $\langle s, u, v, \dots, t \rangle$  onde cada par de vértices consecutivos é conectado por uma aresta.
- Dizemos que o caminho **começa** em **s** e **termina** em **t**.
- O **comprimento** de um caminho é o total de arestas usadas no caminho.
  - Ex:  $\langle 1, 2, 3, 4, 5 \rangle$  é um caminho que começa em 1 e termina em 5, e tem **comprimento** 4.



## BFS: busca em largura

- Um **caminho** de **s** a **t** é um **caminho mínimo** se ele tem comprimento mínimo.
- Ex:
  - **<1,2,3,4>** é um **caminho** de 1 a 4, de **comprimento 3**.

Caminho mínimo? **Não**, mas **<1,5,4>** é caminho mínimo.



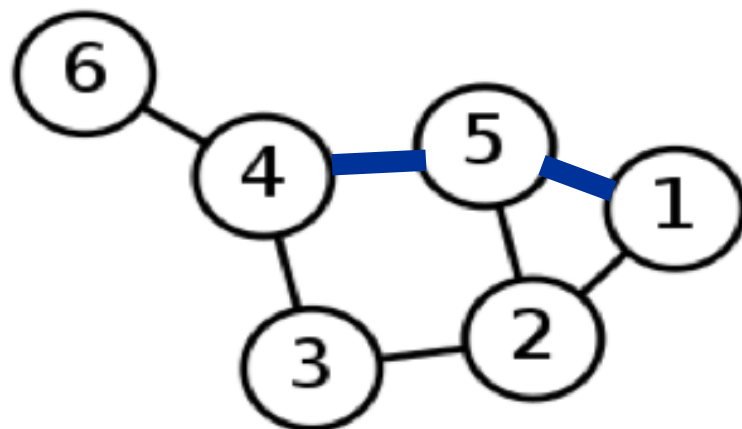
- A **distância** entre dois vértices é o comprimento de um caminho mínimo.

- Ex:

A distância entre 1 e 4 **vale 2**.

- **Notação:**

$$\delta(1,4) = 2$$



BFS: busca em largura

(BFS: Breadth First Search)

- **BFS**( $G, s$ )
  - **Entrada**: um grafo  $G$  e um vértice inicial  $s$
  - **Saída**: **distâncias** em relação ao vértice inicial  $s$
- Atributos para cada vértice  $v \in V$ 
  - $v.d$ : “distância”
  - $v.cor$ 
    - branca (inicial)
    - cinza (visitado)
    - preta (finalizado)



**Fila**: "First-In-First-Out"

### Fila: operações

- $\text{Insere}(Q, x)$ 
  - insere elemento  $x$  no final da fila  $Q$
- $\text{Remove}(Q)$ 
  - remove e devolve o **primeiro** elemento da fila  $Q$



Fila: "First-In-First-Out"

## BFS: busca em largura

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  $Q = \text{VAZIO}$ 
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     $u_i.cor = \text{PRETO}$ 
```

(Inicialização)

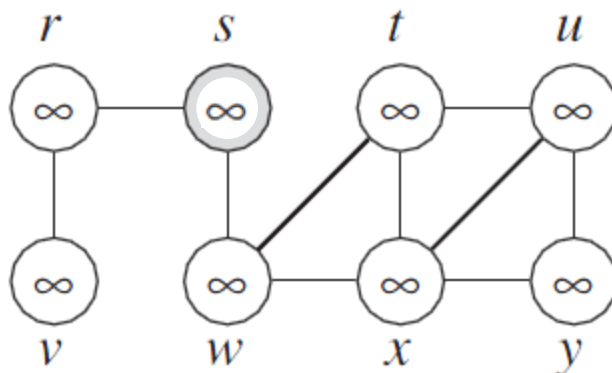
## BFS: busca em largura

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  $Q = \text{VAZIO}$ 
7 Insere( $Q, s$ )
    
```

(Inicialização)





