Busca em Largura
5189-32

Rodrigo Calvo
rcalvo@uem.br

Departamento de Informática – DIN
Universidade Estadual de Maringá – UEM

1° semestre de 2016

Introdução

- Dado um grafo $G = (V, A)$ e um vértice de origem s, a busca em largura (*breadth-first search* – bfs) explora sistematicamente as arestas de um grafo G até descobrir cada vértice acessível a partir de s
- O algoritmo produz uma árvore primeiro em extensão
- O algoritmo é a base para outros algoritmos importantes como o algoritmo de Prim para obter a árvore geradora mínima e o algoritmo de Djkstra para obter o caminho mínimo de um vértice até todos os outros.

Busca em Largura
Funcionamento

- Iniciando a busca em um vértice s, todas as arestas saindo de s são exploradas.
- Uma estrutura do tipo fila é utilizada para armazenar os vértices (adjacentes) recentemente visitados
- Para cada adjacente alcançado (e removido da fila), todas as arestas que saem dele também são exploradas e os novos vértices visitados são inseridos na fila
- Este processo continua até que todos os vértices alcançáveis a partir de s sejam visitados (descobertos)
- Se restarem vértices não descobertos, a busca se repetirá para estes vértices

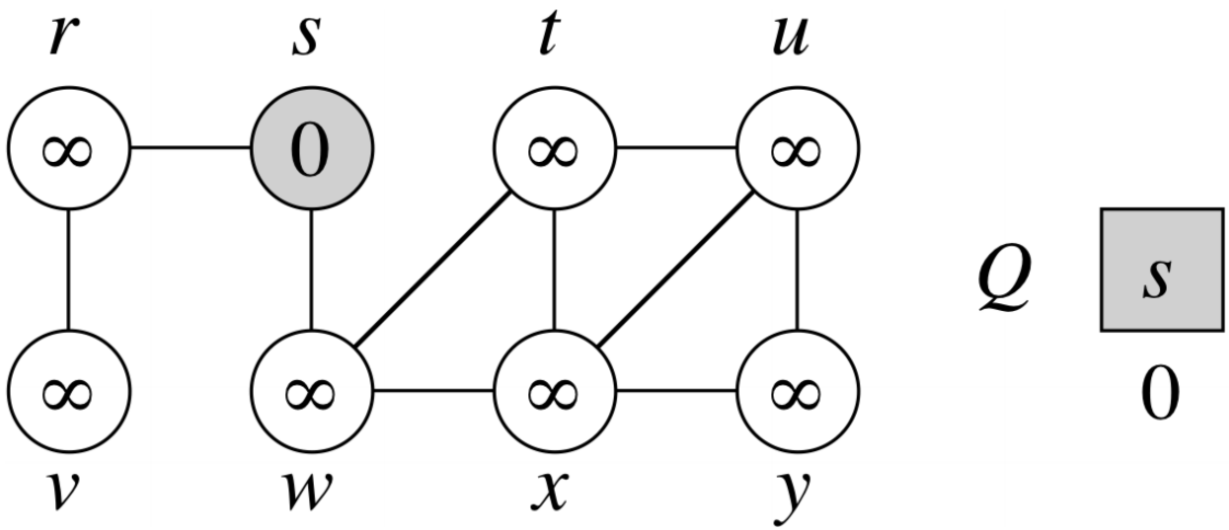
Busca em Lagura

- O algoritmo calcula a distância (menor número de arestas) deste *s* até todos os vértices acessíveis a partir de *s*
- Recebe este nome porque expande a fronteira entre vértices descobertos e não descobertos uniformemente ao longo da extensão da fronteira. Descobre todos os vértices de distância *k* de *s* antes de descobrir quaisquer vértices de distância $k + 1$
- O grafo pode ser direcionado ou não direcionado

