Introdução à Teoria dos Grafos

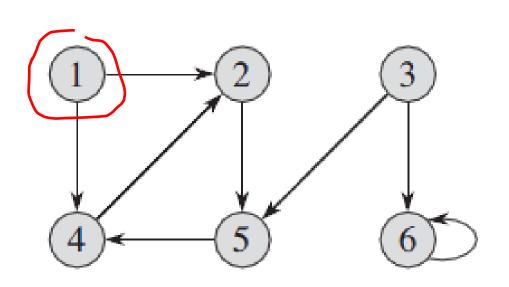
Prof. Alexandre Noma

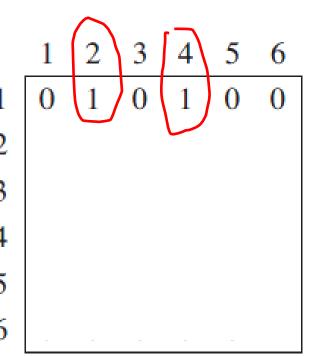
Representar um grafo no computador:

• (1) Matriz de adjacências

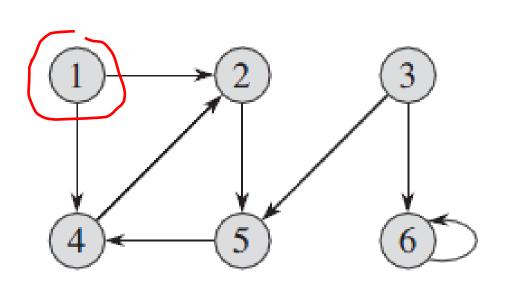
• (2) Listas de adjacências

(1) Matriz de adjacências



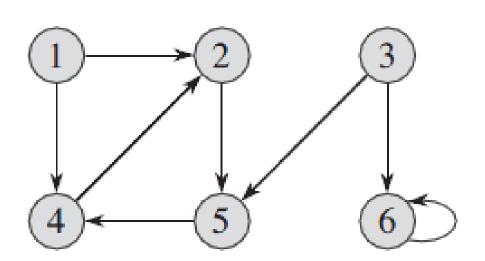


(1) Matriz de adjacências



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

(2) Listas de adjacências



1: 2, 4

2:?

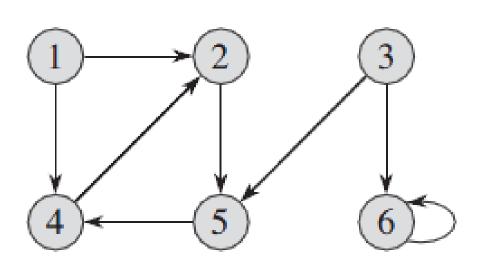
3:?

4:?

5: ?

6: ?

(2) Listas de adjacências



1: 2, 4

2: 5

3: 5, 6

4: 2

5: 4

6: 6

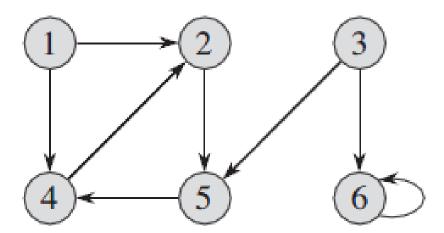
Representar um grafo no computador:

• (1) Matriz de adjacências

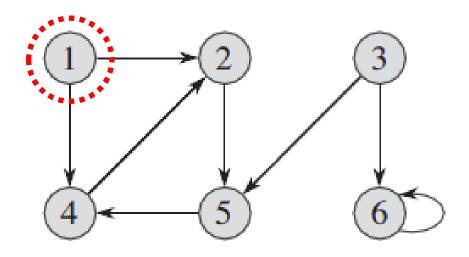
• (2) Listas de adjacências

adjacência = "vizinhança"

• Como visitar TODOS os vértices de um grafo?

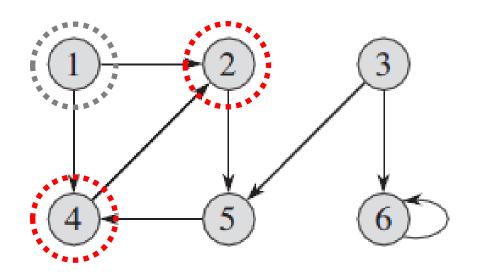


• Como visitar TODOS os vértices de um grafo?



