

Algoritmos e Teoria dos Grafos

Tópico 26: Busca em Grafos Direcionados

Renato Carmo
André Guedes
Murilo Silva

Departamento de Informática da UFPR

2025/1

Busca em Grafos Direcionados

Busca(G)

$F \leftarrow$ grafo direcionado vazio
Para $v \in V(G)$
 $v.\text{estado} \leftarrow 0$
Para $v \in V(G)$
 Se $v.\text{estado} = 0$
 $(T, v) \leftarrow \text{Busca}(G, v)$
 acrescente T a F

Devolva F

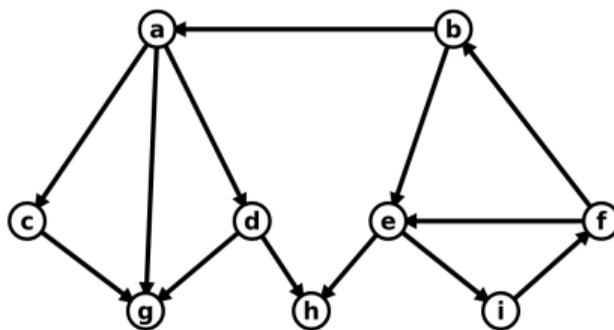
Busca(G, r)

$V_1 \leftarrow \emptyset$
processe r
 $r.\text{pai} \leftarrow \Lambda$
acrescente r a V_1
 $r.\text{estado} \leftarrow 1$
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
 $v.\text{estado} \leftarrow 2$
 Senão
 $w \leftarrow$ "próximo" vértice 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
 $w.\text{estado} \leftarrow 1$

Busca em Grafos Direcionados - Execução

Busca(G, r)

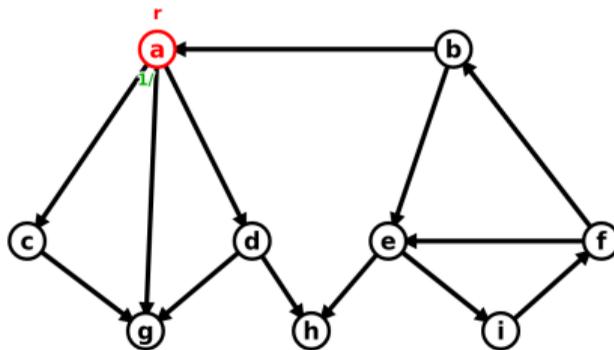
$V_1 \leftarrow \emptyset$
processe r
 $r.pai \leftarrow \Lambda$
acrescente r a V_1
 $r.estado \leftarrow 1$
Enquanto $V_1 \neq \emptyset$
 $v \leftarrow$ vértice em V_1
 Se **não há "próximo" vértice em $\Gamma_G^+(v)$**
 retire v de V_1
 $v.estado \leftarrow 2$
 Senão
 $w \leftarrow$ "próximo" vértice 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
 $w.estado \leftarrow 1$



Busca em Grafos Direcionados - Execução

Busca(G, r)

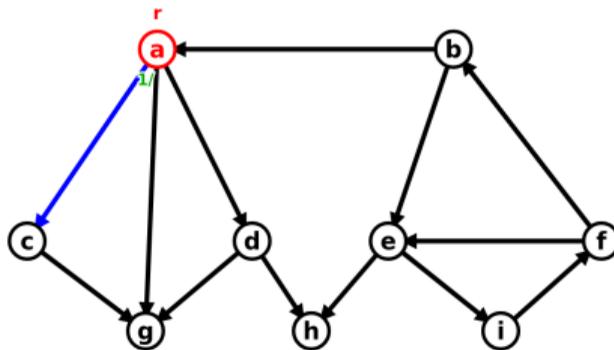
$V_1 \leftarrow \emptyset$
processe r
 $r.pai \leftarrow \Lambda$
acrescente r a V_1
 $r.estado \leftarrow 1$
Enquanto $V_1 \neq \emptyset$
 $v \leftarrow$ vértice em V_1
 Se **não há "próximo" vértice em $\Gamma_G^+(v)$**
 retire v de V_1
 $v.estado \leftarrow 2$
 Senão
 $w \leftarrow$ "próximo" vértice 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
 $w.estado \leftarrow 1$



Busca em Grafos Direcionados - Execução

Busca(G, r)

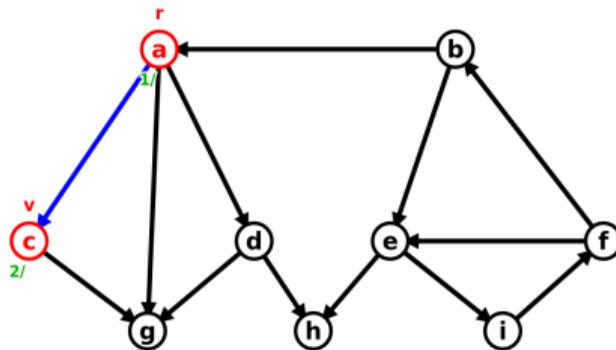
$V_1 \leftarrow \emptyset$
processe r
 $r.pai \leftarrow \Lambda$
acrescente r a V_1
 $r.estado \leftarrow 1$
Enquanto $V_1 \neq \emptyset$
 $v \leftarrow$ vértice em V_1
 Se **não há "próximo" vértice em $\Gamma_G^+(v)$**
 retire v de V_1
 $v.estado \leftarrow 2$
 Senão
 $w \leftarrow$ "próximo" vértice 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
 $w.estado \leftarrow 1$



Busca em Grafos Direcionados - Execução

Busca(G, r)

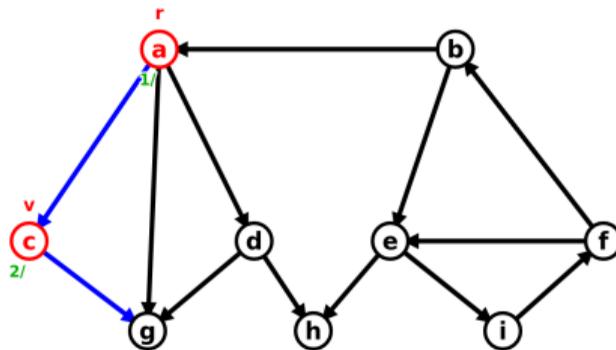
$V_1 \leftarrow \emptyset$
processe r
 $r.pai \leftarrow \Lambda$
acrescente r a V_1
 $r.estado \leftarrow 1$
Enquanto $V_1 \neq \emptyset$
 $v \leftarrow$ vértice em V_1
 Se **não há "próximo" vértice em $\Gamma_G^+(v)$**
 retire v de V_1
 $v.estado \leftarrow 2$
 Senão
 $w \leftarrow$ "próximo" vértice 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
 $w.estado \leftarrow 1$



Busca em Grafos Direcionados - Execução

Busca(G, r)

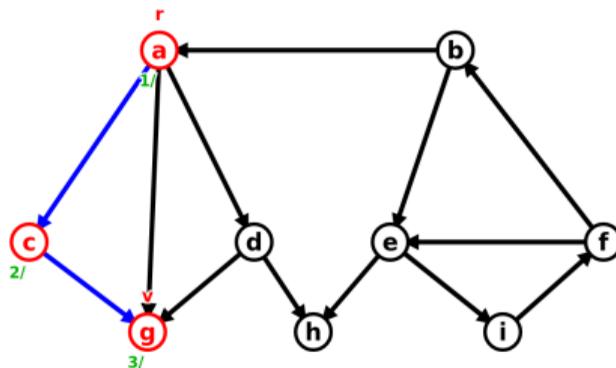
$V_1 \leftarrow \emptyset$
processe r
 $r.pai \leftarrow \Lambda$
acrescente r a V_1
 $r.estado \leftarrow 1$
Enquanto $V_1 \neq \emptyset$
 $v \leftarrow$ vértice em V_1
 Se **não há "próximo" vértice** em $\Gamma_G^+(v)$
 retire v de V_1
 $v.estado \leftarrow 2$
 Senão
 $w \leftarrow$ "próximo" vértice 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
 $w.estado \leftarrow 1$



Busca em Grafos Direcionados - Execução

Busca(G, r)

