

# Algoritmos e Teoria dos Grafos

## Tópico 21: Busca em Grafos

Renato Carmo  
André Guedes  
Murilo Silva

Departamento de Informática da UFPR

2023

# Busca em Grafos

Algoritmos que processam cada vértice e cada aresta de um grafo.

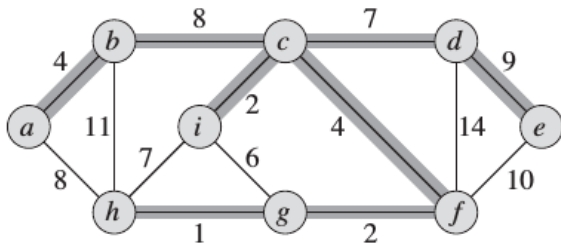


Figura: (Cormen, Leiserson, Rivest, and Stein, 2009, fig 23.1, p. 625)

# Algoritmo Genérico de Busca

---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$ 
processe  $r$ 
 $r.pai \leftarrow \Lambda$ 
acrescente  $r$  a  $V_1$ 
 $r.estado \leftarrow 1$ 
Enquanto  $V_1 \neq \emptyset$ 
     $v \leftarrow$  vértice em  $V_1$ 
    Se não há “próximo” vértice em  $\Gamma_G(v)$ 
        retire  $v$  de  $V_1$ 
         $v.estado \leftarrow 2$ 
    Senão
         $w \leftarrow$  “próximo” vértice em  $\Gamma_G(v)$ 
        Se  $w.estado \neq 0$ 
            Se  $\{v, w\}$  não foi processada
                processe  $\{v, w\}$ 
            Senão
                processe  $\{v, w\}$ 
                processe  $w$ 
                 $w.pai \leftarrow v$ 
                acrescente  $w$  a  $V_1$ 
                 $w.estado \leftarrow 1$ 
```

---

---

## Busca( $G$ )

---

```
Para  $v \in V(G)$ 
     $v.estado \leftarrow 0$ 
Para  $v \in V(G)$ 
    Se  $v.estado = 0$ 
        Busca( $G, v$ )
```

---

$G_r :=$  componente de  $G$  que contém  $r$ ,

$V_0 := \{v \in V(G) \mid v.estado = 0\}$ ,

$V_1 := \{v \in V(G) \mid v.estado = 1\}$ ,

$V_2 := \{v \in V(G) \mid v.estado = 2\}$ ,

$H := G[V_1 \cup V_2]$ ,

$T := G[\{(v.pai, v) \mid v.pai \neq \Lambda\}] + r$ ,

# Algoritmo Genérico de Busca

---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$ 
processe  $r$ 
 $r.pai \leftarrow \Lambda$ 
acrescente  $r$  a  $V_1$ 
 $r.estado \leftarrow 1$ 
Enquanto  $V_1 \neq \emptyset$ 
     $v \leftarrow$  vértice em  $V_1$ 
    Se não há “próximo” vértice em  $\Gamma_G(v)$ 
        retire  $v$  de  $V_1$ 
         $v.estado \leftarrow 2$ 
    Senão
         $w \leftarrow$  “próximo” vértice em  $\Gamma_G(v)$ 
        Se  $w.estado \neq 0$ 
            Se  $\{v, w\}$  não foi processada
                processe  $\{v, w\}$ 
        Senão
            processe  $\{v, w\}$ 
            processe  $w$ 
             $w.pai \leftarrow v$ 
            acrescente  $w$  a  $V_1$ 
             $w.estado \leftarrow 1$ 
```

---

ao início de cada iteração,

# Algoritmo Genérico de Busca

---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$ 
processe  $r$ 
 $r.pai \leftarrow \Lambda$ 
acrescente  $r$  a  $V_1$ 
 $r.estado \leftarrow 1$ 
Enquanto  $V_1 \neq \emptyset$ 
     $v \leftarrow$  vértice em  $V_1$ 
    Se não há “próximo” vértice em  $\Gamma_G(v)$ 
        retire  $v$  de  $V_1$ 
         $v.estado \leftarrow 2$ 
    Senão
         $w \leftarrow$  “próximo” vértice em  $\Gamma_G(v)$ 
        Se  $w.estado \neq 0$ 
            Se  $\{v, w\}$  não foi processada
                processe  $\{v, w\}$ 
        Senão
            processe  $\{v, w\}$ 
            processe  $w$ 
             $w.pai \leftarrow v$ 
            acrescente  $w$  a  $V_1$ 
             $w.estado \leftarrow 1$ 
```

---

ao início de cada iteração,

1.  $(T, r)$  - árvore enraizada geradora de  $H$

# Algoritmo Genérico de Busca

---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$ 
processe  $r$ 
 $r.pai \leftarrow \Lambda$ 
acrescente  $r$  a  $V_1$ 
 $r.estado \leftarrow 1$ 
Enquanto  $V_1 \neq \emptyset$ 
     $v \leftarrow$  vértice em  $V_1$ 
    Se não há “próximo” vértice em  $\Gamma_G(v)$ 
        retire  $v$  de  $V_1$ 
         $v.estado \leftarrow 2$ 
    Senão
         $w \leftarrow$  “próximo” vértice em  $\Gamma_G(v)$ 
        Se  $w.estado \neq 0$ 
            Se  $\{v, w\}$  não foi processada
                processe  $\{v, w\}$ 
        Senão
            processe  $\{v, w\}$ 
            processe  $w$ 
             $w.pai \leftarrow v$ 
            acrescente  $w$  a  $V_1$ 
             $w.estado \leftarrow 1$ 
```

---

ao início de cada iteração,

1.  $(T, r)$  - árvore enraizada geradora de  $H$
2.  $H$  - subgrafo conexo de  $G_r$ .

# Algoritmo Genérico de Busca

---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$ 
processe  $r$ 
 $r.pai \leftarrow \Lambda$ 
acrescente  $r$  a  $V_1$ 
 $r.estado \leftarrow 1$ 
Enquanto  $V_1 \neq \emptyset$ 
     $v \leftarrow$  vértice em  $V_1$ 
    Se não há “próximo” vértice em  $\Gamma_G(v)$ 
        retire  $v$  de  $V_1$ 
         $v.estado \leftarrow 2$ 
    Senão
         $w \leftarrow$  “próximo” vértice em  $\Gamma_G(v)$ 
        Se  $w.estado \neq 0$ 
            Se  $\{v, w\}$  não foi processada
                processe  $\{v, w\}$ 
        Senão
            processe  $\{v, w\}$ 
            processe  $w$ 
             $w.pai \leftarrow v$ 
            acrescente  $w$  a  $V_1$ 
             $w.estado \leftarrow 1$ 
```

---

ao início de cada iteração,

1.  $(T, r)$  - árvore enraizada geradora de  $H$
2.  $H$  - subgrafo conexo de  $G_r$ .
3.  $\{V_0, V_1, V_2\}$  - partição de  $V(G)$ ,

# Algoritmo Genérico de Busca

---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$ 
processe  $r$ 
 $r.pai \leftarrow \Lambda$ 
acrescente  $r$  a  $V_1$ 
 $r.estado \leftarrow 1$ 
Enquanto  $V_1 \neq \emptyset$ 
     $v \leftarrow$  vértice em  $V_1$ 
    Se não há “próximo” vértice em  $\Gamma_G(v)$ 
        retire  $v$  de  $V_1$ 
         $v.estado \leftarrow 2$ 
    Senão
         $w \leftarrow$  “próximo” vértice em  $\Gamma_G(v)$ 
        Se  $w.estado \neq 0$ 
            Se  $\{v, w\}$  não foi processada
                processe  $\{v, w\}$ 
        Senão
            processe  $\{v, w\}$ 
            processe  $w$ 
             $w.pai \leftarrow v$ 
            acrescente  $w$  a  $V_1$ 
             $w.estado \leftarrow 1$ 
```