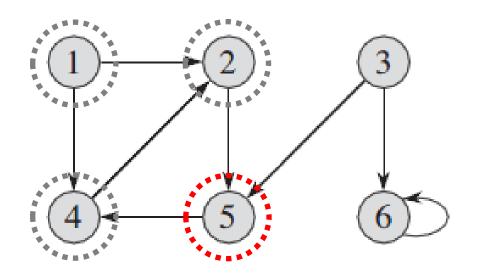
Escolha um vértice inicial.

• Como visitar TODOS os vértices de um grafo?



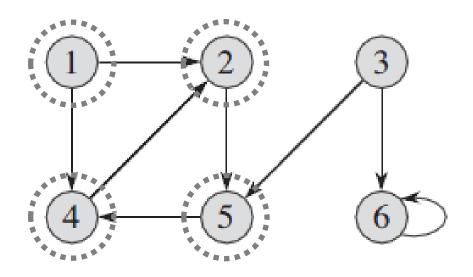
Visite os seus vizinhos.

• Como visitar TODOS os vértices de um grafo?



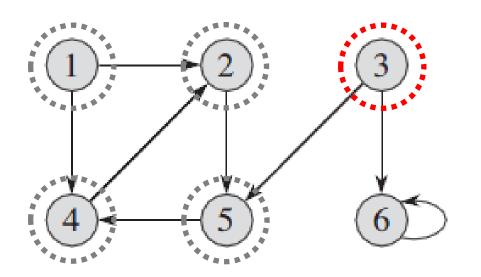
Visite os seus vizinhos.

• Como visitar TODOS os vértices de um grafo?



Acabou vizinhos???

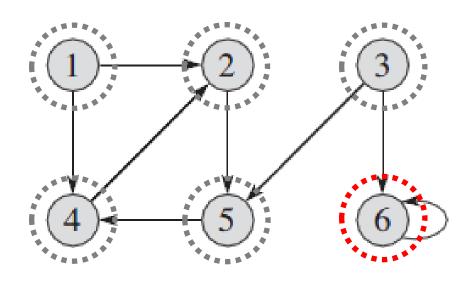
• Como visitar TODOS os vértices de um grafo?



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

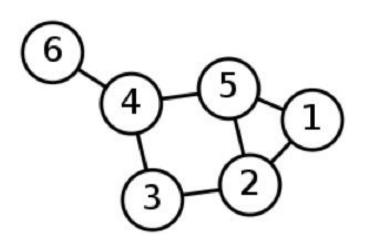
"Busca em largura"

Nosso primeiro algoritmo de busca...

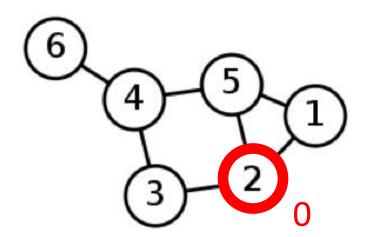
BFS: breadth first search

"Busca em largura"

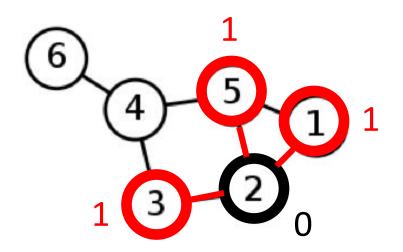
Por que este nome???



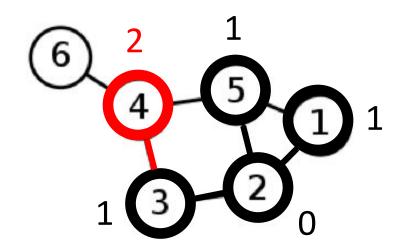
Vértice inicial.



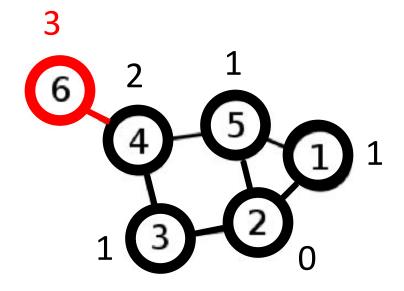
Fogo espalha para vizinhos.



Fogo espalha para vizinhos.



Fogo espalha para vizinhos.



Implementação

• Lembra da estrutura de dados Fila?



Implementação

• Lembra da estrutura de dados Fila?



Algumas operações com Fila:

- Insere(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.