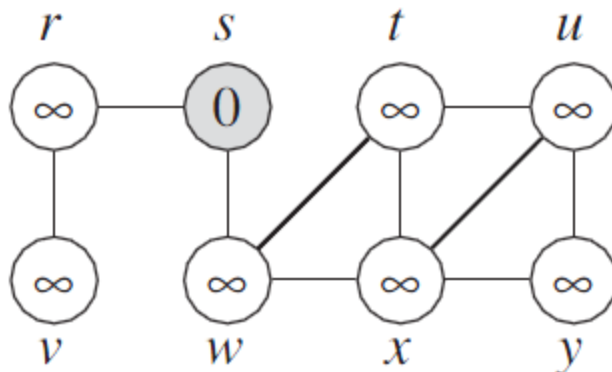
## BFS: busca em largura

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  $Q = \text{VAZIO}$ 
7 Insere( $Q, s$ )
    
```

(Inicialização)



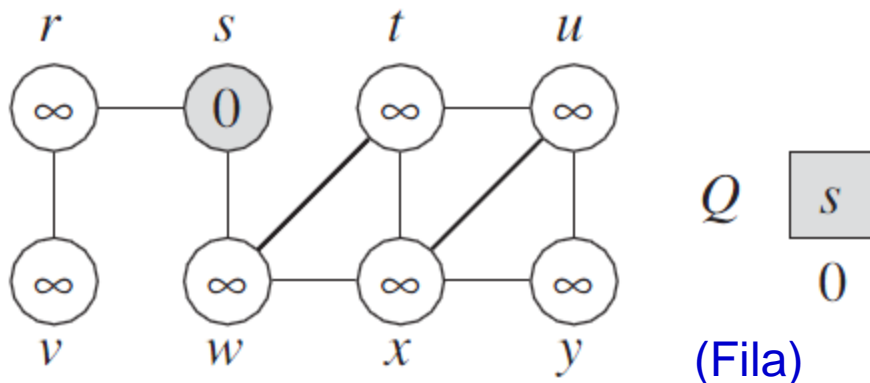
## BFS: busca em largura

BFS( $G, s$ ):

```

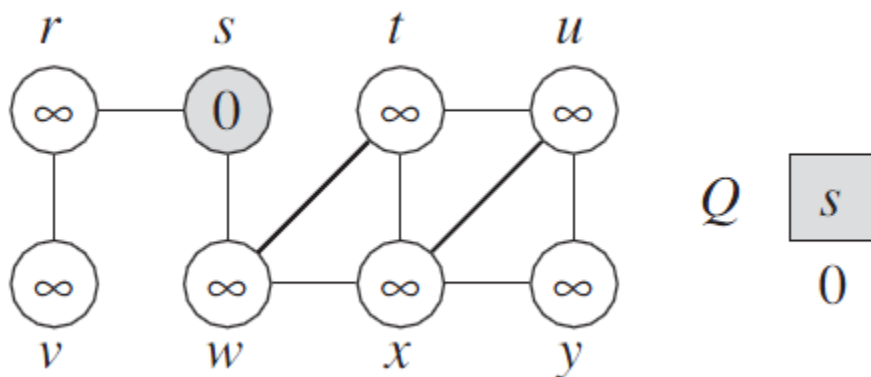
1 para cada vértice  $v_i$  em  $G.V$  faça
2      $v_i.d = \text{INFINITO}$ 
3      $v_i.cor = \text{BRANCO}$ 
4  $s.d = 0$ 
5  $s.cor = \text{CINZA}$ 
6  $Q = \text{VAZIO}$ 
7 Insere( $Q, s$ )
    