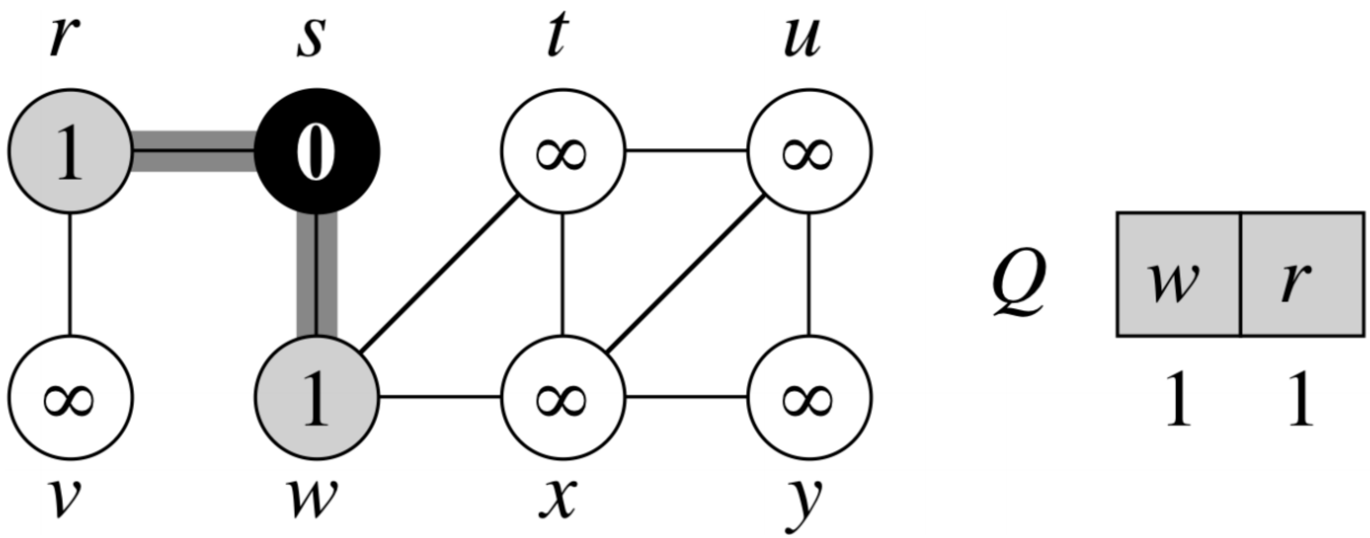
Busca em Largura
Atributos

- Para o algoritmo da Busca em Largura, o significado do atributo *v.d* á alterado
- v.d → indica o comprimento do caminho do vértice por onde a busca foi iniciada até o vértice *v*

