

# Algoritmos e Teoria dos Grafos

## Tópico 22: Busca em Largura

Renato Carmo

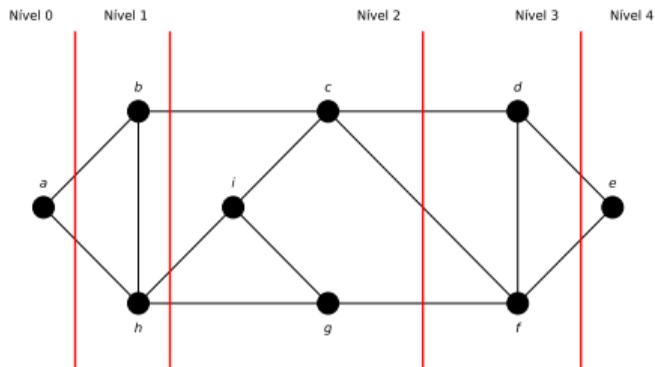
André Guedes

Murilo Silva

Departamento de Informática da UFPR

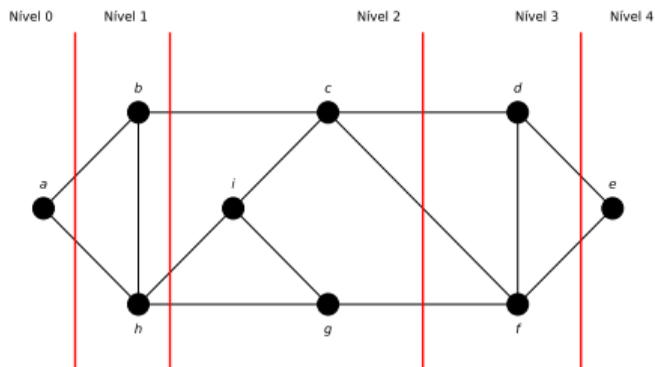
2023

# Busca em Largura



- Vértices processados em ordem não decrescente de distância de  $r$  (raíz)

# Busca em Largura



- Vértices processados em ordem não decrescente de distância de  $r$  (raíz)
- todos os vértices a distância  $d$  de  $r$  são processados antes de todos os vértices a distância  $d + 1$  de  $r$

# Algoritmo de Busca em Largura

---

## BuscaLargura( $G$ )

---

```
Para  $v \in V(G)$   
   $v.\text{estado} \leftarrow 0$   
Para  $v \in V(G)$   
  Se  $v.\text{estado} = 0$   
    BuscaLargura( $G, v$ )
```

---

---

## BuscaLargura( $G, r$ )

---

```
 $V \leftarrow$  fila vazia  
processe  $r$   
 $r.\text{pai} \leftarrow \Lambda$   
enfile  $r$  em  $V$   
 $r.\text{estado} \leftarrow 1$   
Enquanto  $V$  não está vazia  
  desenfile um vértice  $v$  de  $V$   
  Para cada  $w \in \Gamma_G(v)$   
    Se  $w.\text{estado} = 1$   
      processe  $\{v, w\}$  (como aresta fora da árvore)  
    Senão, se  $w.\text{estado} = 0$   
      processe  $w$   
      processe  $\{v, w\}$  (como aresta de árvore)  
       $w.\text{pai} \leftarrow v$   
      enfile  $w$  em  $V$   
       $w.\text{estado} \leftarrow 1$   
  
 $v.\text{estado} \leftarrow 2$ 
```

---

## Teorema

Se  $F$  é a floresta direcionada resultante de uma busca em largura sobre um grafo  $G$ , então toda aresta fora de  $F$  é cruzada com relação a  $F$ .

# Teorema

Se  $F$  é a floresta direcionada resultante de uma busca em largura sobre um grafo  $G$ , então toda aresta fora de  $F$  é cruzada com relação a  $F$ .

Exercício 104

# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

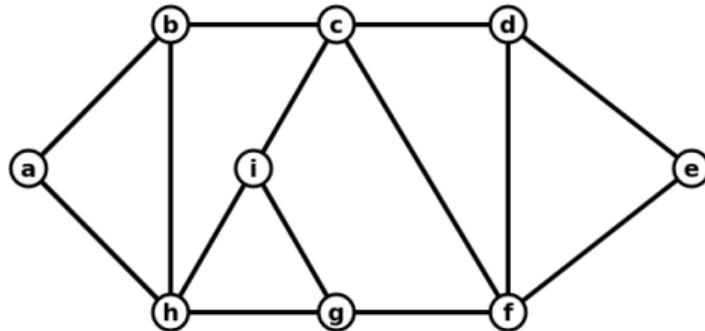
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

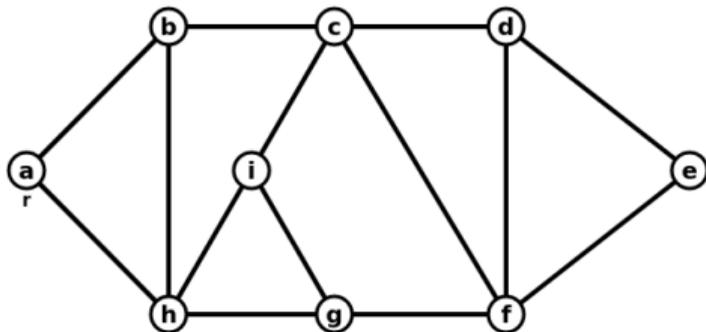
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

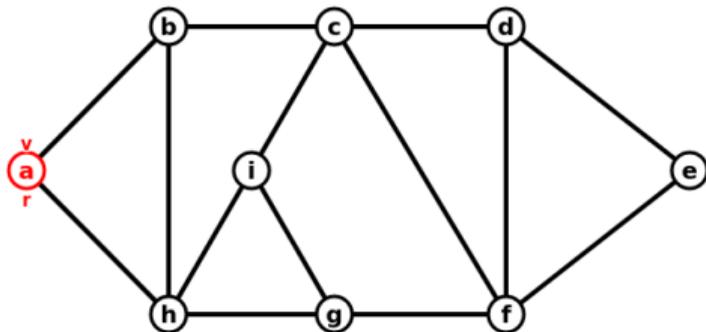
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