```

(Inicialização)



## BFS: busca em largura

BFS( $G, s$ ):



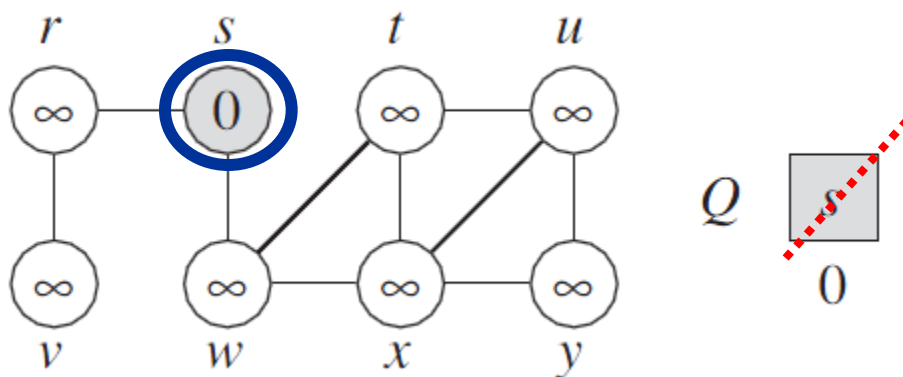
```

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            então  $v_i.d = u_i.d + 1$ 
13                 $v_i.\text{cor} = \text{CINZA}$ 
14                 $\text{Insere}(Q, v_i)$ 
15     $u_i.\text{cor} = \text{PRETO}$ 

```

## BFS: busca em largura

BFS( $G, s$ ):



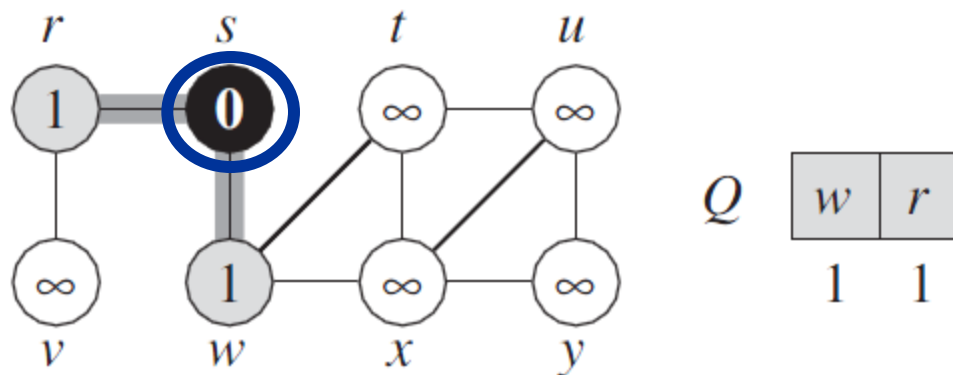
```

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: busca em largura

BFS( $G, s$ ):



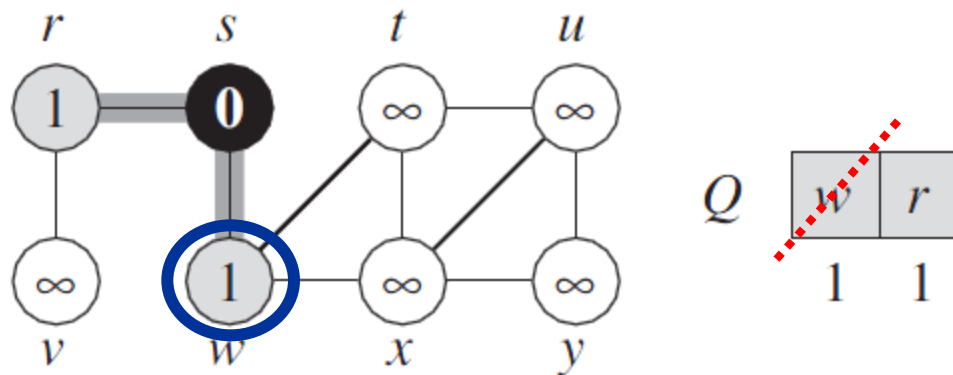
```

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: busca em largura

BFS( $G, s$ ):

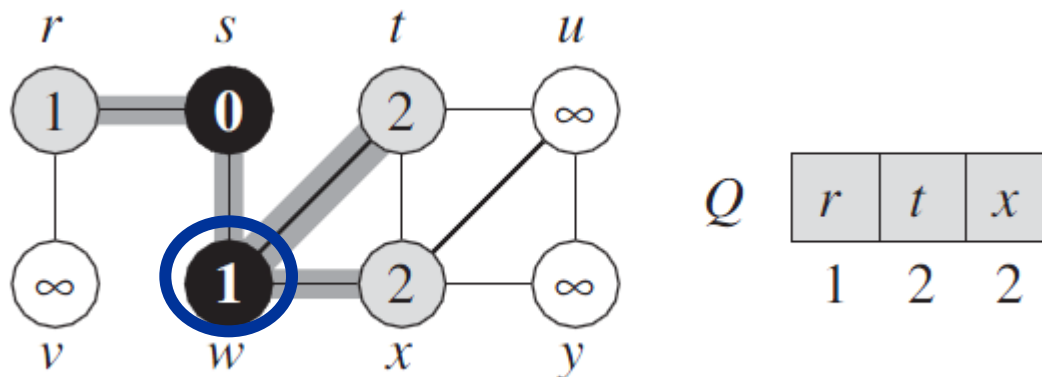


