Teoria dos Grafos

Berilhes

■ Na pesquisa primeiro em largura nós começamos de um vértice específico s e marcamos este como "visitado".

- Na pesquisa primeiro em largura nós começamos de um vértice específico s e marcamos este como "visitado".
- Todos os outros vértices do grafo *G* estão marcados como "não visitados".

- Na pesquisa primeiro em largura nós começamos de um vértice específico s e marcamos este como "visitado".
- Todos os outros vértices do grafo *G* estão marcados como "não visitados".
- Então nós visitamos todos os vértices adjacentes à s.

- Na pesquisa primeiro em largura nós começamos de um vértice específico s e marcamos este como "visitado".
- Todos os outros vértices do grafo *G* estão marcados como "não visitados".
- Então nós visitamos todos os vértices adjacentes à s.
- A seguir, nós visitamos todos os vértices não visitados adjacentes ao primeiro vértice na lista de adjacências de s.



■ Para implementar este método de pesquisa, nós mantemos uma fila (Q) de vértices visitados, e

- Para implementar este método de pesquisa, nós mantemos uma fila (Q) de vértices visitados, e
- nós processamos esta fila de acordo com a disciplina primeiro a entrar - primeiro a sair.

- Para implementar este método de pesquisa, nós mantemos uma fila (Q) de vértices visitados, e
- nós processamos esta fila de acordo com a disciplina primeiro a entrar - primeiro a sair.
- Quando nós visitamos um vértice pela primeira vez, nós o colocamos no final da fila.

- Para implementar este método de pesquisa, nós mantemos uma fila (Q) de vértices visitados, e
- nós processamos esta fila de acordo com a disciplina primeiro a entrar - primeiro a sair.
- Quando nós visitamos um vértice pela primeira vez, nós o colocamos no final da fila.
- Nós tomamos um vértice v da cabeça da fila e atravessamos todas as arestas incidentes em v,

- Para implementar este método de pesquisa, nós mantemos uma fila (Q) de vértices visitados, e
- nós processamos esta fila de acordo com a disciplina primeiro a entrar - primeiro a sair.
- Quando nós visitamos um vértice pela primeira vez, nós o colocamos no final da fila.
- Nós tomamos um vértice v da cabeça da fila e atravessamos todas as arestas incidentes em v,
- fazendo v o pai de todos os vértices adjacentes a ele ainda não visitados,

- Para implementar este método de pesquisa, nós mantemos uma fila (Q) de vértices visitados, e
- nós processamos esta fila de acordo com a disciplina primeiro a entrar - primeiro a sair.
- Quando nós visitamos um vértice pela primeira vez, nós o colocamos no final da fila.
- Nós tomamos um vértice v da cabeça da fila e atravessamos todas as arestas incidentes em v,
- fazendo v o pai de todos os vértices adjacentes a ele ainda não visitados,
- os demais vértices adjacente, ou seja àqueles já visitados permanecem inalterados.



 Uma vez que um vértice v é retirado da fila, todos os seus vizinhos são visitados e

- Uma vez que um vértice v é retirado da fila, todos os seus vizinhos são visitados e
- *v* é marcado como explorado ou finalizado.

- Uma vez que um vértice v é retirado da fila, todos os seus vizinhos são visitados e
- *v* é marcado como explorado ou finalizado.
- Durante a execução da pesquisa primeira em largura existem três tipos de vértices:

- Uma vez que um vértice v é retirado da fila, todos os seus vizinhos são visitados e
- *v* é marcado como explorado ou finalizado.
- Durante a execução da pesquisa primeira em largura existem três tipos de vértices:
- vértices não visitados, àqueles que ainda não foram incluídos na fila;

- Uma vez que um vértice v é retirado da fila, todos os seus vizinhos são visitados e
- *v* é marcado como explorado ou finalizado.
- Durante a execução da pesquisa primeira em largura existem três tipos de vértices:
- vértices não visitados, àqueles que ainda não foram incluídos na fila;
- vértices completamente explorados, ou seja àqueles que já foram incluídos na fila e que agora estão fora da fila;

- Uma vez que um vértice v é retirado da fila, todos os seus vizinhos são visitados e
- *v* é marcado como explorado ou finalizado.
- Durante a execução da pesquisa primeira em largura existem três tipos de vértices:
- vértices não visitados, àqueles que ainda não foram incluídos na fila;
- vértices completamente explorados, ou seja àqueles que já foram incluídos na fila e que agora estão fora da fila;
- 3 vértices visitados mas não completamente explorados, isto é àqueles que estão atualmente na fila.



 Note que todos os vértices alcançáveis do vértice de origem são visitados e

- Note que todos os vértices alcançáveis do vértice de origem são visitados e
- o algoritmo produz uma árvore geradora destes vértices.

- Note que todos os vértices alcançáveis do vértice de origem são visitados e
- o algoritmo produz uma árvore geradora destes vértices.
- Esta árvore, conhecida como árvore primeiro em largura.

- Note que todos os vértices alcançáveis do vértice de origem são visitados e
- o algoritmo produz uma árvore geradora destes vértices.
- Esta árvore, conhecida como árvore primeiro em largura.
- O algoritmo também determina a distância que cada vértice v está do vértice de origem s

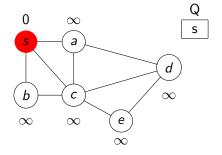
- Note que todos os vértices alcançáveis do vértice de origem são visitados e
- o algoritmo produz uma árvore geradora destes vértices.
- Esta árvore, conhecida como árvore primeiro em largura.
- O algoritmo também determina a distância que cada vértice v está do vértice de origem s
- Arestas que não estão na árvore primeiro em largura sempre conectam vértices no mesmo nível, ou vértices que estão em níveis adjacentes.

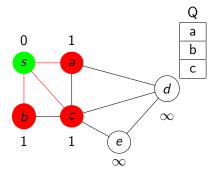
O algoritmo BFS

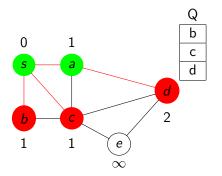
O algoritmo BFS

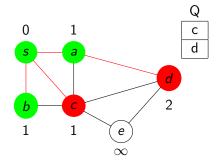
```
BFS(G,s)
     for cada vértice x \in G.V
          x. visitado \leftarrow FALSO
 3
           x.pai \leftarrow NIL
    s. visitado ← VERDADEIRO
 5 s. distancia \leftarrow 0
    InserirFila(Q, s)
     while Q \neq \emptyset
 8
           v \leftarrow \text{RemoverFila}(Q)
 9
           for cada vértice w \in v.Adj
10
                if w.visitado = FALSO
11
                      w.visitado \leftarrow VERDADEIRO
12
                      w.pai \leftarrow v
13
                      w.distancia \leftarrow v.distancia + 1
14
                      INSERIRFILA(Q, w)
                                                   4□ > 4□ > 4 = > 4 = > = 900
```

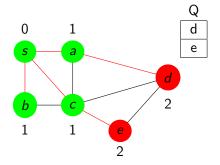
Um exemplo

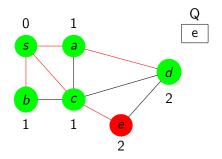


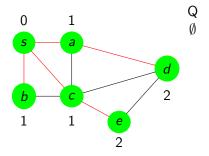


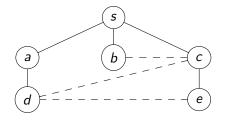












Árvore BFS