---

ao início de cada iteração,

1.  $(T, r)$  - árvore enraizada geradora de  $H$
2.  $H$  - subgrafo conexo de  $G_r$ .
3.  $\{V_0, V_1, V_2\}$  - partição de  $V(G)$ ,
4.  $V(T)$ ,  $E(T)$  e  $E(G[V_2])$  estão processados,

# Algoritmo Genérico de Busca

---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$ 
processe  $r$ 
 $r.pai \leftarrow \Lambda$ 
acrescente  $r$  a  $V_1$ 
 $r.estado \leftarrow 1$ 
Enquanto  $V_1 \neq \emptyset$ 
     $v \leftarrow$  vértice em  $V_1$ 
    Se não há “próximo” vértice em  $\Gamma_G(v)$ 
        retire  $v$  de  $V_1$ 
         $v.estado \leftarrow 2$ 
    Senão
         $w \leftarrow$  “próximo” vértice em  $\Gamma_G(v)$ 
        Se  $w.estado \neq 0$ 
            Se  $\{v, w\}$  não foi processada
                processe  $\{v, w\}$ 
        Senão
            processe  $\{v, w\}$ 
            processe  $w$ 
             $w.pai \leftarrow v$ 
            acrescente  $w$  a  $V_1$ 
             $w.estado \leftarrow 1$ 
```

---

ao início de cada iteração,

1.  $(T, r)$  - árvore enraizada geradora de  $H$
2.  $H$  - subgrafo conexo de  $G_r$ .
3.  $\{V_0, V_1, V_2\}$  - partição de  $V(G)$ ,
4.  $V(T)$ ,  $E(T)$  e  $E(G[V_2])$  estão processados,
5.  $\partial_G(V(H)) = \{\{v_0, v_1\} \in E(G) \mid v_0 \in V_0 \text{ e } v_1 \in V_1\}$ .

# Algoritmo Genérico de Busca

---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$ 
processe  $r$ 
 $r.pai \leftarrow \Lambda$ 
acrescente  $r$  a  $V_1$ 
 $r.estado \leftarrow 1$ 
Enquanto  $V_1 \neq \emptyset$ 
     $v \leftarrow$  vértice em  $V_1$ 
    Se não há “próximo” vértice em  $\Gamma_G(v)$ 
        retire  $v$  de  $V_1$ 
         $v.estado \leftarrow 2$ 
    Senão
         $w \leftarrow$  “próximo” vértice em  $\Gamma_G(v)$ 
        Se  $w.estado \neq 0$ 
            Se  $\{v, w\}$  não foi processada
                processe  $\{v, w\}$ 
        Senão
            processe  $\{v, w\}$ 
            processe  $w$ 
             $w.pai \leftarrow v$ 
            acrescente  $w$  a  $V_1$ 
             $w.estado \leftarrow 1$ 
```

---

ao final do laço

# Algoritmo Genérico de Busca

---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$ 
processe  $r$ 
 $r.pai \leftarrow \Lambda$ 
acrescente  $r$  a  $V_1$ 
 $r.estado \leftarrow 1$ 
Enquanto  $V_1 \neq \emptyset$ 
     $v \leftarrow$  vértice em  $V_1$ 
    Se não há “próximo” vértice em  $\Gamma_G(v)$ 
        retire  $v$  de  $V_1$ 
         $v.estado \leftarrow 2$ 
    Senão
         $w \leftarrow$  “próximo” vértice em  $\Gamma_G(v)$ 
        Se  $w.estado \neq 0$ 
            Se  $\{v, w\}$  não foi processada
                processe  $\{v, w\}$ 
        Senão
            processe  $\{v, w\}$ 
            processe  $w$ 
             $w.pai \leftarrow v$ 
            acrescente  $w$  a  $V_1$ 
             $w.estado \leftarrow 1$ 
```

---

ao final do laço

1.  $V_1 = \emptyset$  e, portanto,

# Algoritmo Genérico de Busca

---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$ 
processe  $r$ 
 $r.pai \leftarrow \Lambda$ 
acrescente  $r$  a  $V_1$ 
 $r.estado \leftarrow 1$ 
Enquanto  $V_1 \neq \emptyset$ 
     $v \leftarrow$  vértice em  $V_1$ 
    Se não há “próximo” vértice em  $\Gamma_G(v)$ 
        retire  $v$  de  $V_1$ 
         $v.estado \leftarrow 2$ 
    Senão
         $w \leftarrow$  “próximo” vértice em  $\Gamma_G(v)$ 
        Se  $w.estado \neq 0$ 
            Se  $\{v, w\}$  não foi processada
                processe  $\{v, w\}$ 
        Senão
            processe  $\{v, w\}$ 
            processe  $w$ 
             $w.pai \leftarrow v$ 
            acrescente  $w$  a  $V_1$ 
             $w.estado \leftarrow 1$ 
```

---

ao final do laço

1.  $V_1 = \emptyset$  e, portanto,
2.  $\partial_G(H) = \emptyset$  e, portanto,