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

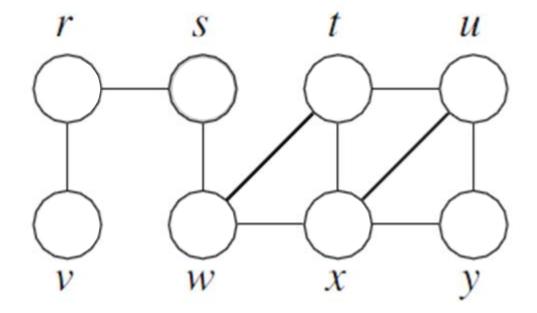
- BFS(G, s)
 - Entrada: um grafo G e um vértice inicial s
 - Saída: "distâncias" em relação ao vértice s

- Atributos
 - v.d: distância
 - v.cor: BRANCO, CINZA, PRETO

 (inicialmente não visitado visitado,

finalizado)

Exemplo



BFS (**G**, **s**): 1 para cada vértice v; em G.V faça $v_i \cdot d = INFINITO$ v_i .cor = BRANCO 4 s.d = 05 s.cor = CINZA $6 \mathbf{Q} = VAZIO$ // Inicialização 7 Insere (Q, s)enquanto Q != VAZIO faça $u_i = Remove(Q)$ 10 para cada v; em G.**Adj**[u;] faça 11 $se v_i.cor == BRANCO$ 12 entao $v_i.d = u_i.d + 1$ 13 $v_{i}.cor = CINZA$

 $u_{i}.cor = PRETO$

Insere (Q, v_i)

14

15

28

BFS(\mathbf{G} , \mathbf{s}): 1 para cada vértice v_i em \mathbf{G} .V faça 2 v_i . \mathbf{d} = INFINITO 3 v_i . \mathbf{cor} = BRANCO 4 s.d = 0 5 s.cor = CINZA

 $6 \ \mathbf{Q} = VAZIO$

7 Insere (Q, s)

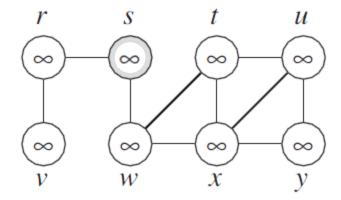
// Inicialização

```
enquanto Q != VAZIO faça
 8
 9
        u_i = Remove(Q)
10
        para cada v<sub>i</sub> em G.Adj[u<sub>i</sub>] faça
11
              se v_i.cor == BRANCO
                entao v_i.d = u_i.d + 1
12
13
                        v_i.cor = CINZA
14
                        Insere (Q, v_i)
15
        u_i.cor = PRETO
```

```
8 enquanto Q != VAZIO faça
9 u_i = Remove(Q)

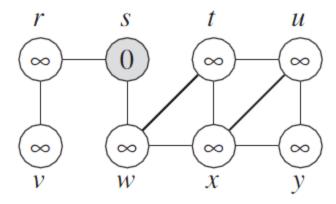
10 para cada v_i em G.Adj[u_i] faça
11 se v_i.cor == BRANCO
12 entao v_i.d = u_i.d + 1
13 v_i.cor = CINZA
14 Insere(Q, v_i)
15 u_i.cor = PRETO
```

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



(Inicialização)

- 1 para cada vértice v_i em G.V faça
- $v_i.d = INFINITO$
- $v_i.cor = BRANCO$
- 4 s.d = 0
- 5 s.cor = CINZA
 - 6 **Q** = VAZIO
 - 7 Insere(Q, s)



(Inicialização)

```
BFS (G, s):
```

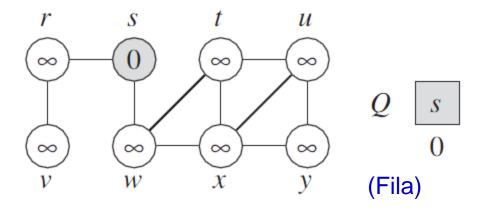
```
1 para cada vértice v_i em G.V faça 2 v_i.d = INFINITO
```

3
$$v_i$$
.cor = BRANCO

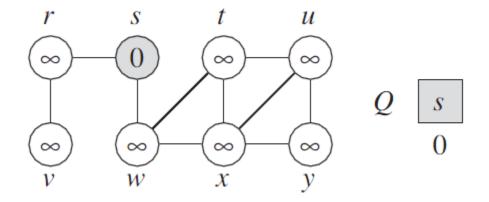
$$4 \text{ s.d} = 0$$

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

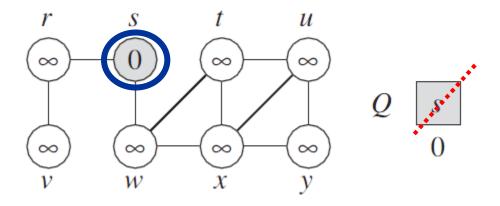
7 Insere
$$(Q, s)$$



(Inicialização)