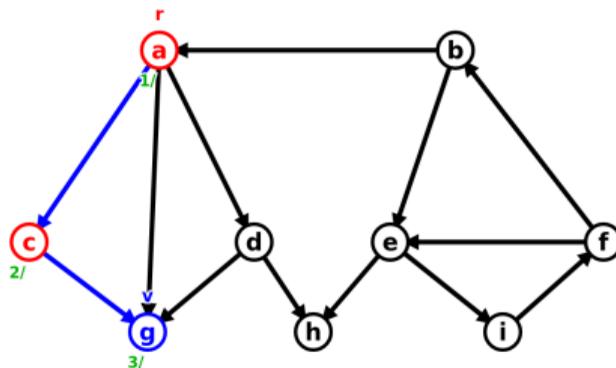
$V_1 \leftarrow \emptyset$
processe r
 $r.pai \leftarrow \Lambda$
acrescente r a V_1
 $r.estado \leftarrow 1$
Enquanto $V_1 \neq \emptyset$
 $v \leftarrow$ vértice em V_1
 Se **não há "próximo" vértice em $\Gamma_G^+(v)$**
 retire v de V_1
 $v.estado \leftarrow 2$
 Senão
 $w \leftarrow$ "próximo" vértice 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
 $w.estado \leftarrow 1$



Busca em Grafos Direcionados - 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
 $w.estado \leftarrow 1$



Busca em Grafos Direcionados - 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

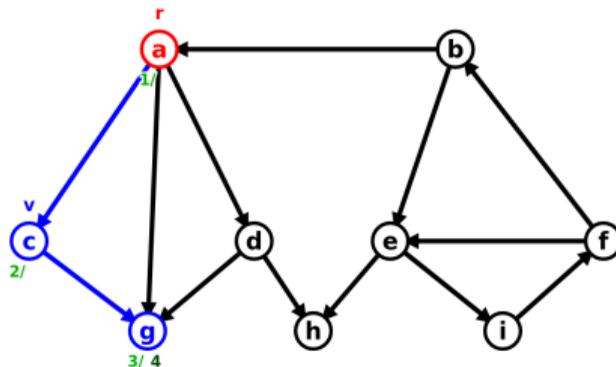
 processe (v, w)

 processe w

$w.pai \leftarrow v$

 acrescente w a V_1

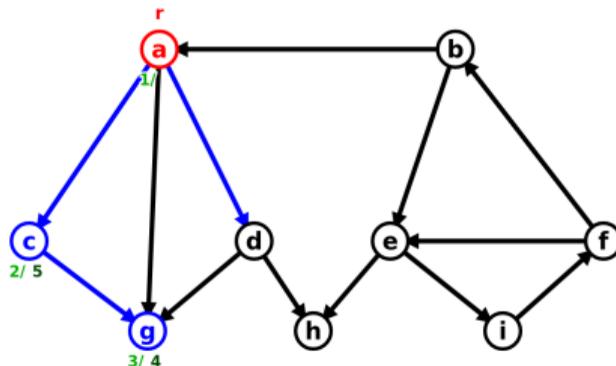
$w.estado \leftarrow 1$



Busca em Grafos Direcionados - Execução

Busca(G, r)

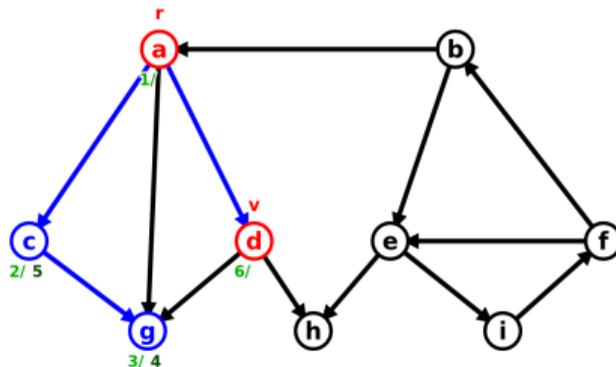
$V_1 \leftarrow \emptyset$
processe r
 $r.pai \leftarrow \Lambda$
acrescente r a V_1
 $r.estado \leftarrow 1$
Enquanto $V_1 \neq \emptyset$
 $v \leftarrow$ vértice em V_1
 Se **não há "próximo" vértice em $\Gamma_G^+(v)$**
 retire v de V_1
 $v.estado \leftarrow 2$
 Senão
 $w \leftarrow$ "próximo" vértice 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
 $w.estado \leftarrow 1$



Busca em Grafos Direcionados - Execução

Busca(G, r)

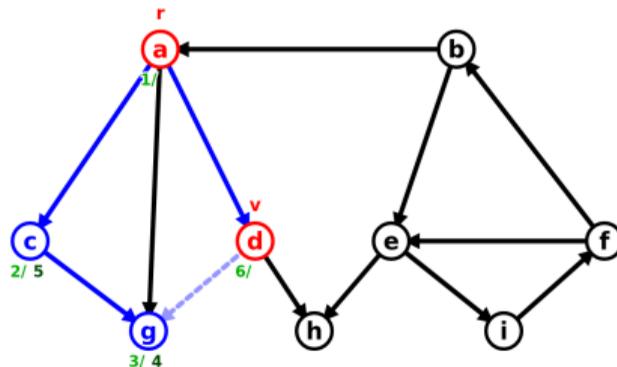
$V_1 \leftarrow \emptyset$
processe r
 $r.pai \leftarrow \Lambda$
acrescente r a V_1
 $r.estado \leftarrow 1$
Enquanto $V_1 \neq \emptyset$
 $v \leftarrow$ vértice em V_1
 Se **não há "próximo" vértice** em $\Gamma_G^+(v)$
 retire v de V_1
 $v.estado \leftarrow 2$
 Senão
 $w \leftarrow$ "próximo" vértice 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
 $w.estado \leftarrow 1$



Busca em Grafos Direcionados - Execução

Busca(G, r)

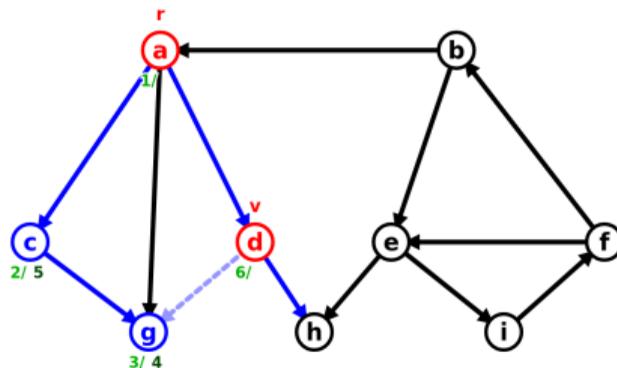
$V_1 \leftarrow \emptyset$
processe r
 $r.pai \leftarrow \Lambda$
acrescente r a V_1
 $r.estado \leftarrow 1$
Enquanto $V_1 \neq \emptyset$
 $v \leftarrow$ vértice em V_1
 Se **não há "próximo" vértice** em $\Gamma_G^+(v)$
 retire v de V_1
 $v.estado \leftarrow 2$
 Senão
 $w \leftarrow$ "próximo" vértice 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
 $w.estado \leftarrow 1$



Busca em Grafos Direcionados - Execução

Busca(G, r)

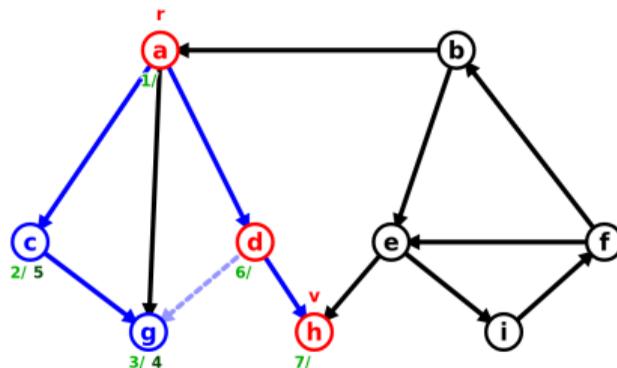
$V_1 \leftarrow \emptyset$
processe r
 $r.pai \leftarrow \Lambda$
acrescente r a V_1
 $r.estado \leftarrow 1$
Enquanto $V_1 \neq \emptyset$
 $v \leftarrow$ vértice em V_1
 Se **não há "próximo" vértice** em $\Gamma_G^+(v)$
 retire v de V_1
 $v.estado \leftarrow 2$
 Senão
 $w \leftarrow$ "próximo" vértice 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
 $w.estado \leftarrow 1$



Busca em Grafos Direcionados - Execução

Busca(G, r)

