



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
SUL-RIO-GRANDENSE  
Campus Passo Fundo



# **ESTRUTURA DE DADOS II**

**Prof. Adilso Nunes de Souza**



# BUSCA EM LARGURA

- O algoritmo de busca em largura BFS (Breadth-first-search) é descrito por BACKES, 2016 como sendo um algoritmo que inicia em um vértice e, então, visita todos os seus vizinhos antes de se aprofundar na busca. Esse processo continua até que o alvo da busca seja alcançado ou não existam mais vértices a ser visitados.
- Faz uso do conceito de FILA.



# BUSCA EM LARGURA

- O algoritmo de busca em largura é o arquétipo de muitos algoritmos de grafos, o algoritmo de árvore geradora mínima de Prim, e o algoritmo de caminhos mínimos de Dijkstra usam ideias semelhantes às que aparecem na busca em largura.
- A busca em largura expande a fronteira entre vértices descobertos e não descobertos uniformemente.



# FUNCIONAMENTO DO ALGORITMO

- O grafo é percorrido de maneira sistemática: primeiro, é marcado como visitado todos os vizinhos de um vértice e, em seguida, começa a visitar os vizinhos de cada vértice na ordem em que eles foram marcados.
- Para realizar essa tarefa, uma fila é utilizada para administrar a visitação dos vértices (o primeiro vértice marcado é o primeiro a ser visitado).



# FUNCIONAMENTO DO ALGORITMO

- Para controlar o progresso, a busca em largura pinta cada vértice de branco, cinza ou preto.
- No início todos os vértices são branco, na sequência eles tornam-se cinzentos e então pretos.
- Vértices cinzentos ou pretos são vértices já descobertos.
- Vértices cinzentos podem ter alguns vértices adjacentes brancos, eles representam a fronteira entre vértices descobertos e não descobertos.



# FUNCIONAMENTO DO ALGORITMO

- Um vértice preto só tem adjacentes já descobertos.
- Quando um vértice muda para cinza entra na fila, quando sai da fila passa a ser preto.



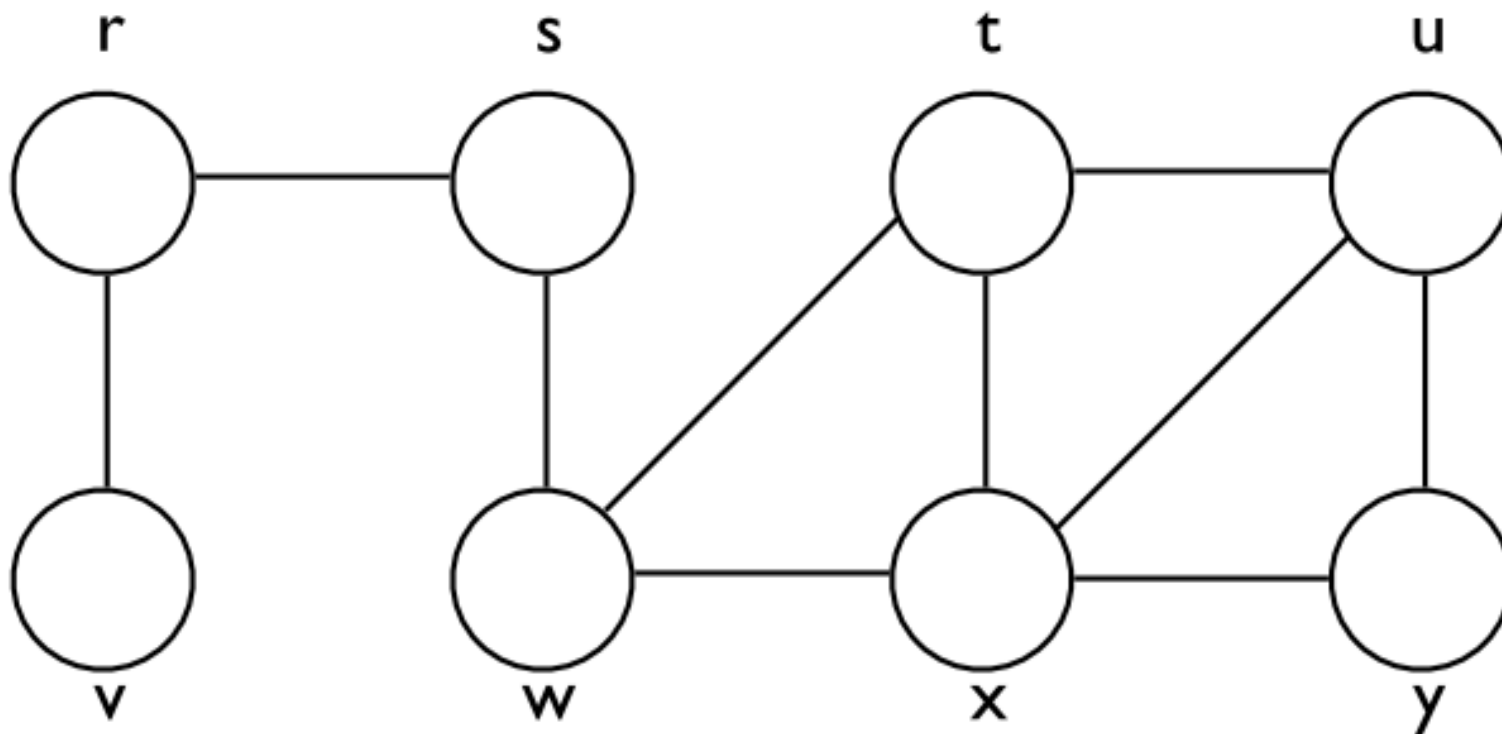
# ALGORITMO

- O algoritmo consiste no seguinte:
  - enquanto a fila não estiver vazia
    - retire um vértice  $x$  da fila
    - para cada vizinho  $y$  de  $x$ 
      - se  $y$  não está numerado
      - então numere  $y$
      - coloque  $y$  na fila
- No começo da primeira iteração, a fila contém o vértice inicial, com número 0.