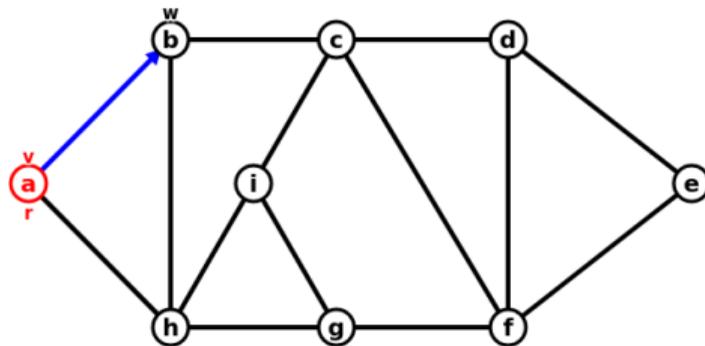
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---





# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

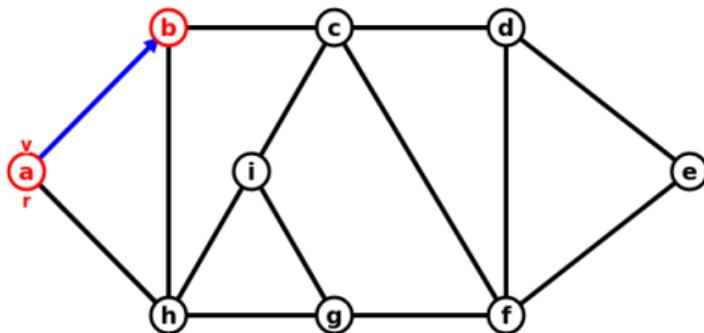
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

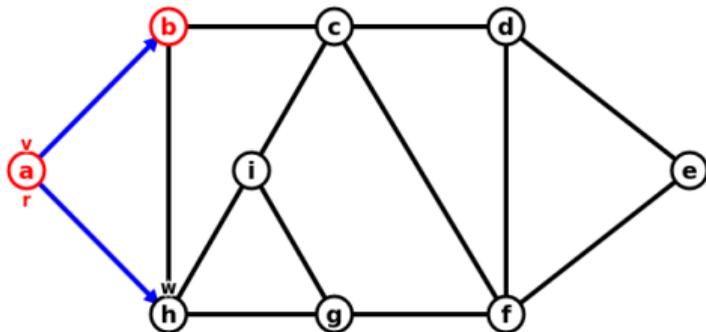
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---





# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

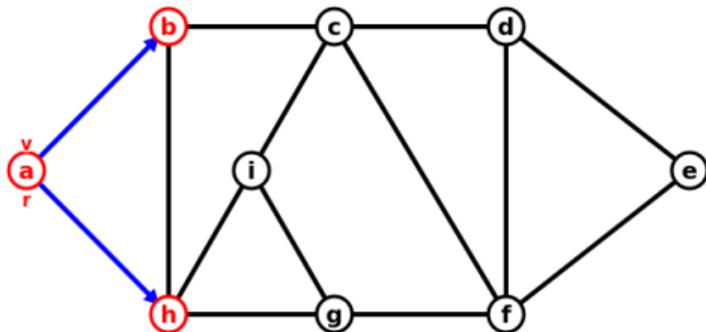
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

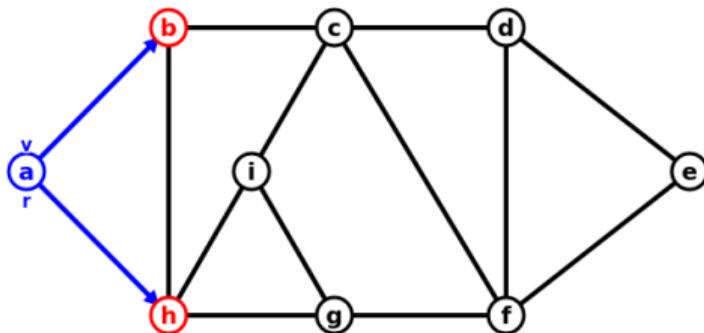
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

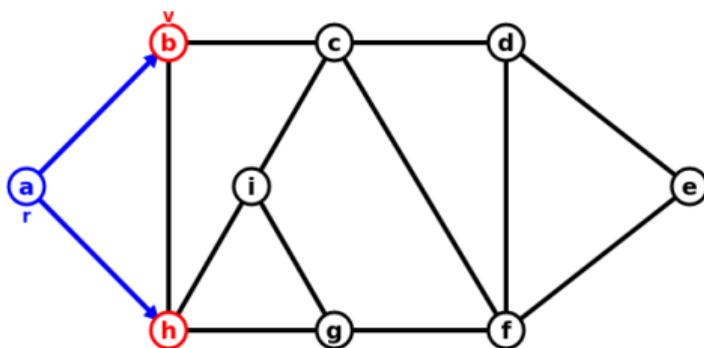
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---





# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

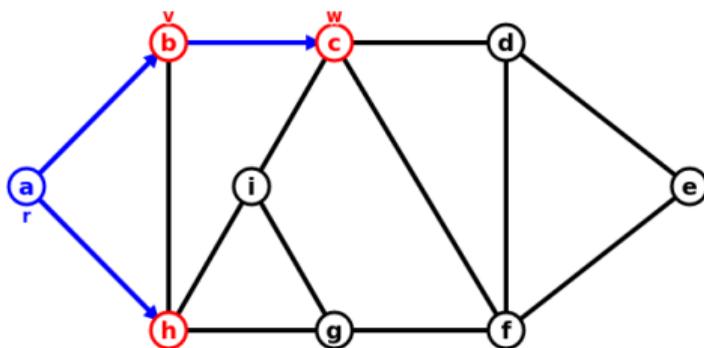
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