# Algoritmo Genérico de Busca

---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$ 
processe  $r$ 
 $r.pai \leftarrow \Lambda$ 
acrescente  $r$  a  $V_1$ 
 $r.estado \leftarrow 1$ 
Enquanto  $V_1 \neq \emptyset$ 
     $v \leftarrow$  vértice em  $V_1$ 
    Se não há “próximo” vértice em  $\Gamma_G(v)$ 
        retire  $v$  de  $V_1$ 
         $v.estado \leftarrow 2$ 
    Senão
         $w \leftarrow$  “próximo” vértice em  $\Gamma_G(v)$ 
        Se  $w.estado \neq 0$ 
            Se  $\{v, w\}$  não foi processada
                processe  $\{v, w\}$ 
        Senão
            processe  $\{v, w\}$ 
            processe  $w$ 
             $w.pai \leftarrow v$ 
            acrescente  $w$  a  $V_1$ 
             $w.estado \leftarrow 1$ 
```

---

ao final do laço

1.  $V_1 = \emptyset$  e, portanto,
2.  $\partial_G(H) = \emptyset$  e, portanto,
3.  $H$  é conexo, então (C. 51)  $H$  é componente de  $G$ ,

# Algoritmo Genérico de Busca

---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$ 
processe  $r$ 
 $r.pai \leftarrow \Lambda$ 
acrescente  $r$  a  $V_1$ 
 $r.estado \leftarrow 1$ 
Enquanto  $V_1 \neq \emptyset$ 
     $v \leftarrow$  vértice em  $V_1$ 
    Se não há “próximo” vértice em  $\Gamma_G(v)$ 
        retire  $v$  de  $V_1$ 
         $v.estado \leftarrow 2$ 
    Senão
         $w \leftarrow$  “próximo” vértice em  $\Gamma_G(v)$ 
        Se  $w.estado \neq 0$ 
            Se  $\{v, w\}$  não foi processada
                processe  $\{v, w\}$ 
        Senão
            processe  $\{v, w\}$ 
            processe  $w$ 
             $w.pai \leftarrow v$ 
            acrescente  $w$  a  $V_1$ 
             $w.estado \leftarrow 1$ 
```

---

ao final do laço

1.  $V_1 = \emptyset$  e, portanto,
2.  $\partial_G(H) = \emptyset$  e, portanto,
3.  $H$  é conexo, então (C. 51)  $H$  é componente de  $G$ ,
4.  $r \in V(H)$ , então  $H = G_r$ ,

# Algoritmo Genérico de Busca

---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$ 
processe  $r$ 
 $r.pai \leftarrow \Lambda$ 
acrescente  $r$  a  $V_1$ 
 $r.estado \leftarrow 1$ 
Enquanto  $V_1 \neq \emptyset$ 
     $v \leftarrow$  vértice em  $V_1$ 
    Se não há “próximo” vértice em  $\Gamma_G(v)$ 
        retire  $v$  de  $V_1$ 
         $v.estado \leftarrow 2$ 
    Senão
         $w \leftarrow$  “próximo” vértice em  $\Gamma_G(v)$ 
        Se  $w.estado \neq 0$ 
            Se  $\{v, w\}$  não foi processada
                processe  $\{v, w\}$ 
        Senão
            processe  $\{v, w\}$ 
            processe  $w$ 
             $w.pai \leftarrow v$ 
            acrescente  $w$  a  $V_1$ 
             $w.estado \leftarrow 1$ 
```

---

ao final do laço

1.  $V_1 = \emptyset$  e, portanto,
2.  $\partial_G(H) = \emptyset$  e, portanto,
3.  $H$  é conexo, então (C. 51)  $H$  é componente de  $G$ ,
4.  $r \in V(H)$ , então  $H = G_r$ ,
5.  $(T, r)$  é uma árvore enraizada geradora de  $G_r$ ,

# Algoritmo Genérico de Busca

---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$ 
processe  $r$ 
 $r.pai \leftarrow \Lambda$ 
acrescente  $r$  a  $V_1$ 
 $r.estado \leftarrow 1$ 
Enquanto  $V_1 \neq \emptyset$ 
     $v \leftarrow$  vértice em  $V_1$ 
    Se não há “próximo” vértice em  $\Gamma_G(v)$ 
        retire  $v$  de  $V_1$ 
         $v.estado \leftarrow 2$ 
    Senão
         $w \leftarrow$  “próximo” vértice em  $\Gamma_G(v)$ 
        Se  $w.estado \neq 0$ 
            Se  $\{v, w\}$  não foi processada
                processe  $\{v, w\}$ 
        Senão
            processe  $\{v, w\}$ 
            processe  $w$ 
             $w.pai \leftarrow v$ 
            acrescente  $w$  a  $V_1$ 
             $w.estado \leftarrow 1$ 
```

---

ao final do laço

1.  $V_1 = \emptyset$  e, portanto,
2.  $\partial_G(H) = \emptyset$  e, portanto,
3.  $H$  é conexo, então (C. 51)  $H$  é componente de  $G$ ,
4.  $r \in V(H)$ , então  $H = G_r$ ,
5.  $(T, r)$  é uma árvore enraizada geradora de  $G_r$ ,
6. todos vértices e arestas de  $G_r$  foram processados.

# Algoritmo Genérico de Busca

---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$ 
processe  $r$ 
 $r.pai \leftarrow \Lambda$ 
acrescente  $r$  a  $V_1$ 
 $r.estado \leftarrow 1$ 
Enquanto  $V_1 \neq \emptyset$ 
     $v \leftarrow$  vértice em  $V_1$ 
    Se não há “próximo” vértice em  $\Gamma_G(v)$ 
        retire  $v$  de  $V_1$ 
         $v.estado \leftarrow 2$ 
    Senão
         $w \leftarrow$  “próximo” vértice em  $\Gamma_G(v)$ 
        Se  $w.estado \neq 0$ 
            Se  $\{v, w\}$  não foi processada
                processe  $\{v, w\}$ 
        Senão
            processe  $\{v, w\}$ 
            processe  $w$ 
             $w.pai \leftarrow v$ 
            acrescente  $w$  a  $V_1$ 
             $w.estado \leftarrow 1$ 
```

---

a cada volta do laço

# Algoritmo Genérico de Busca

---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$ 
processe  $r$ 
 $r.pai \leftarrow \Lambda$ 
acrescente  $r$  a  $V_1$ 
 $r.estado \leftarrow 1$ 
Enquanto  $V_1 \neq \emptyset$ 
     $v \leftarrow$  vértice em  $V_1$ 
    Se não há “próximo” vértice em  $\Gamma_G(v)$ 
        retire  $v$  de  $V_1$ 
         $v.estado \leftarrow 2$ 
    Senão
         $w \leftarrow$  “próximo” vértice em  $\Gamma_G(v)$ 
        Se  $w.estado \neq 0$ 
            Se  $\{v, w\}$  não foi processada
                processe  $\{v, w\}$ 
        Senão
            processe  $\{v, w\}$ 
            processe  $w$ 
             $w.pai \leftarrow v$ 
            acrescente  $w$  a  $V_1$ 
             $w.estado \leftarrow 1$ 
```