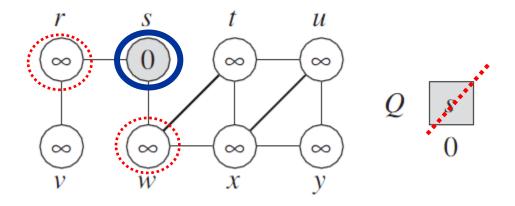
```
8 enquanto Q != VAZIO faça
9 u_i = Remove(Q)
10 para cada v_i em G.Adj[u_i] faça
11 se v_i.cor == BRANCO
12 entao v_i.d = u_i.d + 1
13 v_i.cor = CINZA
14 Insere(Q, v_i)
15 u_i.cor = PRETO
```



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)

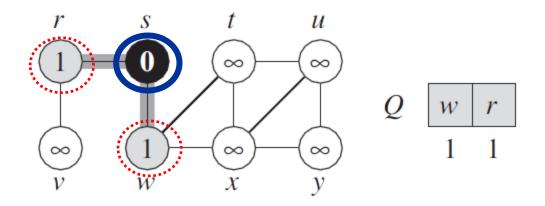
15 u_i.cor = PRETO



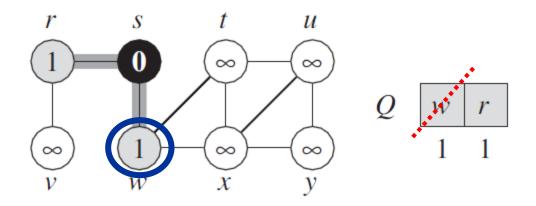
```
8 enquanto Q != VAZIO faça
```

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



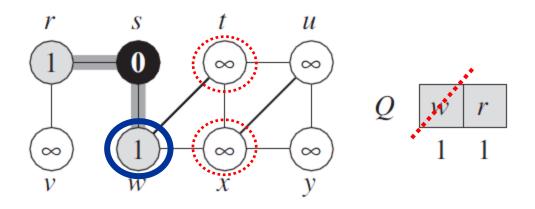
```
8 enquanto Q != VAZIO faça
9 u_i = Remove(Q)
10 para cada v_i em G.Adj[u_i] faça
11 se v_i.cor == BRANCO
12 entao v_i.d = u_i.d + 1
13 v_i.cor = CINZA
14 Insere(Q, v_i)
```



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)
	·	

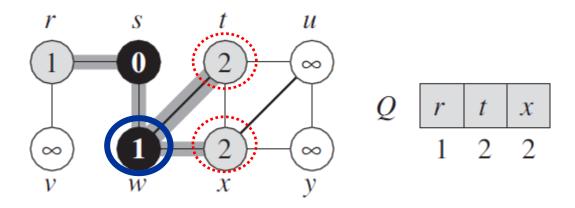
15 $u_i.cor = PRETO$



```
8 enquanto Q != VAZIO faça
```

9 $u_i = Remove(Q)$

	1
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

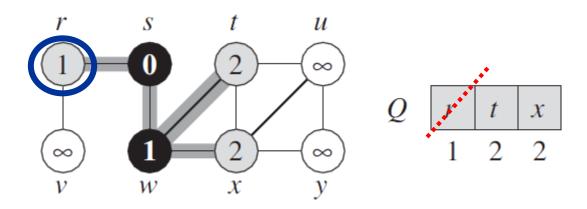


```
8 enquanto Q != VAZIO faça
```

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
Insere(Q, v_i)

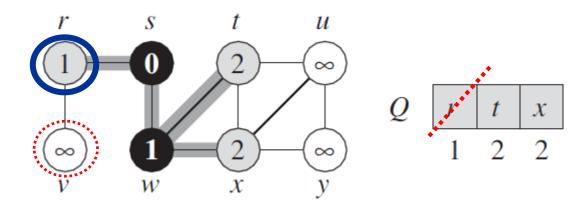
l5 u_i.cor = PRETO



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	11 $C \cap r = DRFT \cap$

42

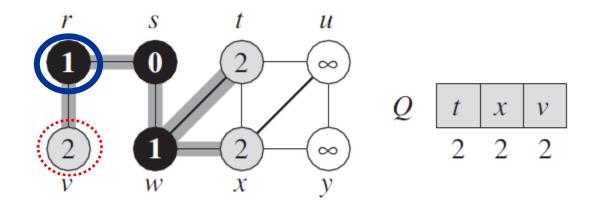


```
8 enquanto Q != VAZIO faça
```

9
$$u_i = Remove(Q)$$

	11 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1
1 0	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)
*4"	

