



Estruturas de Dados II

Busca em Grafos (continuação)

Prof. Leonardo C. R. Soares - *leonardo.soares@ifsudestemg.edu.br*

Instituto Federal do Sudeste de Minas Gerais

23 de agosto de 2024

^aEste material é fortemente baseado nas notas de aula do professor Marco Antonio Moreira de Carvalho - UFOP





Busca em largura - BFS

Características

A **Busca em largura** visita todos os vértices de um grafo usando como critério o vértice *visitado menos recentemente e cuja vizinhança ainda não foi explorada*.

Característica principal: Utiliza uma **fila** para guiar a busca.





Busca em largura - BFS

Características

A **Busca em largura** visita todos os vértices de um grafo usando como critério o vértice *visitado menos recentemente e cuja vizinhança ainda não foi explorada*.

Característica principal: Utiliza uma **fila** para guiar a busca.

- ▶ Inicialmente considera-se os vértices com distância 0 do vértice inicial;
- ▶ Na iteração 1, visita-se todos os vértices com distância 1; prosseguindo de modo genérico, na iteração d será adicionada uma camada com todos os vértices com distância d do vértice inicial;
- ▶ Cada novo vértice visitado é adicionado em uma fila Q ;
- ▶ Cada vértice da fila é removido depois que toda a vizinhança for visitada;
- ▶ A busca termina quando a fila se torna vazia.



Busca em largura - BFS

Entrada: Grafo $G=(V, A)$, vértice inicial v

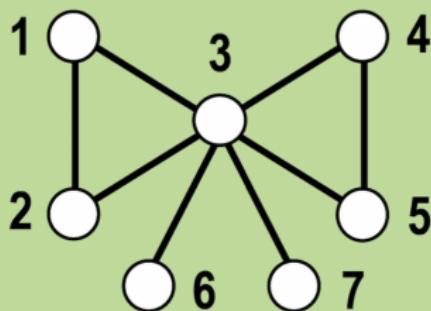
- ```

1 Crie uma fila Q vazia;
2 Marque v como visitado;
3 Insira v em Q;
4 enquanto $Q \neq \emptyset$ faça
5 $v \leftarrow$ remove elemento de Q;
6 para todo vértice w vizinho de v faça
7 se w é marcado como não visitado então
8 Visite a aresta $\{v, w\}$;
9 Insira w em Q;
10 Marque w como visitado;
11 fim
12 senão
13 se $\{v, w\}$ não foi visitada ainda então
14 Visite $\{v, w\}$;
15 fim
16 fim
17 fim
18 fim

```



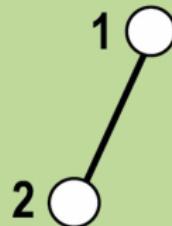
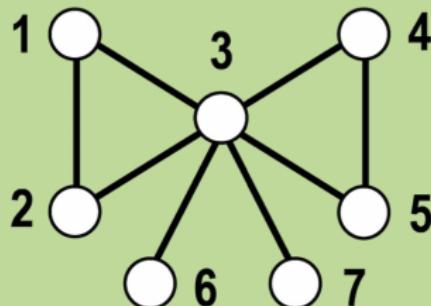
## BFS - Exemplo



1

(1) Inclusão de 1  
 $Q = \{1\}$

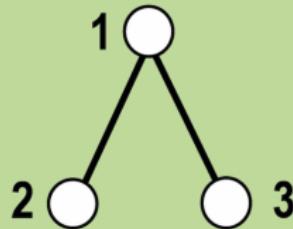
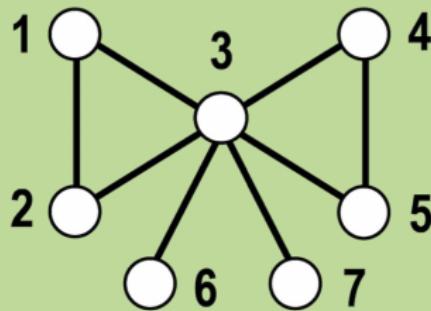
# BFS - Exemplo