```

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: busca em largura

BFS( $G, s$ ):



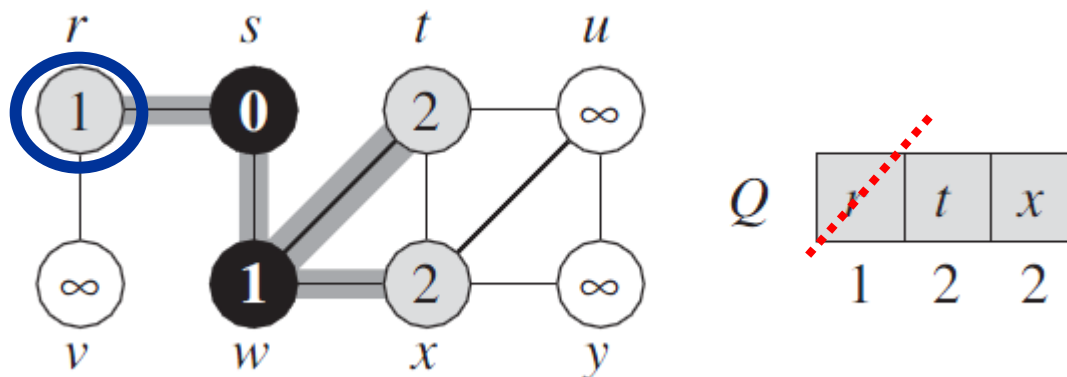
```

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: busca em largura

BFS( $G, s$ ):



