Busca em Profundidade em Grafos Direcionados

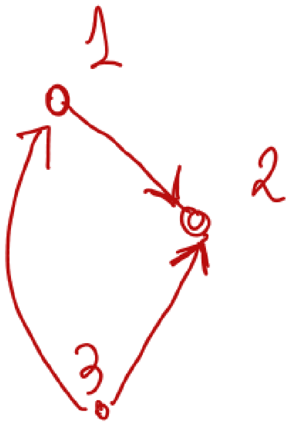
Busca em Largura

```
BuscaLargura(Grafo G, vértice s)
  s.visitado = 1;
  Cria fila vazia F;
  ENFILEIRA(F,s);
  Enquanto F.tamanho >0 faça
    u = DESENFILEIRA(F);
    Para todo vértice v ∈ N+(u) faça
      Se v.visitado==0 então
        v.visitado = 1;
        v.predecessor = u;
        ENFILEIRA(F,s);
```

Busca em Profundidade

Dado um grafo **G**
Desmarcar os vértices
Definir uma pilha **Q**
Definir uma raiz $s \in V$
P(G,s)

P(Grafo G, vértice v)
  marcar v
  colocar v na pilha Q
  para $w \in N^+(v)$
    visitar (v, w)
    se w não é marcado, então P(w)
    retirar v de Q