(2) ArestaAresta{1, 2}  
 $Q = \{2\}$



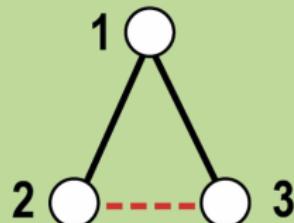
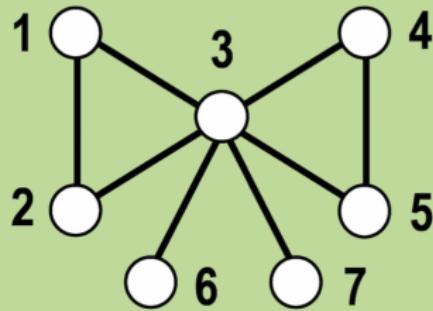
# BFS - Exemplo



(3) Aresta{1, 3}  
 $Q = \{2, 3\}$

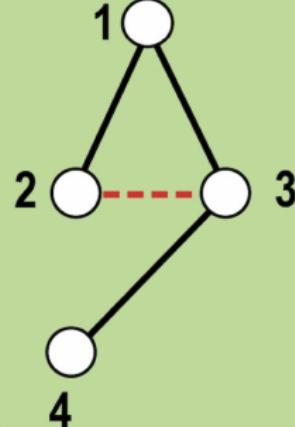
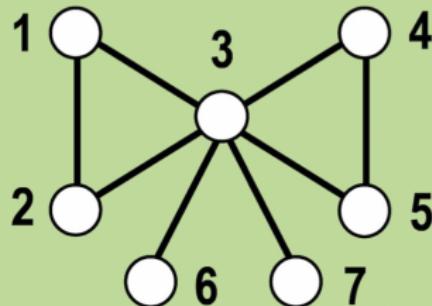


## BFS - Exemplo



(4) Aresta{2, 3}  
 $Q = \{3\}$

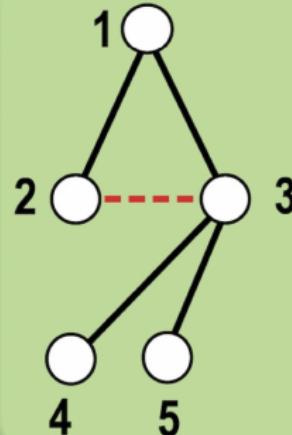
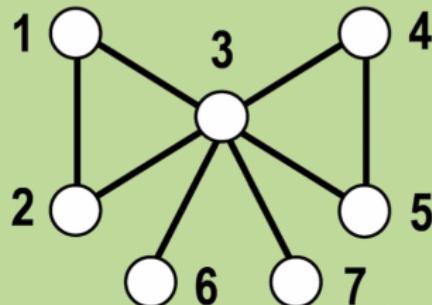
# BFS - Exemplo



(5) Aresta{3, 4}  
 $Q = \{4\}$



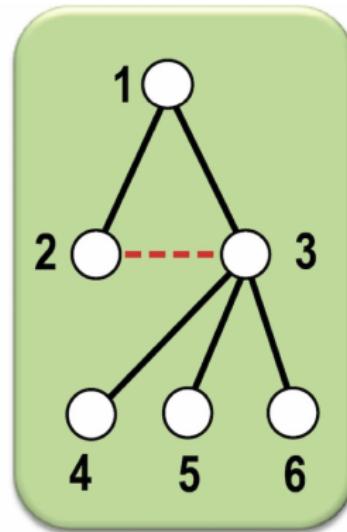
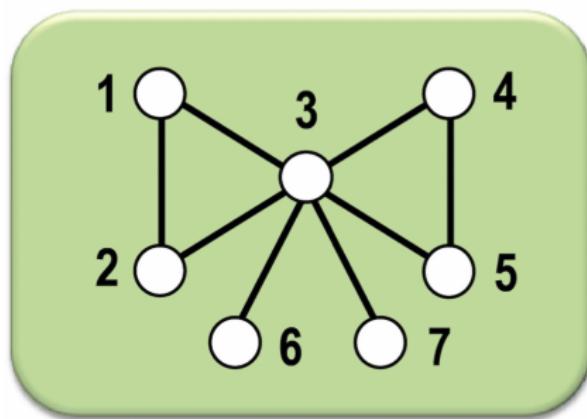
# BFS - Exemplo