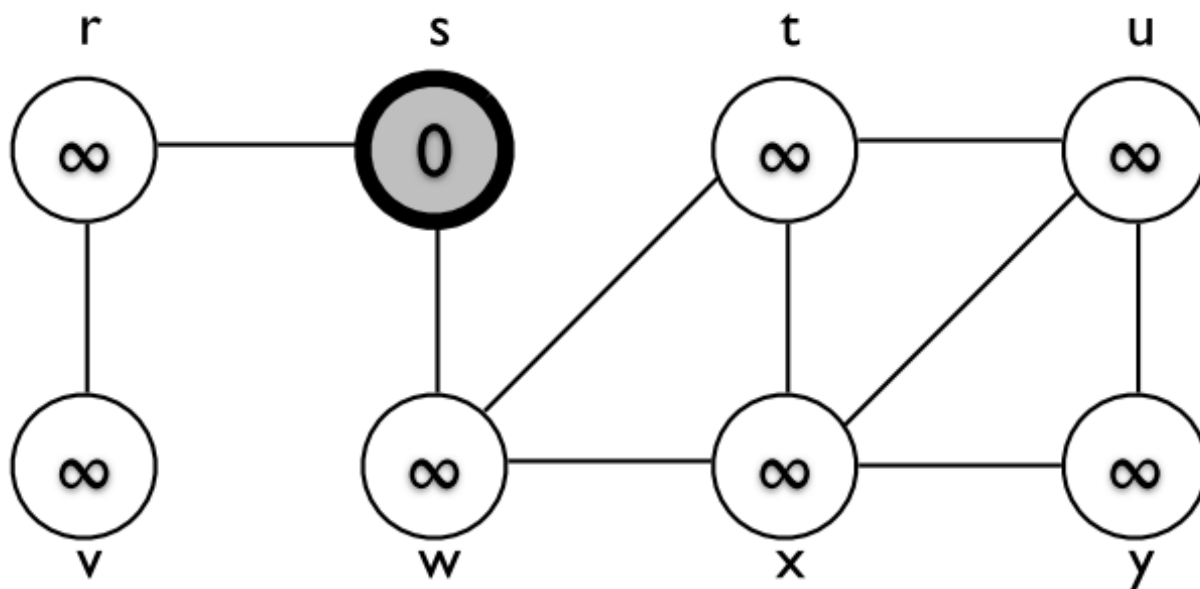
# EXECUÇÃO DO ALGORITMO

- Dado o grafo abaixo:





# EXECUÇÃO DO ALGORITMO



Fila Q

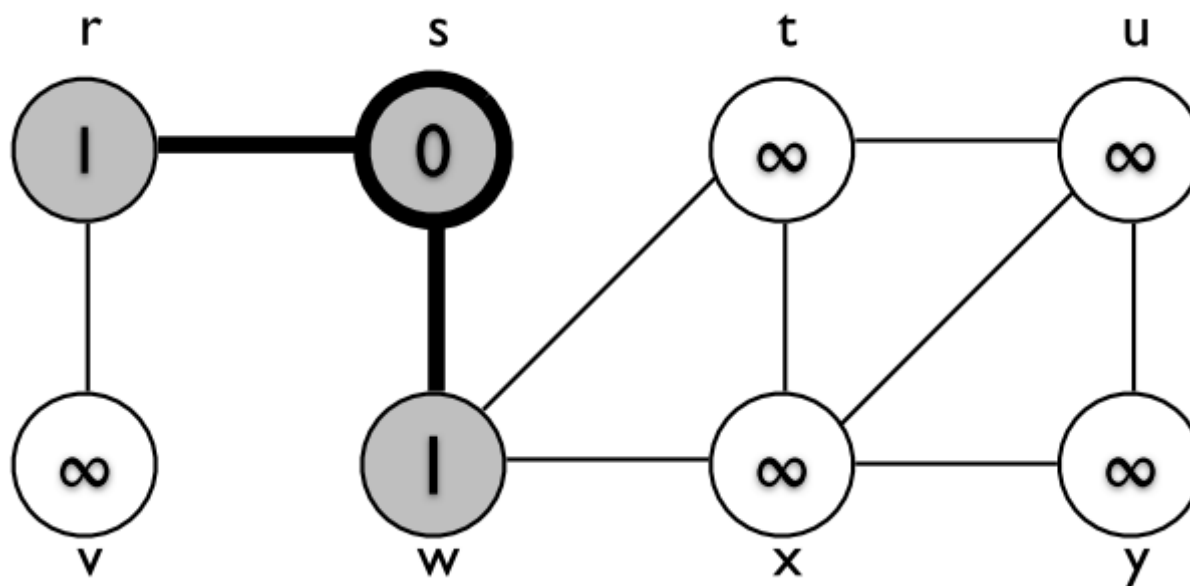
s
0

Nível

0
---



# EXECUÇÃO DO ALGORITMO



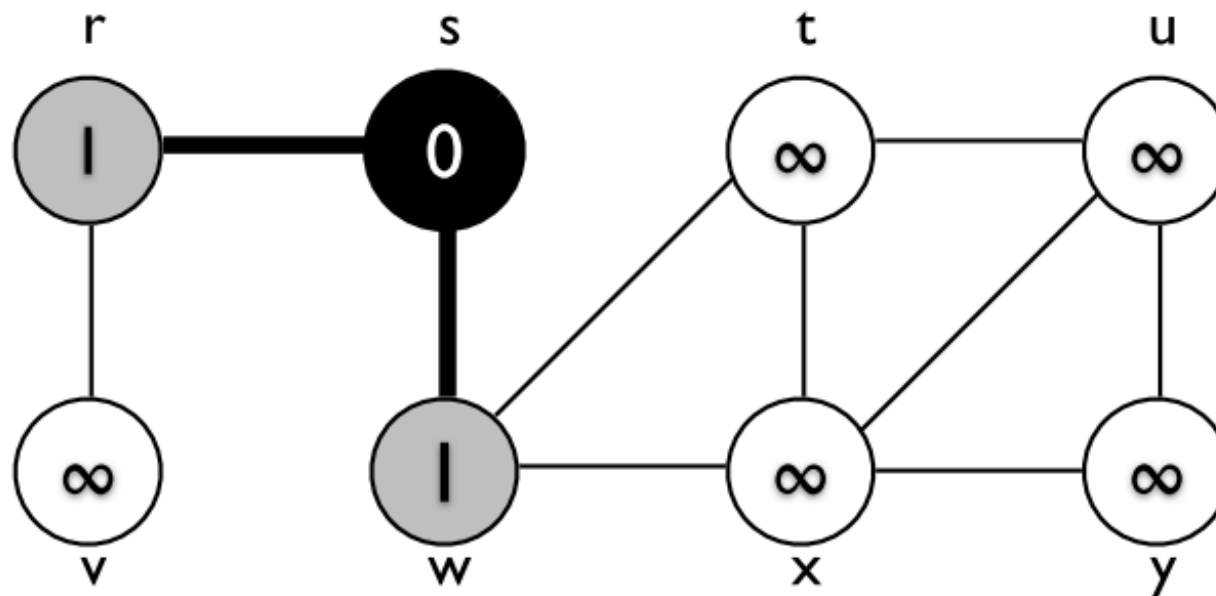
Fila Q

w	r
1	1

Nível



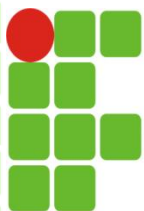
# EXECUÇÃO DO ALGORITMO



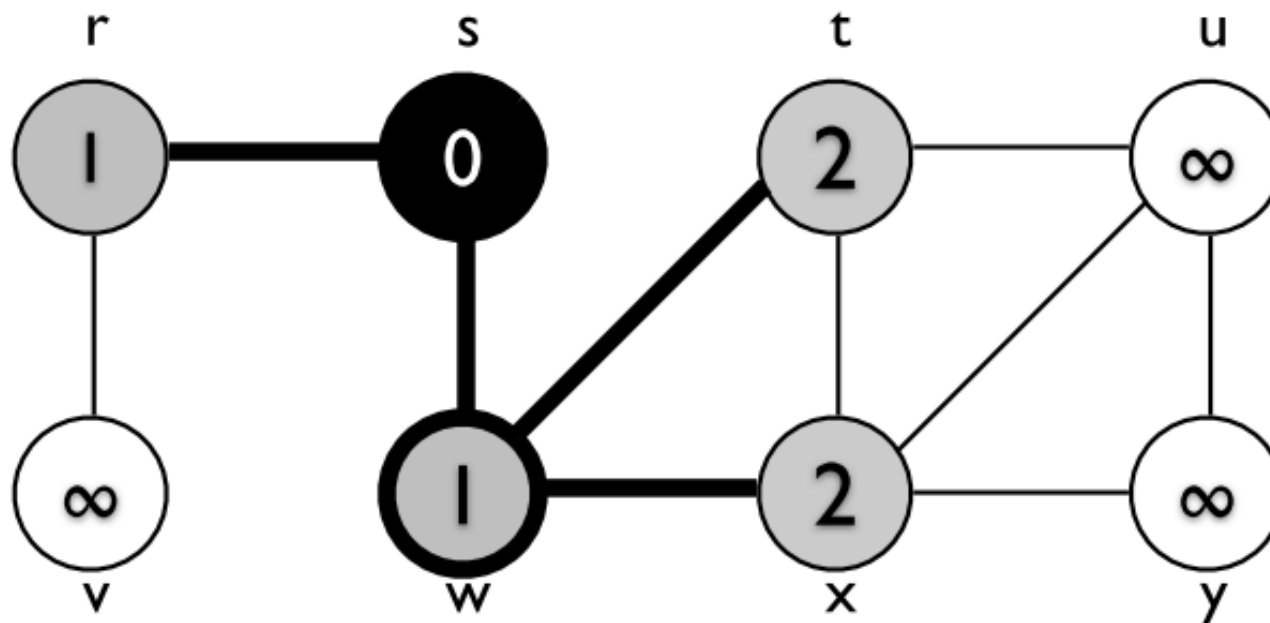
Fila Q

w	r
1	1

Nível



# EXECUÇÃO DO ALGORITMO

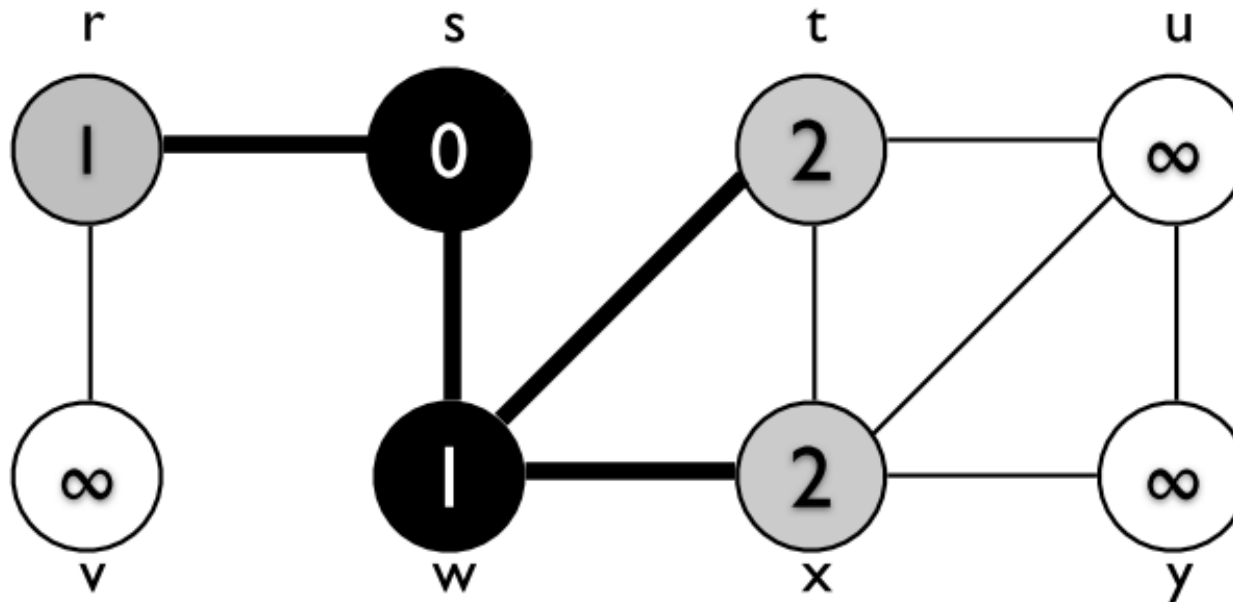


Fila Q  
Nível

r	t	x
1	2	2



# EXECUÇÃO DO ALGORITMO

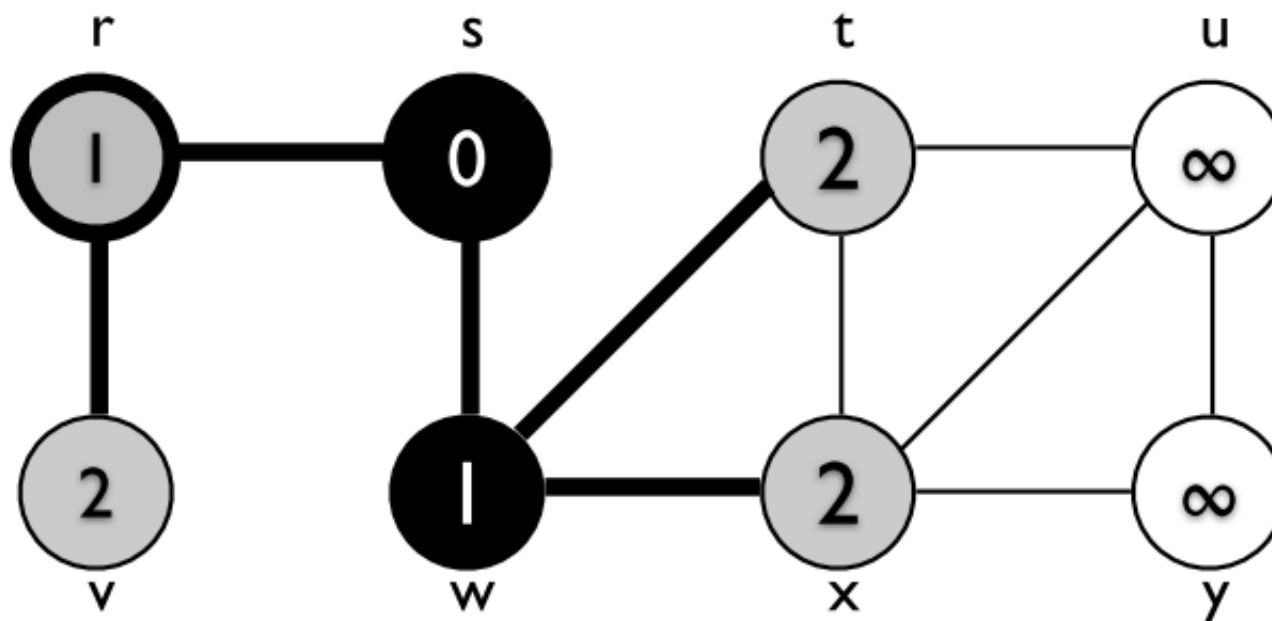


Fila Q  
Nível

r	t	x
1	2	2