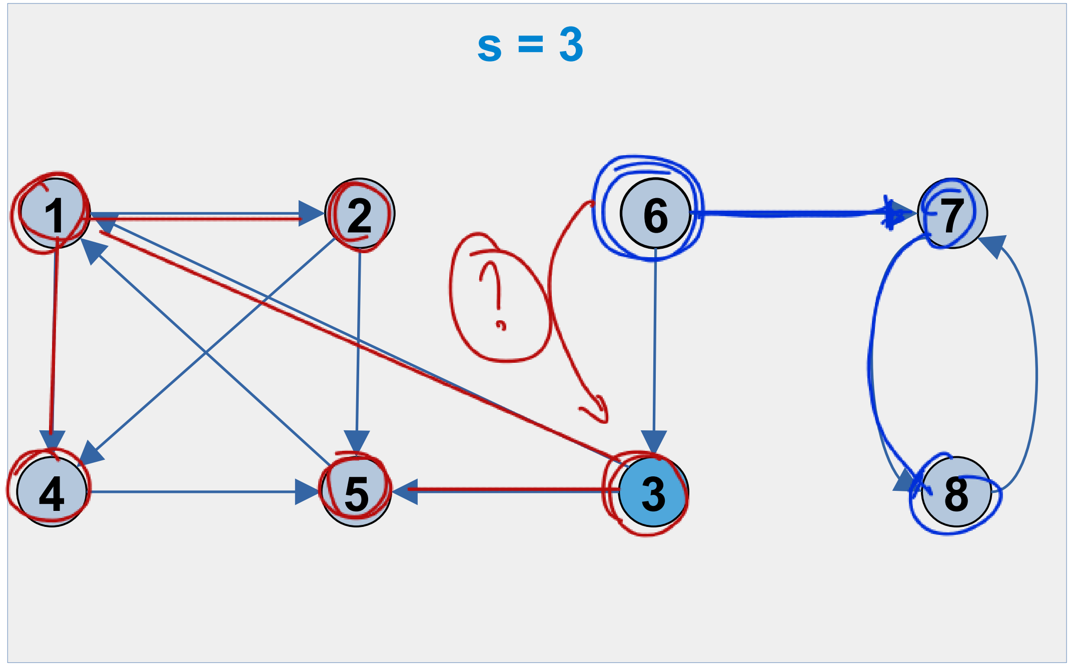
1, 2

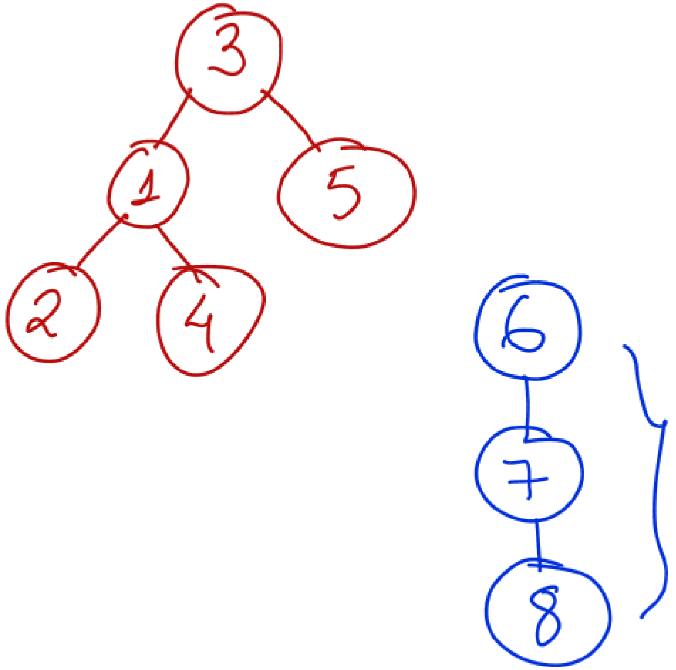
$2 \in N^+(1)$

$N^+(2) = \emptyset$

$N^+(3) = \{2, 1\}$

Percurso em Largura

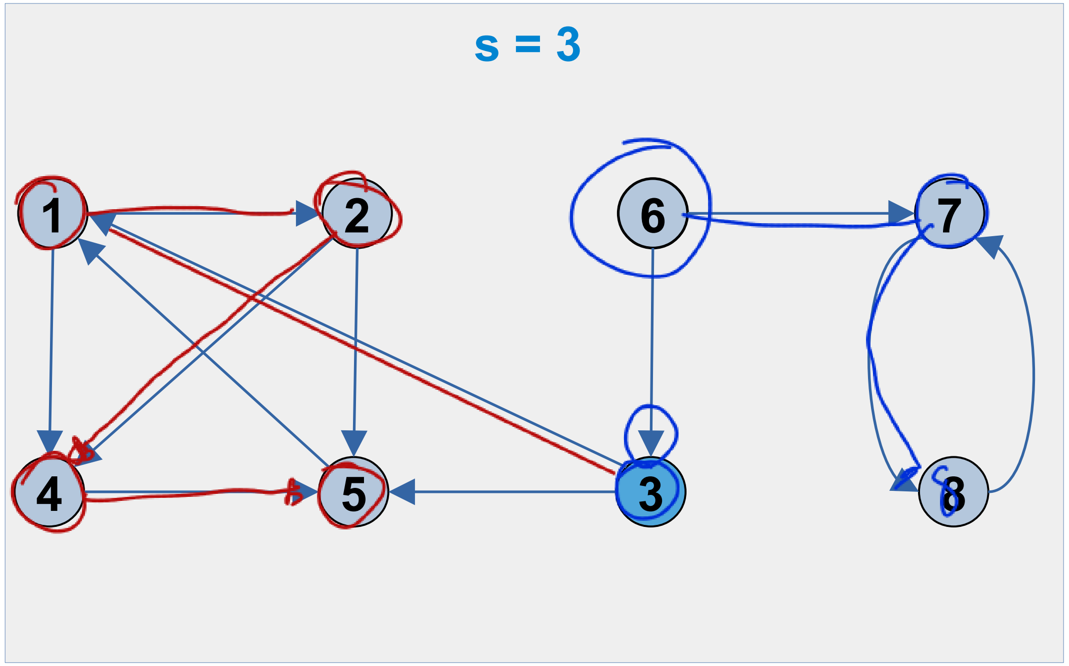


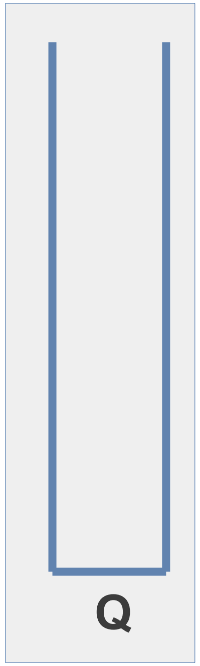


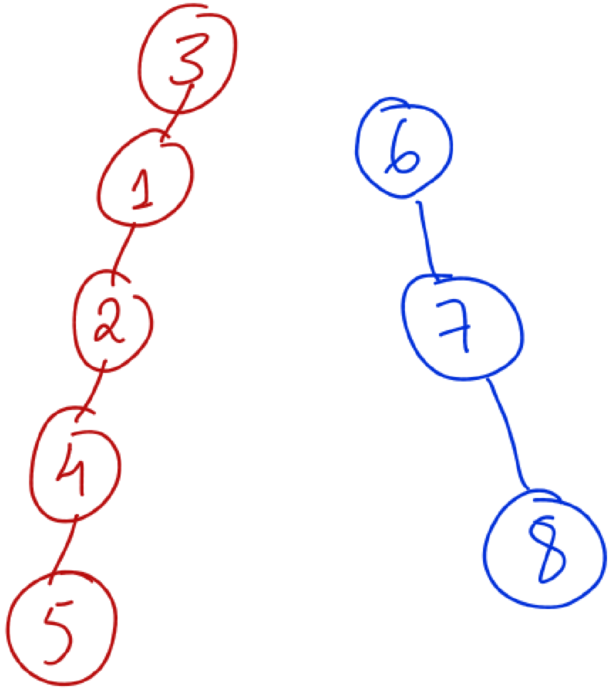
Vértices	1	2	3	4	5	6	7	8
Visitado	1	1	1	1	1	1	1	1
Predecessor	3	1	–	1	3	–	6	7

F

Percurso em Profundidade

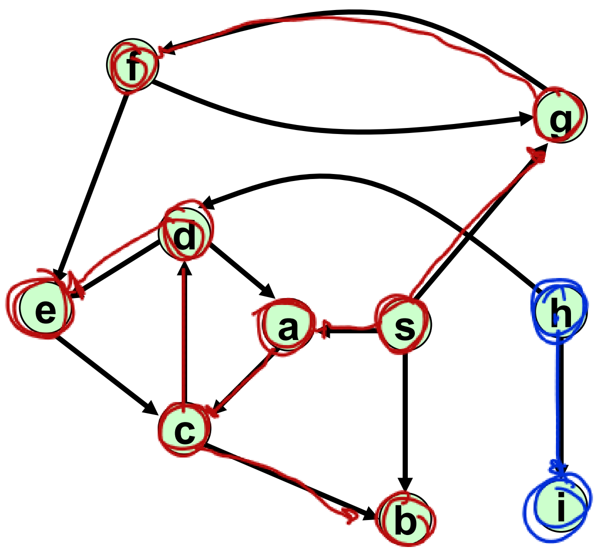






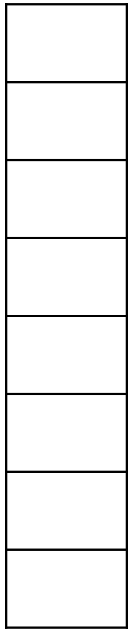
Vértices	1	2	3	4	5	6	7	8
Visitado	1	1	1	1	1	1	1	1
Predecessor	3	1	–	2	4	–	6	7

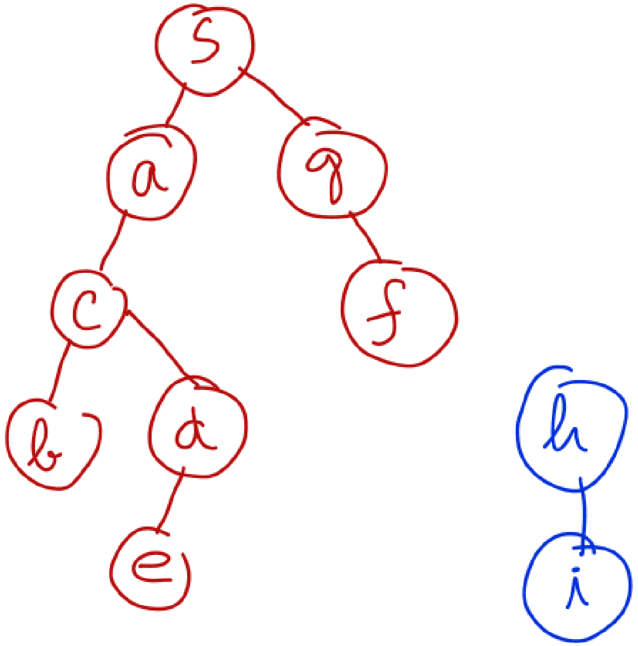
Busca Profundidade - Digrafos



~~A(s) = {a, g, b}~~
~~A(a) = {c}~~
~~A(b) = { }~~
~~A(c) = {b, d}~~
~~A(d) = {a, e}~~
~~A(e) = {c}~~
~~A(f) = {e, g}~~
~~A(g) = {f}~~
~~A(h) = {i, d}~~
~~A(i) = { }~~

Q





P(Grafo G, vértice v)

marcar v

colocar v na pilha Q

para $w \in N^{+}(v)$

visitar (v, w)

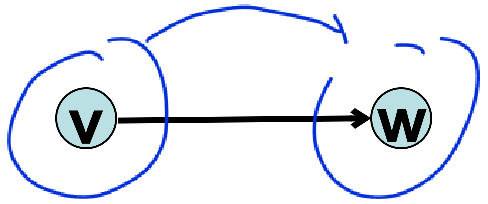
se w não é marcado, **então** P(w)

retirar v de Q

Vértices	s	a	b	c	d	e	f	g	h	i
Visitado	1	1	1	1	1	1	1	1	1	1
Predecessor	–	s	c	a	c	d	g	s	–	h
P(E)	1	2	4	3	5	6	8	7	9	10
P(S)	8	5	1	4	3	2	6	7	10	9

Busca Profundidade - Digrafos

Considere a visita a aresta (*v*,*w*)



Caso 1. *v* é alcançado antes de *w* na busca

1.1 – Se *w* estava desmarcado antes da visita, então (*v*,*w*) é **aresta da árvore** (floresta) de profundidade

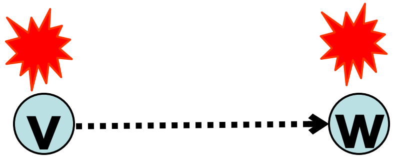
1.2 - Se *w* estava marcado antes da visita, então (*v*,*w*) é **aresta de avanço**

Busca Profundidade - Digrafos

Considere a visita a aresta (v,w)

Caso 2. w é alcançado antes de v na busca

2.1 – Se $w \in Q$ no momento da visita, então (v,w) denomina-se **aresta de retorno**

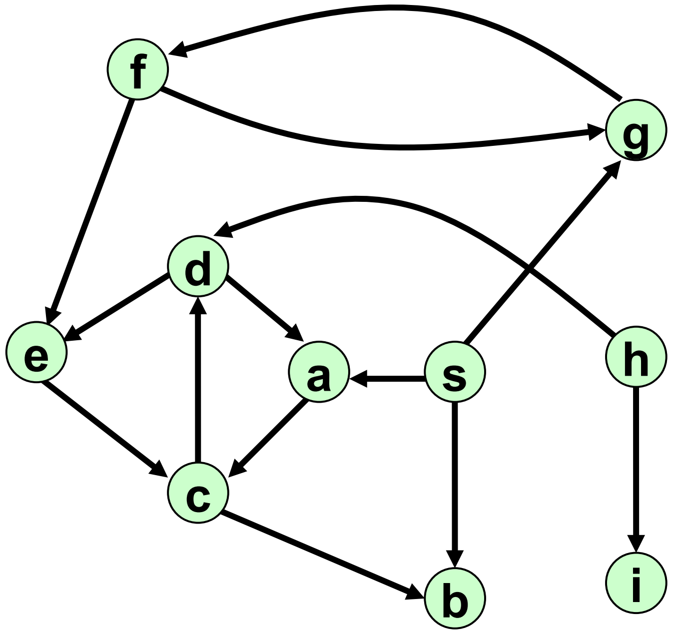


2.2 - Se $w \notin Q$ no momento da visita, então (v,w) é denominada **aresta de cruzamento**

