Algoritmos e Estruturas de Dados II

Grafos - Busca

Profa. Elaine Parros Machado de Sousa
alterações: Cristina Dutra de Aguiar Ciferri

Material baseado em aulas dos professores:
Gustavo Batista, Robson Cordeiro, Moacir Ponti Jr.,
Maria Cristina Oliveira e Thiago A. S. Pardo

Busca em Grafos: Motivação

- Percorrer um grafo é um problema fundamental
  - deve-se ter uma forma sistemática de visitar as arestas e os vértices
  - o algoritmo deve ser suficientemente flexível para se adequar à diversidade de grafos
- Requisitos
  - não deve haver repetições (desnecessárias) de visitas a um vértice e/ou aresta
  - todos os vértices e/ou arestas devem ser visitados

BUSCA EM GRAFOS: TIPOS DE BUSCA

- Exemplos:
  - dado um grafo **G = (V, A)** e um vértice **v** ∈ **V** => encontrar todos os vértices em **G** que estão conectados a **v**.
  - dado um grafo **G = (V, A)** => visitar todos os vértices de **G**.
- Duas maneiras principais de realizar essas tarefas:
  - **Busca em profundidade**
  - **Busca em largura**

Busca em Largura: Definição

- *Breadth-First Search – BFS*
  - expande a fronteira entre vértices descobertos e não descobertos uniformemente através da largura da fronteira.
- Características
  - o algoritmo descobre todos os vértices a uma distância *k* do vértice origem antes de se descobrir qualquer vértice a uma distância *k+1*
  - a busca em largura permite descobrir todos os vértices alcançáveis a partir de um vértice de origem **u**, com o menor número de arestas entre **u** e todos os outros vértices

Busca em Largura: Estratégia

- Cada vértice é colorido de branco, cinza ou preto.
  - todos os vértices são inicialmente brancos.
  - quando um vértice **v** é "descoberto" pela primeira vez ele torna-se cinza.
  - quando todos os vértices adjacentes a **v** forem "descobertos", **v** torna-se preto.

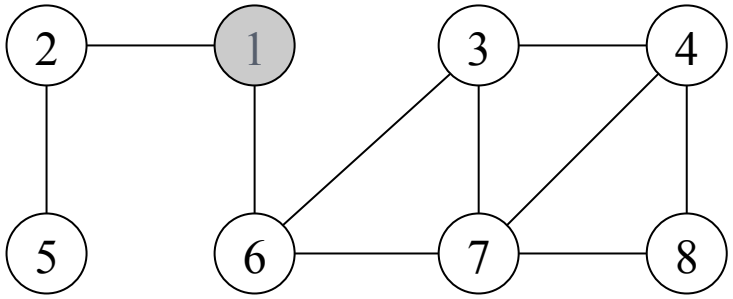
Busca em Largura: Estratégia

- Observações
  - vértices cinza e preto já foram "descobertos", mas são diferenciados para assegurar que a busca ocorra em largura
  - se **(u,v)** $\in$ **A** e o vértice **u** é preto, então o vértice ***v*** tem que ser cinza ou preto.
    - todos os vértices adjacentes a um vértice preto já foram "descobertos"
  - vértices cinza podem ter alguns vértices adjacentes brancos, representando a fronteira entre vértices "descobertos" e não "descobertos".

Busca em Largura: Fila

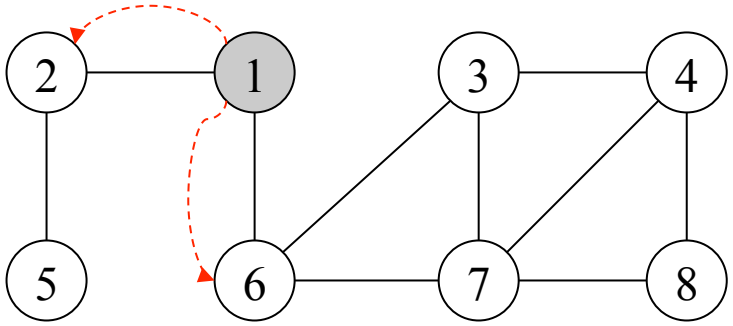
- Uso de uma fila para organizar os vértices que devem ser descobertos
  1. a fila começa com o vértice origem
  2. o primeiro vértice da fila é recuperado e processado, sendo que seus vértices adjacentes são inseridos no final da fila
  3. se a fila está vazia, o processo termina. Caso contrário, volta-se ao passo 2