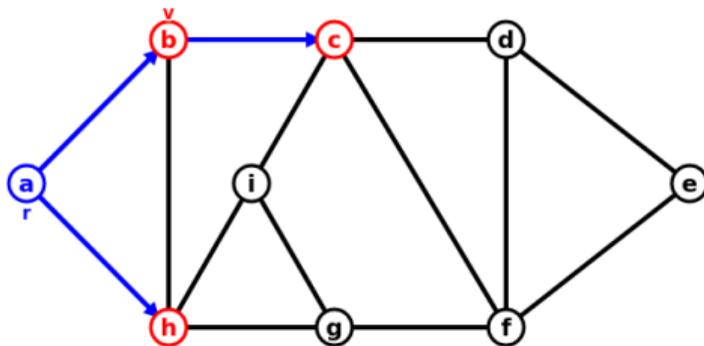
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---





# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

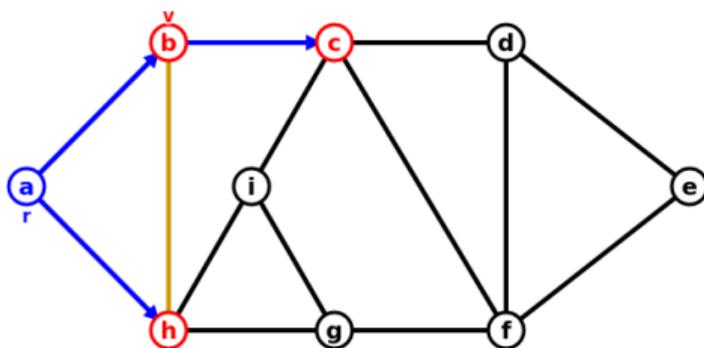
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

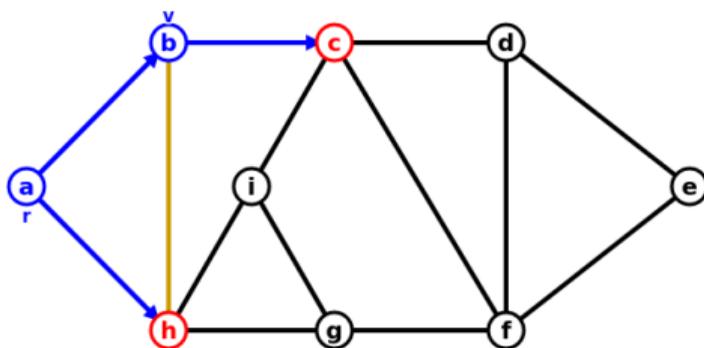
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

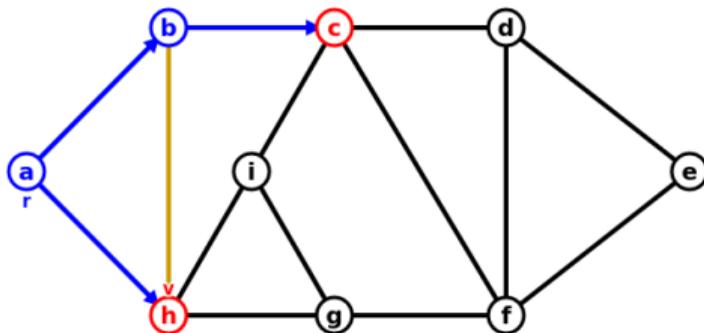
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---





# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

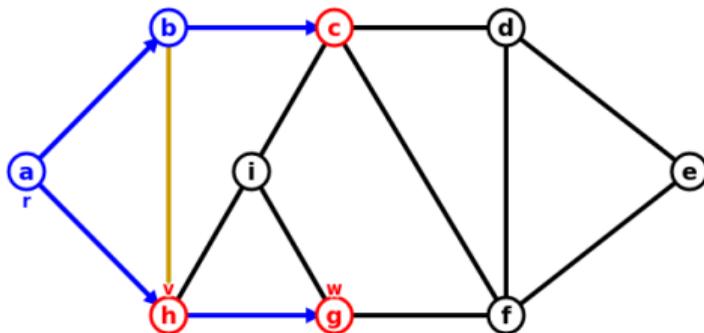
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

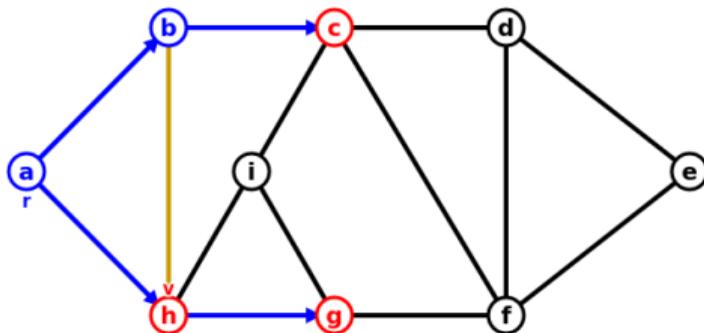
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---









# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

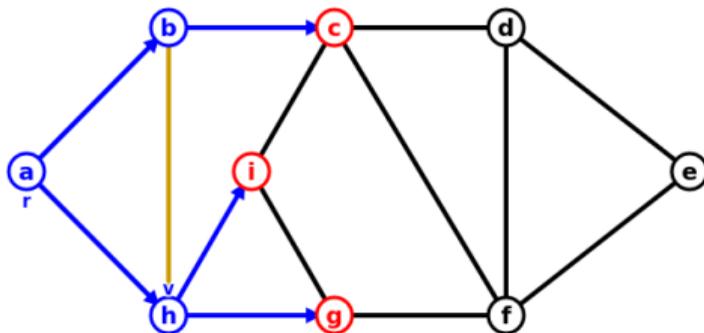
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

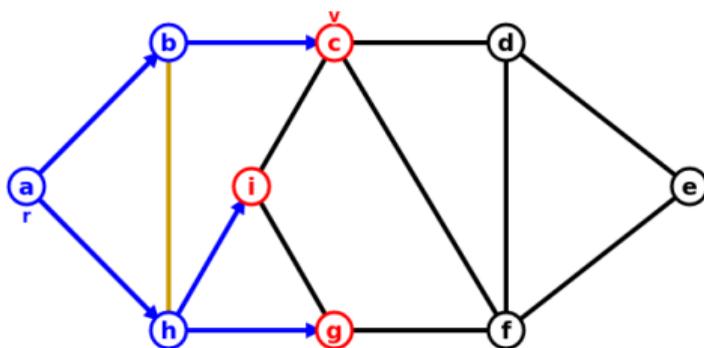
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