15 u_i .cor = PRETO



```
8 enquanto Q != VAZIO faça
```

9
$$u_i = Remove(Q)$$

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

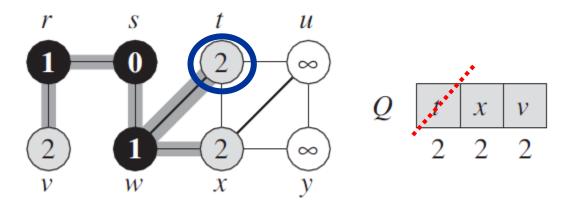
11 se $v_i.cor == BRANCO$

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

13 $v_i.cor = CINZA$

Insere(Q, v_i)

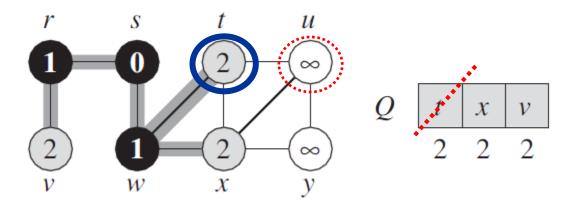
u;.cor = PRETO



8 enquanto Q != VAZIO faça

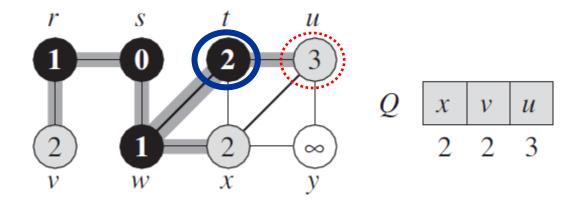
→	9	$u_i = Remove(Q)$
	10	para cada v _i em G.Adj[u _i] faça
	11	se v _i .cor == BRANCO
	12	entao $v_i.d = u_i.d + 1$
	13	$v_{i}.cor = CINZA$
	14	Insere(Q, v_i)
	·	DD D D D D D D D D D D D D D D D D D D

15 $u_i.cor = PRETO$



```
8 enquanto Q != VAZIO faça
```

9
$$u_i = Remove(Q)$$



```
8 enquanto Q != VAZIO faça
```

9
$$u_i = Remove(Q)$$

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

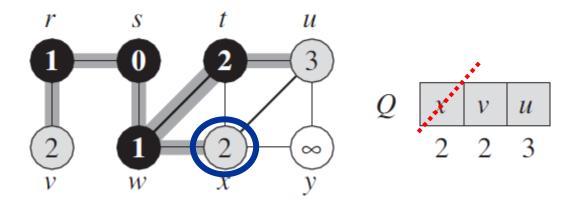
11 se v_i .cor == BRANCO

12 entao v_i .d = u_i .d + 1

13 v_i .cor = CINZA

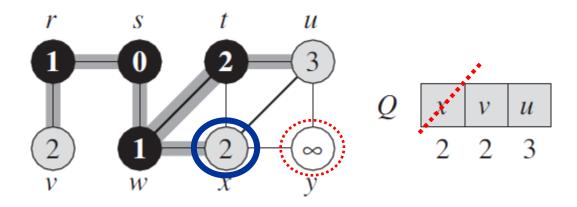
Insere(Q, v_i)

15 u_i.cor = PRETO



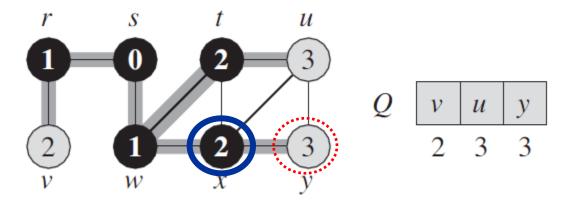
8 enquanto Q != VAZIO faça

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



```
8 enquanto Q != VAZIO faça u_i = Remove(Q)
```

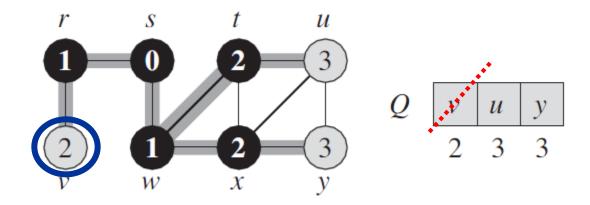
10	para cada v _i em G.Adj[u _i] faça
11	se v _i .cor == BRANCO
12	$entao v_i.d = u_i.d + 1$
13	v;.cor = CINZA
14	Insere(Q, v _i)
15	u _i .cor = PRETO



```
8 enquanto Q != VAZIO faça
9 u_i = Remove(Q)

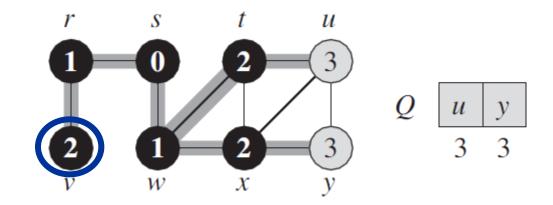
10 para cada v_i em G.Adj[u_i] faça
11 se v_i.cor == BRANCO
12 entao v_i.d = u_i.d + 1
13 v_i.cor = CINZA
14 Insere(Q, v_i)
```