```

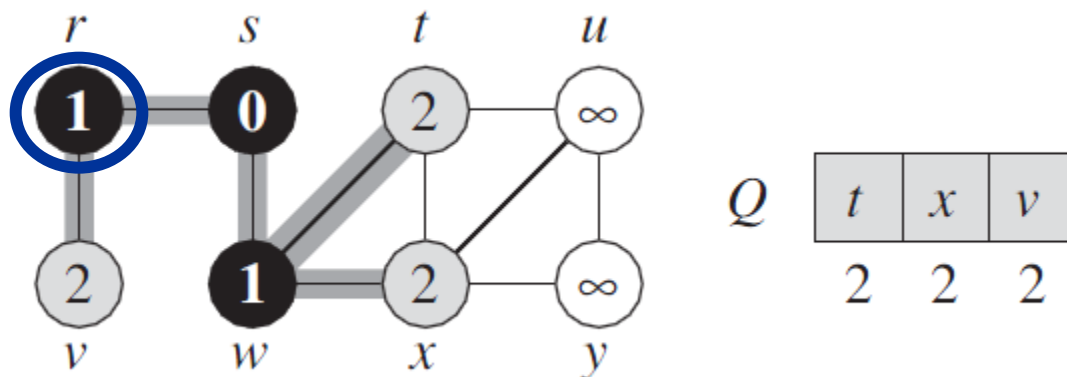
8 enquanto Q != VAZIO faça
9      $u_i = \text{Remove}(Q)$ 
10    para cada  $v_i$  em  $G.\text{Adj}[u_i]$  faça
11        se  $v_i.\text{cor} == \text{BRANCO}$ 
12            entao  $v_i.d = u_i.d + 1$ 
13             $v_i.\text{cor} = \text{CINZA}$ 
14             $\text{Insere}(Q, v_i)$ 
15     $u_i.\text{cor}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

```



## BFS: busca em largura

BFS( $G, s$ ):



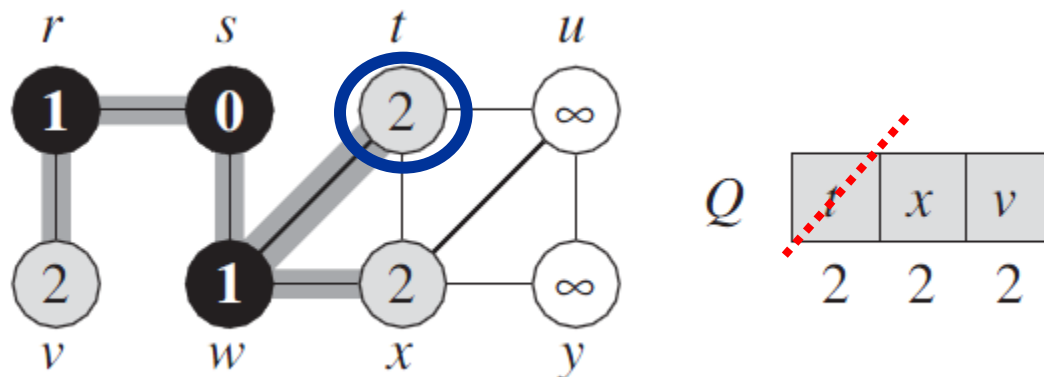
```

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: busca em largura

BFS( $G, s$ ):



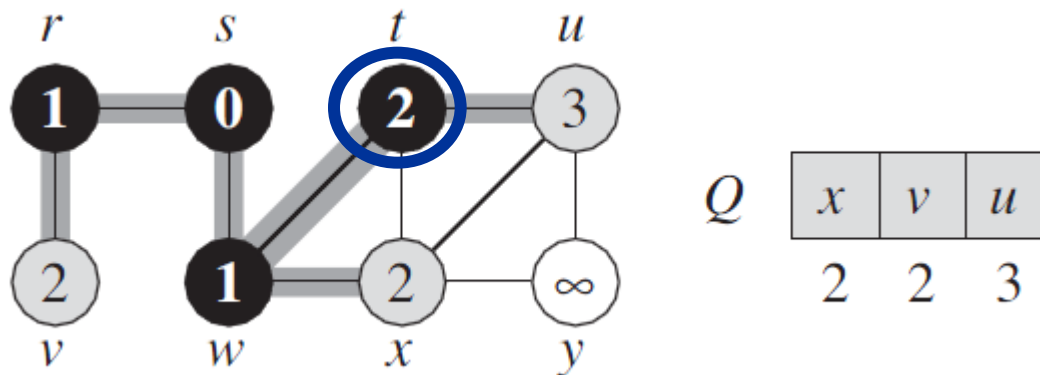
```

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: busca em largura

BFS( $G, s$ ):



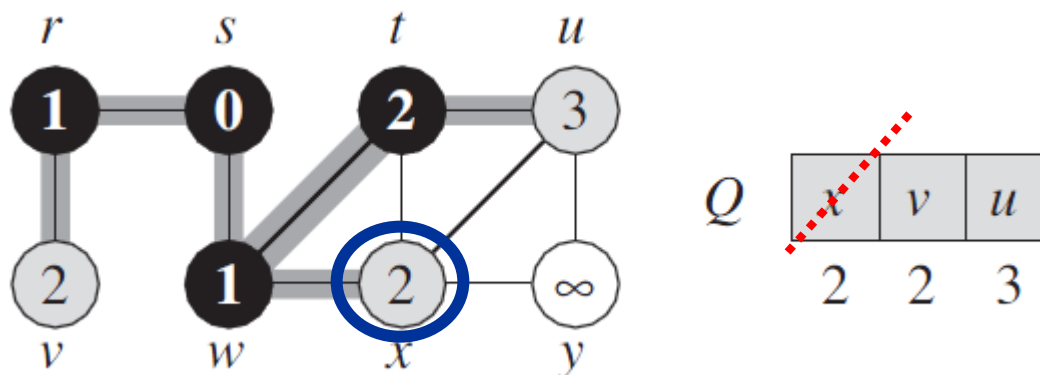
```