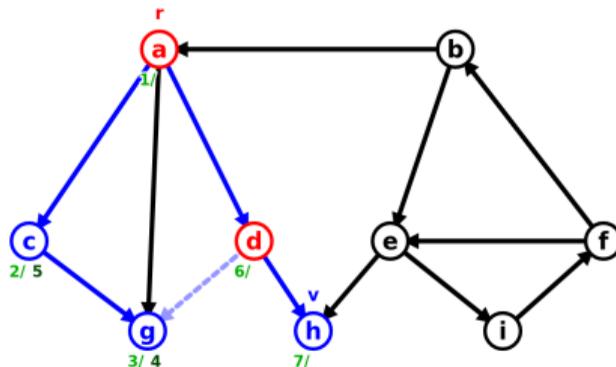
$V_1 \leftarrow \emptyset$
processe r
 $r.pai \leftarrow \Lambda$
acrescente r a V_1
 $r.estado \leftarrow 1$
Enquanto $V_1 \neq \emptyset$
 $v \leftarrow$ vértice em V_1
 Se **não há "próximo" vértice** em $\Gamma_G^+(v)$
 retire v de V_1
 $v.estado \leftarrow 2$
 Senão
 $w \leftarrow$ "próximo" vértice 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
 $w.estado \leftarrow 1$



Busca em Grafos Direcionados - Execução

Busca(G, r)

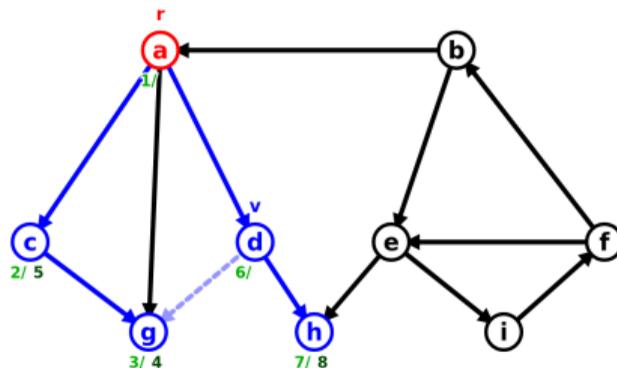
$V_1 \leftarrow \emptyset$
processe r
 $r.pai \leftarrow \Lambda$
acrescente r a V_1
 $r.estado \leftarrow 1$
Enquanto $V_1 \neq \emptyset$
 $v \leftarrow$ vértice em V_1
 Se **não há "próximo" vértice** em $\Gamma_G^+(v)$
 retire v de V_1
 $v.estado \leftarrow 2$
 Senão
 $w \leftarrow$ "próximo" vértice 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
 $w.estado \leftarrow 1$



Busca em Grafos Direcionados - 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
 $w.estado \leftarrow 1$



Busca em Grafos Direcionados - 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

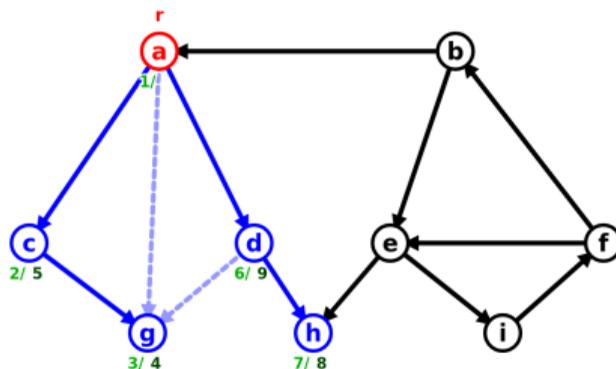
 processe (v, w)

 processe w

$w.pai \leftarrow v$

 acrescente w a V_1

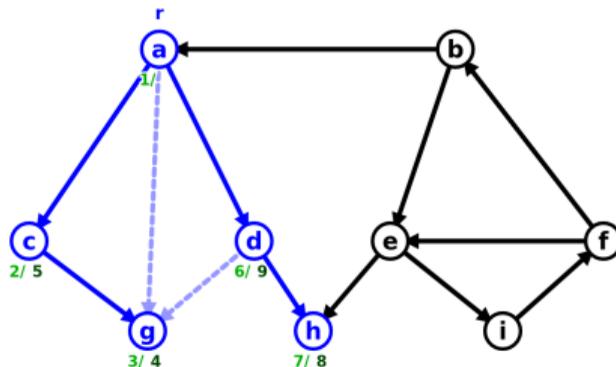
$w.estado \leftarrow 1$



Busca em Grafos Direcionados - Execução

Busca(G, r)

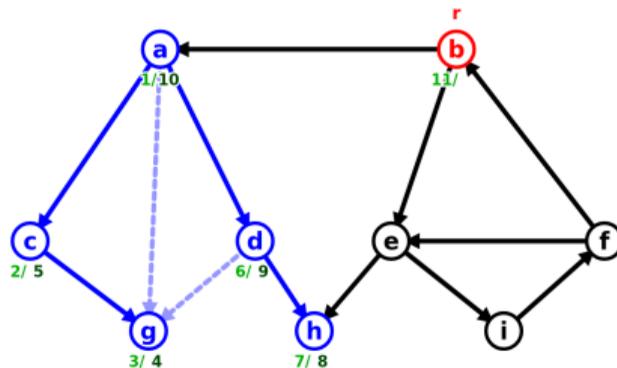
$V_1 \leftarrow \emptyset$
processe r
 $r.pai \leftarrow \Lambda$
acrescente r a V_1
 $r.estado \leftarrow 1$
Enquanto $V_1 \neq \emptyset$
 $v \leftarrow$ vértice em V_1
 Se **não há "próximo" vértice em $\Gamma_G^+(v)$**
 retire v de V_1
 $v.estado \leftarrow 2$
 Senão
 $w \leftarrow$ "próximo" vértice 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
 $w.estado \leftarrow 1$



Busca em Grafos Direcionados - Execução

Busca(G, r)

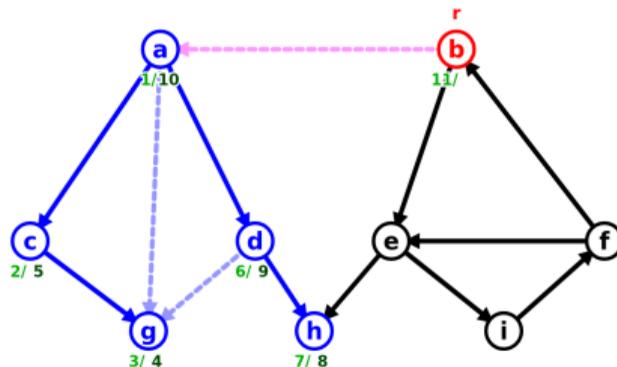
$V_1 \leftarrow \emptyset$
processe r
 $r.pai \leftarrow \Lambda$
acrescente r a V_1
 $r.estado \leftarrow 1$
Enquanto $V_1 \neq \emptyset$
 $v \leftarrow$ vértice em V_1
 Se **não há "próximo" vértice em $\Gamma_G^+(v)$**
 retire v de V_1
 $v.estado \leftarrow 2$
 Senão
 $w \leftarrow$ "próximo" vértice 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
 $w.estado \leftarrow 1$



Busca em Grafos Direcionados - Execução

Busca(G, r)

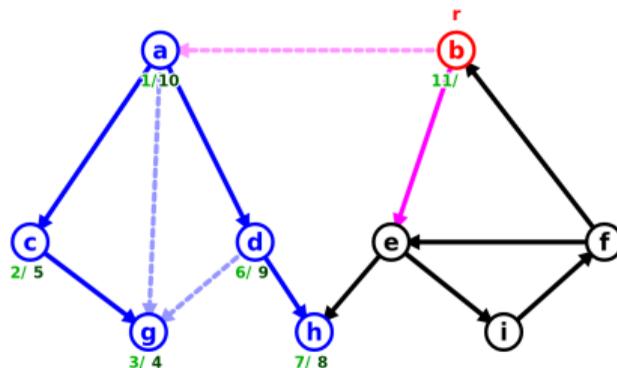
$V_1 \leftarrow \emptyset$
processe r
 $r.pai \leftarrow \Lambda$
acrescente r a V_1
 $r.estado \leftarrow 1$
Enquanto $V_1 \neq \emptyset$
 $v \leftarrow$ vértice em V_1
 Se **não há "próximo" vértice em $\Gamma_G^+(v)$**
 retire v de V_1
 $v.estado \leftarrow 2$
 Senão
 $w \leftarrow$ "próximo" vértice 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
 $w.estado \leftarrow 1$



Busca em Grafos Direcionados - 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
 $w.estado \leftarrow 1$



Busca em Grafos Direcionados - 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