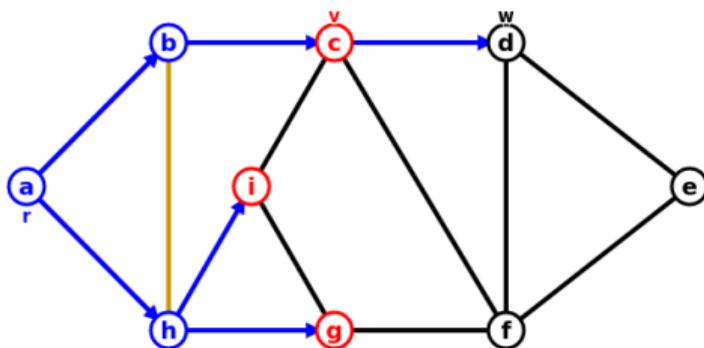
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---





# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

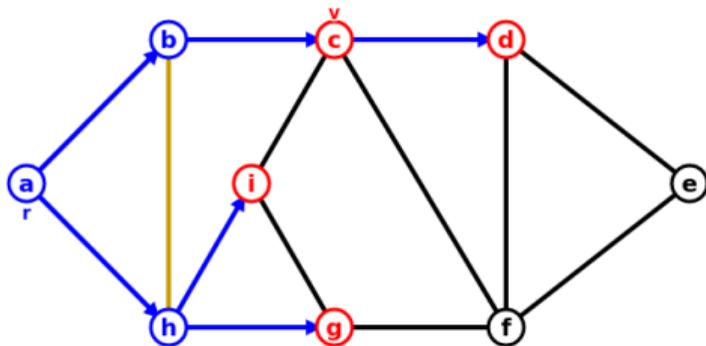
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

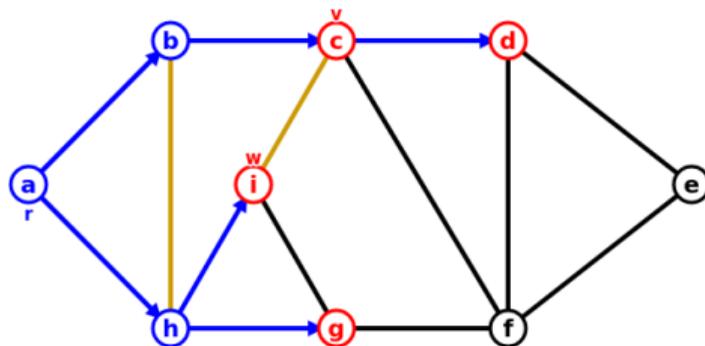
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

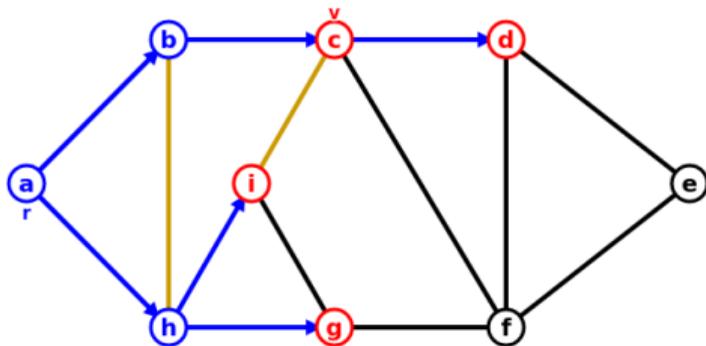
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

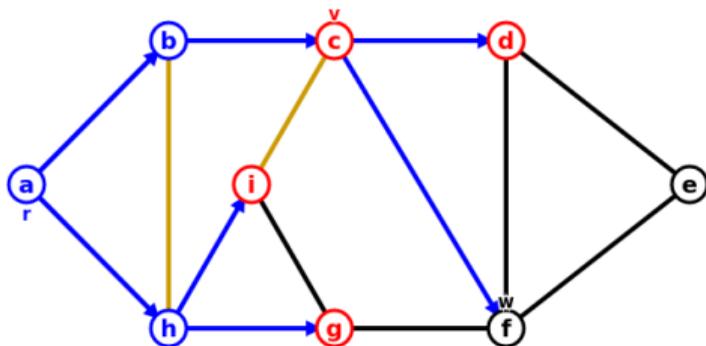
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

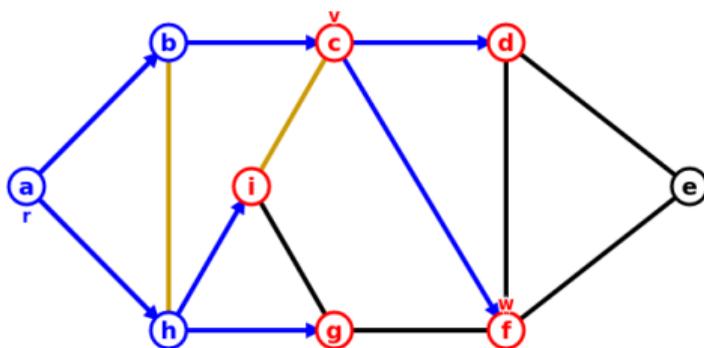
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---





# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

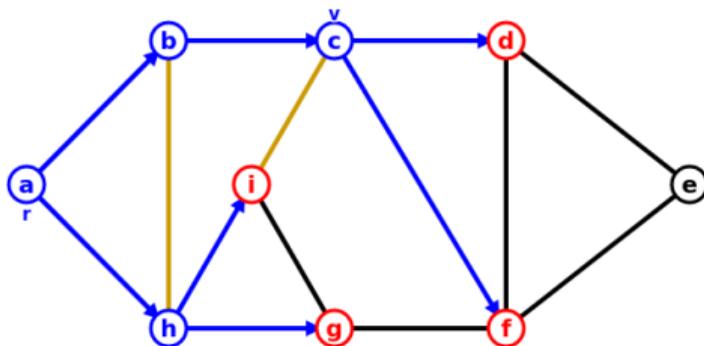
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