(6) Aresta{3, 5}  
 $Q = \{4, 5\}$

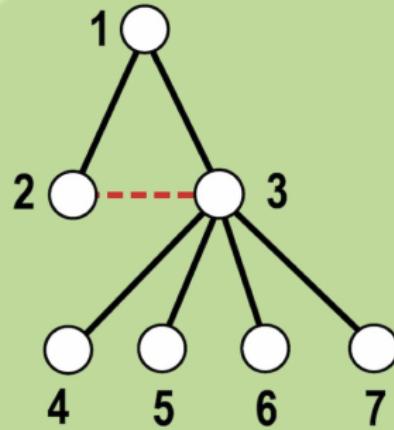
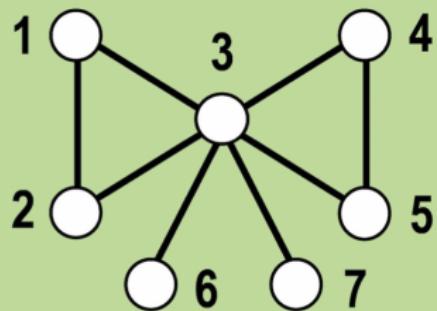


# BFS - Exemplo



(7) Aresta{3, 6}  
 $Q = \{4, 5, 6\}$

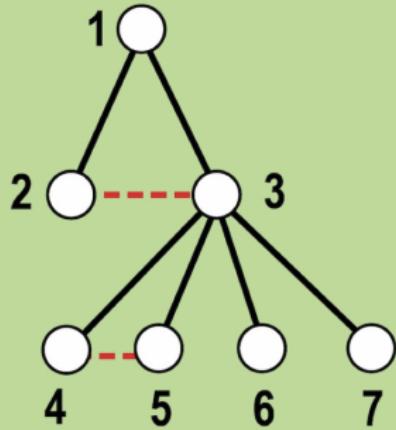
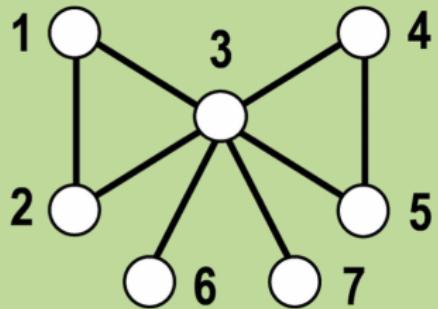
## BFS - Exemplo



(8) Aresta{3, 7}  
 $Q = \{4, 5, 6, 7\}$



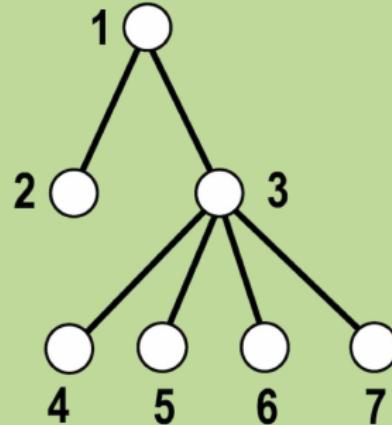
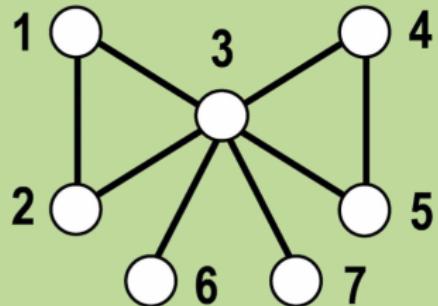
## BFS - Exemplo



(9) Aresta{4, 5}  
 $Q = \{5, 6, 7\}$



# BFS - Exemplo



Grafo original e respectiva árvore.