Busca Profundidade - Digrafos

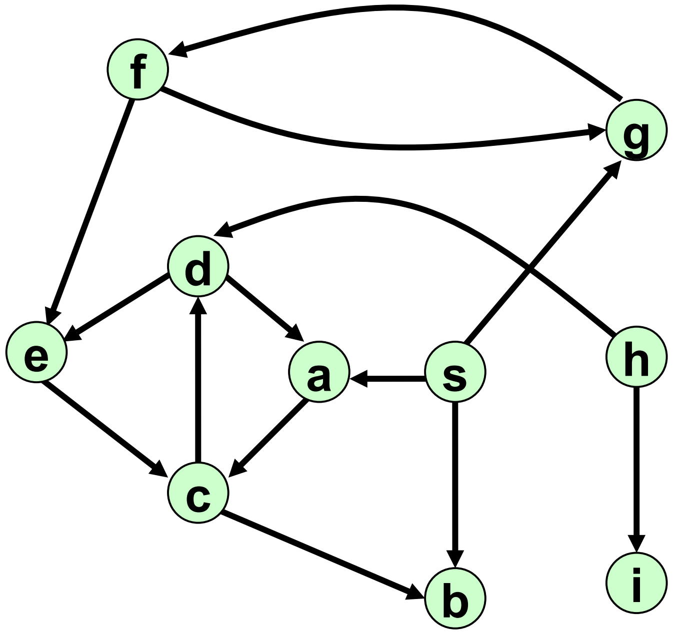
Dado um grafo **G**
Desmarcar os vértices
Definir uma pilha **Q**
Definir uma raiz $s \in V$
P(G,s)

P(Grafo G, vértice v)
  marcar v
  colocar v na pilha Q
  para $w \in N^{+}(v)$
    visitar (v, w)
    se w não é marcado, **então** P(w)
    retirar v de Q



Digrafo

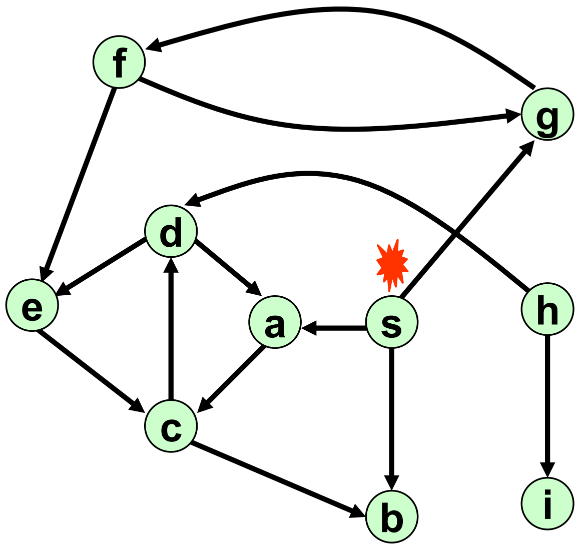
Busca Profundidade - Digrafos



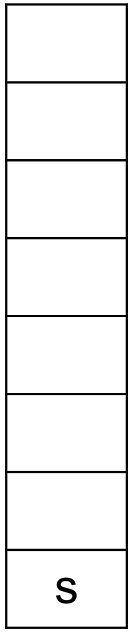
Digrafo

A(s) = {a, g, b}
A(a) = {c}
A(b) = { }
A(c) = {b, d}
A(d) = {a, e}
A(e) = {c}
A(f) = {e, g}
A(g) = {f}
A(h) = {i, d}
A(i) = { }

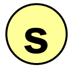
Busca Profundidade - Digrafos



A(s) = {a, g, b}
A(a) = {c}
A(b) = { }
A(c) = {b, d}
A(d) = {a, e}
A(e) = {c}
A(f) = {e, g}
A(g) = {f}
A(h) = {i, d}
A(i) = { }

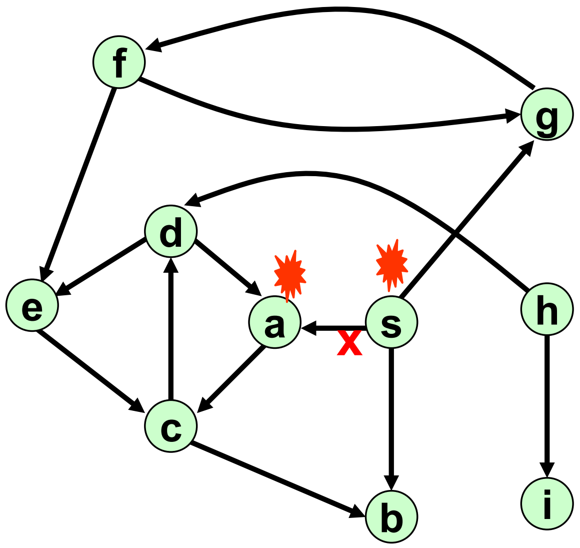


Q

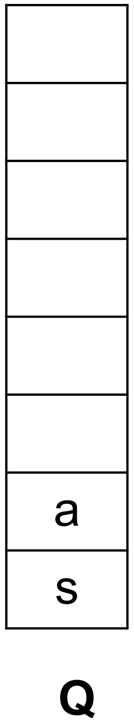


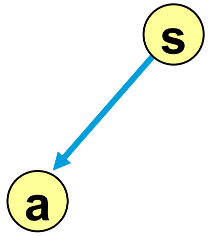
P(Grafo ***G***, vértice ***v***)
  marcar ***v***
  colocar ***v*** na pilha ***Q***
  para ***w*** ∈ ***N⁺ (v)***
    visitar ***(v, w)***
    se ***w*** não é marcado, **então** P(***w***)
    retirar ***v*** de ***Q***

Busca Profundidade - Digrafos



A(s) = {a, g, b}
A(a) = {c}
A(b) = { }
A(c) = {b, d}
A(d) = {a, e}
A(e) = {c}
A(f) = {e, g}
A(g) = {f}
A(h) = {i, d}
A(i) = { }

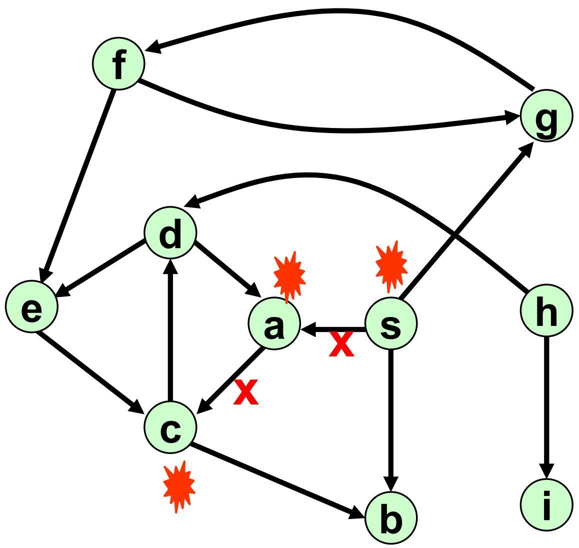




P(Grafo ***G***, vértice ***v***)
  marcar ***v***
  colocar ***v*** na pilha ***Q***
  para ***w*** ∈ ***N⁺ (v)***
    visitar ***(v, w)***
    se ***w*** não é marcado, **então** P(***w***)
    retirar ***v*** de ***Q***

Considere a aresta (*v*,*w*). Se *w* estava desmarcado antes da visita, então (*v*,*w*) é **aresta da árvore** (floresta) de profundidade

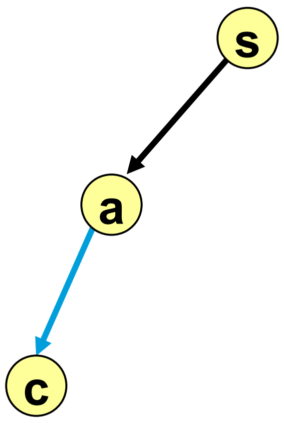
Busca Profundidade - Digrafos



A(s) = {a, g, b}
A(a) = {c}
A(b) = { }
A(c) = {b, d}
A(d) = {a, e}
A(e) = {c}
A(f) = {e, g}
A(g) = {f}
A(h) = {i, d}
A(i) = { }