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: busca em largura

BFS( $G, s$ ):



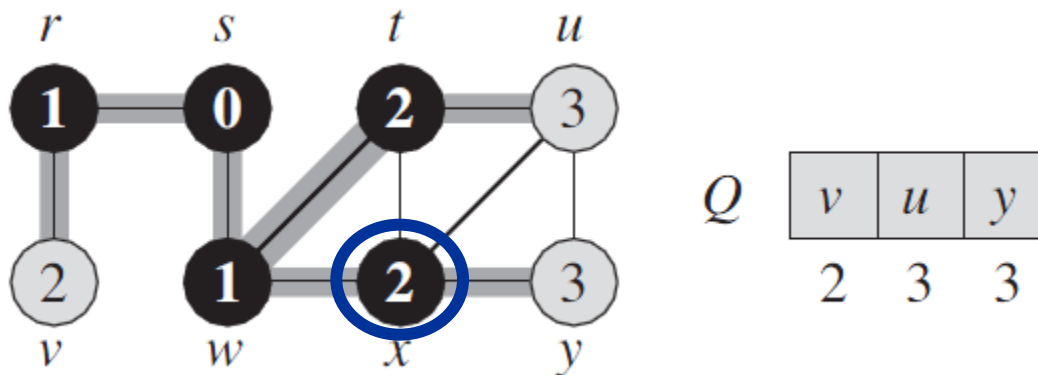
```

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: busca em largura

BFS( $G, s$ ):



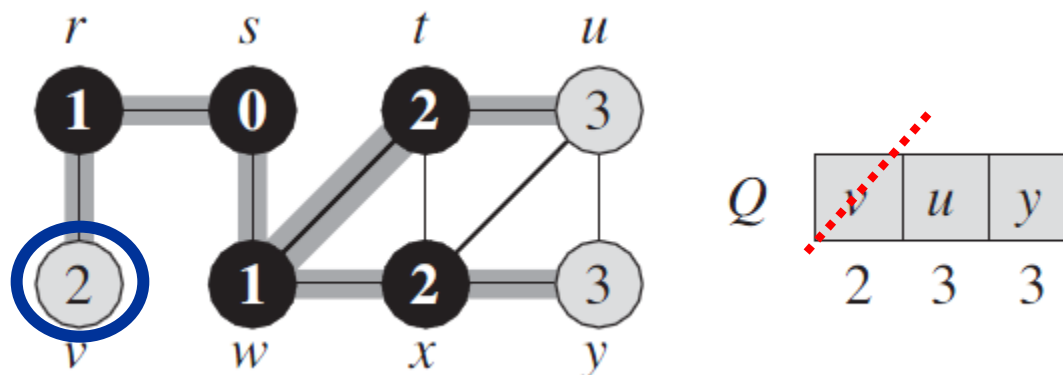
```

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: busca em largura

BFS( $G, s$ ):