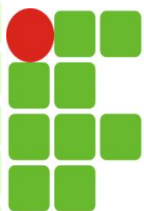


# EXECUÇÃO DO ALGORITMO

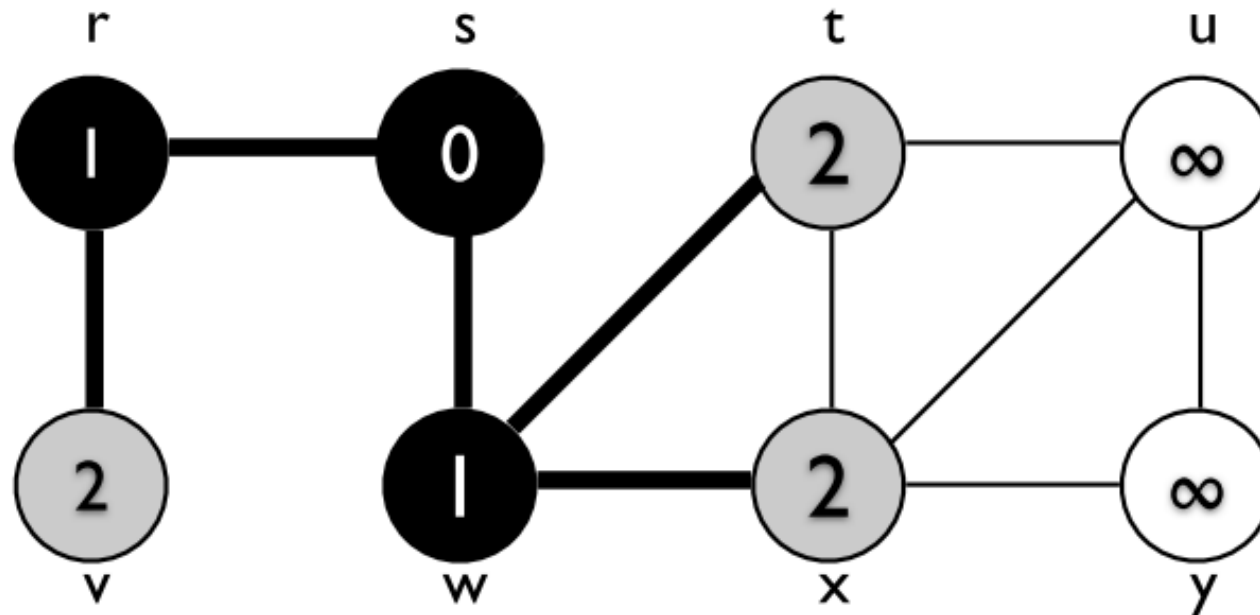


Fila Q  
Nível

t	x	v
2	2	2



# EXECUÇÃO DO ALGORITMO

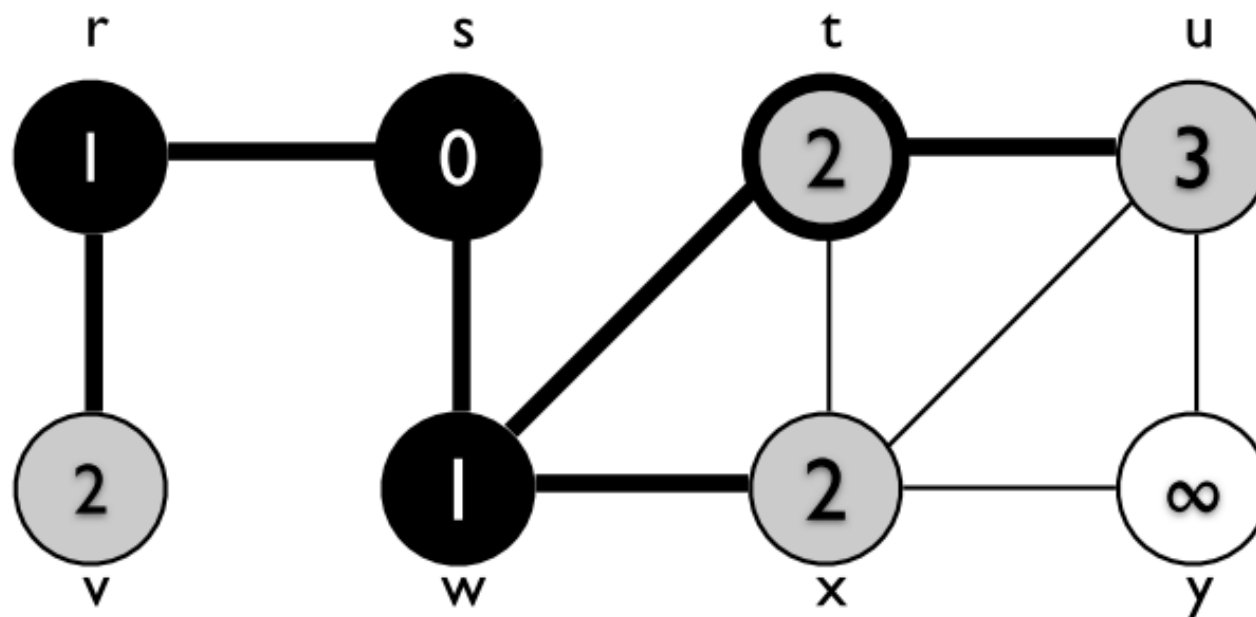


Fila Q  
Nível

t	x	v
2	2	2



# EXECUÇÃO DO ALGORITMO



Fila Q

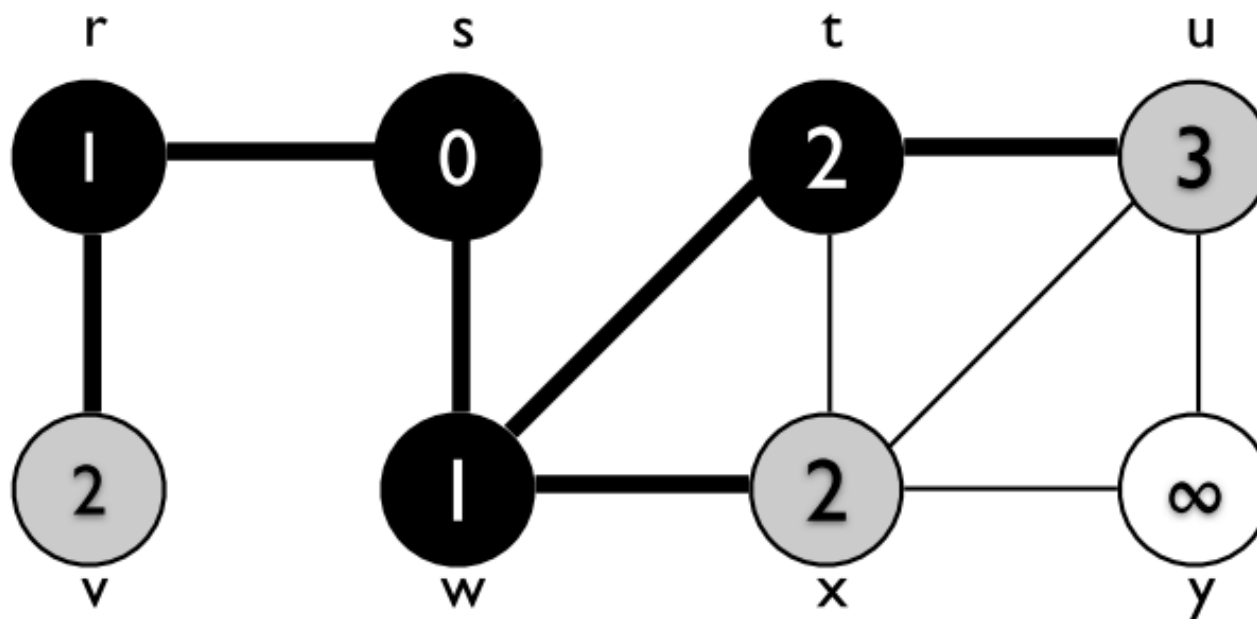
x	v	u
2	2	3

Nível



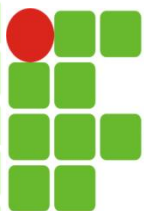


# EXECUÇÃO DO ALGORITMO

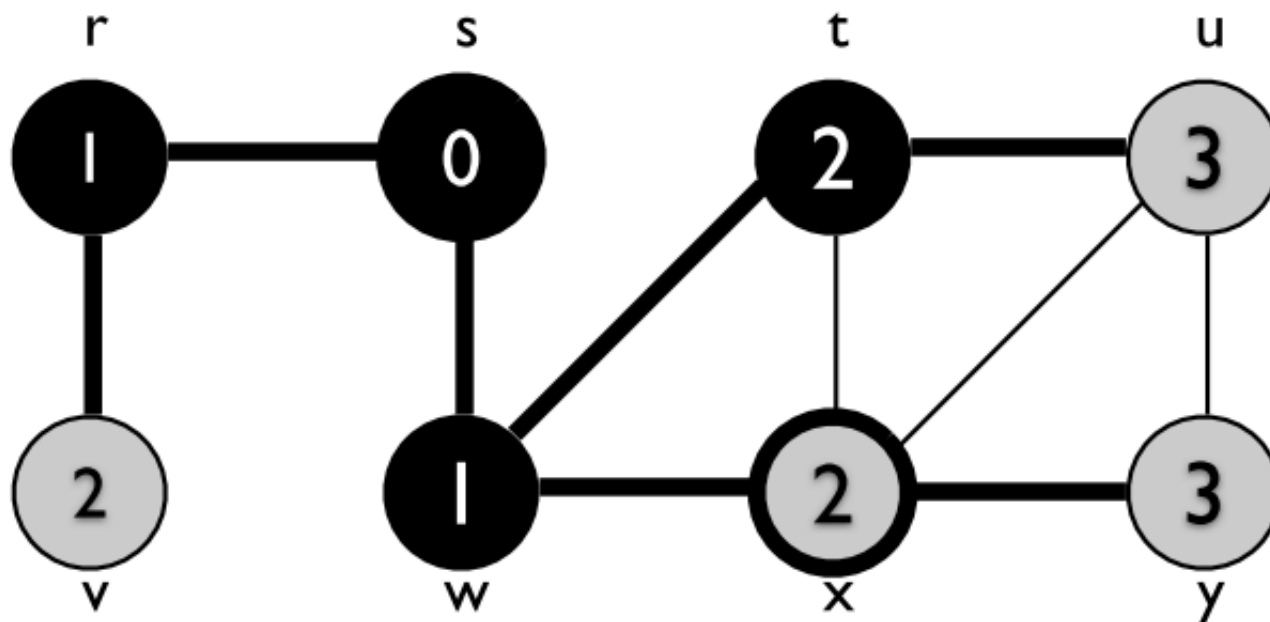


Fila Q  
Nível

x	v	u
2	2	3



# EXECUÇÃO DO ALGORITMO

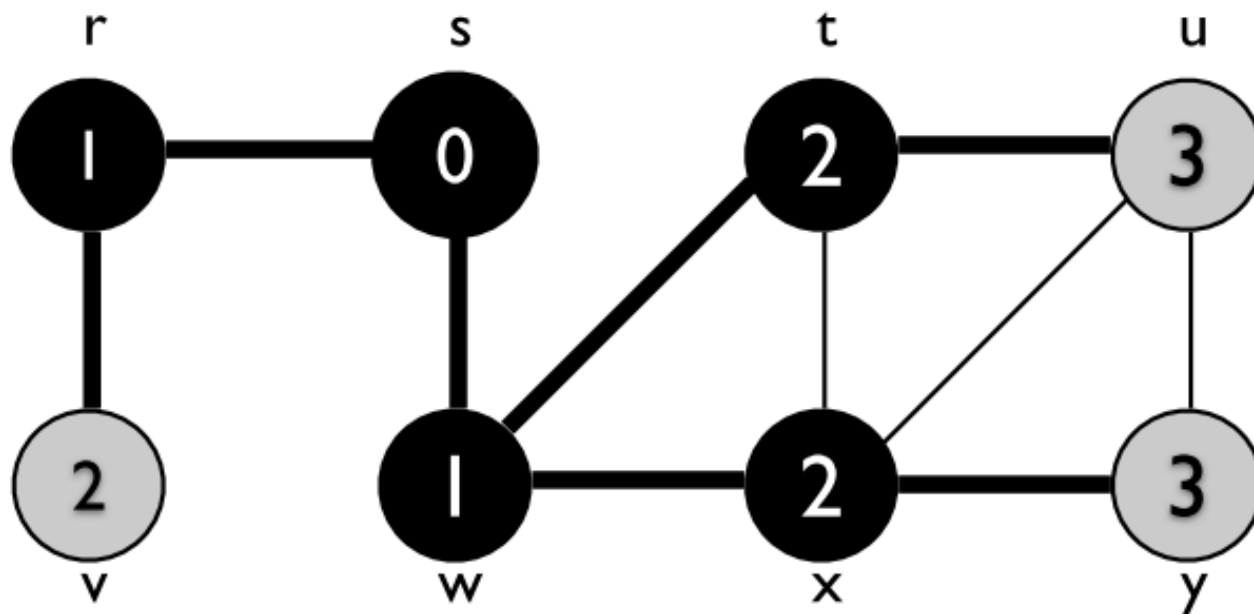