---

a cada volta do laço

1. um vértice  $v \in V_1$  é escolhido e

# Algoritmo Genérico de Busca

---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$ 
processe  $r$ 
 $r.pai \leftarrow \Lambda$ 
acrescente  $r$  a  $V_1$ 
 $r.estado \leftarrow 1$ 
Enquanto  $V_1 \neq \emptyset$ 
     $v \leftarrow$  vértice em  $V_1$ 
    Se não há “próximo” vértice em  $\Gamma_G(v)$ 
        retire  $v$  de  $V_1$ 
         $v.estado \leftarrow 2$ 
    Senão
         $w \leftarrow$  “próximo” vértice em  $\Gamma_G(v)$ 
        Se  $w.estado \neq 0$ 
            Se  $\{v, w\}$  não foi processada
                processe  $\{v, w\}$ 
        Senão
            processe  $\{v, w\}$ 
            processe  $w$ 
             $w.pai \leftarrow v$ 
            acrescente  $w$  a  $V_1$ 
             $w.estado \leftarrow 1$ 
```

---

a cada volta do laço

1. um vértice  $v \in V_1$  é escolhido e

1.1 um novo vizinho  $w$  de  $v$  é escolhido, ou

# Algoritmo Genérico de Busca

---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$ 
processe  $r$ 
 $r.pai \leftarrow \Lambda$ 
acrescente  $r$  a  $V_1$ 
 $r.estado \leftarrow 1$ 
Enquanto  $V_1 \neq \emptyset$ 
     $v \leftarrow$  vértice em  $V_1$ 
    Se não há “próximo” vértice em  $\Gamma_G(v)$ 
        retire  $v$  de  $V_1$ 
         $v.estado \leftarrow 2$ 
    Senão
         $w \leftarrow$  “próximo” vértice em  $\Gamma_G(v)$ 
        Se  $w.estado \neq 0$ 
            Se  $\{v, w\}$  não foi processada
                processe  $\{v, w\}$ 
        Senão
            processe  $\{v, w\}$ 
            processe  $w$ 
             $w.pai \leftarrow v$ 
            acrescente  $w$  a  $V_1$ 
             $w.estado \leftarrow 1$ 
```

---

a cada volta do laço

1. um vértice  $v \in V_1$  é escolhido e

1.1 um novo vizinho  $w$  de  $v$  é escolhido, ou

1.2  $v$  sai de  $V_1$  e vai para  $V_2$ , de forma que

# Algoritmo Genérico de Busca

---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$ 
processe  $r$ 
 $r.pai \leftarrow \Lambda$ 
acrescente  $r$  a  $V_1$ 
 $r.estado \leftarrow 1$ 
Enquanto  $V_1 \neq \emptyset$ 
     $v \leftarrow$  vértice em  $V_1$ 
    Se não há “próximo” vértice em  $\Gamma_G(v)$ 
        retire  $v$  de  $V_1$ 
         $v.estado \leftarrow 2$ 
    Senão
         $w \leftarrow$  “próximo” vértice em  $\Gamma_G(v)$ 
        Se  $w.estado \neq 0$ 
            Se  $\{v, w\}$  não foi processada
                processe  $\{v, w\}$ 
        Senão
            processe  $\{v, w\}$ 
            processe  $w$ 
             $w.pai \leftarrow v$ 
            acrescente  $w$  a  $V_1$ 
             $w.estado \leftarrow 1$ 
```

---

a cada volta do laço

1. um vértice  $v \in V_1$  é escolhido e
  - 1.1 um novo vizinho  $w$  de  $v$  é escolhido, ou
  - 1.2  $v$  sai de  $V_1$  e vai para  $V_2$ , de forma que
2. cada vértice  $v$  é escolhido exatamente  $\delta_G(v) + 1$  vezes

# Algoritmo Genérico de Busca

---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$ 
processe  $r$ 
 $r.pai \leftarrow \Lambda$ 
acrescente  $r$  a  $V_1$ 
 $r.estado \leftarrow 1$ 
Enquanto  $V_1 \neq \emptyset$ 
     $v \leftarrow$  vértice em  $V_1$ 
    Se não há “próximo” vértice em  $\Gamma_G(v)$ 
        retire  $v$  de  $V_1$ 
         $v.estado \leftarrow 2$ 
    Senão
         $w \leftarrow$  “próximo” vértice em  $\Gamma_G(v)$ 
        Se  $w.estado \neq 0$ 
            Se  $\{v, w\}$  não foi processada
                processe  $\{v, w\}$ 
        Senão
            processe  $\{v, w\}$ 
            processe  $w$ 
             $w.pai \leftarrow v$ 
            acrescente  $w$  a  $V_1$ 
             $w.estado \leftarrow 1$ 
```

---

a cada volta do laço

1. um vértice  $v \in V_1$  é escolhido e
  - 1.1 um novo vizinho  $w$  de  $v$  é escolhido, ou
  - 1.2  $v$  sai de  $V_1$  e vai para  $V_2$ , de forma que
2. cada vértice  $v$  é escolhido exatamente  $\delta_G(v) + 1$  vezes
3. o laço é executado exatamente  $\sum_{v \in V(G_r)} (\delta_G(v) + 1) = 2|E(G_r)| + |V(G_r)|$  vezes.

# Algoritmo Genérico de Busca - Execução

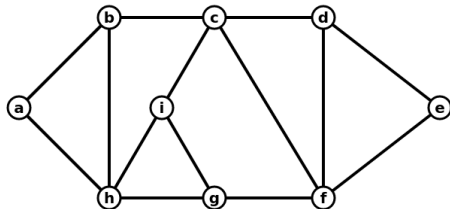
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

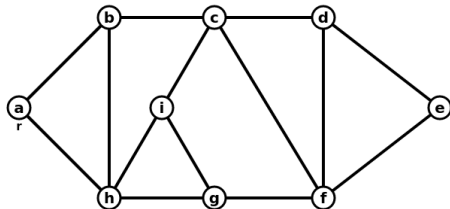
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

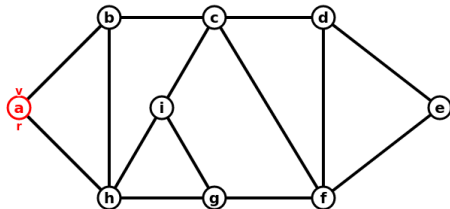
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

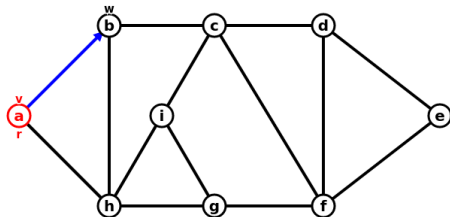
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

---

## Busca( $G, r$ )

---

$V_1 \leftarrow \emptyset$

processe  $r$

$r.pai \leftarrow \Lambda$

acrescente  $r$  a  $V_1$

$r.estado \leftarrow 1$

Enquanto  $V_1 \neq \emptyset$

$v \leftarrow$  vértice em  $V_1$

    Se não há "próximo" vértice em  $\Gamma_G(v)$

         retire  $v$  de  $V_1$

$v.estado \leftarrow 2$

    Senão

$w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$

        Se  $w.estado \neq 0$

            Se  $\{v, w\}$  não foi processada

                processe  $\{v, w\}$

        Senão

            processe  $\{v, w\}$

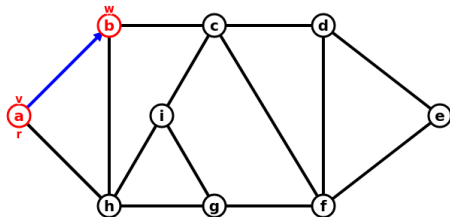
            processe  $w$

$w.pai \leftarrow v$

            acrescente  $w$  a  $V_1$

$w.estado \leftarrow 1$

---



# Algoritmo Genérico de Busca - Execução

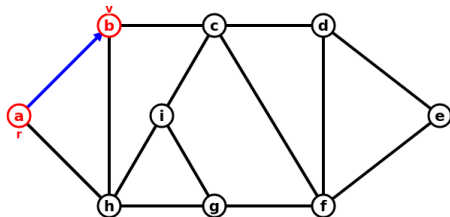
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