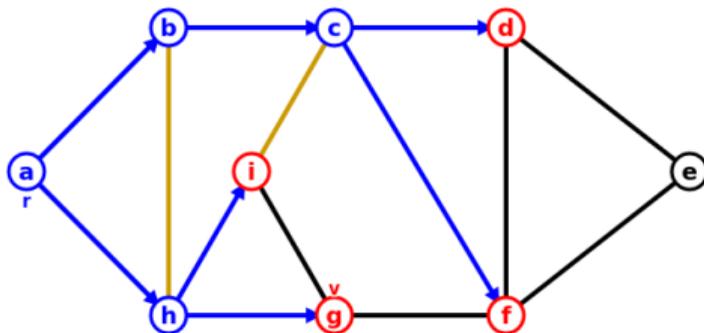
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

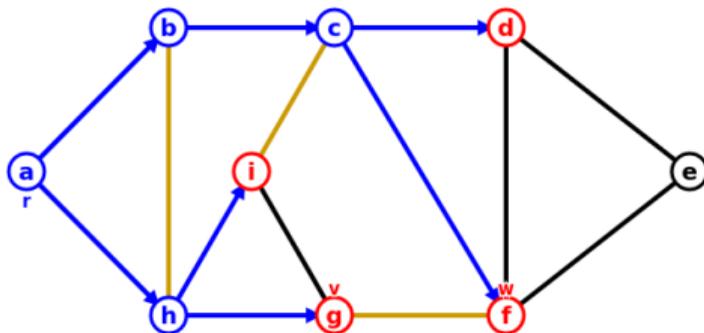
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

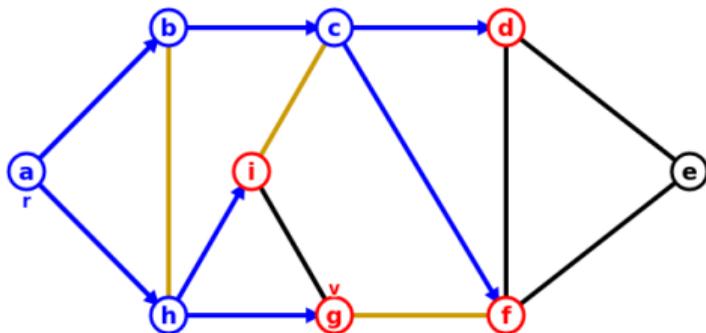
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

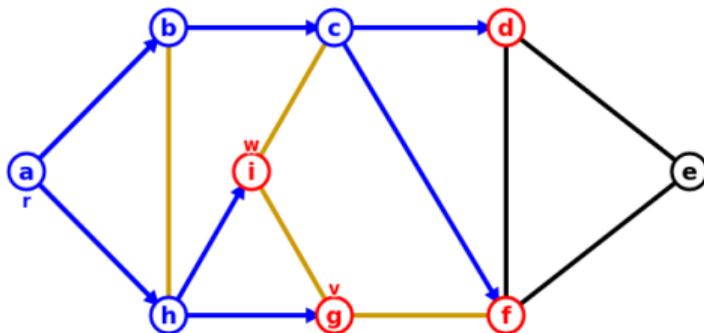
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

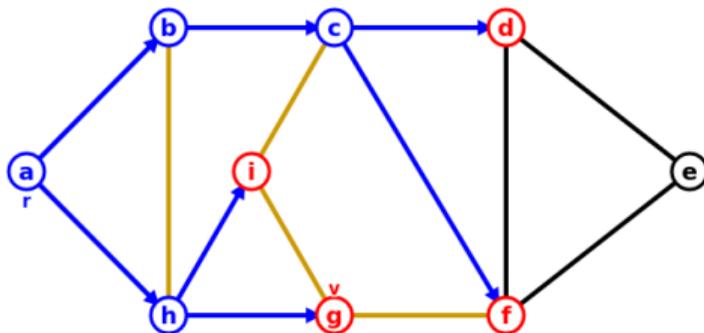
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

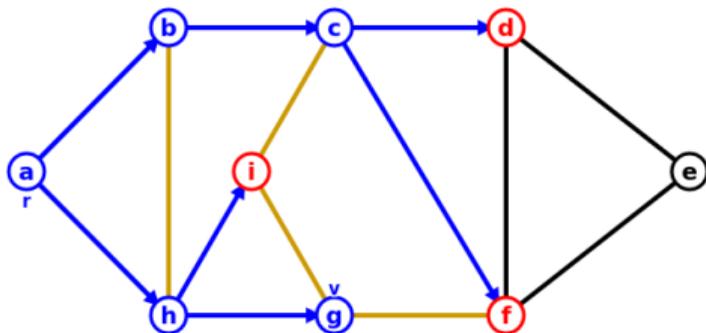
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

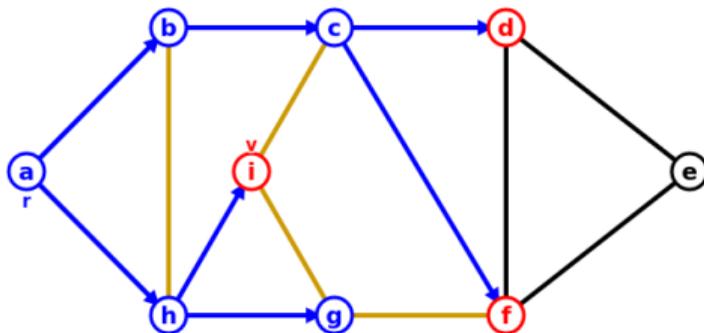
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

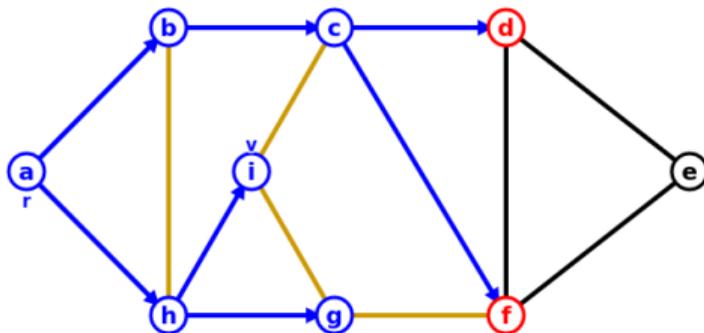
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