u_i.cor = PRETO



8 enquanto Q != VAZIO faça

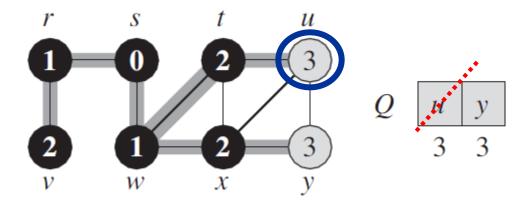
9	$u_i = Remove(Q)$
10	para cada v _i em G.Adj[u _i] faça
11	se v _i .cor == BRANCO
12	$entao v_i.d = u_i.d + 1$
13	$v_{i}.cor = CINZA$
14	Insere(Q, v_i)
15	u _i .cor = PRETO



```
8 enquanto Q != VAZIO faça
```

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

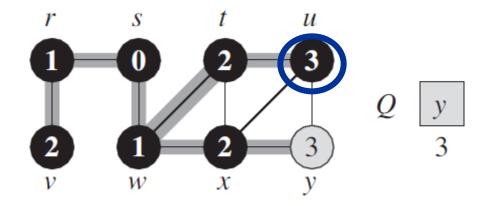
u_i.cor = PRETO



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)

 $u_{i}.cor = PRETO$

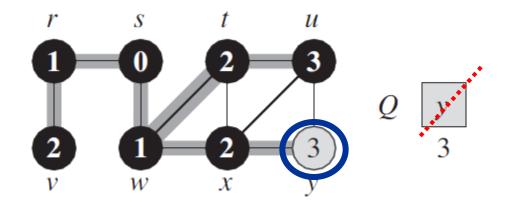


```
8 enquanto Q != VAZIO faça
```

$$9 u_i = Remove(Q)$$

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)

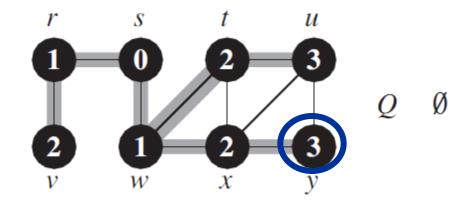
u_i.cor = PRETO



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)

 $u_i.cor = PRETO$



```
8 enquanto Q != VAZIO faça
```

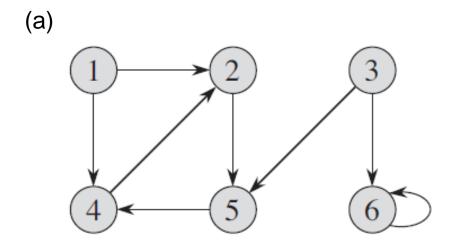
9
$$u_i = Remove(Q)$$

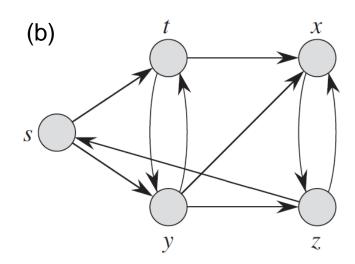
10	para cada v _i em G.Adj[u _i] faça
11	se v _i .cor == BRANCO
12	entao $v_i.d = u_i.d + 1$
13	$v_{i}.cor = CINZA$
14	Insere(Q, v_i)