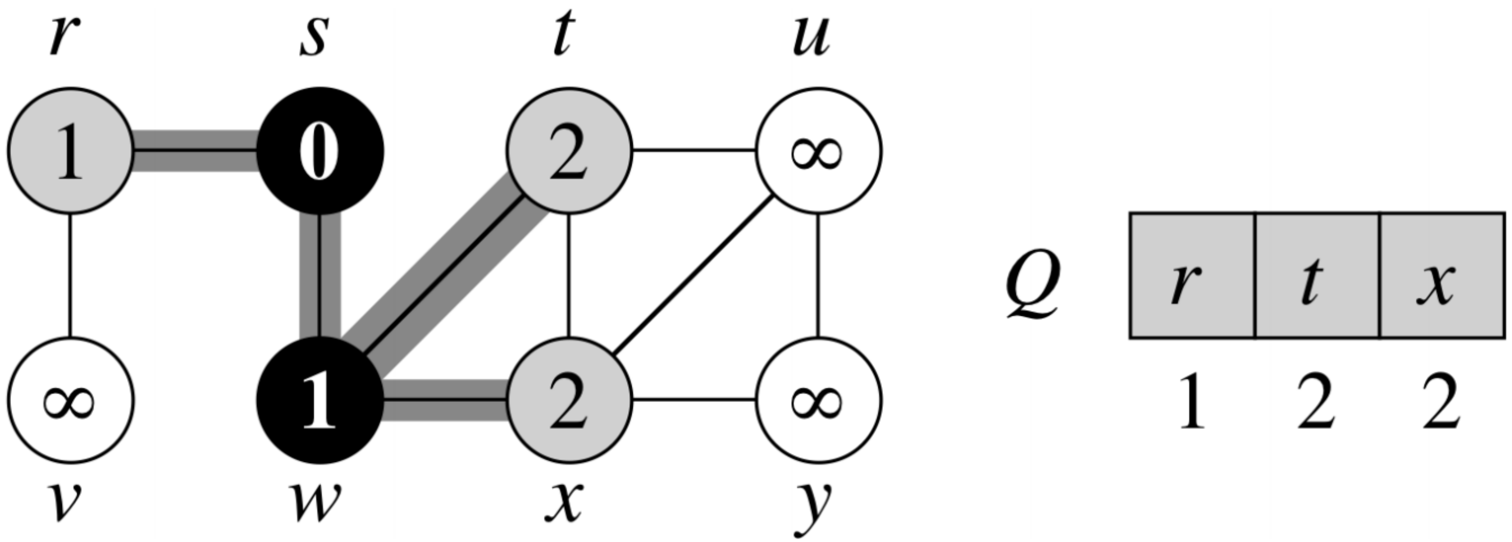
Busca em Largura: Exemplo



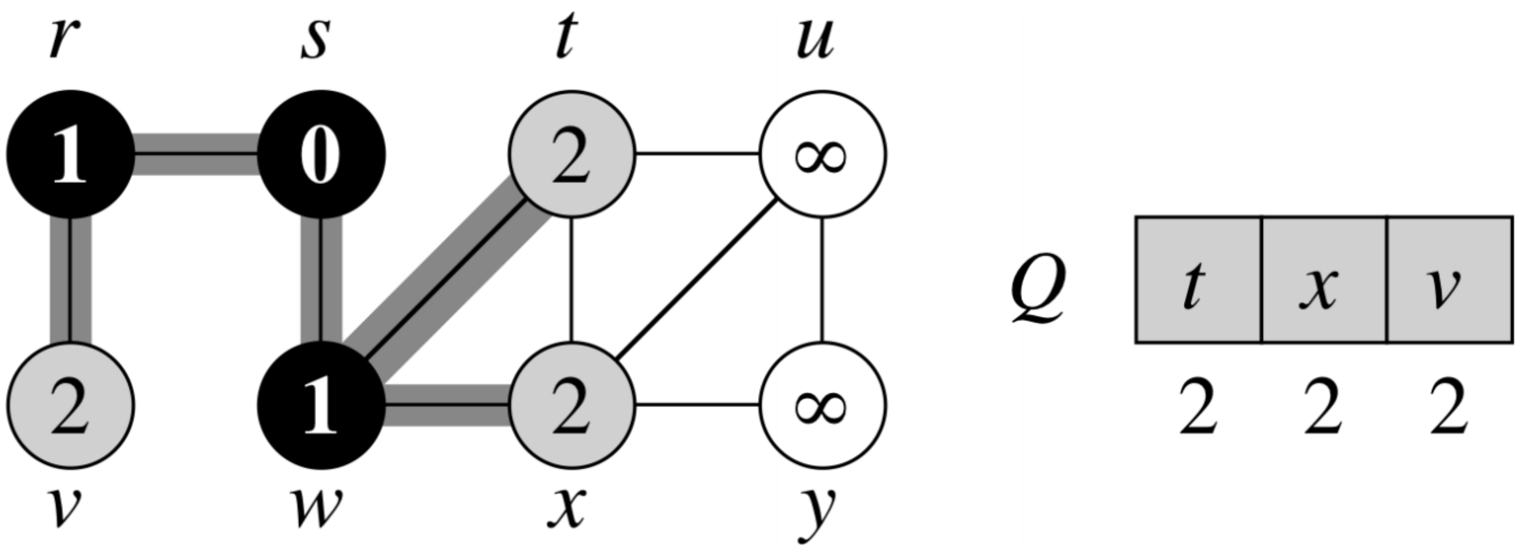
Busca em Largura: Exemplo



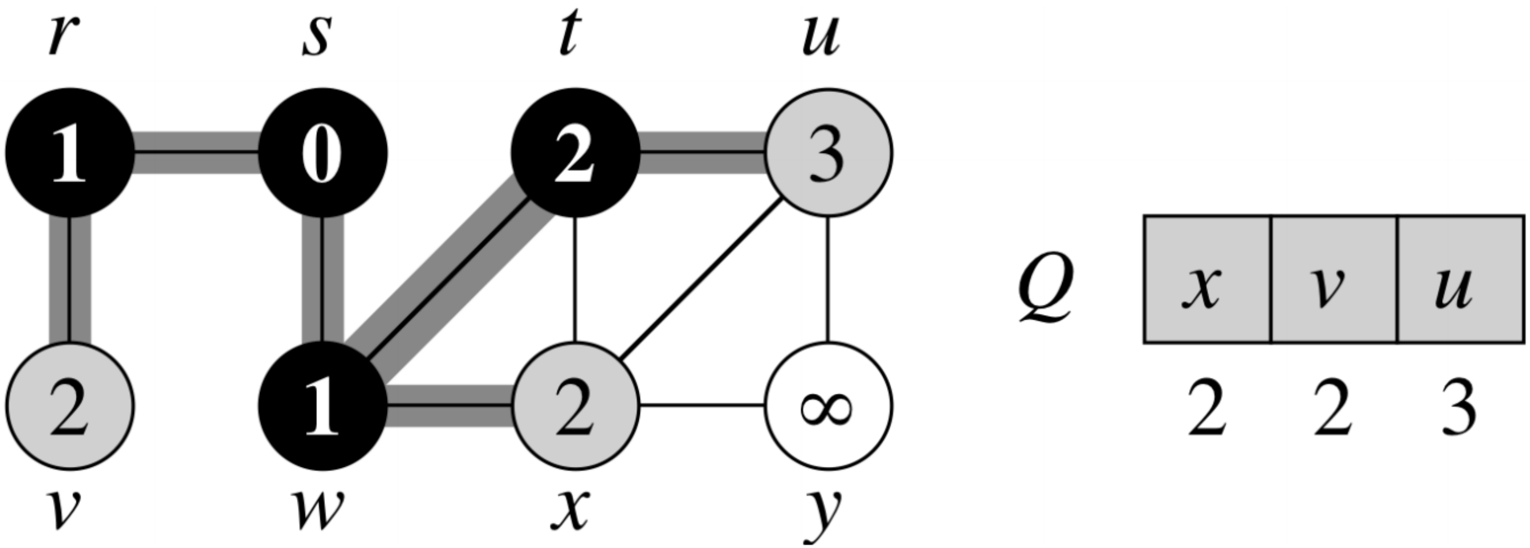
Busca em Largura: Exemplo



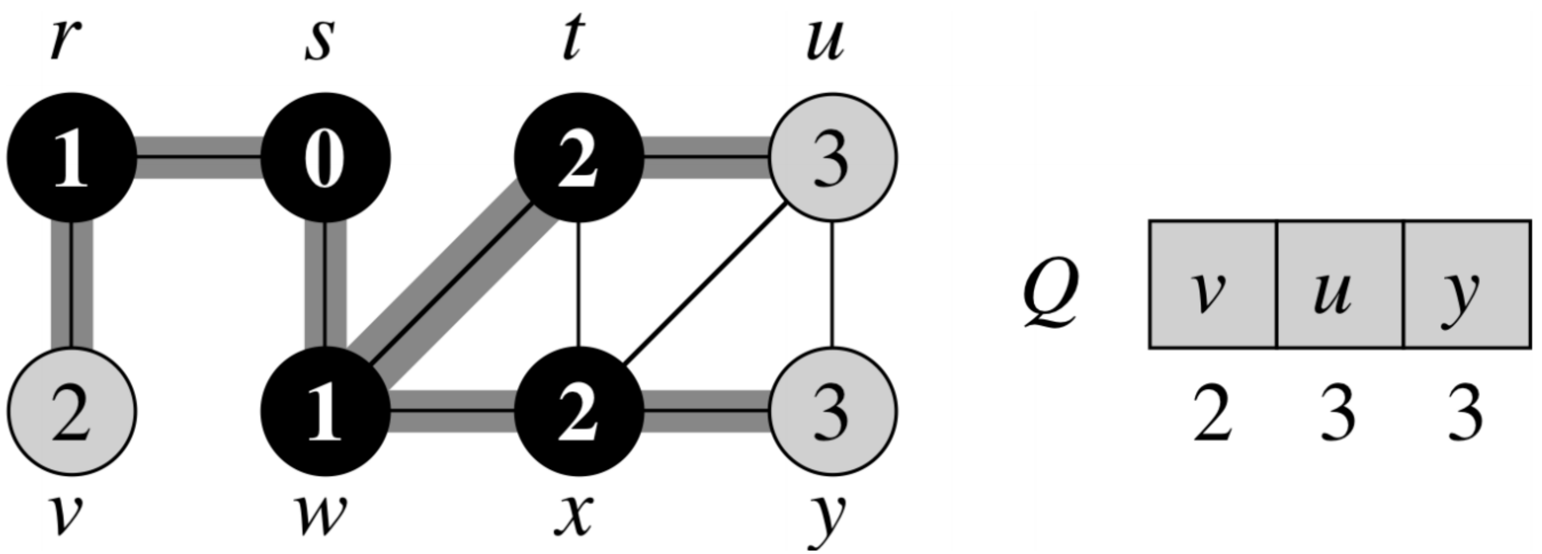
