

# Busca em Largura

Profa. Sheila Moraes de Almeida

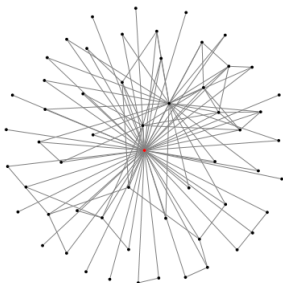
DAINF-UTFPR-PG

junho - 2018

# Busca em Grafos

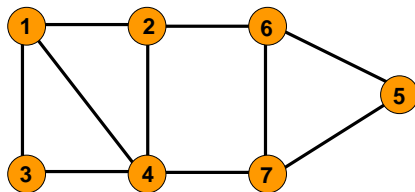
Problema fundamental em grafos:

Como explorar um grafo de forma sistemática?



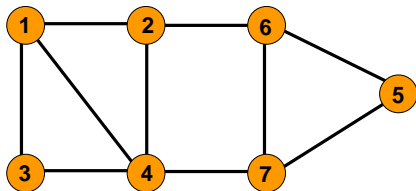
- Muitas aplicações são abstraídas como problemas de busca.
- Muitos algoritmos usam fundamentos similares.

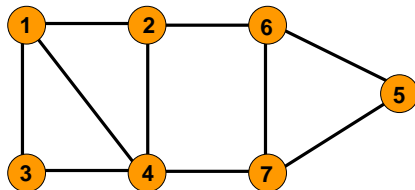
Como saber se existe caminho entre dois vértices?



Como saber se existe caminho entre dois vértices?

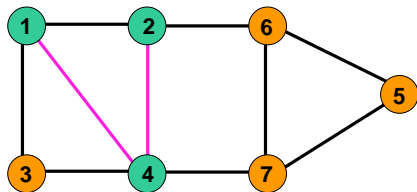
- de maneira eficiente?





**Idéia:** Evitar explorar vértices já explorados.

- marcar os vértices!



**Idéia:** Evitar explorar vértices já explorados.

- marcar os vértices: descoberto ou explorado.

## Idéias:

- escolher um vértice inicial;
- explorar e marcar vértices:
  - Descoberto: vértice foi descoberto (visitado pela primeira vez).
  - Explorado: todas as arestas incidentes ao vértice foram exploradas e vizinhos descobertos.

- Passo inicial
  - Desmarcar todos os vértices.
  - Escolher vértice inicial e marcá-lo como descoberto.
- Passo geral (enquanto houver vértice descoberto)
  - Selecionar vértice descoberto  $u$ .
  - Considerar aresta não explorada  $(u, v)$ .
  - Se  $v$  não estiver marcado, marcar  $v$  como descoberto.
  - Marcar  $u$  como explorado quando não houver mais arestas incidentes a  $u$  para serem exploradas.

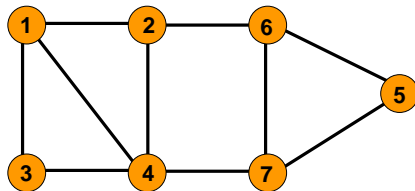


Qual a ordem de visita dos vértices?

Duas abordagens:

- explorar o vértice que foi descoberto "mais antigo"; ou
- explorar o vértice que foi descoberto "mais recente".

- Explorar vértices descobertos mais antigos primeiro.



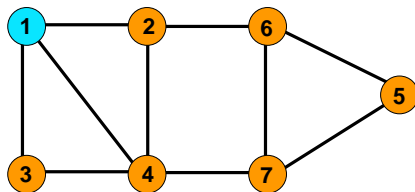
- Origem: vértice 1.
- Em que ordem os vértices são descobertos?

Assumir que as arestas são exploradas em ordem crescente dos vértices adjacentes.