Busca em Largura: Exemplo



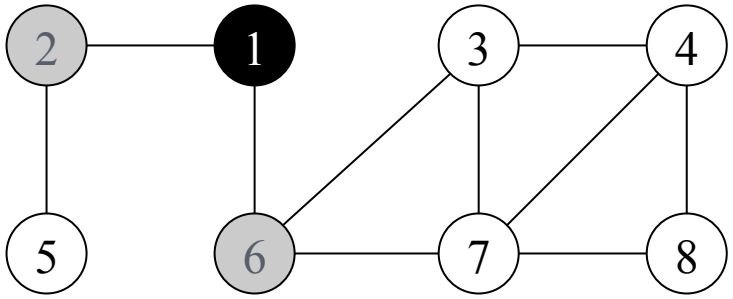
Vértice origem: 1
Distância k do vértice origem: 0
Ação: vértice 1 torna-se cinza

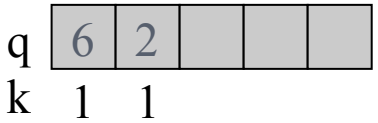
Busca em Largura: Exemplo



Vértices não descobertos adjacentes a 1: 2, 6
Distância k do vértice origem: 1

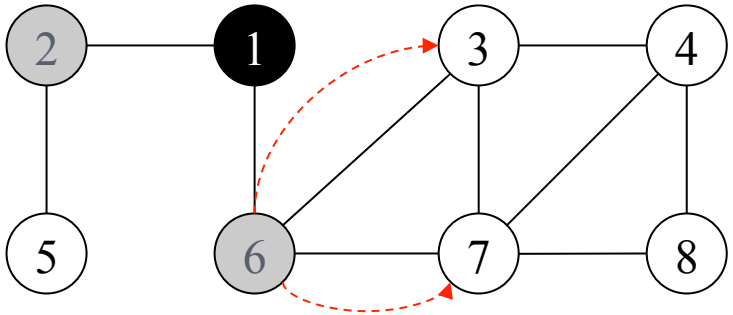
Busca em Largura: Exemplo

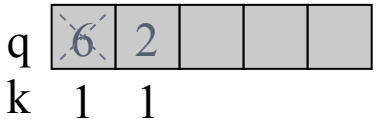




Vértices não descobertos adjacentes a 1: 2, 6
Distância k do vértice origem: 1
Ação: vértice 1 torna-se preto e vértices 2 e 6 tornam-se cinza