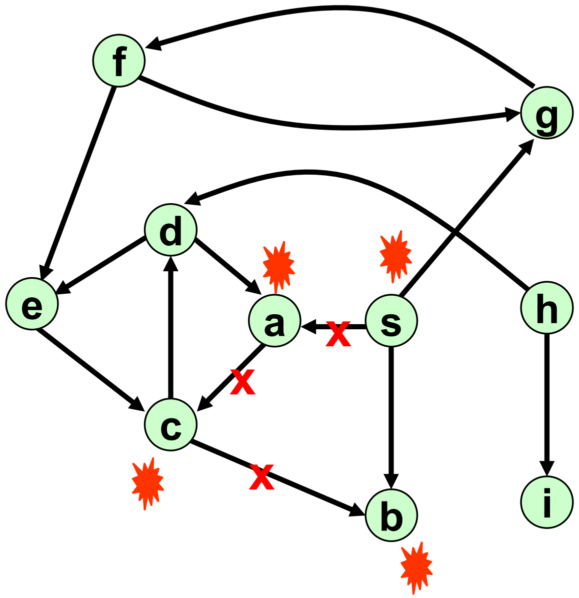
c
a
s

Q



Considere a aresta (*v*,*w*). Se *w* estava desmarcado antes da visita, então (*v*,*w*) é **aresta da árvore** (floresta) de profundidade

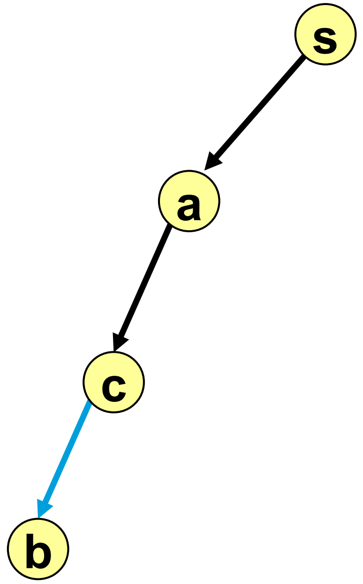
Busca Profundidade - Digrafos



A(s) = {a, g, b}
A(a) = {c}
~~A(b) = { }~~
A(c) = {b, d}
A(d) = {a, e}
A(e) = {c}
A(f) = {e, g}
A(g) = {f}
A(h) = {i, d}
A(i) = { }

b
c
a
s

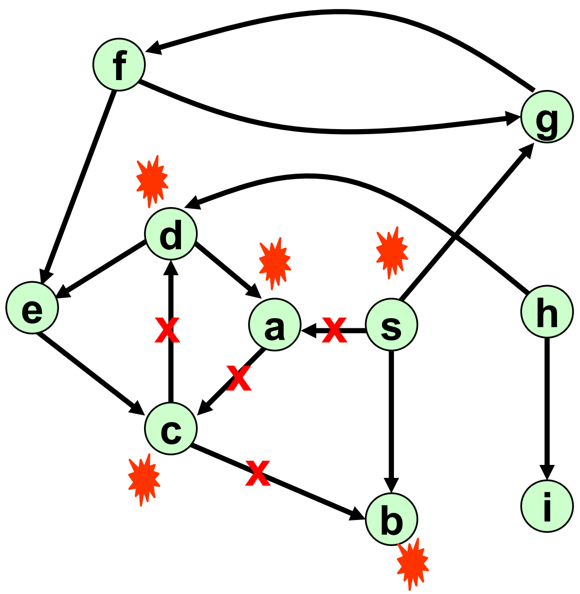
Q



b é removido da pilha

Considere a aresta (*v*,*w*). Se *w* estava desmarcado antes da visita, então (*v*,*w*) é **aresta da árvore** (floresta) de profundidade

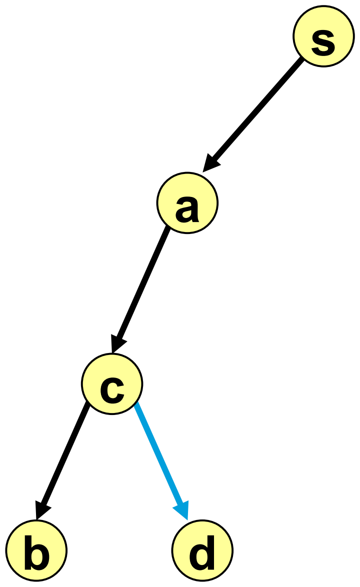
Busca Profundidade - Digrafos



A(s) = {a, g, b}
A(a) = {c}
~~A(b) = { }~~
A(c) = {b, d}
A(d) = {a, e}
A(e) = {c}
A(f) = {e, g}
A(g) = {f}
A(h) = {i, d}
A(i) = { }

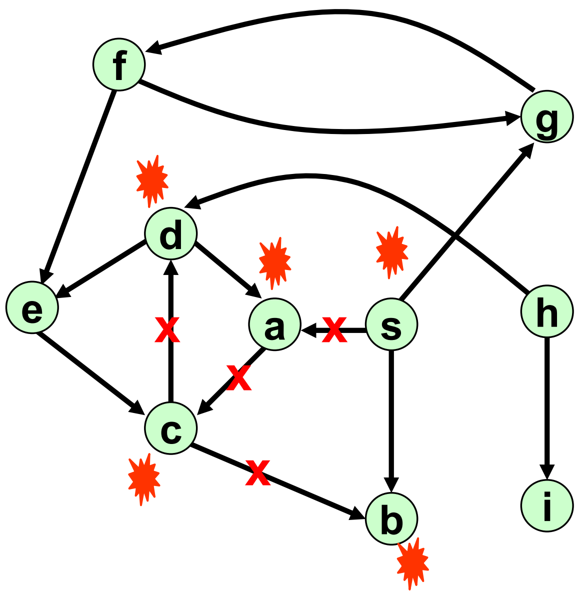
d
c
a
s

Q

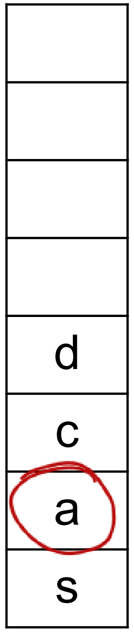


Considere a aresta (*v*,*w*). Se *w* estava desmarcado antes da visita, então (*v*,*w*) é **aresta da árvore** (floresta) de profundidade

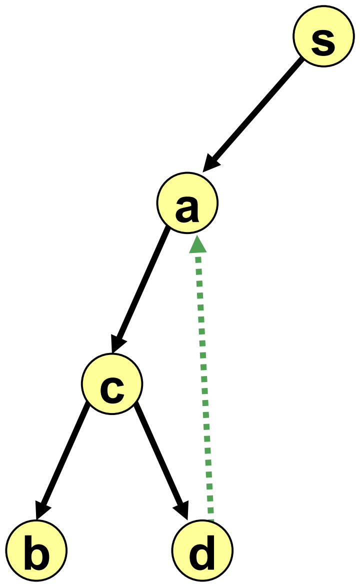
Busca Profundidade - Digrafos



A(s) = {a, g, b}
A(a) = {c}
~~A(b) = { }~~
A(c) = {b, d}
A(d) = {a, e}
A(e) = {c}
A(f) = {e, g}
A(g) = {f}
A(h) = {i, d}
A(i) = { }