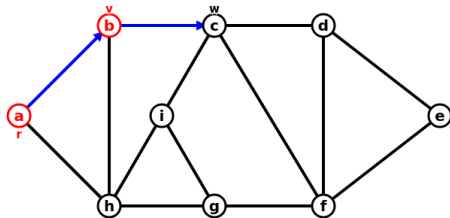
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

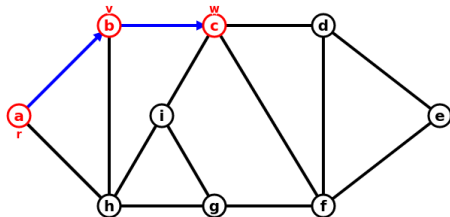
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há “próximo” vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  “próximo” vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

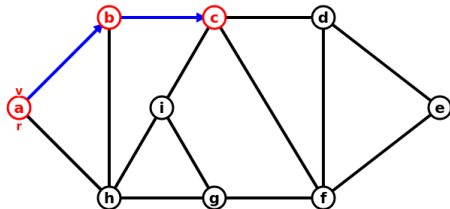
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há “próximo” vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  “próximo” vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

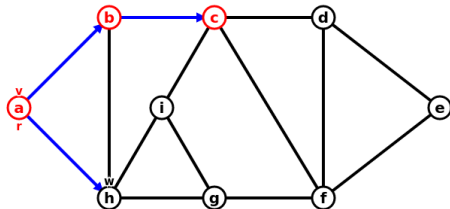
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

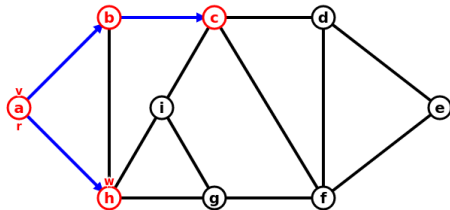
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$ 
processe  $r$ 
 $r.pai \leftarrow \Lambda$ 
acrescente  $r$  a  $V_1$ 
 $r.estado \leftarrow 1$ 
Enquanto  $V_1 \neq \emptyset$ 
   $v \leftarrow$  vértice em  $V_1$ 
  Se não há "próximo" vértice em  $\Gamma_G(v)$ 
    retire  $v$  de  $V_1$ 
     $v.estado \leftarrow 2$ 
  Senão
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$ 
    Se  $w.estado \neq 0$ 
      Se  $\{v, w\}$  não foi processada
        processe  $\{v, w\}$ 
    Senão
      processe  $\{v, w\}$ 
      processe  $w$ 
       $w.pai \leftarrow v$ 
      acrescente  $w$  a  $V_1$ 
       $w.estado \leftarrow 1$ 
```

---



## Algoritmo Genérico de Busca - Execução

$$\text{Busca}(G, r)$$
$$V_1 \leftarrow \emptyset$$

processe  $r$

$$r.pai \leftarrow \Lambda$$

acrescente  $r$  a  $V_1$

```
r.estado ← 1
```

Enquanto  $V_1 \neq \emptyset$

$$v \leftarrow \text{vértice em } V_1$$

Se não há “próximo” vértice em  $\Gamma_G(v)$

retire  $v$  de  $V_1$

```
v.estado ← 2
```

Senão

$$w \leftarrow \text{"pr\u00f3ximo" v\u00e9rtice em } \Gamma_G(v)$$

Se  $w.\text{estado} \neq 0$

Se  $\{v, w\}$  não foi processada

processe  $\{v, w\}$

Senão

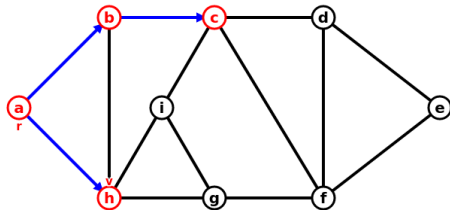
processe  $\{v, w\}$

processe w

$$w.\text{pai} \leftarrow v$$

acrescente w a  $V_1$

```
w.estado ← 1
```



## Algoritmo Genérico de Busca - Execução

$$\text{Busca}(G, r)$$
$$V_1 \leftarrow \emptyset$$

processe  $r$

$$r.pai \leftarrow \Lambda$$

acrescente  $r$  a  $V_1$

```
r.estado ← 1
```

Enquanto  $V_1 \neq \emptyset$

$$v \leftarrow \text{vértice em } V_1$$

Se não há “próximo” vértice em  $\Gamma_G(v)$

retire  $v$  de  $V_1$

```
v.estado ← 2
```

Senão

$$w \leftarrow \text{"pr\u00f3ximo" v\u00e9rtice em } \Gamma_G(v)$$

Se  $w.\text{estado} \neq 0$

Se  $\{v, w\}$  não foi processada

processe  $\{v, w\}$

Senão

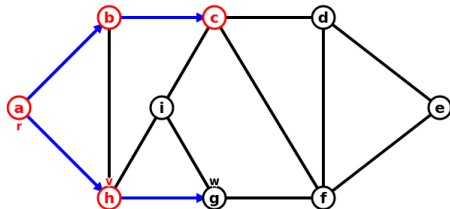
processe  $\{v, w\}$

processe w

$$w.pai \leftarrow v$$

acrescente w a  $V_1$