# DFS x BFS

## DFS - Busca em profundidade

- ▶ Incursões profundas no grafo, voltando somente quando não existem mais vértices desconhecidos pela frente;
- ▶ Marca o vértice antes de visitar toda sua vizinhança;
- ▶ Uso de pilha.

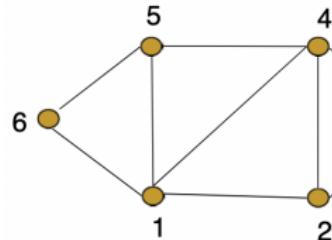
## BFS - Busca em largura

- ▶ Busca progride em “largura”: certifica-se de que vizinhos próximos sejam visitados primeiro;
- ▶ Marca o vértice depois de visitar toda sua vizinhança;
- ▶ Uso de fila.



# Exercícios

Encontre a árvore de profundidade (árvore geradora) para o grafo abaixo.  
Inicie o percurso pelo vértice 1.



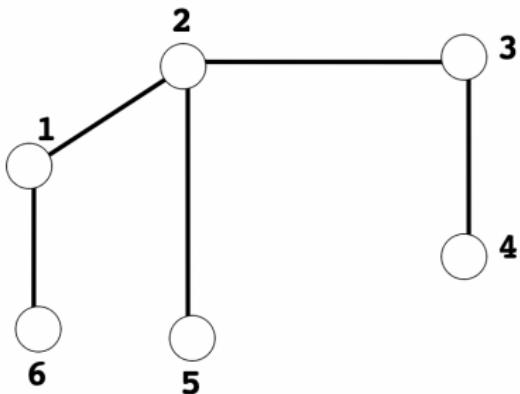
Listas de Adjacências:

- 1: (2,6,4,5)
- 2: (4,3,1)
- 3: (2,4)
- 4: (5,3,2,1)
- 5: (6,4,1)
- 6: (1,5)



# Exercícios

Encontre a árvore de profundidade (árvore geradora) para o grafo abaixo.  
Inicie o percurso pelo vértice 1. Mostre a ordem de visita aos vértices.



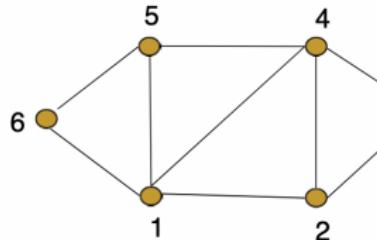
Lista de adjacências: 1: (2, 6), 2: (1, 3, 5), 3: (2, 4), 4: (3), 5: (2), 6: (1)





# Exercícios

Qual é a ordem de visita dos vértices do grafo abaixo, começando-se pelo vértice 1, em uma busca em largura? E em uma busca em profundidade?



Listas de Adjacências:

- 1: (2,6,4,5)
- 2: (4,3,1)
- 3: (2,4)
- 4: (5,3,2,1)
- 5: (6,4,1)
- 6: (1,5)





## Exercícios

- ▶ Qual dos dois métodos pode ser utilizado para se encontrar o caminho mais curto entre dois vértices em um grafo não direcionado?
- ▶ Implemente dois métodos em Java para dizer se existe ou não um caminho entre dois vértices. Utilize DFS em um e BFS no outro.





A large, colorful word cloud centered around the words "thank you". The word "thank" is at the top center in blue, "you" is below it in green, and "thank you" is repeated in red. Surrounding these are numerous other words in different languages, each with its phonetic pronunciation in parentheses. The languages include German (danke), English (thank you), Spanish (gracias), French (merci), Italian (grazie), Portuguese (obrigado), Polish (dziękuje), Russian (спасибо), Chinese (感谢), Japanese (ありがとうございます), Korean (감사합니다), and many others like Dutch, Swedish, Danish, Norwegian, etc.