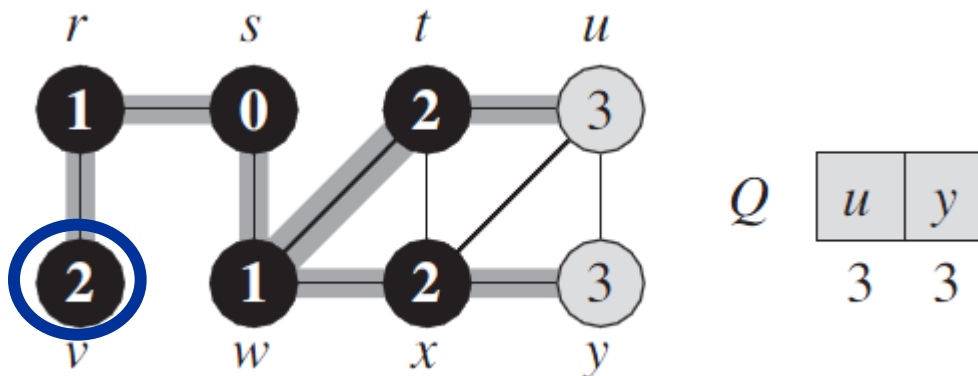


```

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: busca em largura

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 então  $v_i.d = u_i.d + 1$

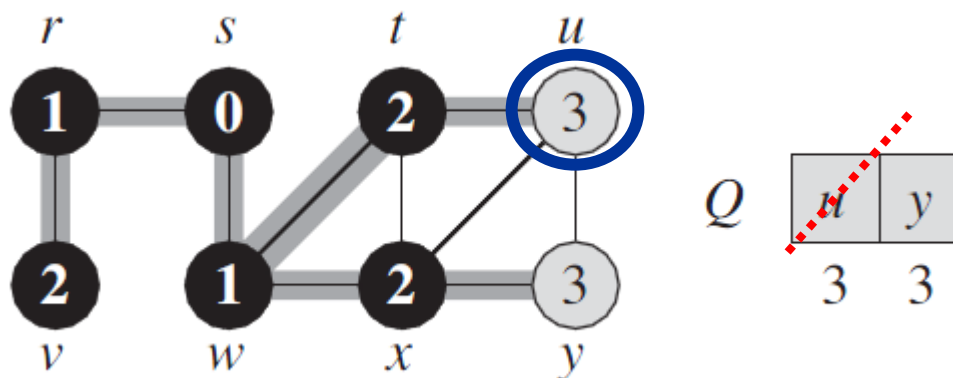
13  $v_i.\text{cor} = \text{CINZA}$

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

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

## BFS: busca em largura

BFS( $G, s$ ):