```
w.estado ← 1
```



# Algoritmo Genérico de Busca - Execução

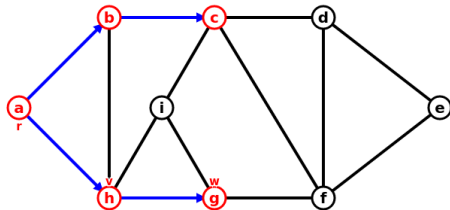
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

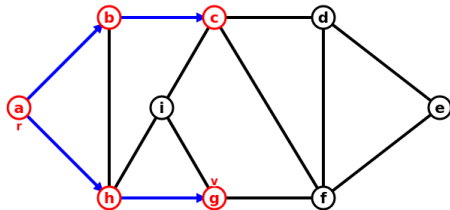
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

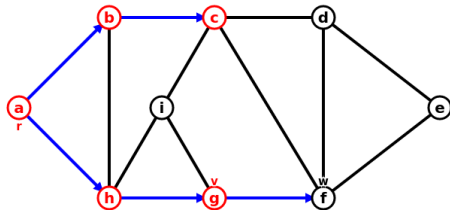
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

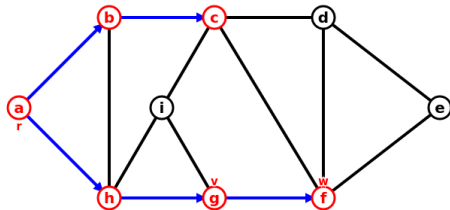
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

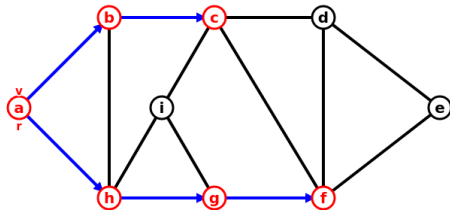
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

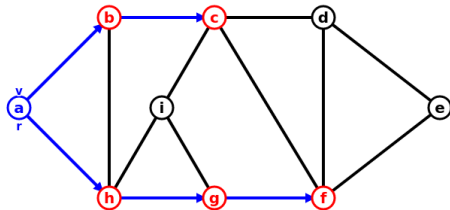
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

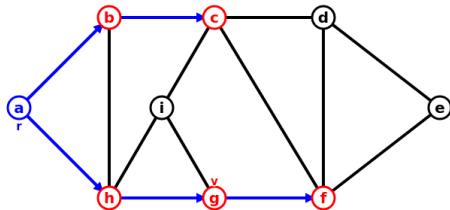
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

---

## Busca( $G, r$ )

---

$V_1 \leftarrow \emptyset$

processe  $r$

$r.pai \leftarrow \Lambda$

acrescente  $r$  a  $V_1$

$r.estado \leftarrow 1$

Enquanto  $V_1 \neq \emptyset$

$v \leftarrow$  vértice em  $V_1$

    Se não há “próximo” vértice em  $\Gamma_G(v)$

        retire  $v$  de  $V_1$

$v.estado \leftarrow 2$

    Senão

$w \leftarrow$  “próximo” vértice em  $\Gamma_G(v)$

        Se  $w.estado \neq 0$

            Se  $\{v, w\}$  não foi processada

                processe  $\{v, w\}$

        Senão

            processe  $\{v, w\}$

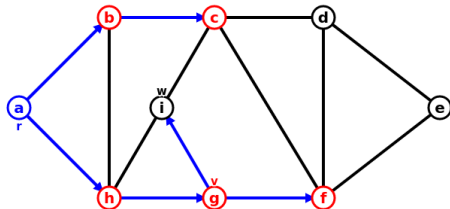
            processe  $w$

$w.pai \leftarrow v$

            acrescente  $w$  a  $V_1$

$w.estado \leftarrow 1$

---



# Algoritmo Genérico de Busca - Execução

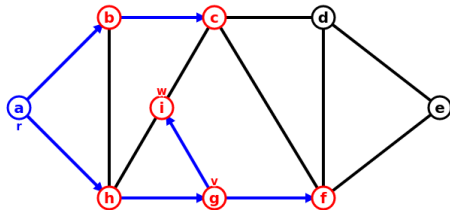
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

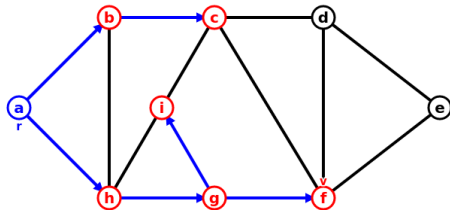
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

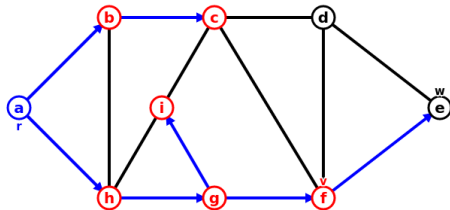
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

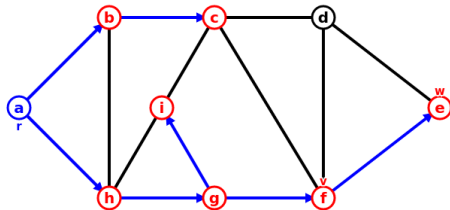
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

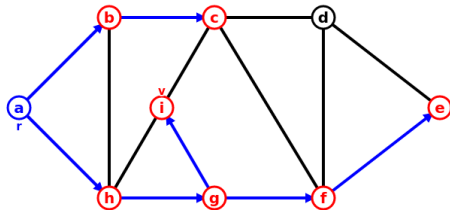
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

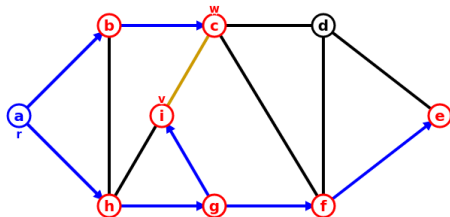
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

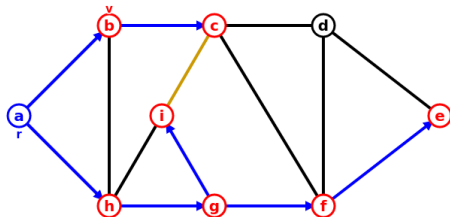