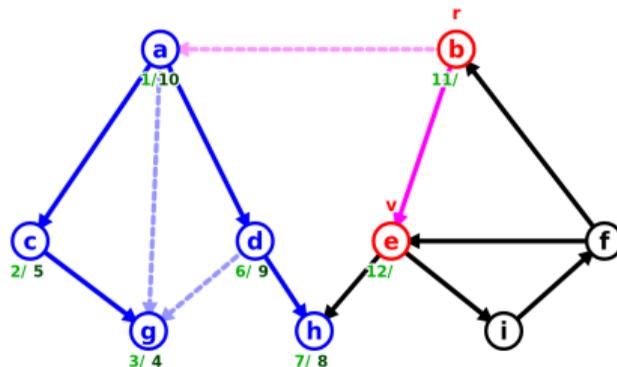
processe (v, w)

processe w

$w.pai \leftarrow v$

acrescente w a V_1

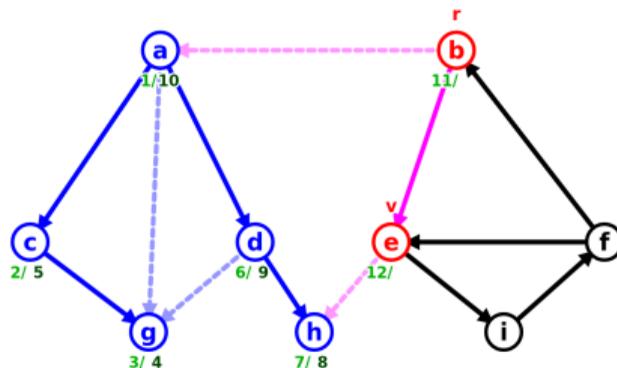
$w.estado \leftarrow 1$



Busca em Grafos Direcionados - Execução

Busca(G, r)

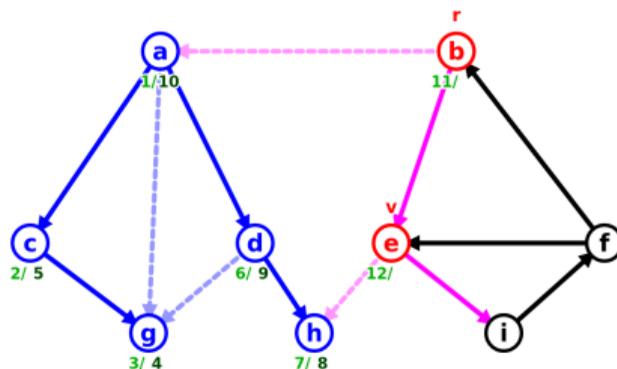
$V_1 \leftarrow \emptyset$
processe r
 $r.pai \leftarrow \Lambda$
acrescente r a V_1
 $r.estado \leftarrow 1$
Enquanto $V_1 \neq \emptyset$
 $v \leftarrow$ vértice em V_1
 Se **não há "próximo" vértice em $\Gamma_G^+(v)$**
 retire v de V_1
 $v.estado \leftarrow 2$
 Senão
 $w \leftarrow$ "próximo" vértice 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
 $w.estado \leftarrow 1$



Busca em Grafos Direcionados - Execução

Busca(G, r)

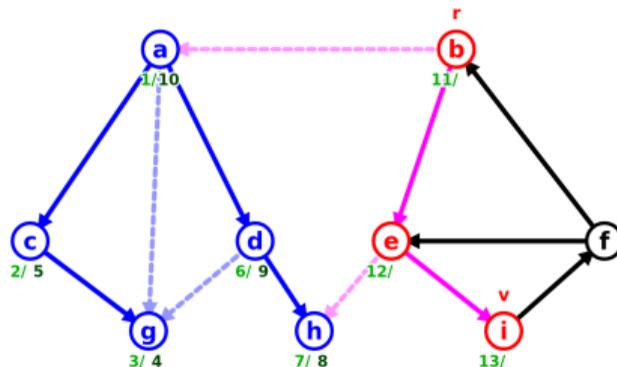
$V_1 \leftarrow \emptyset$
processe r
 $r.pai \leftarrow \Lambda$
acrescente r a V_1
 $r.estado \leftarrow 1$
Enquanto $V_1 \neq \emptyset$
 $v \leftarrow$ vértice em V_1
 Se **não há "próximo" vértice** em $\Gamma_G^+(v)$
 retire v de V_1
 $v.estado \leftarrow 2$
 Senão
 $w \leftarrow$ "próximo" vértice 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
 $w.estado \leftarrow 1$



Busca em Grafos Direcionados - 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
 $w.estado \leftarrow 1$



Busca em Grafos Direcionados - 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

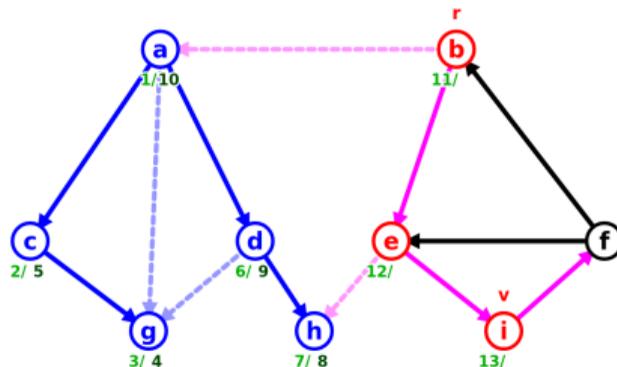
processe (v, w)

processe w

$w.pai \leftarrow v$

acrescente w a V_1

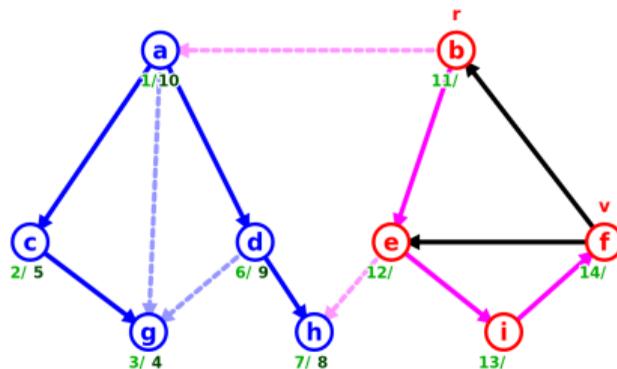
$w.estado \leftarrow 1$



Busca em Grafos Direcionados - Execução

Busca(G, r)

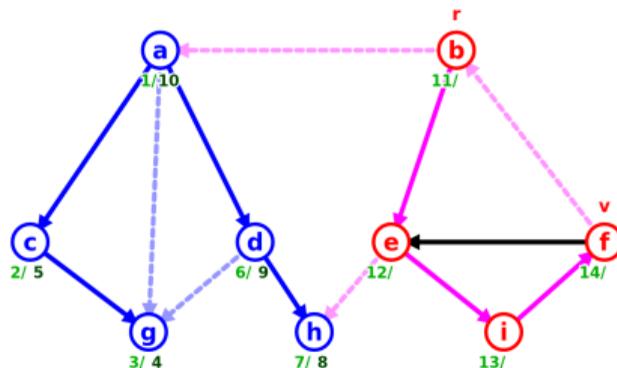
$V_1 \leftarrow \emptyset$
processe r
 $r.pai \leftarrow \Lambda$
acrescente r a V_1
 $r.estado \leftarrow 1$
Enquanto $V_1 \neq \emptyset$
 $v \leftarrow$ vértice em V_1
 Se **não há "próximo" vértice em $\Gamma_G^+(v)$**
 retire v de V_1
 $v.estado \leftarrow 2$
 Senão
 $w \leftarrow$ "próximo" vértice 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
 $w.estado \leftarrow 1$



Busca em Grafos Direcionados - Execução

Busca(G, r)

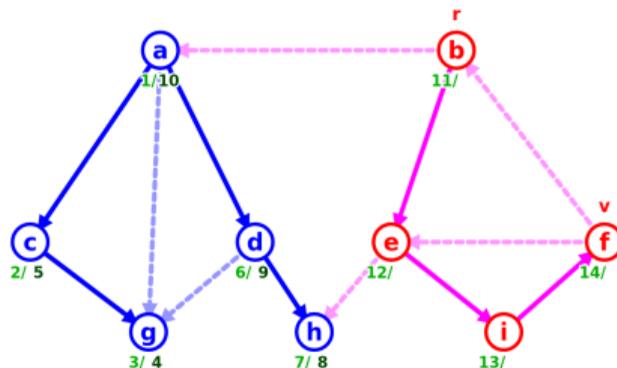
$V_1 \leftarrow \emptyset$
processe r
 $r.pai \leftarrow \Lambda$
acrescente r a V_1
 $r.estado \leftarrow 1$
Enquanto $V_1 \neq \emptyset$
 $v \leftarrow$ vértice em V_1
 Se **não há "próximo" vértice em $\Gamma_G^+(v)$**
 retire v de V_1
 $v.estado \leftarrow 2$
 Senão
 $w \leftarrow$ "próximo" vértice 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
 $w.estado \leftarrow 1$