Busca em Largura: Exemplo



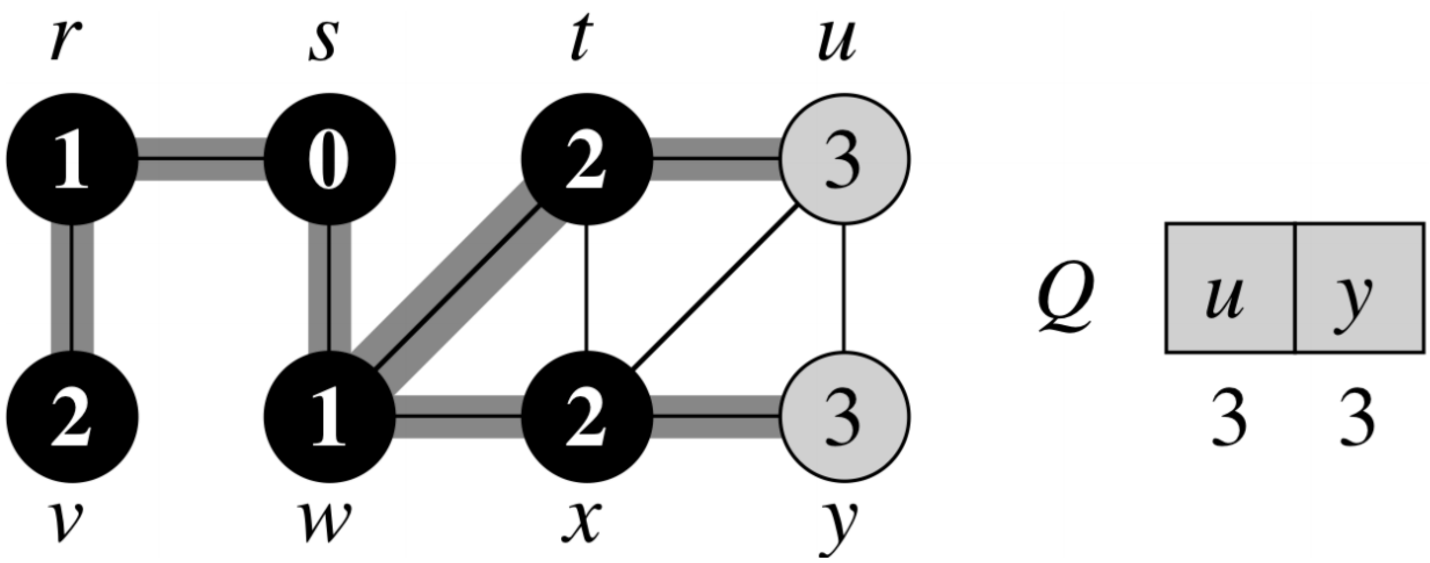
Busca em Largura: Exemplo



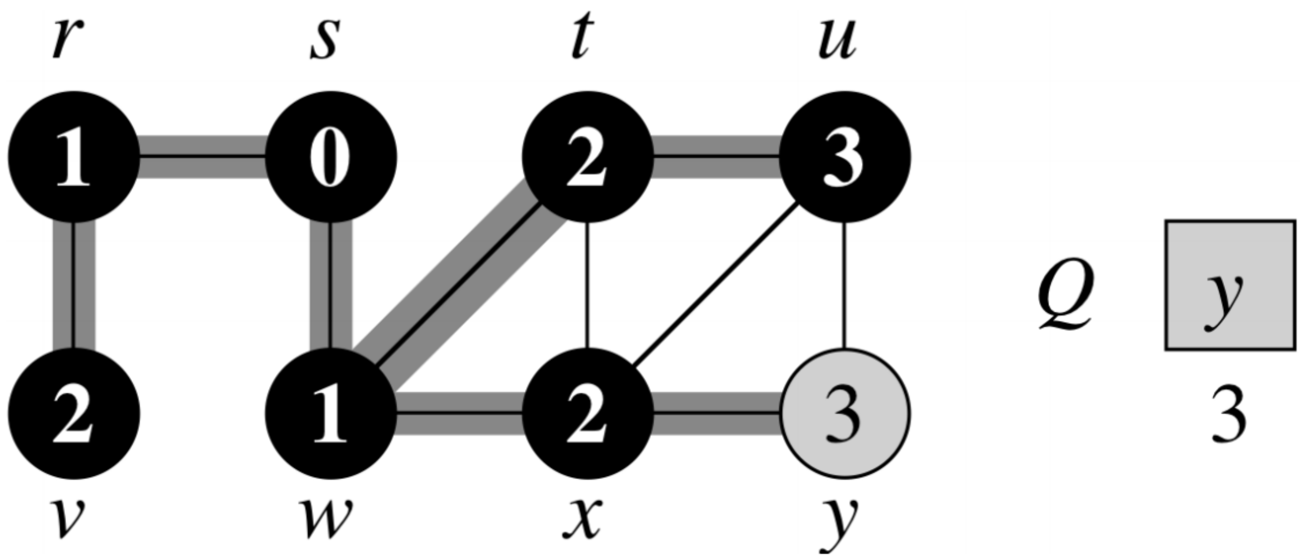
Busca em Largura: Exemplo



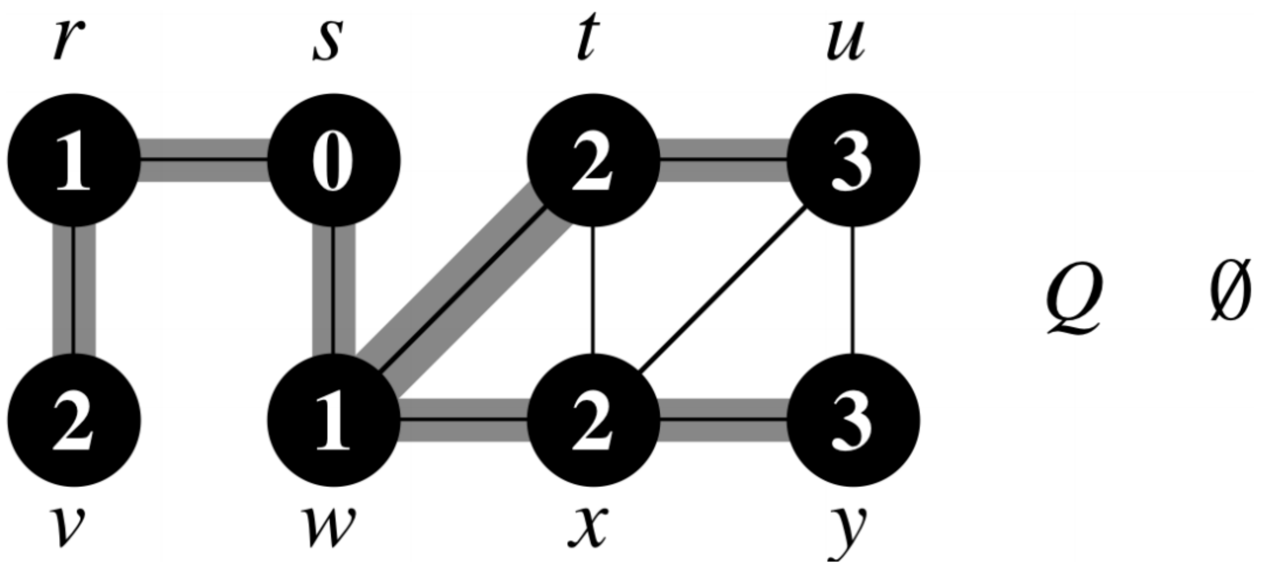
Busca em Largura: Exemplo



Busca em Largura: Exemplo



Busca em Largura: Exemplo



Algoritmo

```
bfs(G, s)
  for cada vértice u em G.V - {s}
    u.d = infinito
    u.π = nil
    u.cor = branco
  s.d = 0
  s.π = nil
  s.cor = cinza
  Q = newQueue()
  enqueue(Q,s)
  while !isEmpty(Q)
    u = dequeue(Q)
    for cada vértice v em u.adj
      if v.cor == branco
        v.d = u.d + 1
        v.π = u
        v.cor = cinza
        enqueue(Q,v)
    u.cor = preto
```