u_i.cor = PRETO

Exercícios

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





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:
   para cada vértice v; em G.V faça
                                                          333
         v_i.d = INFINITO
                                                           333
         v_i.cor = BRANCO
                                                          333
 4 \text{ s.d} = 0
                                                          0(1)
 5 \text{ s.cor} = \text{CINZA}
                                                          0(1)
 6 Q = VAZIO
                                                          0(1)
                                                          0(1)
   Insere (Q, s)
    enquanto Q != VAZIO faça
                                                           333
 9
         u_i = Remove(Q)
                                                          333
10
         para cada v<sub>i</sub> em G.Adj[u<sub>i</sub>] faça
                                                          333
11
               se v<sub>i</sub>.cor == BRANCO
                                                          333
12
                 entao v_i \cdot d = u_i \cdot d + 1
                                                          555
13
                         v_{i}.cor = CINZA
                                                          333
14
                         Insere (Q, v_i)
                                                          333
15
         u_{i}.cor = PRETO
                                                           333
                                                            60
```

Algumas operações com Fila:

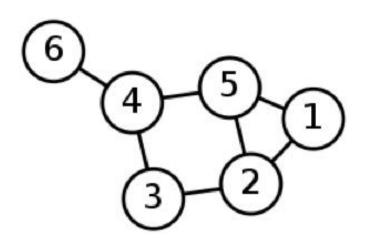
- Insere(Q, x) consome O(1) unidades de tempo.
 - insere elemento x no final da fila Q
- Remove(Q) consome O(1) unidades de tempo.
 - remove e devolve o elemento do início da fila Q
- FilaVazia(Q) consome O(1) unidades de tempo.
 - devolve verdadeiro se a fila estiver vazia,
 falso caso contrário.
-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

```
BFS (G, s):
                                                 Consumo
                                                de tempo:
 1 para cada vértice v; em G.V faça
                                                      O(n)
        v_i.d = INFINITO
                                                O(n) *O(1)
        v_i.cor = BRANCO
                                                O(n) *O(1)
                                                     0(1)
 4 \text{ s.d} = 0
 5 \text{ s.cor} = CINZA
                                                     0(1)
 6 \mathbf{Q} = VAZIO
                                                      0(1)
                                                     0(1)
   Insere (Q, s)
   enquanto Q != VAZIO faça
                                                      333
 9
        u_i = Remove(Q)
                                                      333
10
        para cada v; em G.Adj[u;] faça
                                                      555
11
             se v<sub>i</sub>.cor == BRANCO
                                                      333
12
                entao v_i.d = u_i.d + 1
                                                      333
13
                       v_i.cor = CINZA
                                                      333
14
                       Insere (Q, v_i)
                                                      333
15
        u_{i}.cor = PRETO
                                                      333
                                                        62
```

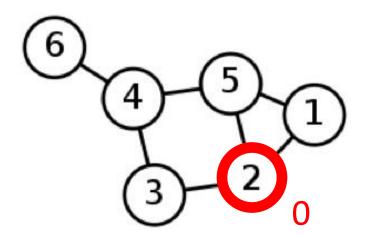
```
BFS (G, s):
                                                   Consumo
                                                   de tempo:
   para cada vértice v; em G.V faça
                                                         O(n)
        v_i.d = INFINITO
                                                         O(n)
        v_i.cor = BRANCO
                                                         O(n)
 4 \text{ s.d} = 0
                                                        0(1)
 5 \text{ s.cor} = CINZA
                                                        0(1)
   \mathbf{Q} = VAZIO
                                                        0(1)
                                                        0(1)
    Insere (Q, s)
   enquanto Q != VAZIO faça
                                                        O(n)
 9
                                                  O(n) *O(1)
         u_i = Remove(Q)
10
         para cada v<sub>i</sub> em G.Adj[u<sub>i</sub>] faça
                                                        555
11
              se v<sub>i</sub>.cor == BRANCO
                                                         333
12
                 entao v_i.d = u_i.d + 1
                                                        333
13
                        v_{i}.cor = CINZA
                                                         333
14
                        Insere (Q, v_i)
                                                         333
15
         u_{i}.cor = PRETO
                                                  O(n) *O(1)
```

```
BFS (G, s):
                                                 Consumo
                                                de tempo:
   para cada vértice v; em G.V faça
                                                      O(n)
        v_i.d = INFINITO
                                                      O(n)
        v_i.cor = BRANCO
                                                      O(n)
                                                     0(1)
 4 \text{ s.d} = 0
 5 \text{ s.cor} = CINZA
                                                     0(1)
   \mathbf{Q} = VAZIO
                                                      0(1)
                                                     0(1)
    Insere (Q, s)
                                                     O(n)
    enquanto Q != VAZIO faça
 9
                                                     O(n)
         u_i = Remove(Q)
10
         para cada v; em G.Adj[u;] faça
                                                     O (m)
11
              se v<sub>i</sub>.cor == BRANCO
                                                      333
12
                entao v_i.d = u_i.d + 1
                                                      555
13
                        v_i.cor = CINZA
                                                      333
14
                        Insere (Q, v_i)
                                                      555
15
         u_{i}.cor = PRETO
                                                     O(n)
```

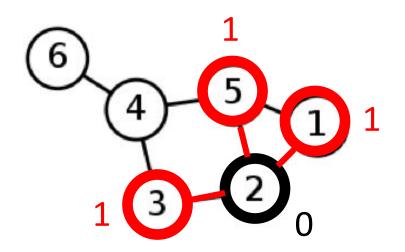
64



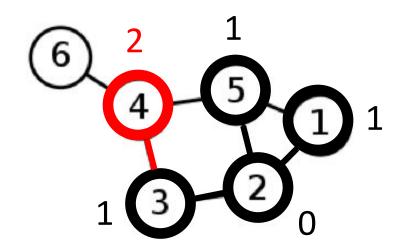
Vértice inicial.



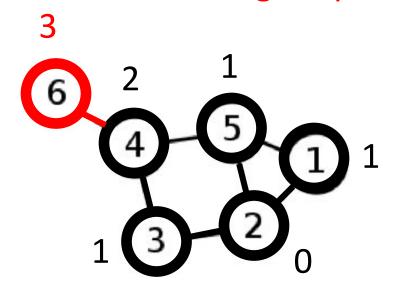
Fogo espalha para vizinhos.



Fogo espalha para vizinhos.



Fogo espalha para vizinhos.