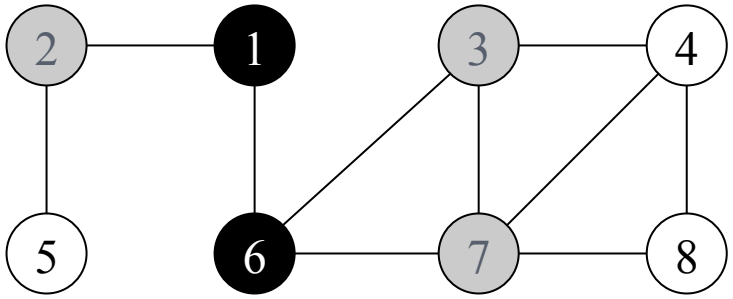
Busca em Largura: Exemplo





Vértices não descobertos adjacentes a 6: 3, 7
Distância k do vértice origem: 2

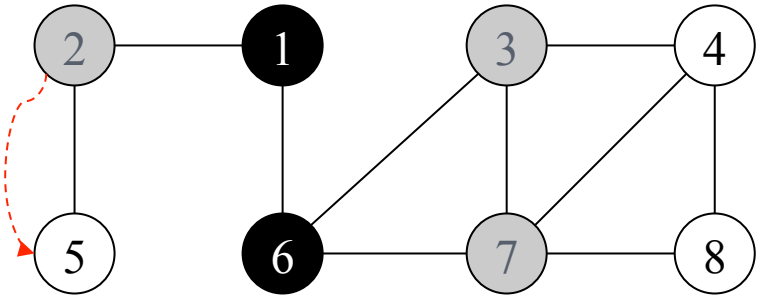
Busca em Largura: Exemplo

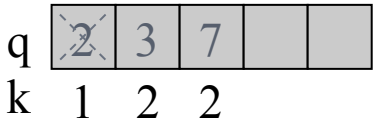


q	2	3	7		
k	1	2	2		

Vértices não descobertos adjacentes a 6: 3, 7
Distância k do vértice origem: 2
Ação: vértice 6 torna-se preto e vértices 3 e 7 tornam-se cinza

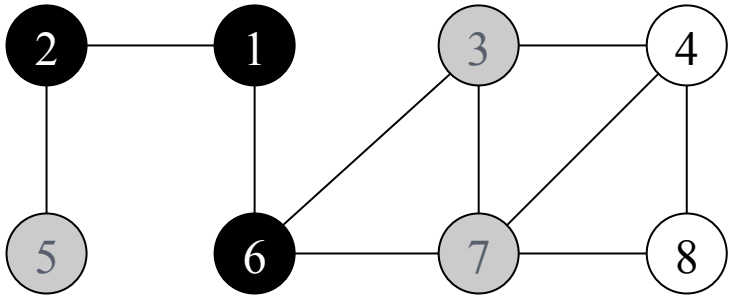
Busca em Largura: Exemplo





Vértices não descobertos adjacentes a 2: 5
Distância k do vértice origem: 2

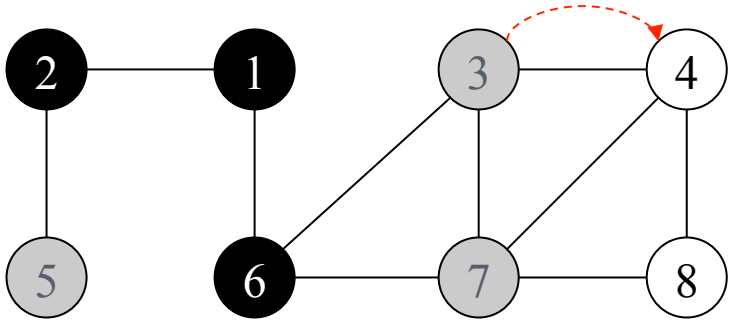
Busca em Largura: Exemplo



q	3	7	5		
k	2	2	2		

Vértices não descobertos adjacentes a 2: 5
Distância k do vértice origem: 2
Ação: vértice 2 torna-se preto e vértice 5 torna-se cinza