(matriz ou lista de adjacências)

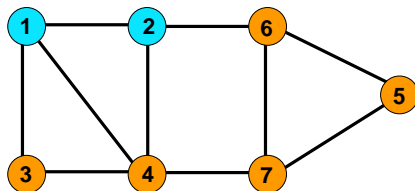
# Busca em Largura

mais antigo: 1



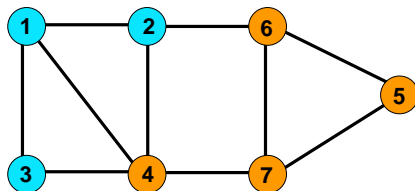
# Busca em Largura

mais antigo: 1 - 2



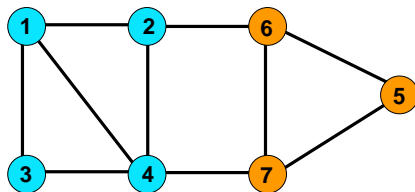
# Busca em Largura

mais antigo: 1 - 2 - 3



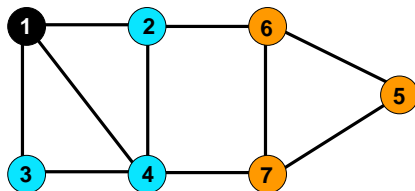
# Busca em Largura

mais antigo: 1 - 2 - 3 - 4



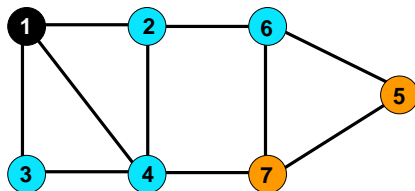
# Busca em Largura

mais antigo: 2 - 3 - 4



# Busca em Largura

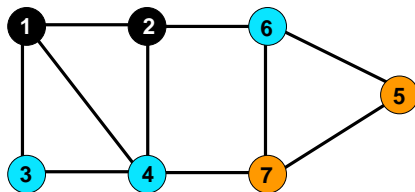
mais antigo: 2 - 3 - 4 - 6





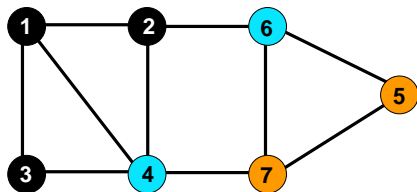
# Busca em Largura

mais antigo: 3 - 4 - 6



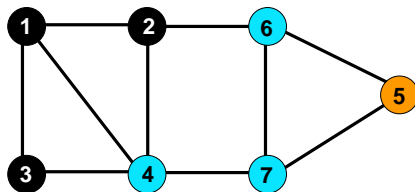
# Busca em Largura

mais antigo: 4 - 6



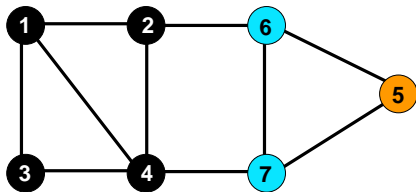
# Busca em Largura

mais antigo: 4 - 6 - 7



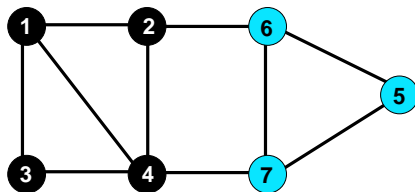
# Busca em Largura

mais antigo: 6 - 7



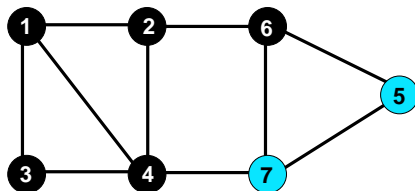
# Busca em Largura

mais antigo: 6 - 7 - 5



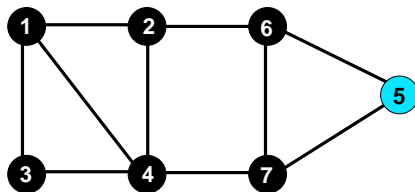
# Busca em Largura

mais antigo: 7 - 5



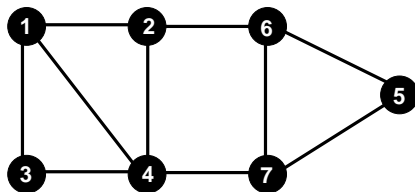
# Busca em Largura

mais antigo: 5



# Busca em Largura

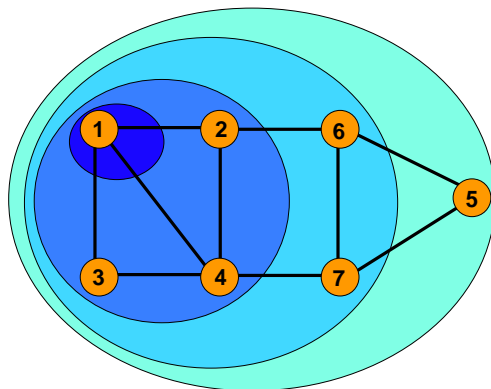
mais antigo:

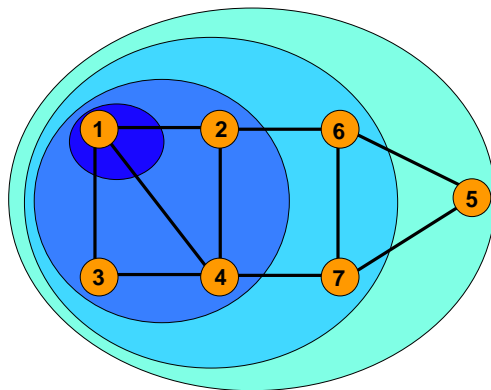




## Interpretação

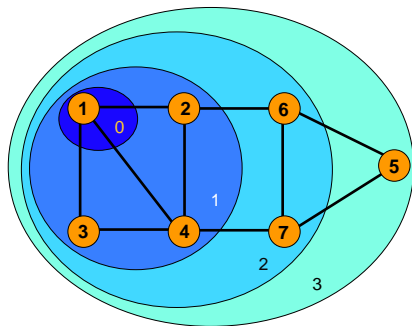
- onda é propagada a partir da raiz.
- onda expande em círculos, descobrindo vértices alcançáveis!





## Busca em Largura - Breadth First Search (BFS)

# Busca em Largura



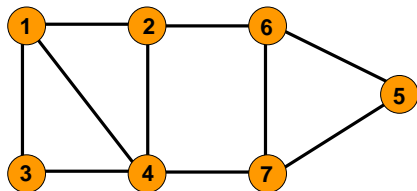
$L_0$ : vértice de origem

$L_i$ : conjunto de vértices da camada  $i$ ,  $i = 0, 1, 2, \dots$

$L_{i+1}$ : conjunto de vértices que **não fazem parte** das camadas anteriores e **são adjacentes** a algum vértice da camada  $L_i$ .

# Busca em Largura

Exemplo:



$L_0$ : vértice 2

$L_1$ : ?

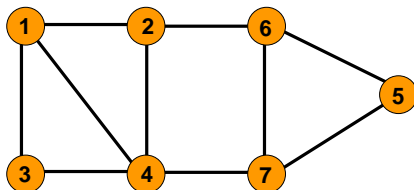
$L_2$ : ?

# Distância

**Distância** é o comprimento do **menor** caminho simples entre dois vértices.

- É uma função  $d(u, v)$ , onde  $u$  e  $v$  são vértices.
  - infinito quando não há caminho

Exemplo:



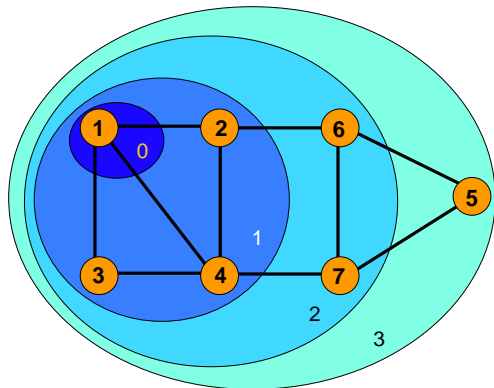
$$d(1, 2) = 1$$

$$d(6, 3) = ?$$

$$d(7, 1) = ?$$

# Camadas e Distância

Qual a relação entre esses conceitos?



$$d(1,1) = ?$$

$$d(1,2) = ?$$

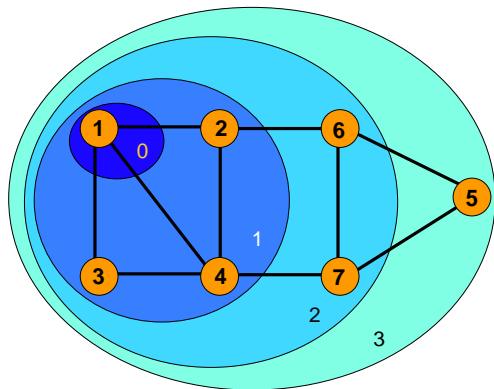
$$d(1,3) = ?$$

$$d(1,7) = ?$$

$$d(1,5) = ?$$

# Camadas e Distância

Qual a relação entre esses conceitos?



Vértices pertencentes a  $L_i$  têm distância  $i$  da origem!