```
BFS (G, s):
                                                  Consumo
                                                 de tempo:
   para cada vértice v; em G.V faça
                                                       O(n)
         v_i.d = INFINITO
                                                       O(n)
                                                       O(n)
         v_i.cor = BRANCO
                                                      0(1)
 4 \text{ s.d} = 0
 5 \text{ s.cor} = CINZA
                                                      0(1)
   \mathbf{Q} = VAZIO
                                                       0(1)
                                                      0(1)
    Insere (Q, s)
                                                      O(n)
    enquanto Q != VAZIO faça
 9
                                                      O(n)
         u_i = Remove(Q)
10
                                                      O (m)
         para cada v; em G.Adj[u;] faça
11
                                                 O(m) *O(1)
              se v<sub>i</sub>.cor == BRANCO
12
                                                 O(m) *O(1)
                entao v_i \cdot d = u_i \cdot d + 1
                                                 O(m) *O(1)
13
                        v_i.cor = CINZA
14
                        Insere (Q, v_i)
                                                 O(m) *O(1)
15
         u_{i}.cor = PRETO
                                                      O(n)
```

70

```
BFS (G, s):
                                                Consumo
                                               de tempo:
 1 para cada vértice v<sub>i</sub> em G.V faça
                                                     O(n)
       v_i.d = INFINITO
                                                     O(n)
    v_i.cor = BRANCO
                                                     O(n)
 4 \text{ s.d} = 0
                                                    0(1)
 5 \text{ s.cor} = CINZA
                                                    0(1)
 6 \mathbf{Q} = VAZIO
                                                    0(1)
 7 Insere (Q, s)
                                                    0(1)
 8 enquanto Q != VAZIO faça
                                                    O(n)
                                                    O(n)
        u_i = Remove(Q)
10
        para cada v; em G.Adj[u;] faça
                                                   O(m)
11
                                                   O(m)
             se v;.cor == BRANCO
12
                                                   O (m)
               entao v_i \cdot d = u_i \cdot d + 1
13
                       v_{i}.cor = CINZA
                                                   O(m)
14
                       Insere (Q, v_i)
                                                   O(m)
15
                                                    O(n)
        u_i.cor = PRETO
        Total:
                             T(n,m) = 6*O(n) + 5*O(m) + 4*O(1)
```

71

```
BFS (G, s):
                                                  Consumo
                                                 de tempo:
 1 para cada vértice v<sub>i</sub> em G.V faça
                                                       O(n)
        v_i.d = INFINITO
                                                       O(n)
    v_i.cor = BRANCO
                                                       O(n)
 4 \text{ s.d} = 0
                                                      0(1)
 5 \text{ s.cor} = CINZA
                                                      0(1)
 6 \mathbf{Q} = VAZIO
                                                       0(1)
 7 Insere (Q, s)
                                                      0(1)
 8 enquanto Q != VAZIO faça
                                                      O(n)
                                                      O(n)
        u_i = Remove(Q)
10
        para cada v<sub>i</sub> em G.Adj[u<sub>i</sub>] faça
                                                     O(m)
11
                                                      O(m)
              se v;.cor == BRANCO
12
                                                     O (m)
                entao v_i \cdot d = u_i \cdot d + 1
13
                        v_{i}.cor = CINZA
                                                      O(m)
14
                        Insere (Q, v_i)
                                                      O(m)
15
                                                      O(n)
        u_i.cor = PRETO
        Total:
                              T(n,m) = 6*O(n) + 5*O(m) + 4*O(1)
                                       = O(n+m)
```

Exercício Programa

• 02-bfs.py