Fila Q  
Nível

v	u	y
2	3	3



# EXECUÇÃO DO ALGORITMO

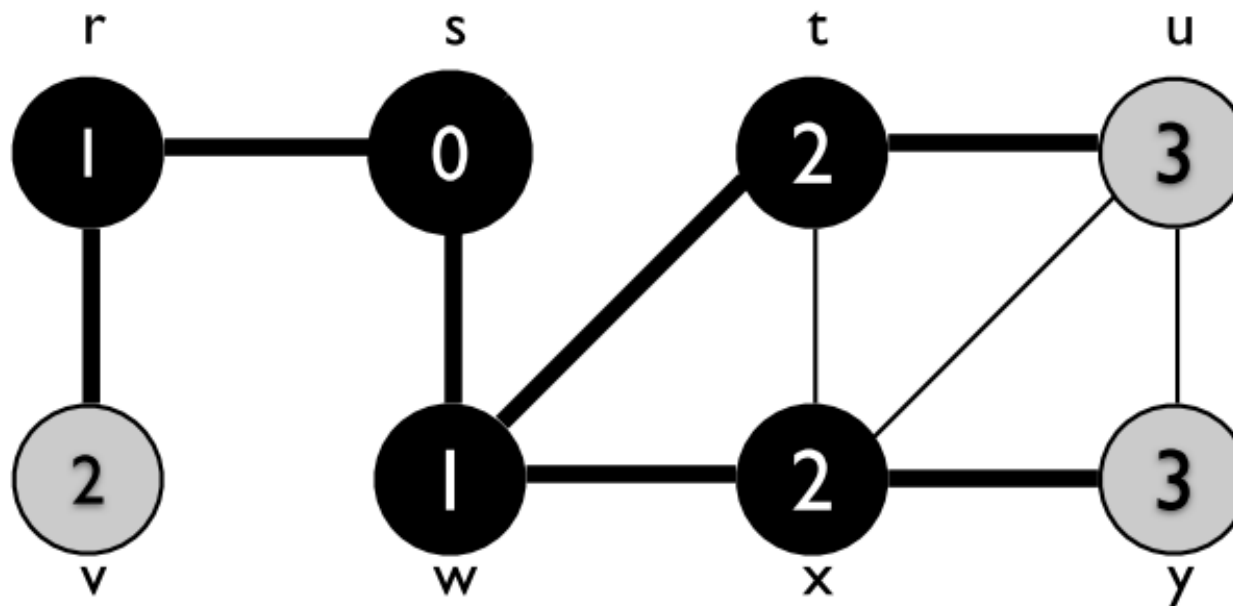


Fila Q  
Nível

v	u	y
2	3	3

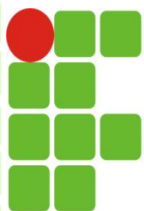


# EXECUÇÃO DO ALGORITMO

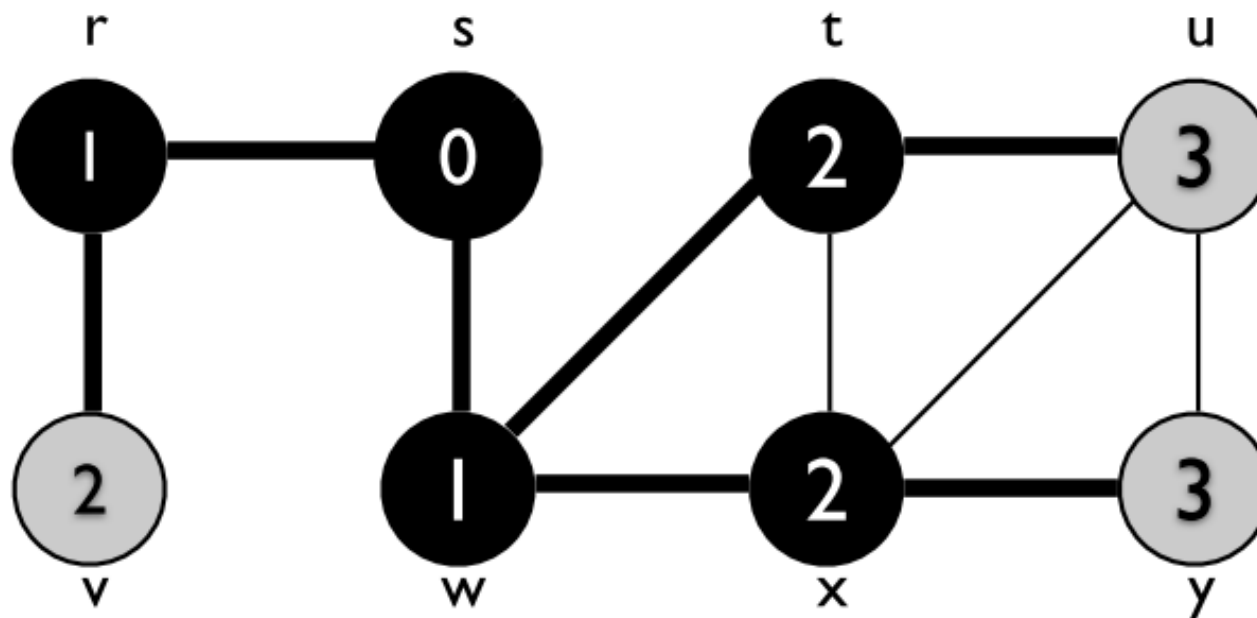


Fila Q  
Nível

u	y
3	3



# EXECUÇÃO DO ALGORITMO



Fila Q

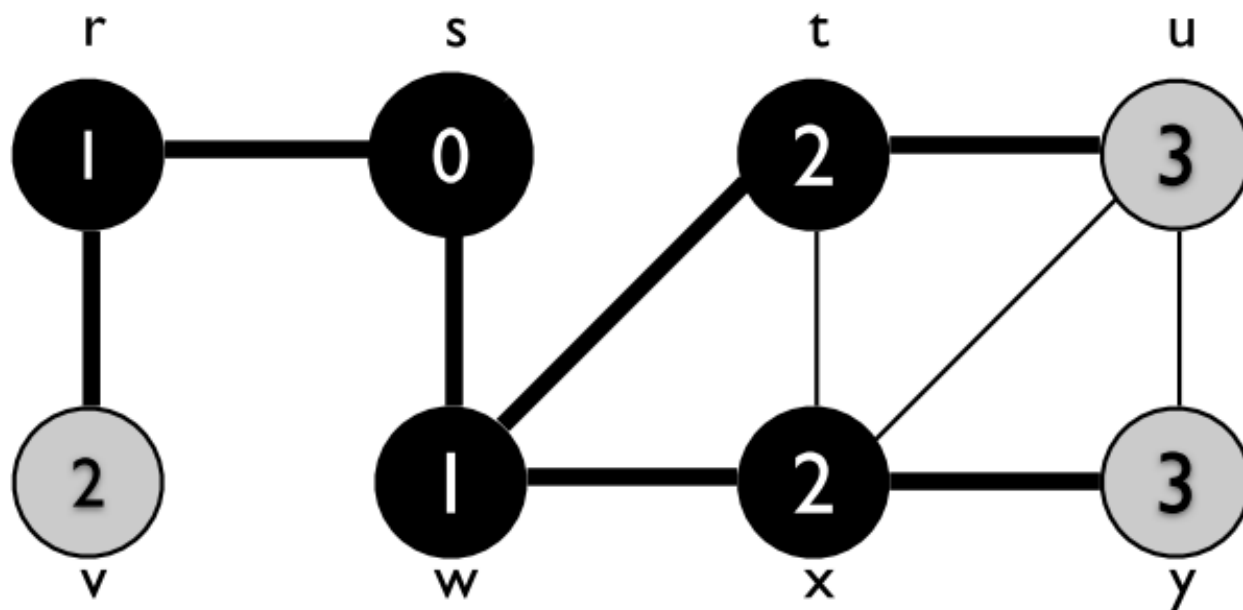
y

Nível

3



# EXECUÇÃO DO ALGORITMO



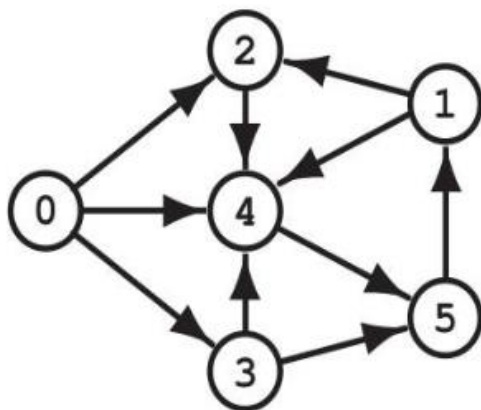
Fila Q  
Nível

$\emptyset$
—



## EXEMPLO 2

### ■ Dado o Digrafo:



Fila		0	1	2	3	4	5