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

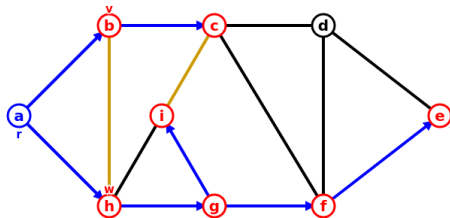
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

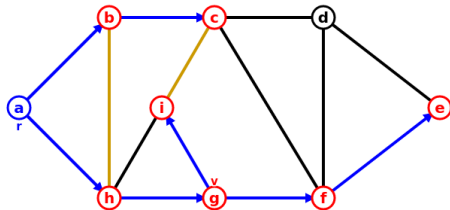
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

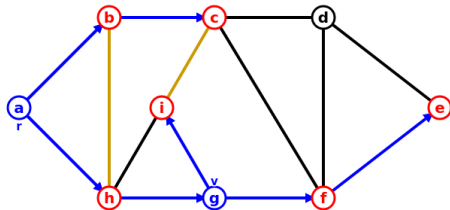
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há “próximo” vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  “próximo” vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

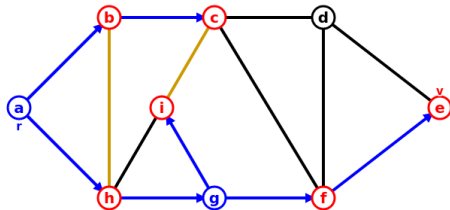
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

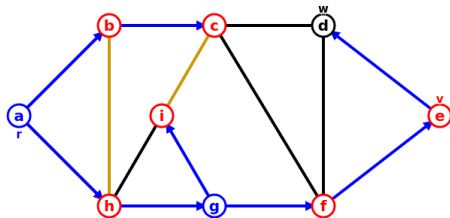
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

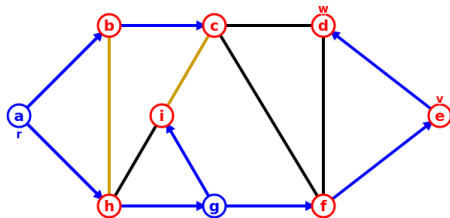
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

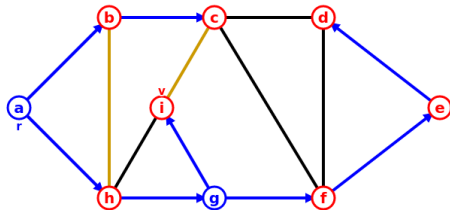
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

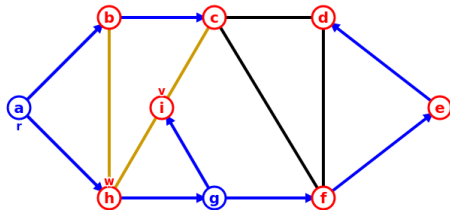
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

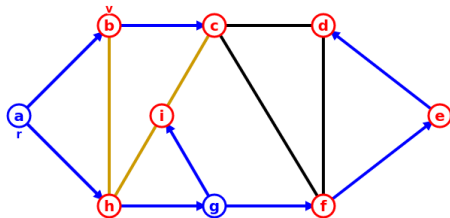
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

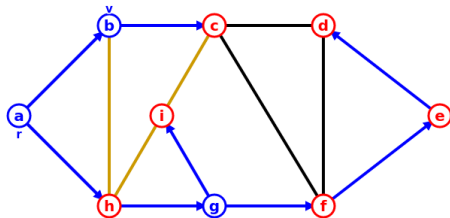
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

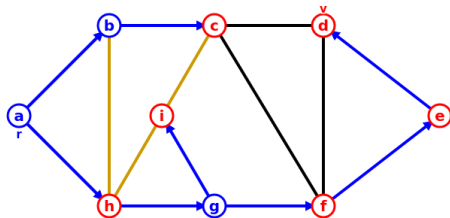
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

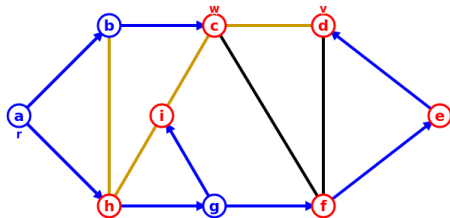
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

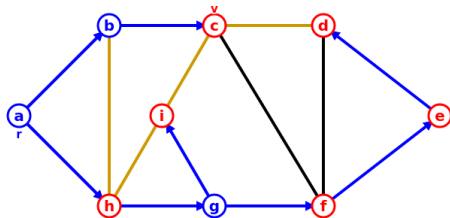
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

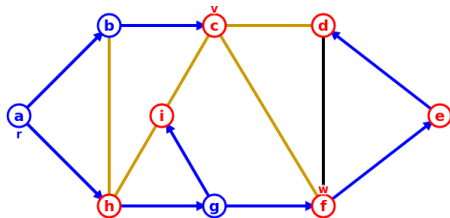
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

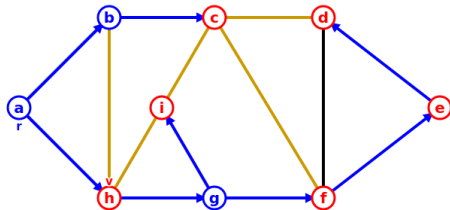
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

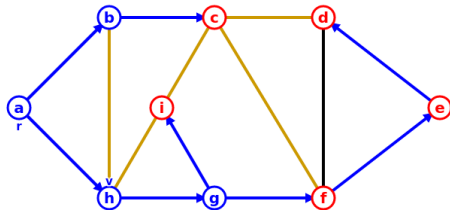
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

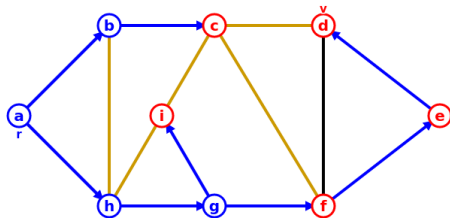
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

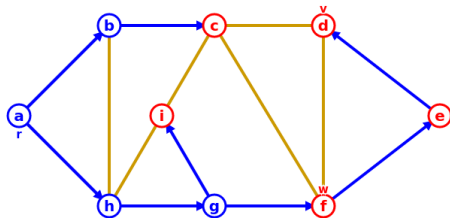
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



## Algoritmo Genérico de Busca - Execução

$$\text{Busca}(G, r)$$
$$V_1 \leftarrow \emptyset$$

processe  $r$

$$r.pai \leftarrow \Lambda$$

acrescente  $r$  a  $V_1$