Busca em Largura calcula a distância da origem!



# Busca em Largura: Implementação

Algoritmo simples, sem camadas, somente para percorrer o grafo utilizando **fila**.

desmarcar todos os vértices

definir fila  $Q$  vazia

marcar  $s$  e inserir  $s$  na fila  $Q$  //  $s$  é o vértice origem

enquanto  $Q$  não estiver vazia

    retirar  $v$  de  $Q$

    para todo vizinho  $w$  de  $v$  faça:

        se  $w$  não estiver marcado, então

            marcar  $w$

            inserir  $w$  em  $Q$

Qual é a complexidade desse Algoritmo?

- utilizando lista de adjacências

# Busca em Largura: Complexidade

desmarcar todos os vértices  $O(n)$

definir fila  $Q$  vazia

marcar  $s$  e inserir  $s$  na fila  $Q$  //  $s$  é o vértice raiz

enquanto  $Q$  não estiver vazia **cada vértice vai a  $Q$  uma única vez**

retirar  $v$  de  $Q$

para todo vizinho  $w$  de  $v$  faça: **percorre vizinhos de cada vértice**

se  $w$  não estiver marcado, então

marcar  $w$

inserir  $w$  em  $Q$

**Observe:** o laço enquanto percorre todas as arestas.

- passa 1 vez em cada ponta da aresta

- $$\sum_{v \in V(G)} d(v) = 2m \rightarrow O(m)$$

# Busca em Largura: Complexidade

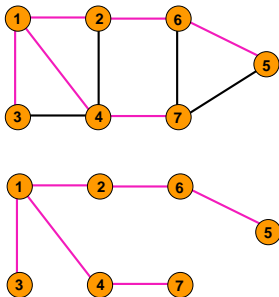
- $O(n + m)$
- Complexidade linear
  - mesma ordem de grandeza do tempo necessário para ler o grafo!
- Se o grafo for denso,  $m \sim n^2$

Uma **árvore geradora** de um grafo  $G$  é uma árvore cujo conjunto de vértices é igual ao do grafo  $G$ .

Árvores geradoras são subgrafos conexos com o menor número de arestas possível.

# Busca em Largura: Aplicação

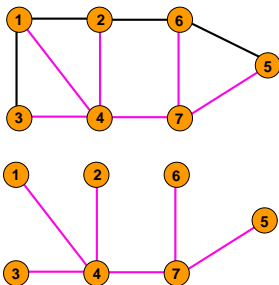
Busca em Largura apresenta uma **árvore geradora do grafo!**



- a raiz da árvore é a origem da busca em largura;
- neste exemplo: raiz = 1;

# Busca em Largura: Aplicação

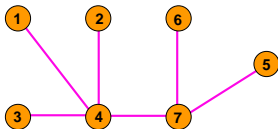
Busca em Largura apresenta uma **árvore geradora do grafo!**



- a raiz da árvore é a origem da busca em largura;
- neste exemplo: raiz = 7;

# Busca em Largura: Aplicação

Árvore Geradora define o menor caminho entre a origem (raiz) e qualquer vértice.



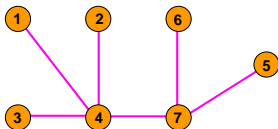
Árvore geradora com raiz no vértice 7.

O menor caminho é definido pelo seqüência de pais do vértice até a raiz.

Qual o menor caminho de 3 a 7?



# Busca em Largura: Aplicação



Árvore geradora com raiz no vértice 7.

Qual o menor caminho de 3 a 5?

- **Cuidado:** a árvore define o menor caminho para a raiz!

**Problema:** Como saber se existe caminho entre dois vértices?  
ex: como ir de São José dos Campos para Barra do Garça?

Use BFS para resolver o problema!

Como?

- marque o vértice que é a origem como raiz
- realiza a BFS
- ao terminar a BFS, se o vértice que é o destino estiver marcado, então há caminho, caso contrário, não há.

**Problema:** Como determinar se um grafo  $G$  é conexo?

**Aplicação:** Transporte aéreo comercial

- voar de qualquer cidade para qualquer cidade

**Idéia:** Utilizar a BFS!

- Escolher vértice  $v$  qualquer de  $G$ .
- Executar BFS a partir de  $v$ .
- Verificar se todos os vértices foram marcados.

Dúvidas?