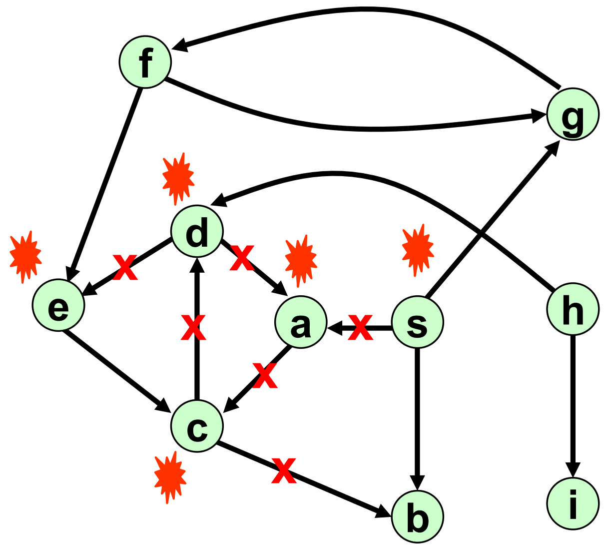


Q

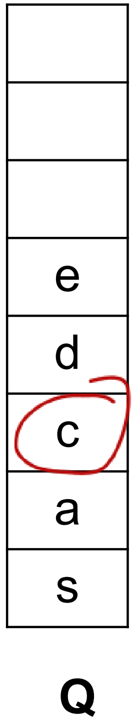


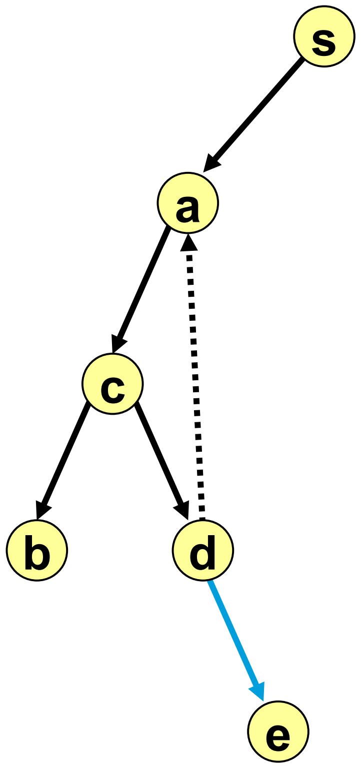
Considere a aresta (*v*,*w*). Se *w* $\in$ Q no momento da visita, então (*v*,*w*) denomina-se **aresta de retorno**

Busca Profundidade - Digrafos



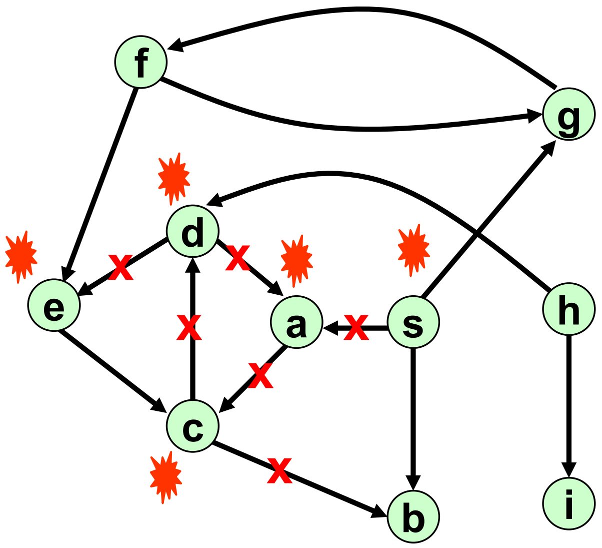
A(s) = {a, g, b}
A(a) = {c}
~~A(b) = { }~~
A(c) = {b, d}
A(d) = {a, e}
A(e) = {c}
A(f) = {e, g}
A(g) = {f}
A(h) = {i, d}
A(i) = { }



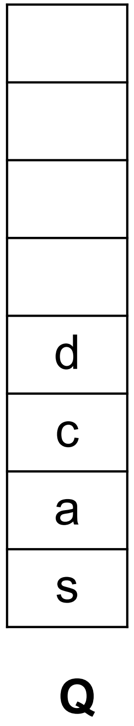


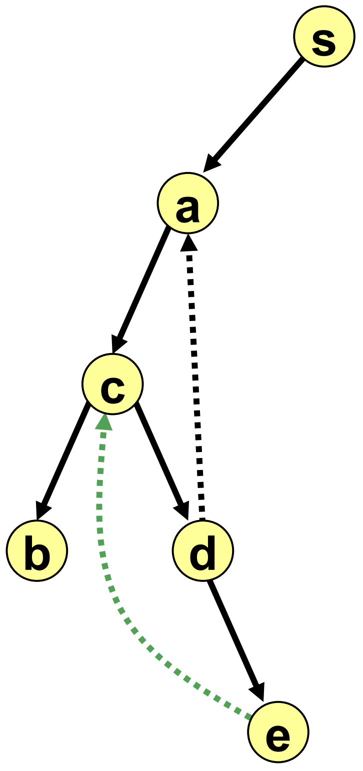
Considere a aresta (*v*,*w*). Se *w* estava desmarcado antes da visita, então (*v*,*w*) é **aresta da árvore** (floresta) de profundidade

Busca Profundidade - Digrafos



A(s) = {a, g, b}
A(a) = {c}
~~A(b) = { }~~
A(c) = {b, d}
A(d) = {a, e}
~~A(e) = {c}~~
A(f) = {e, g}
A(g) = {f}
A(h) = {i, d}
A(i) = { }

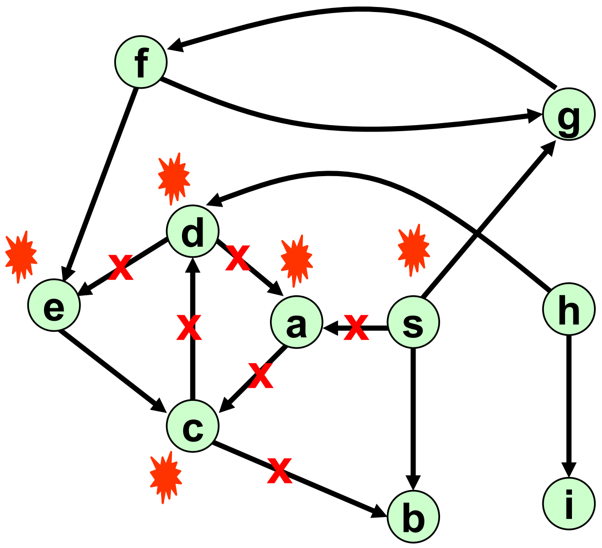




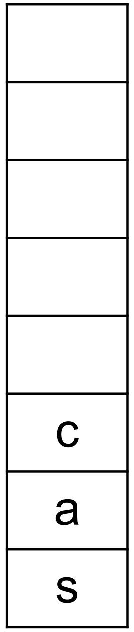
e é removido da pilha

Considere a aresta (v,w). Se $w \in Q$ no momento da visita, então (v,w) denomina-se **aresta de retorno**

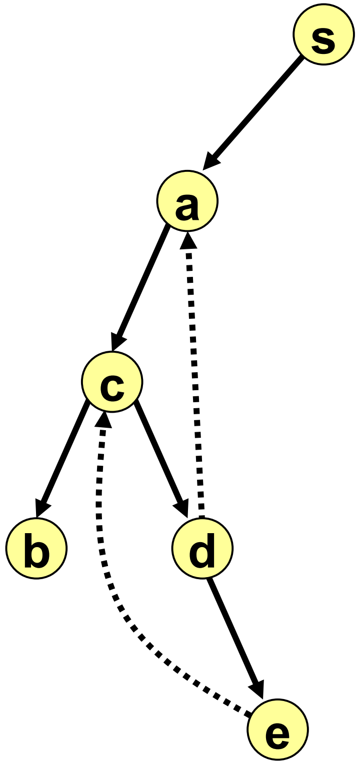
Busca Profundidade - Digrafos



A(s) = {a, g, b}
A(a) = {c}
~~A(b) = { }~~
A(c) = {b, d}
~~A(d) = {a, e}~~
~~A(e) = {c}~~
A(f) = {e, g}
A(g) = {f}
A(h) = {i, d}
A(i) = { }