0		0	-	-	-	-	-
	2	3	4				
		3	4				
			4	5			
				5			
					1		
1		0	5	1	2	3	4
		0	5	1	2	3	4
		0	5	1	2	3	4
		0	5	1	2	3	4
		0	5	1	2	3	4
		0	5	1	2	3	4
		0	5	1	2	3	4

- Percebe-se que que, no processo de busca em largura, os vértices foram descobertos na ordem: 0 2 3 4 5 1, ou seja a última linha do vetor utilizado para numeração.



# CONCLUSÃO

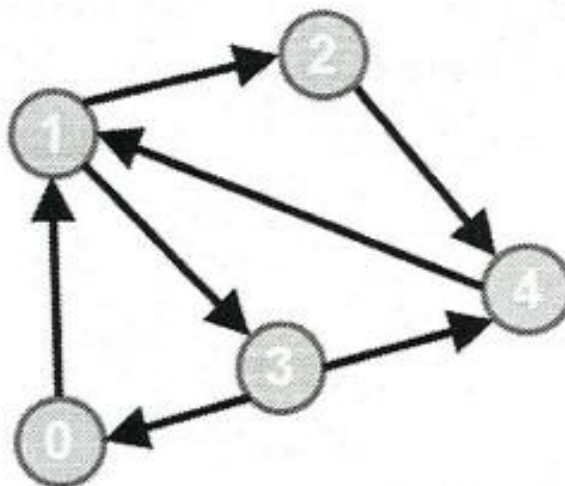
- Todo vértice que está na fila já foi numerado;