Busca em Grafos Direcionados - Execução

Busca(G, r)

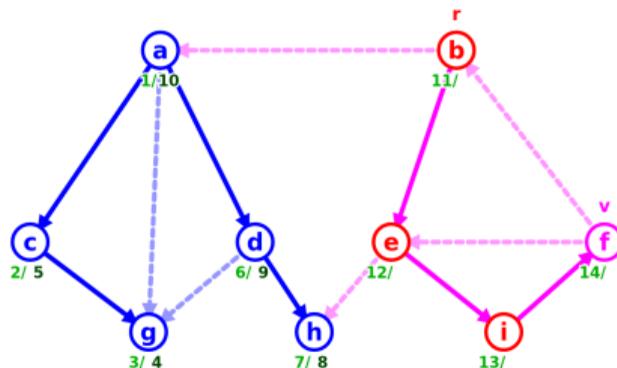
$V_1 \leftarrow \emptyset$
processe r
 $r.pai \leftarrow \Lambda$
acrescente r a V_1
 $r.estado \leftarrow 1$
Enquanto $V_1 \neq \emptyset$
 $v \leftarrow$ vértice em V_1
 Se **não há "próximo" vértice em $\Gamma_G^+(v)$**
 retire v de V_1
 $v.estado \leftarrow 2$
 Senão
 $w \leftarrow$ "próximo" vértice 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
 $w.estado \leftarrow 1$



Busca em Grafos Direcionados - 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
 $w.estado \leftarrow 1$



Busca em Grafos Direcionados - 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

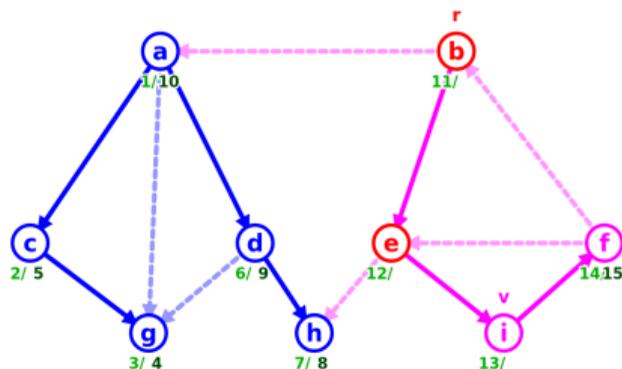
processe (v, w)

processe w

$w.pai \leftarrow v$

acrescente w a V_1

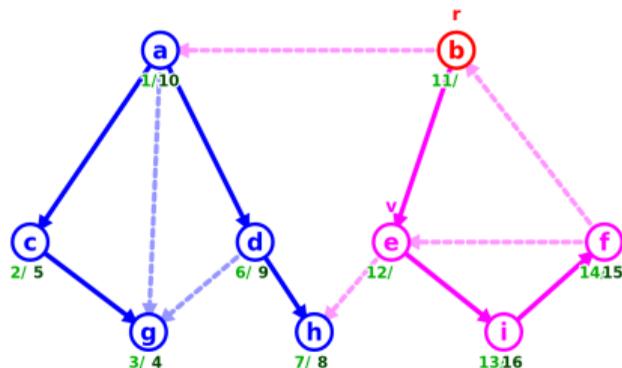
$w.estado \leftarrow 1$



Busca em Grafos Direcionados - Execução

Busca(G, r)

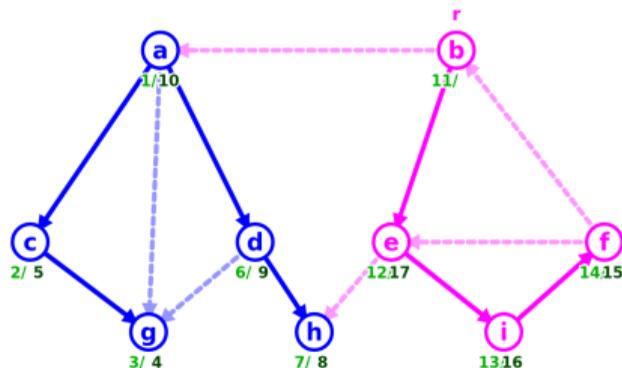
$V_1 \leftarrow \emptyset$
processe r
 $r.pai \leftarrow \Lambda$
acrescente r a V_1
 $r.estado \leftarrow 1$
Enquanto $V_1 \neq \emptyset$
 $v \leftarrow$ vértice em V_1
 Se **não há "próximo" vértice em $\Gamma_G^+(v)$**
 retire v de V_1
 $v.estado \leftarrow 2$
 Senão
 $w \leftarrow$ "próximo" vértice 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
 $w.estado \leftarrow 1$



Busca em Grafos Direcionados - Execução

Busca(G, r)

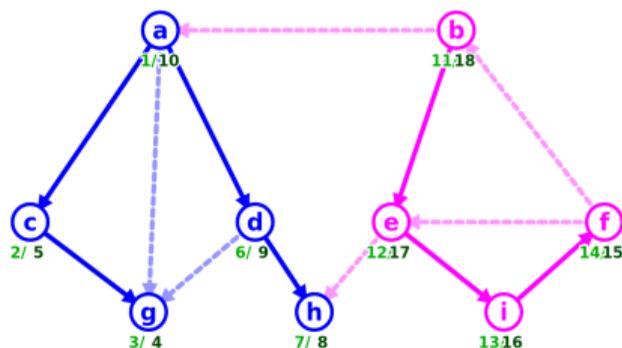
$V_1 \leftarrow \emptyset$
processe r
 $r.pai \leftarrow \Lambda$
acrescente r a V_1
 $r.estado \leftarrow 1$
Enquanto $V_1 \neq \emptyset$
 $v \leftarrow$ vértice em V_1
 Se **não há "próximo" vértice em $\Gamma_G^+(v)$**
 retire v de V_1
 $v.estado \leftarrow 2$
 Senão
 $w \leftarrow$ "próximo" vértice 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
 $w.estado \leftarrow 1$



Busca em Grafos Direcionados - 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
 $w.estado \leftarrow 1$



Busca em Grafos Direcionados

1. O tempo de execução de uma busca em grafo direcionado é a mesmo do caso não direcinado

Busca em Grafos Direcionados

1. O tempo de execução de uma busca em grafo direcionado é a mesmo do caso não direcinado (Obs: no caso não direcionado, cada aresta é “inspecionada” a partir das duas pontas)

Busca em Grafos Direcionados

1. O tempo de execução de uma busca em grafo direcionado é a mesmo do caso não direcinado (Obs: no caso não direcionado, cada aresta é “inspecionada” a partir das duas pontas)
2. Pode devolver florestas direcionadas, mesmo quando o grafo subjacente é conexo.

Busca em Grafos Direcionados

1. O tempo de execução de uma busca em grafo direcionado é a mesmo do caso não direcinado (Obs: no caso não direcionado, cada aresta é “inspecionada” a partir das duas pontas)
2. Pode devolver florestas direcionadas, mesmo quando o grafo subjacente é conexo.
3. Pode haver arcos cruzados em uma floresta direcionada gerada por uma busca em profundidade.

Busca em Grafos Direcionados

1. O tempo de execução de uma busca em grafo direcionado é a mesmo do caso não direcinado (Obs: no caso não direcionado, cada aresta é “inspecionada” a partir das duas pontas)
2. Pode devolver florestas direcionadas, mesmo quando o grafo subjacente é conexo.
3. Pode haver arcos cruzados em uma floresta direcionada gerada por uma busca em profundidade.
4. Se T é a arborescência resultante de uma busca no grafo direcionado G , então $V[T]$ é o conjunto dos vértices de G alcançáveis a partir de v em G .

Busca em Grafos Direcionados - pré-ordem e pós-ordem

Seja F uma floresta direcionada.

Busca em Grafos Direcionados - pré-ordem e pós-ordem

Seja F uma floresta direcionada.

Uma **pré-ordem** de F é uma permutação de $V(F)$ na qual ancestrais são anteriores a seus descendentes.

Busca em Grafos Direcionados - pré-ordem e pós-ordem

Seja F uma floresta direcionada.

Uma **pré-ordem** de F é uma permutação de $V(F)$ na qual ancestrais são anteriores a seus descendentes.

Uma **pós-ordem** de F é uma permutação de $V(F)$ na qual ancestrais são posteriores a seus descendentes.

Teorema 97

F : floresta direcionada resultante de uma busca em profundidade sobre o grafo direcionado G .

Teorema 97

F : floresta direcionada resultante de uma busca em profundidade sobre o grafo direcionado G .