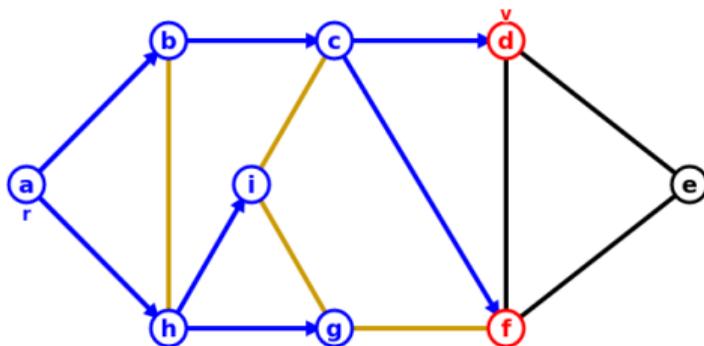
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

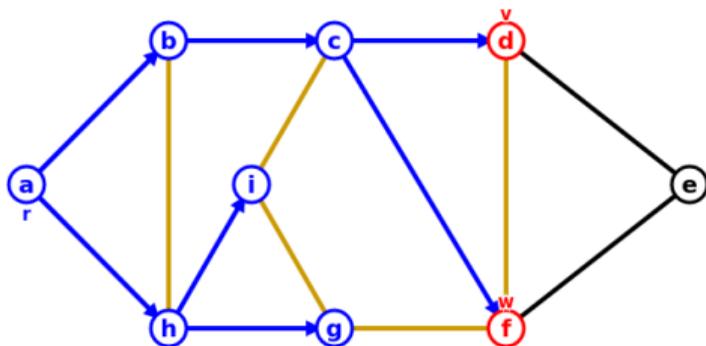
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

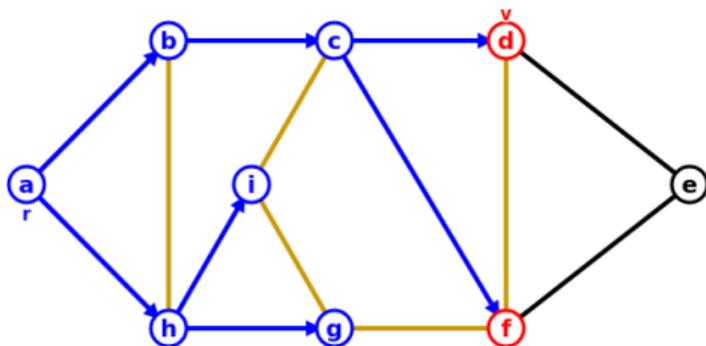
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

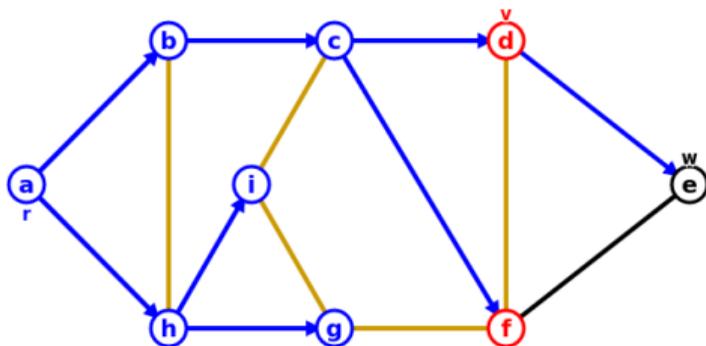
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---





# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

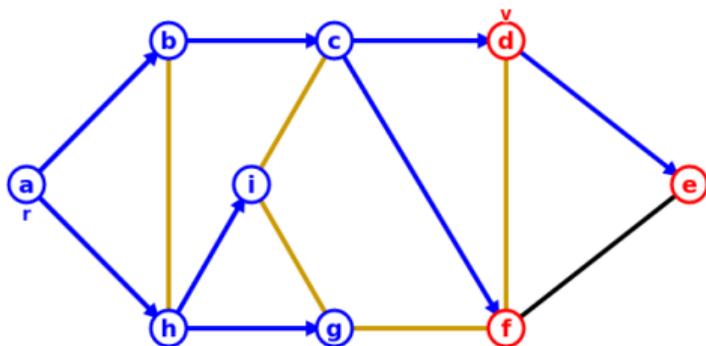
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

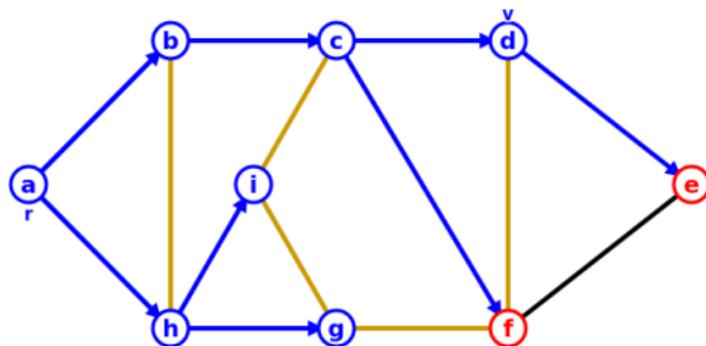
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

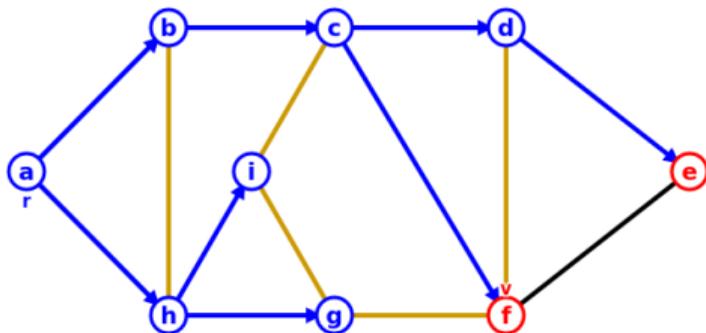
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