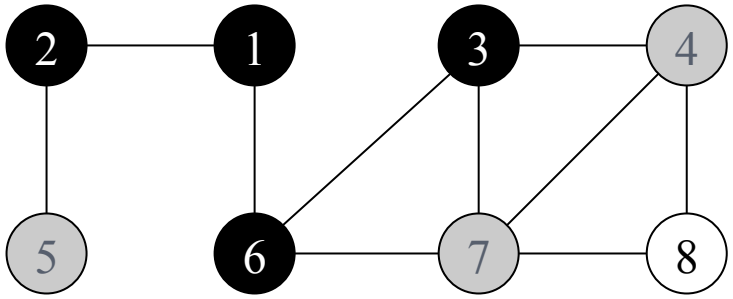
Busca em Largura: Exemplo



q	~~3~~	7	5		
k	2	2	2		

Vértices não descobertos adjacentes a 3: 4
Distância k do vértice origem: 3

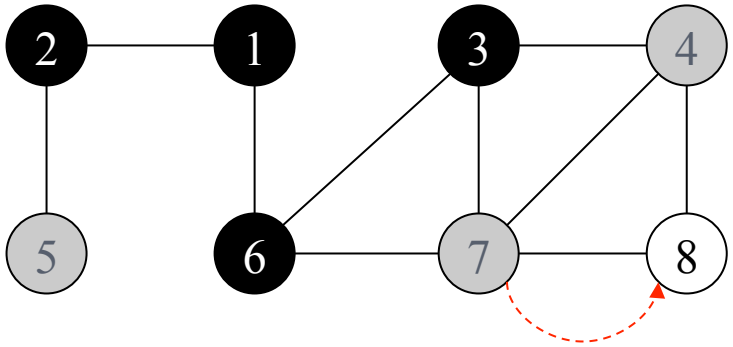
Busca em Largura: Exemplo

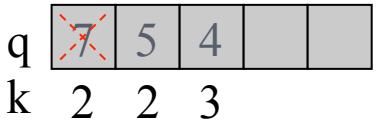


q	7	5	4		
k	2	2	3		

Vértices não descobertos adjacentes a 3: 4
Distância k do vértice origem: 3
Ação: vértice 3 torna-se preto e vértice 4 torna-se cinza

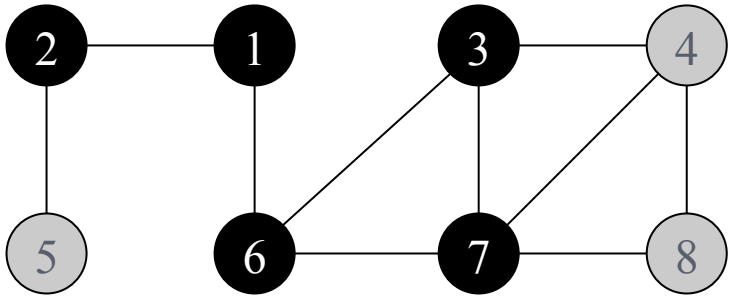
Busca em Largura: Exemplo





Vértices não descobertos adjacentes a 7: 8
Distância k do vértice origem: 3

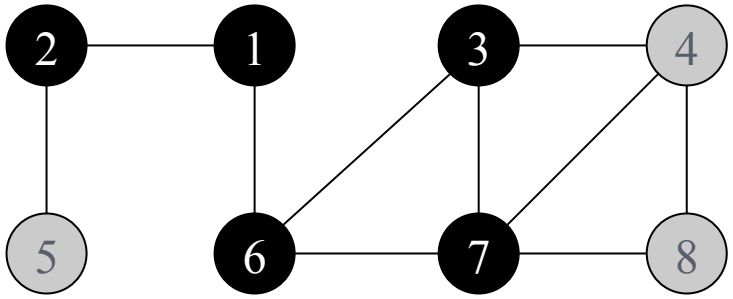
Busca em Largura: Exemplo



q	5	4	8		
k	2	3	3		

Vértices não descobertos adjacentes a 7: 8
Distância k do vértice origem: 3
Ação: vértice 7 torna-se preto e vértice 8 torna-se cinza

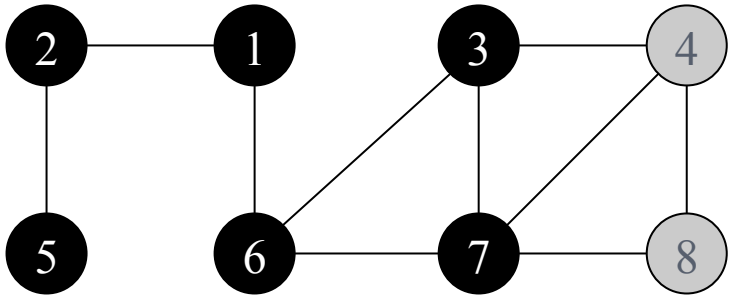
Busca em Largura: Exemplo



q	~~5~~	4	8		
k	2	3	3		

Vértices não descobertos adjacentes a 5: nenhum
Distância k do vértice origem: -

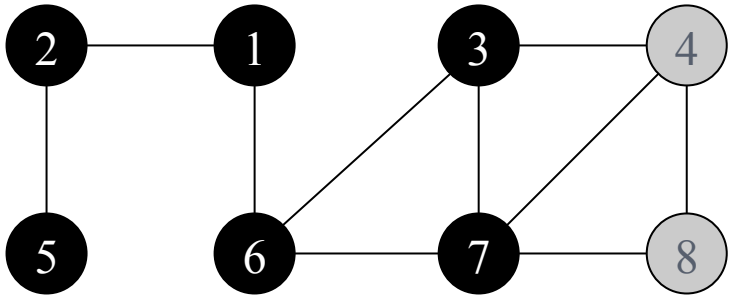
Busca em Largura: Exemplo

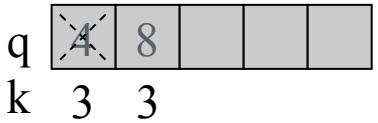


q	4	8			
k	3	3			

Vértices não descobertos adjacentes a 5: nenhum
Distância k do vértice origem: -
Ação: vértice 5 torna-se preto