$v.pre$ e $v.pos$: índices de pré-ordem e pós-ordem computados.

Teorema 97

F : floresta direcionada resultante de uma busca em profundidade sobre o grafo direcionado G .

$v.pre$ e $v.pos$: índices de pré-ordem e pós-ordem computados.

Arco (u, v) , com relação a F é:

Teorema 97

F : floresta direcionada resultante de uma busca em profundidade sobre o grafo direcionado G .

$v.pre$ e $v.pos$: índices de pré-ordem e pós-ordem computados.

Arco (u, v) , com relação a F é:

1. arco de F ou arco de avanço se e somente se
 $u.pre < v.pre < v.pos < u.pos$.

Teorema 97

F : floresta direcionada resultante de uma busca em profundidade sobre o grafo direcionado G .

$v.pre$ e $v.pos$: índices de pré-ordem e pós-ordem computados.

Arco (u, v) , com relação a F é:

1. arco de F ou arco de avanço se e somente se
 $u.pre < v.pre < v.pos < u.pos$.
2. arco cruzado se e somente se
 $v.pre < v.pos < u.pre < u.pos$.

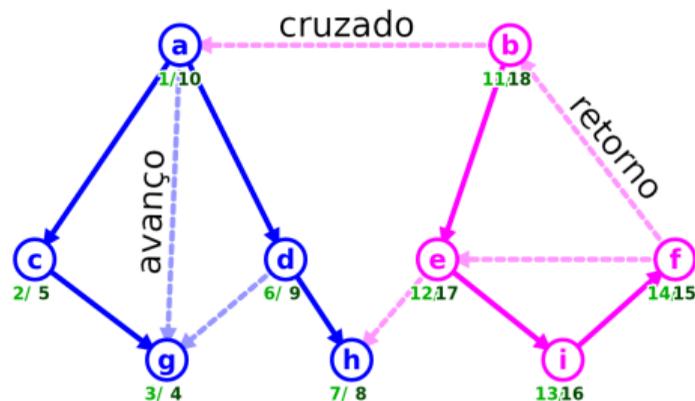
Teorema 97

F : floresta direcionada resultante de uma busca em profundidade sobre o grafo direcionado G .

$v.pre$ e $v.pos$: índices de pré-ordem e pós-ordem computados.

Arco (u, v) , com relação a F é:

1. arco de F ou arco de avanço se e somente se
 $u.pre < v.pre < v.pos < u.pos$.
2. arco cruzado se e somente se
 $v.pre < v.pos < u.pre < u.pos$.
3. arco de retorno se e somente se
 $v.pre < u.pre < u.pos < v.pos$.



A ordem $<$ induzida sobre $V(G)$ dada por:

A ordem $<$ induzida sobre $V(G)$ dada por:

1. $u < v := u.\text{pre} < v.\text{pre}$, para todo $u, v \in V(G)$, é uma pré-ordem de F .

A ordem $<$ induzida sobre $V(G)$ dada por:

1. $u < v := u.pre < v.pre$, para todo $u, v \in V(G)$, é uma pré-ordem de F .

2. $u < v := u.pos < v.pos$, para todo $u, v \in V(G)$, é uma pós-ordem de F .

Prove isso: Exercício 128

Ordenação Topológica

Dado um grafo direcionado G

Ordenação Topológica

Dado um grafo direcionado G

Uma **ordenação topológica** de G é uma permutação (v_1, \dots, v_n) de $V(G)$ que “respeita a direção dos arcos” de G , isto é,

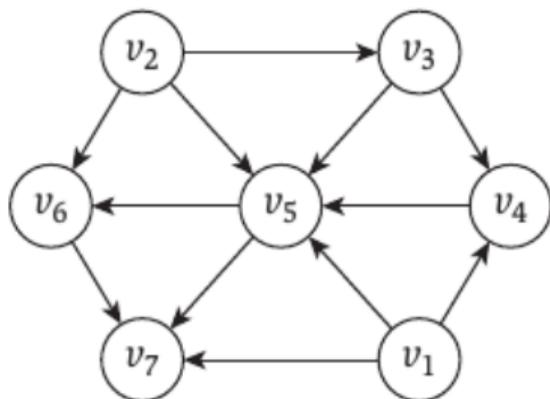
$$i < j, \text{ para todo } (v_i, v_j) \in A(G).$$

Ordenação Topológica

Dado um grafo direcionado G

Uma **ordenação topológica** de G é uma permutação (v_1, \dots, v_n) de $V(G)$ que “respeita a direção dos arcos” de G , isto é,

$$i < j, \text{ para todo } (v_i, v_j) \in A(G).$$



Um grafo direcionado G admite ordenação topológica se e somente se é acíclico.

Um grafo direcionado G admite ordenação topológica se e somente se é acíclico.

Prova: Exercício 130

Um grafo direcionado G é acíclico se e somente se qualquer floresta direcionada resultante de uma busca em profundidade sobre G não tem arcos de retorno.

Um grafo direcionado G é acíclico se e somente se qualquer floresta direcionada resultante de uma busca em profundidade sobre G não tem arcos de retorno.

Prova:

Exercício 129

Ordenação Topológica - Teorema 101

O reverso da pós-ordem de uma floresta direcionada resultante de uma busca em profundidade em um grafo direcionado acíclico G , é uma ordenação topológica de G .

Ordenação Topológica - Teorema 101

O reverso da pós-ordem de uma floresta direcionada resultante de uma busca em profundidade em um grafo direcionado acíclico G , é uma ordenação topológica de G .

Demonstração.

1. Seja (v_1, \dots, v_n) a permutação de $V(G)$ em ordem decrescente de pós-ordem,

Ordenação Topológica - Teorema 101

O reverso da pós-ordem de uma floresta direcionada resultante de uma busca em profundidade em um grafo direcionado acíclico G , é uma ordenação topológica de G .

Demonstração.

1. Seja (v_1, \dots, v_n) a permutação de $V(G)$ em ordem decrescente de pós-ordem, isto é,

$i < j$ se e somente se $v_i.\text{pos} > v_j.\text{pos}$, para todo $1 \leq i < j \leq n \in A(G)$,

Ordenação Topológica - Teorema 101

O reverso da pós-ordem de uma floresta direcionada resultante de uma busca em profundidade em um grafo direcionado acíclico G , é uma ordenação topológica de G .

Demonstração.

1. Seja (v_1, \dots, v_n) a permutação de $V(G)$ em ordem decrescente de pós-ordem, isto é,

$$i < j \text{ se e somente se } v_i.\text{pos} > v_j.\text{pos}, \text{ para todo } 1 \leq i < j \leq n \in A(G),$$

2. Esta permutação é uma ordenação topológica de G .

Ordenação Topológica - Teorema 101

O reverso da pós-ordem de uma floresta direcionada resultante de uma busca em profundidade em um grafo direcionado acíclico G , é uma ordenação topológica de G .

Demonstração.

1. Seja (v_1, \dots, v_n) a permutação de $V(G)$ em ordem decrescente de pós-ordem, isto é,

$$i < j \text{ se e somente se } v_i.\text{pos} > v_j.\text{pos}, \text{ para todo } 1 \leq i < j \leq n \in A(G),$$

2. Esta permutação é uma ordenação topológica de G . Basta provar que

$$i < j \text{ para todo } (v_i, v_j) \in A(G),$$

Ordenação Topológica - Teorema 101

O reverso da pós-ordem de uma floresta direcionada resultante de uma busca em profundidade em um grafo direcionado acíclico G , é uma ordenação topológica de G .

Demonstração.

1. Seja (v_1, \dots, v_n) a permutação de $V(G)$ em ordem decrescente de pós-ordem, isto é,

$$i < j \text{ se e somente se } v_i.\text{pos} > v_j.\text{pos}, \text{ para todo } 1 \leq i < j \leq n \in A(G),$$

2. Esta permutação é uma ordenação topológica de G . Basta provar que

$$i < j \text{ para todo } (v_i, v_j) \in A(G),$$

ou seja, que

$$v_i.\text{pos} > v_j.\text{pos} \text{ para todo } (v_i, v_j) \in A(G),$$

Ordenação Topológica - Teorema 101

O reverso da pós-ordem de uma floresta direcionada resultante de uma busca em profundidade em um grafo direcionado acíclico G , é uma ordenação topológica de G .

Demonstração.