- Se um vértice  $x$  já foi numerado mas algum de seus vizinhos  $y$  ainda não foi numerado, então  $x$  está na fila;
- Cada vértice do grafo entra na fila no máximo uma vez;





# EXERCÍCIO

- Dado o Grafo apresentado execute a busca em largura, partindo do vértice 0, registrando cada etapa do processo.





# REFERÊNCIAS

- PEREIRA, Silvio do Lago. Estrutura de Dados Fundamentais: Conceitos e Aplicações, 12. Ed. São Paulo, Érica, 2008.
- BACKES, André Ricardo, Estrutura de dados descomplicada: em linguagem C, 1 Ed. – Rio de Janeiro: Elsevier, 2016.
- ROCHA, Anderson, Grafos – Representações, buscas e Aplicações.
- SENGHER, H., Notas de Aula, Universidade de São Judas Tadeu, 1999.
- WALDEMAR Celes, Renato Cerqueira, José Lucas Rangel, Introdução a Estruturas de Dados, Editora Campus (2004).
- VELOSO, Paulo. SANTOS, Celso dos. AZEVEDO, Paulo. FURTADO, Antonio. Estrutura de dados. Rio de Janeiro: Ed. Elsevier, 1983 27ª reimpressão.
- <https://www.cs.usfca.edu/~galles/visualization/Algorithms.html>
- ZIVIANI, Nivio. Projeto de Algoritmos com implementações em Java e C++, 2007.