```
r.estado ← 1
```

Enquanto  $V_1 \neq \emptyset$

$$v \leftarrow \text{vértice em } V_1$$

Se não há “próximo” vértice em  $\Gamma_G(v)$

retire  $v$  de  $V_1$

```
v.estado ← 2
```

Senão

$$w \leftarrow \text{"pr\u00f3ximo" v\u00e9rtice em } \Gamma_G(v)$$

Se  $w.\text{estado} \neq 0$

Se  $\{v, w\}$  não foi processada

processe  $\{v, w\}$

Senão

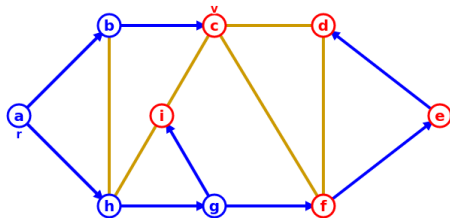
processe  $\{v, w\}$

processe w

$$w.pai \leftarrow v$$

acrescente w a  $V_1$

```
w.estado ← 1
```



# Algoritmo Genérico de Busca - Execução

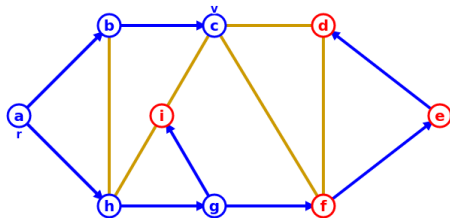
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

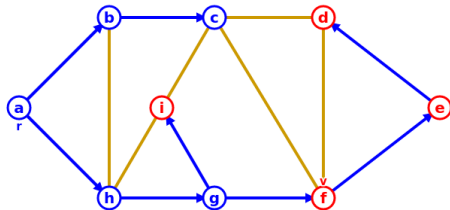
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

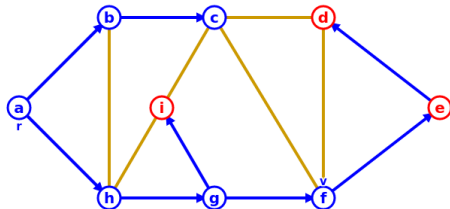
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

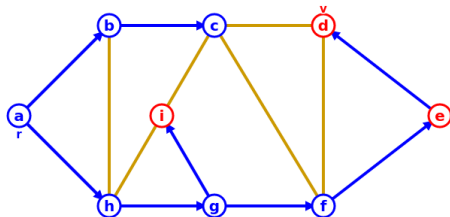
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

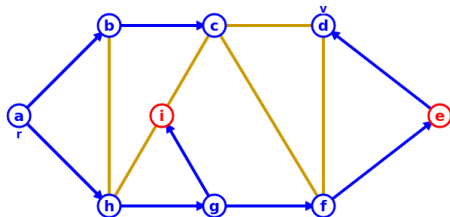
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

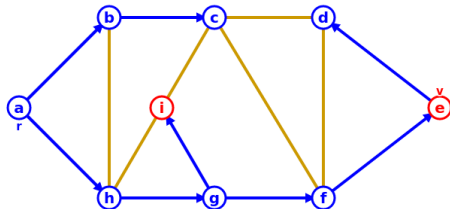
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

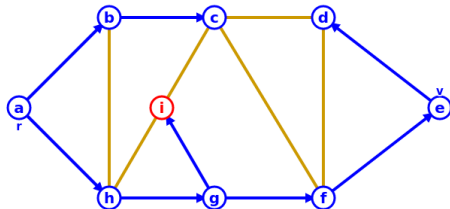
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

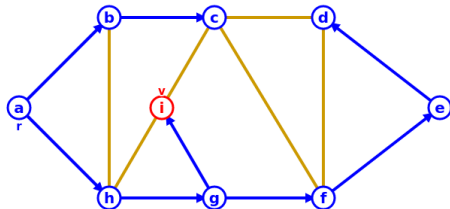
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há "próximo" vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  "próximo" vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



# Algoritmo Genérico de Busca - Execução

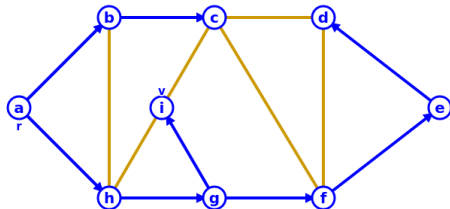
---

## Busca( $G, r$ )

---

```
 $V_1 \leftarrow \emptyset$   
processe  $r$   
 $r.pai \leftarrow \Lambda$   
acrescente  $r$  a  $V_1$   
 $r.estado \leftarrow 1$   
Enquanto  $V_1 \neq \emptyset$   
   $v \leftarrow$  vértice em  $V_1$   
  Se não há “próximo” vértice em  $\Gamma_G(v)$   
    retire  $v$  de  $V_1$   
     $v.estado \leftarrow 2$   
  Senão  
     $w \leftarrow$  “próximo” vértice em  $\Gamma_G(v)$   
    Se  $w.estado \neq 0$   
      Se  $\{v, w\}$  não foi processada  
        processe  $\{v, w\}$   
    Senão  
      processe  $\{v, w\}$   
      processe  $w$   
       $w.pai \leftarrow v$   
      acrescente  $w$  a  $V_1$   
       $w.estado \leftarrow 1$ 
```

---



## Teorema 52

$G_r$  : componente de  $G$  que contém o vértice  $r$

## Teorema 52

$G_r$  : componente de  $G$  que contém o vértice  $r$

Busca( $G, r$ ) devolve árvore enraizada  $(T, r)$  geradora de  $G_r$

## Teorema 52

$G_r$  : componente de  $G$  que contém o vértice  $r$

Busca( $G, r$ ) devolve árvore enraizada  $(T, r)$  geradora de  $G_r$

processa todos os vértices e arestas de  $G_r$

## Teorema 52

$G_r$  : componente de  $G$  que contém o vértice  $r$

$\text{Busca}(G, r)$  devolve árvore enraizada  $(T, r)$  geradora de  $G_r$

processa todos os vértices e arestas de  $G_r$

$2|E(G_r)| + |V(G_r)|$  voltas do laço

## Teorema 52

$G_r$  : componente de  $G$  que contém o vértice  $r$

$\text{Busca}(G, r)$  devolve árvore enraizada  $(T, r)$  geradora de  $G_r$

processa todos os vértices e arestas de  $G_r$

$2|E(G_r)| + |V(G_r)|$  voltas do laço

Diferentes árvores geradoras de  $G_r$  são construídas, dependendo dos vértices escolhidos em cada iteração.

# Algoritmo Genérico de Busca - Versão Compacta

Uma versão mais compacta e restrita:

---

$\text{Busca}(G, r)$

---

```
processe  $r$ 
 $r.\text{pai} \leftarrow \Lambda$ 
 $r.\text{estado} \leftarrow 1$ 
 $V \leftarrow \{r\}$ 
Enquanto  $V \neq \emptyset$ 
    retire um vértice  $v$  de  $V$ 
    Para cada  $w \in \Gamma_G(v)$ 
        Se  $w.\text{estado} = 1$ 
            processe  $\{v, w\}$ 
        Senão, se  $w.\text{estado} = 0$ 
            processe  $w$ 
            processe  $\{v, w\}$ 
             $w.\text{pai} \leftarrow v$ 
             $w.\text{estado} \leftarrow 1$ 
            acrescente  $w$  a  $V$ 
     $v.\text{estado} \leftarrow 2$ 
```

---

# Componentes Conexos

---

## Componentes( $G$ )

---

```
Para cada  $v \in V(G)$   
   $v.estado \leftarrow 0$   
   $v.componente \leftarrow 0$   
 $c \leftarrow 0$   
Para cada  $v \in V(G)$   
  Se  $v.estado = 0$   
     $v.componente \leftarrow ++c$   
    Componente( $G, v$ )
```

---

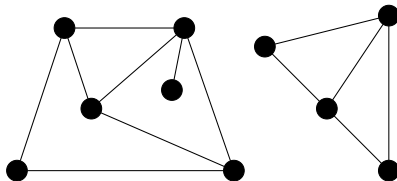
---

## Componente( $G, r$ )

---

```
 $V \leftarrow \emptyset$   
acrescente  $r$  a  $V$   
 $r.estado \leftarrow 1$   
Enquanto  $V \neq \emptyset$   
  retire um vértice  $v$  de  $V$   
  Para cada  $w \in \Gamma_G(v)$   
    Se  $w.estado = 0$   
       $w.componente \leftarrow r.componente$   
      acrescente  $w$  a  $V$   
       $w.estado \leftarrow 1$   
  
   $v.estado \leftarrow 2$ 
```

---



## Teorema

É possível determinar os componentes de um grafo de  $n$  vértices e  $m$  arestas em tempo  $O(n + m)$ .

# Teorema

É possível determinar os componentes de um grafo de  $n$  vértices e  $m$  arestas em tempo  $O(n + m)$ .

Demonstração:

## Teorema

É possível determinar os componentes de um grafo de  $n$  vértices e  $m$  arestas em tempo  $O(n + m)$ .

Demonstração:

Basta implementar o conjunto  $V$  no Algoritmo Componentes( $G, r$ ) por meio de uma lista encadeada.