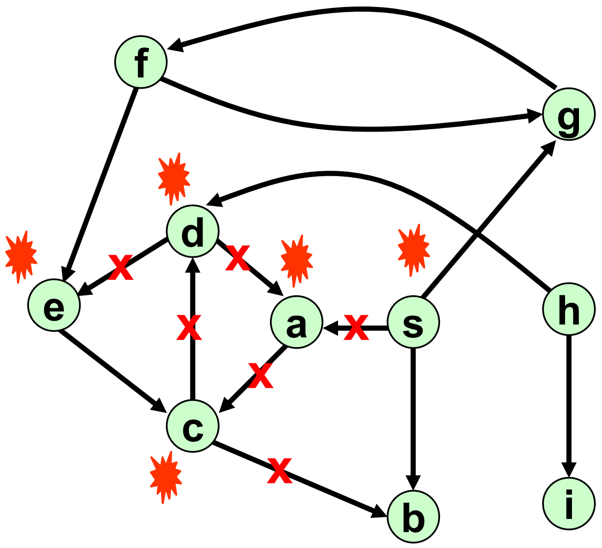


Q



d é removido da pilha

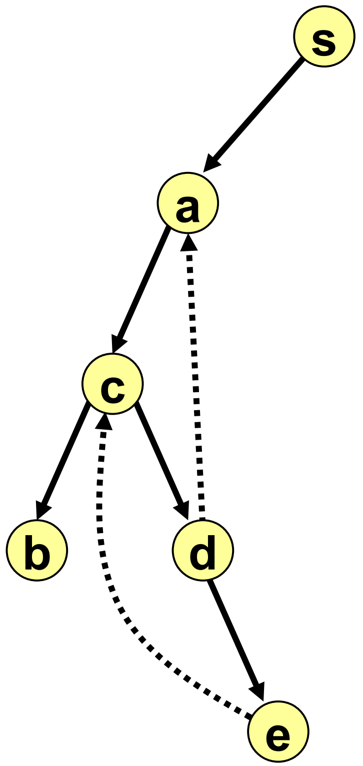
Busca Profundidade - Digrafos



A(s) = {a, g, b}
A(a) = {c}
~~A(b) = { }~~
~~A(c) = {b, d}~~
~~A(d) = {a, e}~~
~~A(e) = {c}~~
A(f) = {e, g}
A(g) = {f}
A(h) = {i, d}
A(i) = { }

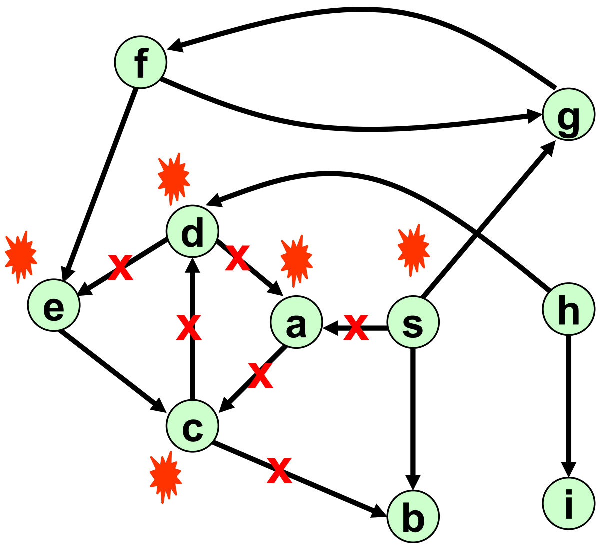
a
s

Q

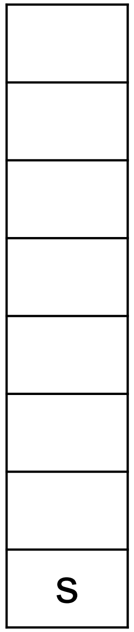


c é removido da pilha

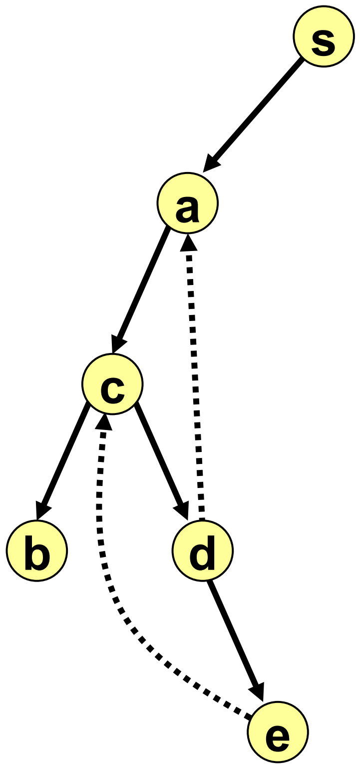
Busca Profundidade - Digrafos



A(s) = {a, g, b}
~~A(a) = {c}~~
~~A(b) = { }~~
~~A(c) = {b, d}~~
~~A(d) = {a, e}~~
~~A(e) = {c}~~
A(f) = {e, g}
A(g) = {f}
A(h) = {i, d}
A(i) = { }

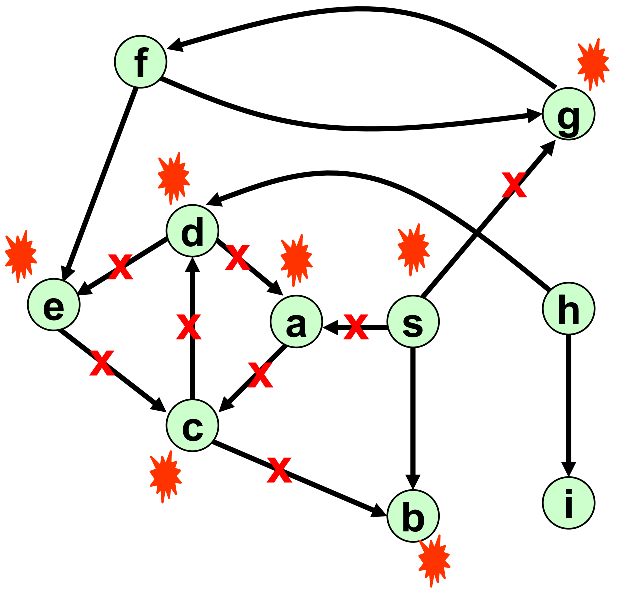


Q

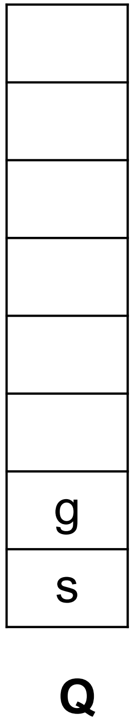


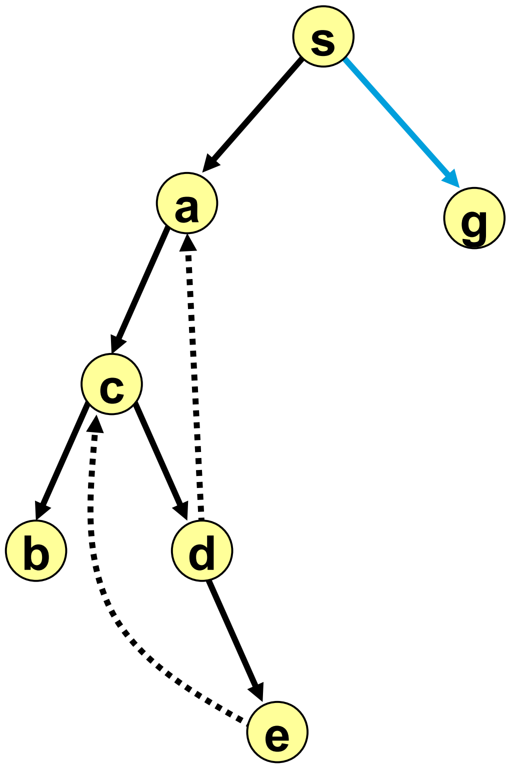
a é removido da pilha

Busca Profundidade - Digrafos



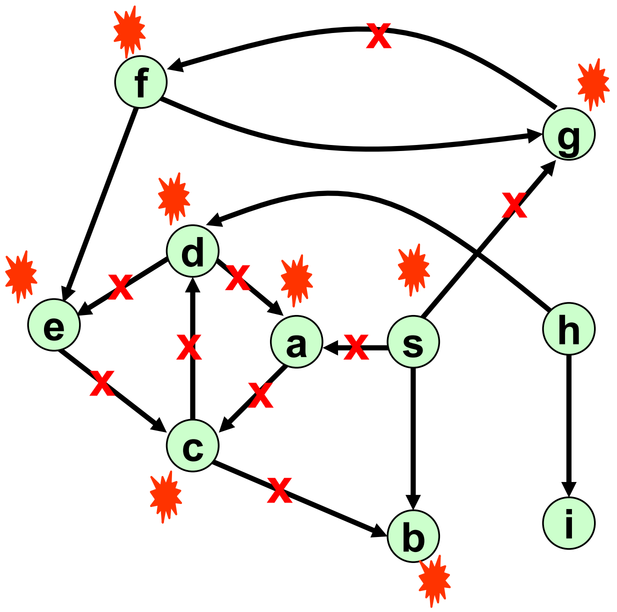
A(s) = {a, g, b}
~~A(a) = {c}~~
~~A(b) = { }~~
~~A(c) = {b, d}~~
~~A(d) = {a, e}~~
~~A(e) = {c}~~
A(f) = {e, g}
A(g) = {f}
A(h) = {i, d}
A(i) = { }