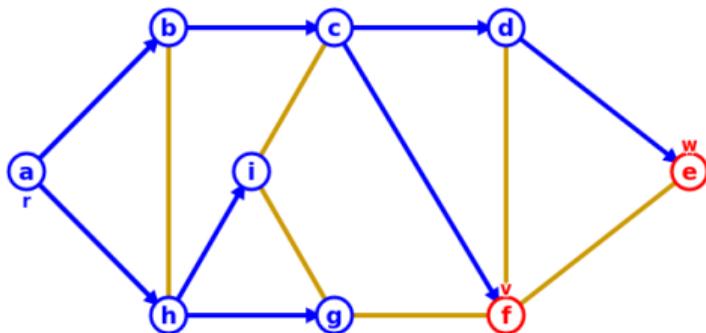
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

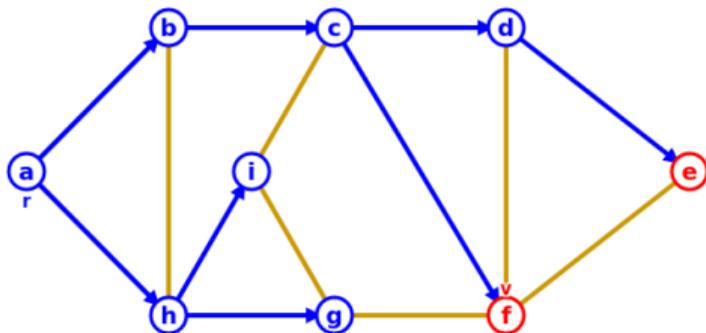
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

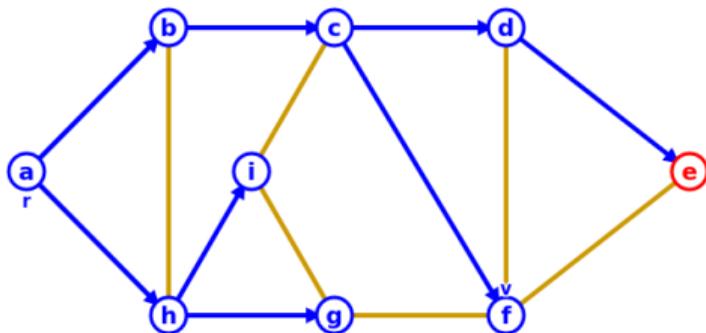
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

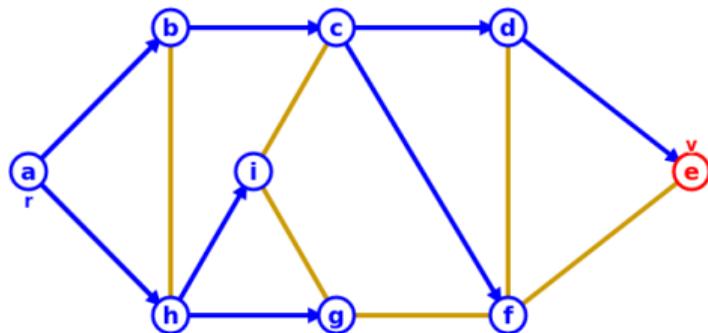
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

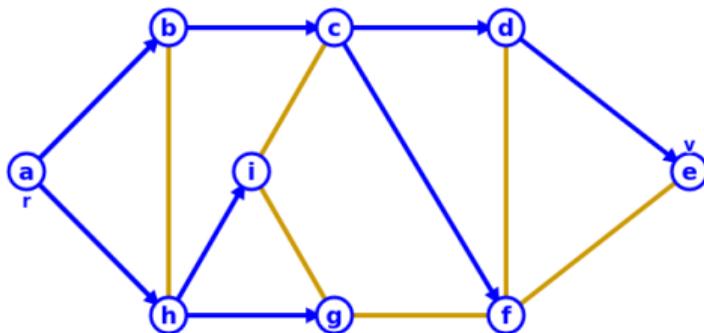
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



# Algoritmo de Busca em Largura - Execução

---

## BuscaLargura( $G, r$ )

---

$V \leftarrow$  fila vazia

processe  $r$

$r.pai \leftarrow \Lambda$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V$  não está vazia

  desenfile um vértice  $v$  de  $V$

  Para cada  $w \in \Gamma_G(v)$

    Se  $w.estado = 1$

      processe  $\{v, w\}$  (como  
      aresta fora da árvore)

    Senão, se  $w.estado = 0$

      processe  $w$

      processe  $\{v, w\}$  (como  
      aresta de árvore)

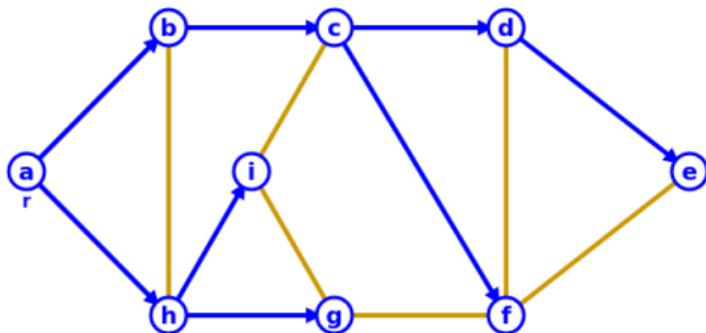
$w.pai \leftarrow v$

      enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



## Distâncias - Teorema 55

Se  $T$  é a arborescência resultante de uma busca em largura a partir do vértice  $r$  em um grafo conexo  $G$ , então

$$d_T(r, v) = d_G(r, v), \text{ para todo } v \in V(G).$$

## Distâncias - Teorema 55

Se  $T$  é a arborescência resultante de uma busca em largura a partir do vértice  $r$  em um grafo conexo  $G$ , então

$$d_T(r, v) = d_G(r, v), \text{ para todo } v \in V(G).$$

Demonstração:

## Distâncias - Teorema 55

Se  $T$  é a arborescência resultante de uma busca em largura a partir do vértice  $r$  em um grafo conexo  $G$ , então

$$d_T(r, v) = d_G(r, v), \text{ para todo } v \in V(G).$$

Demonstração:

Exercício 105

# Distâncias

---

## CaminhosMinimos( $G, r$ )

---

$V \leftarrow$  fila vazia

$r.dist \leftarrow 0$

enfile  $r$  em  $V$

$r.estado \leftarrow 1$

Enquanto  $V \neq \emptyset$

    desenfile um vértice  $v$  de  $V$

    Para **cada**  $w \in \Gamma_G(v)$

        Se  $w.estado = 0$

$w.pai \leftarrow v$

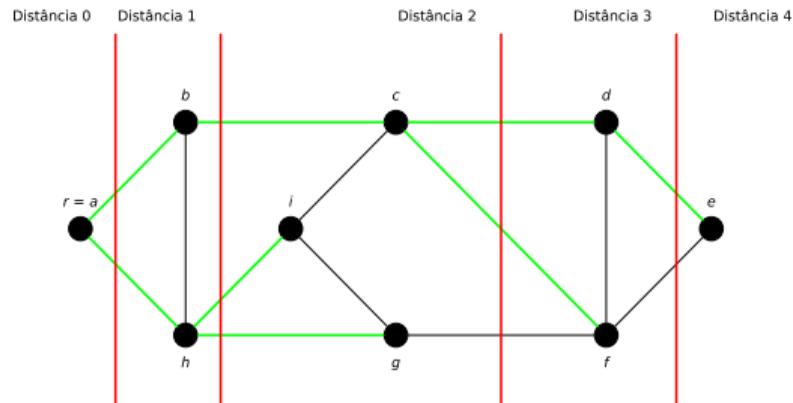
$w.dist \leftarrow w.pai.dist + 1$

            enfile  $w$  em  $V$

$w.estado \leftarrow 1$

$v.estado \leftarrow 2$

---



Se  $(T, r)$  é a árvore enraizada resultante de uma busca em largura em um grafo conexo  $G$ , então  $rTv$  é um caminho mínimo em  $G$  para todo  $v \in V(G)$ .

# Árvore de Caminhos Mínimos

É uma árvore enraizada  $(T, r)$  onde  $rTv$  é um caminho mínimo em  $G$  para todo  $v \in V(G)$ .

# Árvore de Caminhos Mínimos

É uma árvore enraizada  $(T, r)$  onde  $rTv$  é um caminho mínimo em  $G$  para todo  $v \in V(G)$ .

Corolário: É possível computar uma árvore de caminhos mínimos geradora de um grafo conexo com  $m$  arestas em tempo  $O(m)$ .

## Distâncias e Diâmetro

Corolário: É possível computar as distâncias e os caminhos mínimos entre todos os pares de vértices de um grafo com  $n$  vértices e  $m$  arestas em tempo  $O(nm)$ .

## Distâncias e Diâmetro

Corolário: É possível computar as distâncias e os caminhos mínimos entre todos os pares de vértices de um grafo com  $n$  vértices e  $m$  arestas em tempo  $O(nm)$ .

Corolário: É possível computar o diâmetro de um grafo conexo com  $n$  vértices e  $m$  arestas em tempo  $O(nm)$ .

## Bipartição

Teorema: Se  $(T, r)$  é a árvore enraizada resultante de uma busca em largura sobre um grafo conexo  $G$ , então

$$|d_G(r, u) - d_G(r, v)| \leq 1, \text{ para todo } \{u, v\} \in E(G - T).$$

# Bipartição

Teorema: Se  $(T, r)$  é a árvore enraizada resultante de uma busca em largura sobre um grafo conexo  $G$ , então

$$|d_G(r, u) - d_G(r, v)| \leq 1, \text{ para todo } \{u, v\} \in E(G - T).$$

Demonstração: Exercício 99.

# Bipartição

Teorema: Se  $(T, r)$  é a árvore enraizada resultante de uma busca em largura sobre um grafo conexo  $G$ , então

$$|d_G(r, u) - d_G(r, v)| \leq 1, \text{ para todo } \{u, v\} \in E(G - T).$$

Demonstração: Exercício 99.

Teorema: Seja  $(T, r)$  a árvore enraizada resultante de uma busca em largura sobre um grafo conexo  $G$ . O grafo  $G$  é bipartido se e somente se  $d(r, u)$  e  $d(r, v)$  tem paridades diferentes para toda aresta  $\{u, v\} \in E(G - T)$ .

# Bipartição

Teorema: Se  $(T, r)$  é a árvore enraizada resultante de uma busca em largura sobre um grafo conexo  $G$ , então

$$|d_G(r, u) - d_G(r, v)| \leq 1, \text{ para todo } \{u, v\} \in E(G - T).$$

Demonstração: Exercício 99.

Teorema: Seja  $(T, r)$  a árvore enraizada resultante de uma busca em largura sobre um grafo conexo  $G$ . O grafo  $G$  é bipartido se e somente se  $d(r, u)$  e  $d(r, v)$  tem paridades diferentes para toda aresta  $\{u, v\} \in E(G - T)$ .

Demonstração: Exercício 100.

# Bipartição

Teorema: Se  $(T, r)$  é a árvore enraizada resultante de uma busca em largura sobre um grafo conexo  $G$ , então

$$|d_G(r, u) - d_G(r, v)| \leq 1, \text{ para todo } \{u, v\} \in E(G - T).$$

Demonstração: Exercício 99.

Teorema: Seja  $(T, r)$  a árvore enraizada resultante de uma busca em largura sobre um grafo conexo  $G$ . O grafo  $G$  é bipartido se e somente se  $d(r, u)$  e  $d(r, v)$  tem paridades diferentes para toda aresta  $\{u, v\} \in E(G - T)$ .

Demonstração: Exercício 100.

Corolário: É possível decidir se um grafo de  $n$  vértices e  $m$  arestas é bipartido e, em caso positivo, computar uma sua bipartição em tempo  $O(n + m)$ .