1. Seja (v_1, \dots, v_n) a permutação de $V(G)$ em ordem decrescente de pós-ordem, isto é,

$$i < j \text{ se e somente se } v_i.\text{pos} > v_j.\text{pos}, \text{ para todo } 1 \leq i < j \leq n \in A(G),$$

2. Esta permutação é uma ordenação topológica de G . Basta provar que

$$i < j \text{ para todo } (v_i, v_j) \in A(G),$$

ou seja, que

$$v_i.\text{pos} > v_j.\text{pos} \text{ para todo } (v_i, v_j) \in A(G),$$

ou seja, que

$$v.\text{pos} < u.\text{pos} \text{ para todo } (u, v) \in A(G).$$

Ordenação Topológica - Teorema ??

Demonstração (continuação).

3. se tivéssemos $u.pos < v.pos$ então (u, v) seria arco de retorno em G .

Ordenação Topológica - Teorema ??

Demonstração (continuação).

3. se tivéssemos $u.pos < v.pos$ então (u, v) seria arco de retorno em G . (T. 97)

Ordenação Topológica - Teorema ??

Demonstração (continuação).

3. se tivéssemos $u.pos < v.pos$ então (u, v) seria arco de retorno em G . (T. 97)
4. Como G é acíclico, não tem arco de retorno

Ordenação Topológica - Teorema ??

Demonstração (continuação).

3. se tivéssemos $u.pos < v.pos$ então (u, v) seria arco de retorno em G . (T. 97)
4. Como G é acíclico, não tem arco de retorno (T. 99)

Ordenação Topológica - Teorema ??

Demonstração (continuação).

3. se tivéssemos $u.pos < v.pos$ então (u, v) seria arco de retorno em G . (T. 97)
4. Como G é acíclico, não tem arco de retorno (T. 99)
5. e portanto, $v.pos < u.pos$ para todo $(u, v) \in A(G)$, e (v_1, \dots, v_n) é ordenação topológica de G .



Algoritmo de Ordenação Topológica

Ordena(G)

Para **cada** $v \in V(G)$

$v.\text{estado} \leftarrow 0$

$G.l \leftarrow$ lista vazia

Para **cada** $v \in V(G)$

 Se $v.\text{estado} = 0$

 Ordena(G, v)

Ordena(G, r)

$r.\text{estado} \leftarrow 1$

Para **cada** $v \in \Gamma_G^+(r)$

 Se $v.\text{estado} = 0$

 Ordena(G, v)