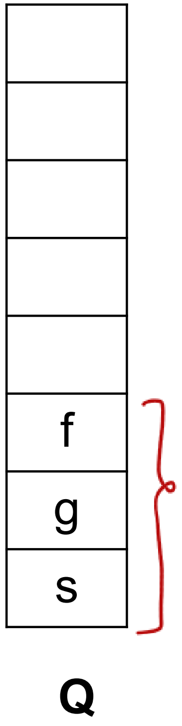


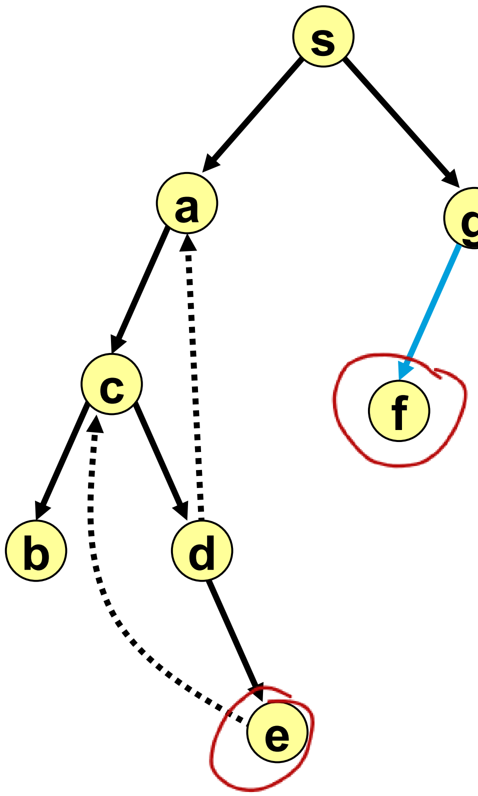
Considere a aresta (*v*,*w*). Se *w* estava desmarcado antes da visita, então (*v*,*w*) é **aresta da árvore** (floresta) de profundidade

Busca Profundidade - Digrafos



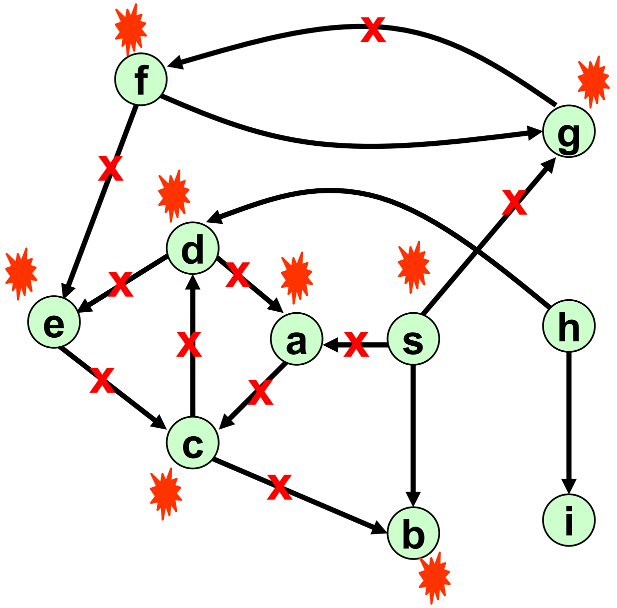
A(s) = {a, g, b}
~~A(a) = {c}~~
~~A(b) = { }~~
~~A(c) = {b, d}~~
~~A(d) = {a, e}~~
~~A(e) = {c}~~
A(f) = {e, g}
A(g) = {f}
A(h) = {i, d}
A(i) = { }



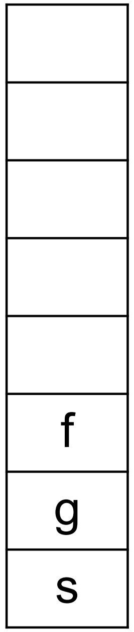


Considere a aresta (*v*,*w*). Se *w* estava desmarcado antes da visita, então (*v*,*w*) é **aresta da árvore** (floresta) de profundidade

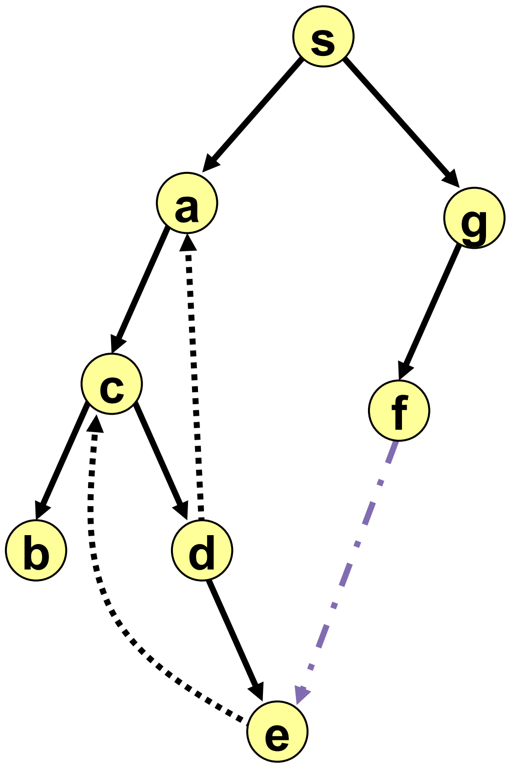
Busca Profundidade - Digrafos



A(s) = {a, g, b}
~~A(a) = {c}~~
~~A(b) = { }~~
~~A(c) = {b, d}~~
~~A(d) = {a, e}~~
~~A(e) = {c}~~
A(f) = {e, g}
A(g) = {f}
A(h) = {i, d}
A(i) = { }