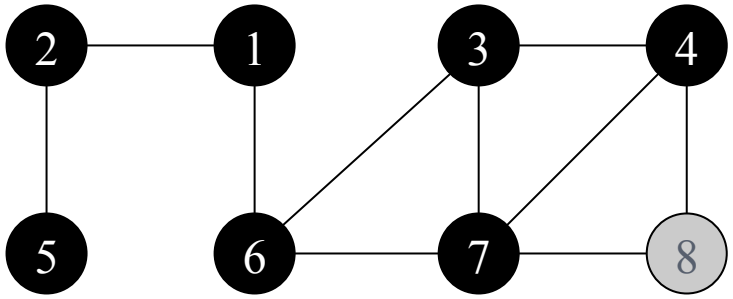
Busca em Largura: Exemplo





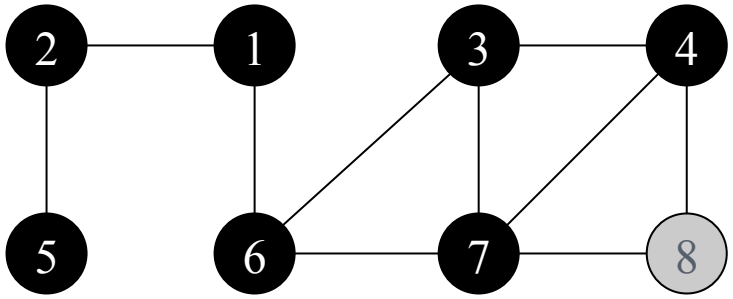
Vértices não descobertos adjacentes a 4: nenhum
Distância k do vértice origem: -

Busca em Largura: Exemplo



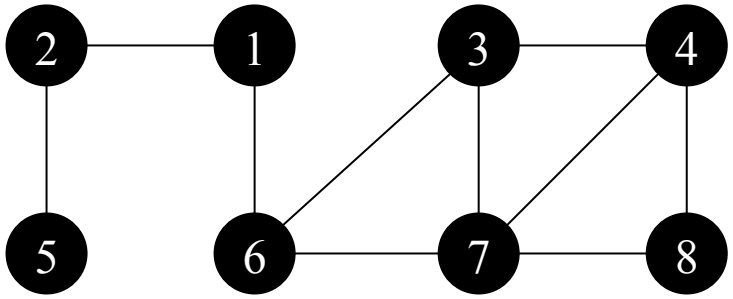
Vértices não descobertos adjacentes a 4: nenhum
Distância k do vértice origem: -
Ação: vértice 4 torna-se preto

Busca em Largura: Exemplo



Vértices não descobertos adjacentes a 8: nenhum
Distância k do vértice origem: -

Busca em Largura: Exemplo



Vértices não descobertos adjacentes a 8: nenhum
Distância k do vértice origem: -
Ação: vértice 8 torna-se preto

Busca em Largura: Complexidade

$$O(|V| + |A|)$$

- Característica
  - linear em relação ao tamanho da representação do grafo usando listas de adjacência
- $O(|V|)$
  - todos os vértices são colocados na fila no máximo uma vez, ou seja, são executadas $|V|$ iterações com custo $O(1)$ cada uma delas
- $O(|A|)$
  - cada lista de adjacência é percorrida no máximo uma vez quando o vértice é retirado da fila

Busca em Largura: Uso

- O algoritmo é base para outros algoritmos importantes:
  - encontrar a árvore geradora mínima (**MST**) – **Algoritmo de Prim**
  - encontrar o caminho mais curto de um vértice v a todos os outros – **Algoritmo de Dijkstra**

a busca em largura resulta no caminho mais curto entre o vértice origem u e um vértice qualquer v.