Análise do tempo de execução

- Análise agregada
- O teste da linha 13 garante que cada vértice é colocado na fila no máximo uma vez, e portanto, é retirado da fila no máximo uma vez
- As operações de colocar e retirar da fila demoram O(1), assim, o tempo total das operações com filas é O(V)
- A lista de adjacência de cada vértice é examinada apenas quando o vértice é retirado da fila, desta forma, no máximo uma vez
- Como a soma dos comprimentos das listas de adjacências é Θ(A), o tempo para percorrer todas as listas é no máximo O(A)
- O tempo de inicialização é O(V)
- Tempo total de execução do **bfs** é O($V + A$)

Observações

- Cada vértice é colorido de branco, cinza ou preto.
- Todos os vértices são inicializados branco.
- Quando um vértice é descoberto pela primeira vez ele torna-se cinza.
- Vértices cinza e preto já foram descobertos, mas são distinguidos para assegurar que a busca ocorra em largura.
- Se $(u, v) \in A$ e o vértice u é preto, qual é a cor do vértice v?

Observações

- Cada vértice é colorido de branco, cinza ou preto.
- Todos os vértices são inicializados branco.
- Quando um vértice é descoberto pela primeira vez ele torna-se cinza.
- Vértices cinza e preto já foram descobertos, mas são distinguidos para assegurar que a busca ocorra em largura.
- Se $(u, v) \in A$ e o vértice u é preto, qual é a cor do vértice v?
  - cinza ou preto

Observações

- Cada vértice é colorido de branco, cinza ou preto.
- Todos os vértices são inicializados branco.
- Quando um vértice é descoberto pela primeira vez ele torna-se cinza.
- Vértices cinza e preto já foram descobertos, mas são distinguidos para assegurar que a busca ocorra em largura.
- Se $(u, v) \in A$ e o vértice u é preto, qual é a cor do vértice v?
  - cinza ou preto
- Vértices cinza podem ter alguns vértices adjacentes brancos, e eles representam a fronteira entre vértices descobertos e não descobertos.

Árvore primeiro na extensão

- **`bfs`** constrói uma árvore primeiro na extensão
- A árvore é definida pelo campo pai (π) em cada vértice
- Para um grafo $G = (V, A)$ e um vértice de origem s, definimos o **subgrafo predecessor** de G como $G_\pi = (V_\pi, A_\pi)$ onde:
  - $V_\pi = \{v \in V \mid v.\pi \neq \text{NIL}\} \cup \{s\}$
  - $A_\pi = \{(v.\pi, v) \mid v \in V_\pi - \{s\}\}$

Árvore primeiro na extensão

- O **subgrafo predecessor** G_π é uma árvore primeiro na extensão
  - V_π consiste nos vértices acessíveis a partir de s
  - Para todo $v \in V_\pi$, existe um caminho único simples desde s até v em $V\pi$, que também é um caminho mais curto de s até v em G

- Uma árvore primeiro na extensão é de fato uma árvore, pois é conectada $|A_\pi| = |V_\pi| - 1$

Algoritmo

```
imprimir-caminho(G, s, v)
if v == s
  imprimir s
else if v.π == nil
       imprimir "nenhum caminho existente de" s "para" v
else
  imprimir-caminho(G, s, v.π)
  imprimir v
```

Algoritmo

```
imprimir-caminho(G, s, v)
if v == s
  imprimir s
else if v.π == nil
     imprimir "nenhum caminho existente de" s "para" v
else
  imprimir-caminho(G, s, v.π)
  imprimir v
```

- Executado em tempo linear no número de vértices no caminho impresso, pois cada chamada recursiva é feita para um caminho com um vértice menor que o atual

Bibliografia

- Thomas H. Cormen et al. Introduction to Algorithms. 3rd edition. Capítulo 22.4.

- Nivio Ziviani. Projeto de Algoritmos com Implementações em Pascal e C. 3a Edição Revista e Ampliada, Cengage Learning, 2010. Capítulo 7. Seção 7.4