acrescente r ao início de $G.l$

$r.\text{estado} \leftarrow 2$

Algoritmo de Ordenação Topológica - Execução

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

$r.\text{estado} \leftarrow 1$

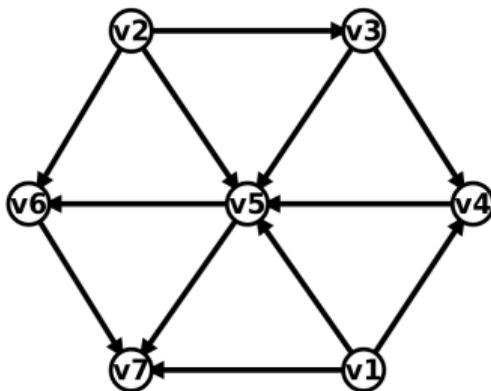
Para cada $v \in \Gamma_G^+(r)$

 Se $v.\text{estado} = 0$

$\text{ordena}(G, v)$

acrescente r ao início de $G.I$

$r.\text{estado} \leftarrow 2$



Algoritmo de Ordenação Topológica - Execução

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

$r.\text{estado} \leftarrow 1$

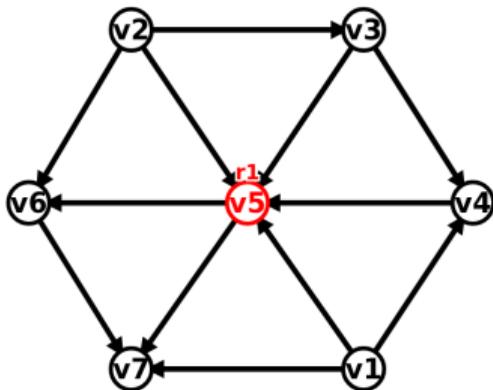
Para cada $v \in \Gamma_G^+(r)$

 Se $v.\text{estado} = 0$

$\text{Ordena}(G, v)$

 acrescente r ao início de $G.I$

$r.\text{estado} \leftarrow 2$



Algoritmo de Ordenação Topológica - Execução

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

$r.\text{estado} \leftarrow 1$

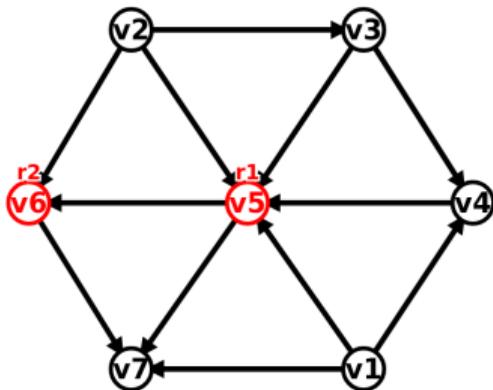
Para cada $v \in \Gamma_G^+(r)$

 Se $v.\text{estado} = 0$

$\text{ordena}(G, v)$

 acrescente r ao início de $G.I$

$r.\text{estado} \leftarrow 2$



Algoritmo de Ordenação Topológica - Execução

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

$r.\text{estado} \leftarrow 1$

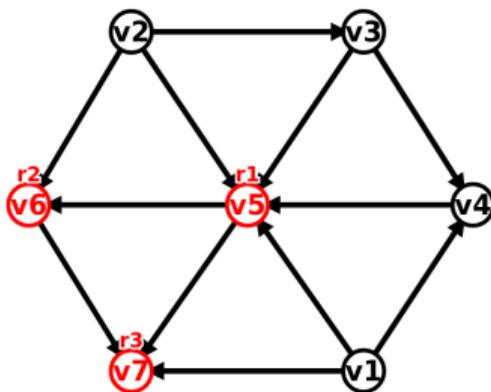
Para cada $v \in \Gamma_G^+(r)$

 Se $v.\text{estado} = 0$

$\text{ordena}(G, v)$

acrescente r ao início de $G.l$

$r.\text{estado} \leftarrow 2$



Algoritmo de Ordenação Topológica - Execução

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

$r.\text{estado} \leftarrow 1$

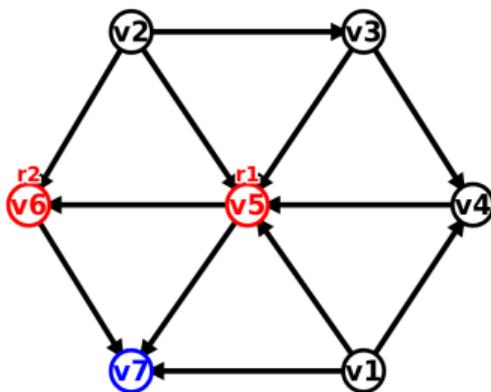
Para cada $v \in \Gamma_G^+(r)$

 Se $v.\text{estado} = 0$

$\text{ordena}(G, v)$

acrescente r ao início de $G.l$

$r.\text{estado} \leftarrow 2$



$G.l = [v7]$

Algoritmo de Ordenação Topológica - Execução

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

$r.\text{estado} \leftarrow 1$

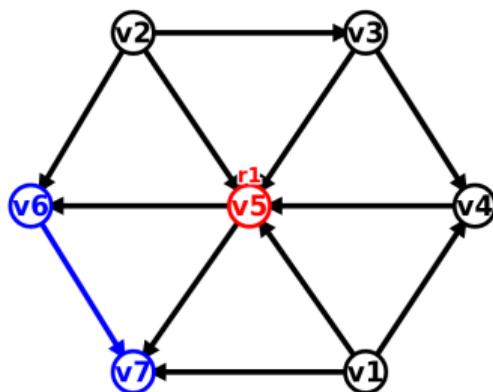
Para cada $v \in \Gamma_G^+(r)$

 Se $v.\text{estado} = 0$

$\text{Ordena}(G, v)$

acrescente r ao início de $G.l$

$r.\text{estado} \leftarrow 2$



$G.l = [v6, v7]$

Algoritmo de Ordenação Topológica - Execução

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

$r.\text{estado} \leftarrow 1$

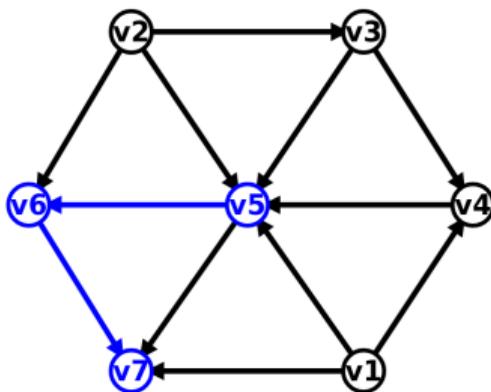
Para cada $v \in \Gamma_G^+(r)$

 Se $v.\text{estado} = 0$

$\text{ordena}(G, v)$

acrescente r ao início de $G.l$

$r.\text{estado} \leftarrow 2$



$G.l = [v5, v6, v7]$

Algoritmo de Ordenação Topológica - Execução

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

$r.\text{estado} \leftarrow 1$

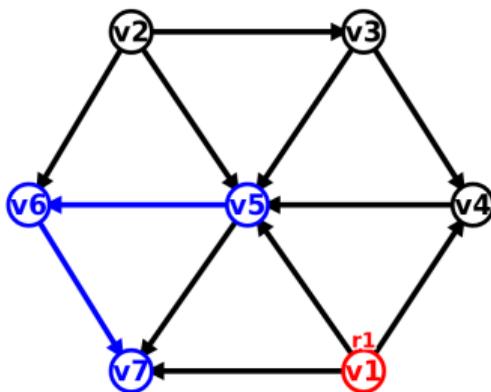
Para cada $v \in \Gamma_G^+(r)$

 Se $v.\text{estado} = 0$

$\text{ordena}(G, v)$

 acrescente r ao início de $G.l$

$r.\text{estado} \leftarrow 2$



$G.l = [v5, v6, v7]$

Algoritmo de Ordenação Topológica - Execução

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

$r.\text{estado} \leftarrow 1$

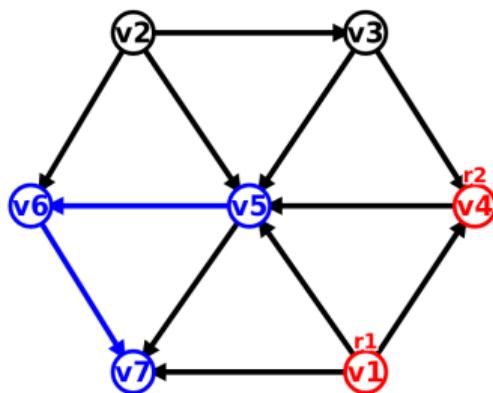
Para cada $v \in \Gamma_G^+(r)$

 Se $v.\text{estado} = 0$

$\text{Ordena}(G, v)$

acrescente r ao início de $G.l$

$r.\text{estado} \leftarrow 2$



$G.l = [v5, v6, v7]$

Algoritmo de Ordenação Topológica - Execução

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

$r.\text{estado} \leftarrow 1$

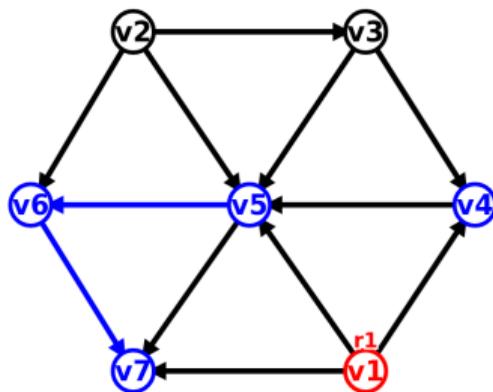
Para cada $v \in \Gamma_G^+(r)$

 Se $v.\text{estado} = 0$

$\text{Ordena}(G, v)$

 acrescente r ao início de $G.l$

$r.\text{estado} \leftarrow 2$



$G.l = [v4, v5, v6, v7]$

Algoritmo de Ordenação Topológica - Execução

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

$r.\text{estado} \leftarrow 1$

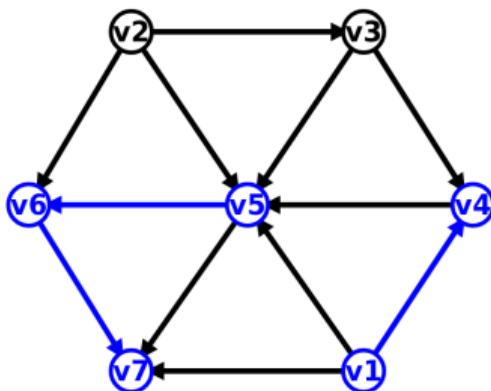
Para cada $v \in \Gamma_G^+(r)$

 Se $v.\text{estado} = 0$

$\text{Ordena}(G, v)$

acrescente r ao início de $G.l$

$r.\text{estado} \leftarrow 2$



$G.l = [v1, v4, v5, v6, v7]$

Algoritmo de Ordenação Topológica - Execução

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

$r.\text{estado} \leftarrow 1$

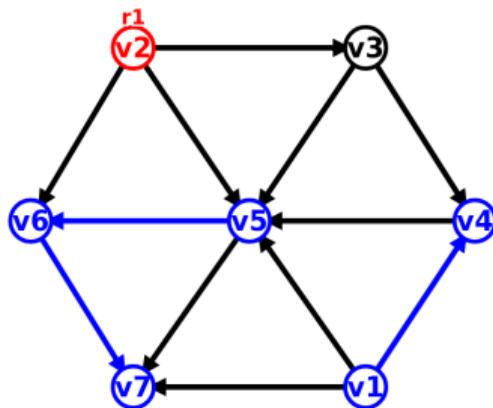
Para cada $v \in \Gamma_G^+(r)$

 Se $v.\text{estado} = 0$

$\text{Ordena}(G, v)$

 acrescente r ao início de $G.l$

$r.\text{estado} \leftarrow 2$



$G.l = [v1, v4, v5, v6, v7]$

Algoritmo de Ordenação Topológica - Execução

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

$r.\text{estado} \leftarrow 1$

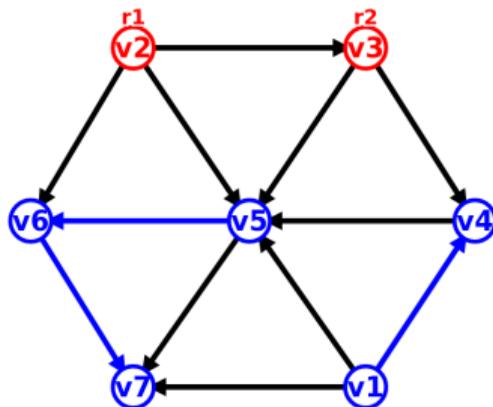
Para **cada** $v \in \Gamma_G^+(r)$

 Se $v.\text{estado} = 0$

$\text{Ordena}(G, v)$

 acrescente r ao início de $G.l$

$r.\text{estado} \leftarrow 2$



$G.l = [v1, v4, v5, v6, v7]$

Algoritmo de Ordenação Topológica - Execução

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

$r.\text{estado} \leftarrow 1$

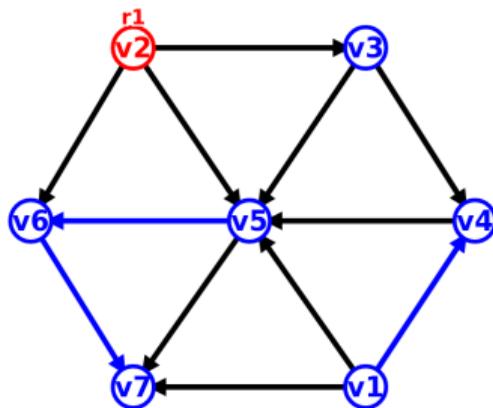
Para cada $v \in \Gamma_G^+(r)$

 Se $v.\text{estado} = 0$

$\text{Ordena}(G, v)$

 acrescente r ao início de $G.l$

$r.\text{estado} \leftarrow 2$



$G.l = [v3, v1, v4, v5, v6, v7]$

Algoritmo de Ordenação Topológica - Execução

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

$r.\text{estado} \leftarrow 1$

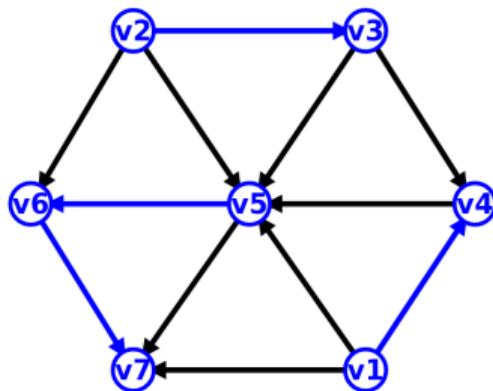
Para **cada** $v \in \Gamma_G^+(r)$

 Se $v.\text{estado} = 0$

$\text{Ordena}(G, v)$

 acrescente r ao início de $G.l$

$r.\text{estado} \leftarrow 2$



$G.l = [v2, v3, v1, v4, v5, v6, v7]$