```

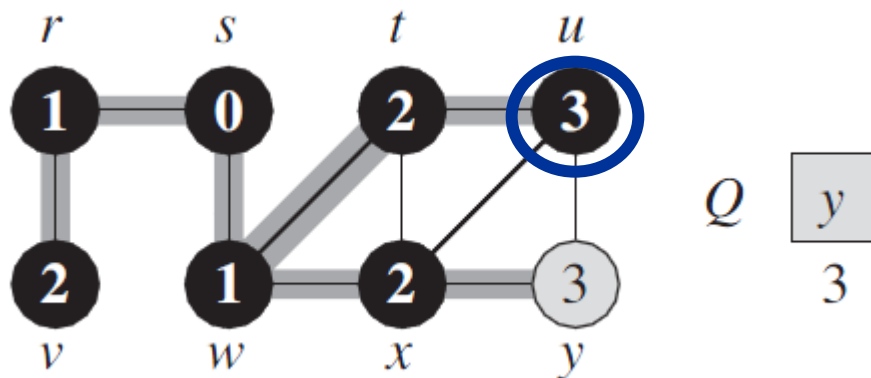
8 enquanto  $Q \neq \text{VAZIO}$  faça
9      $u_i = \text{Remove}(Q)$ 
10    para cada  $v_i$  em  $G.\text{Adj}[u_i]$  faça
11        se  $v_i.\text{cor} == \text{BRANCO}$ 
12            entao  $v_i.d = u_i.d + 1$ 
13             $v_i.\text{cor} = \text{CINZA}$ 
14             $\text{Insere}(Q, v_i)$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

```



## BFS: busca em largura

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 então  $v_i.d = u_i.d + 1$

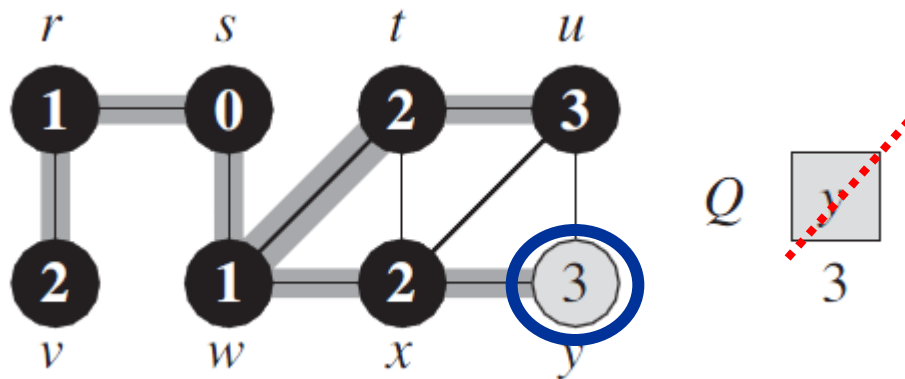
13  $v_i.\text{cor} = \text{CINZA}$

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

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

## BFS: busca em largura

BFS( $G, s$ ):

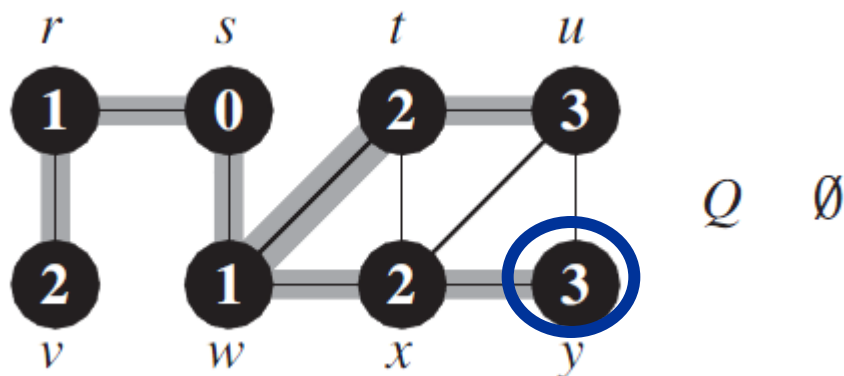


```

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: busca em largura

BFS( $G, s$ ):



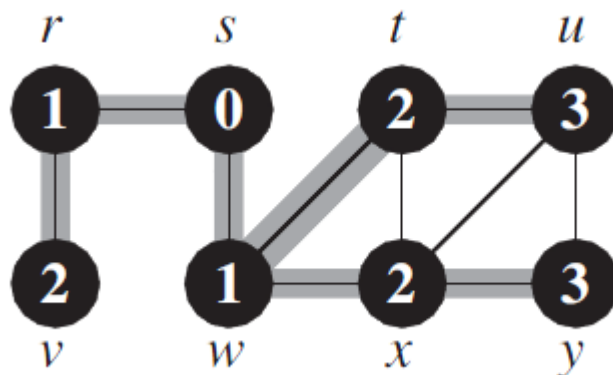
```

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: busca em largura

BFS( $G, s$ ):



**Distâncias em relação a  $s$ :**

$$\delta(s, r) = 1 \quad \delta(s, v) = 2$$

$$\delta(s, s) = 0 \quad \delta(s, w) = 1$$

$$\delta(s, t) = 2 \quad \delta(s, x) = 2$$

$$\delta(s, u) = 3 \quad \delta(s, y) = 3$$

```

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
    
```

### Tarefa



- EP 1

- Página da disciplina:

- <https://sites.google.com/site/alexnomma/home/grafos>