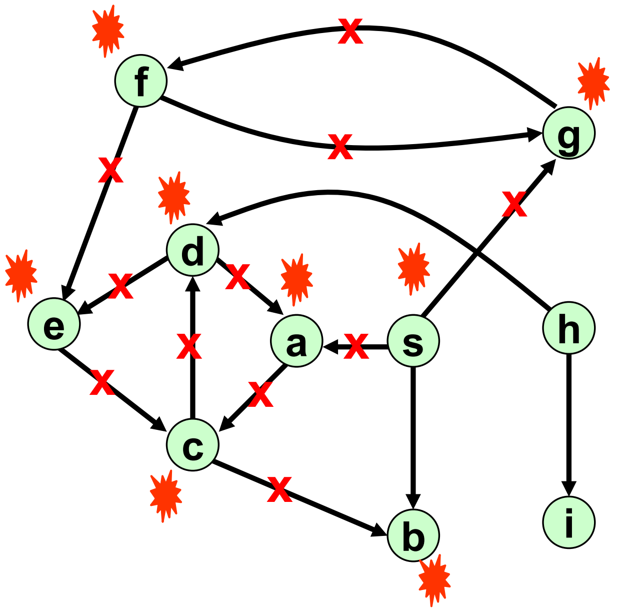


Q

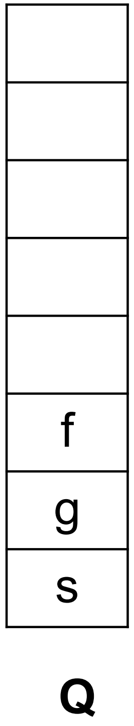


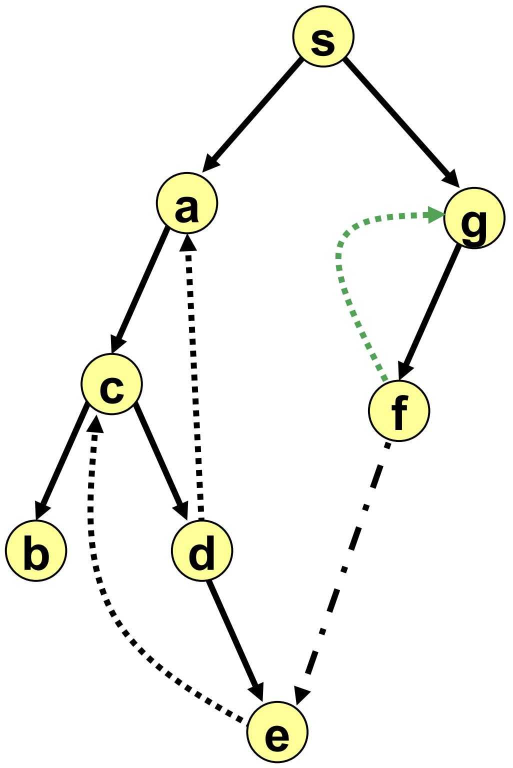
Considere a aresta (v,w). Se $w \notin Q$ no momento da visita, então (v,w) é denominada **aresta de cruzamento**

Busca Profundidade - Digrafos



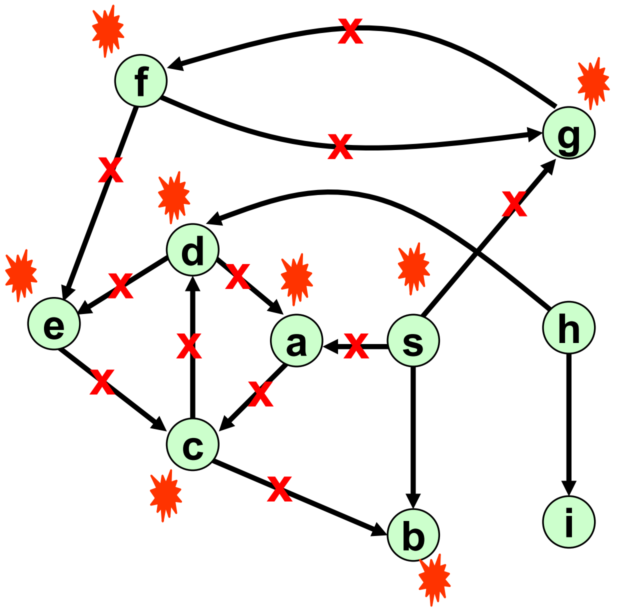
A(s) = {a, g, b}
~~A(a) = {c}~~
~~A(b) = { }~~
~~A(c) = {b, d}~~
~~A(d) = {a, e}~~
~~A(e) = {c}~~
A(f) = {e, g}
A(g) = {f}
A(h) = {i, d}
A(i) = { }



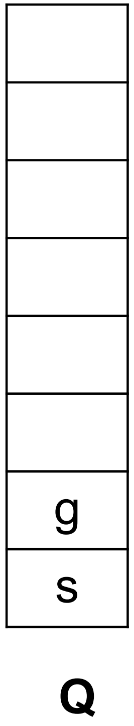


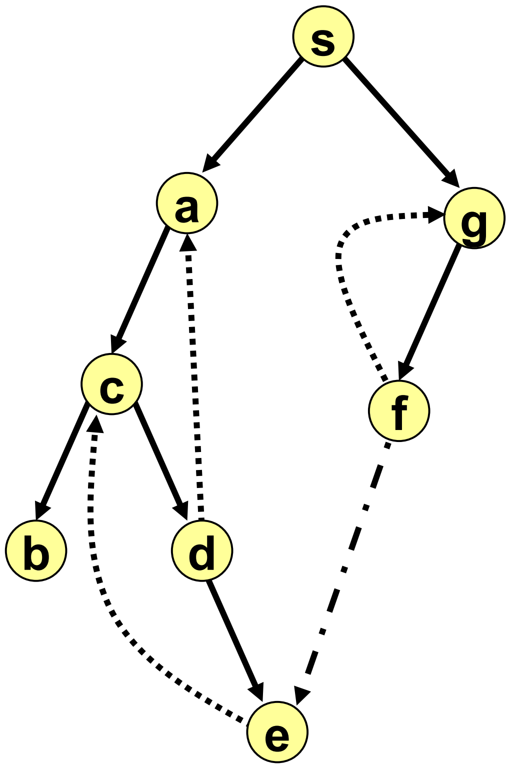
Considere a aresta (*v*,*w*). Se $w \in Q$ no momento da visita, então (*v*,*w*) denomina-se **aresta de retorno**

Busca Profundidade - Digrafos



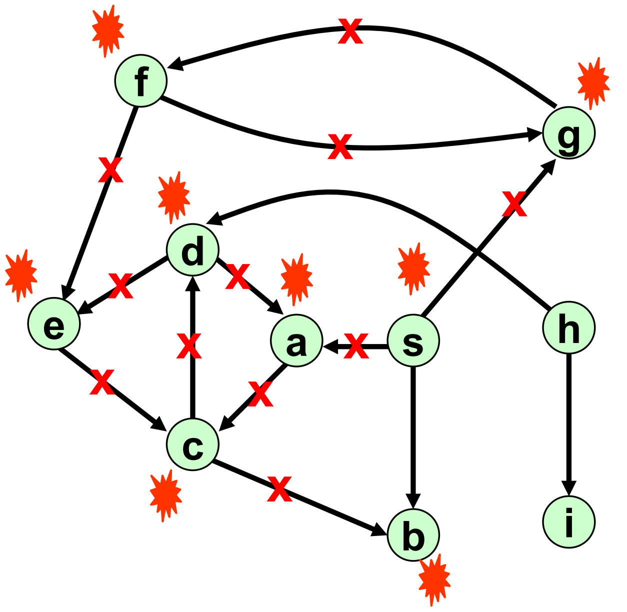
A(s) = {a, g, b}
~~A(a) = {c}~~
~~A(b) = { }~~
~~A(c) = {b, d}~~
~~A(d) = {a, e}~~
~~A(e) = {c}~~
~~A(f) = {c, g}~~
A(g) = {f}
A(h) = {i, d}
A(i) = { }





f é removido da pilha

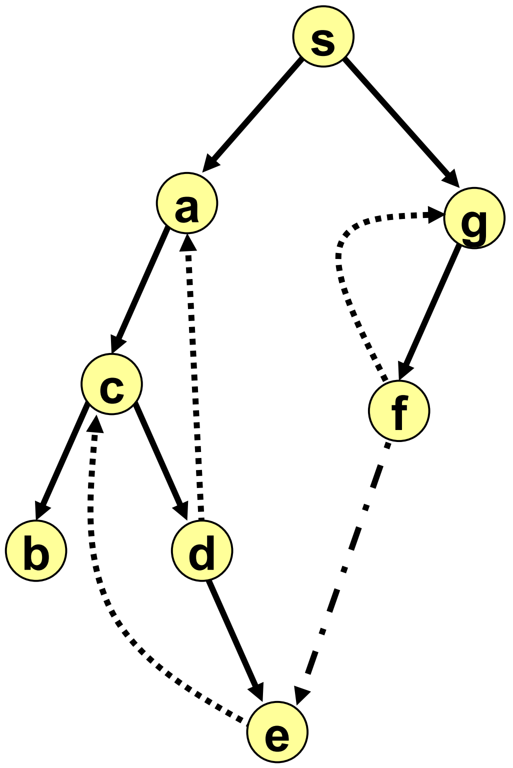
Busca Profundidade - Digrafos



A(s) = {a, g, b}
~~A(a) = {c}~~
~~A(b) = { }~~
~~A(c) = {b, d}~~
~~A(d) = {a, e}~~
~~A(e) = {c}~~
~~A(f) = {e, g}~~
~~A(g) = {f}~~
A(h) = {i, d}
A(i) = { }

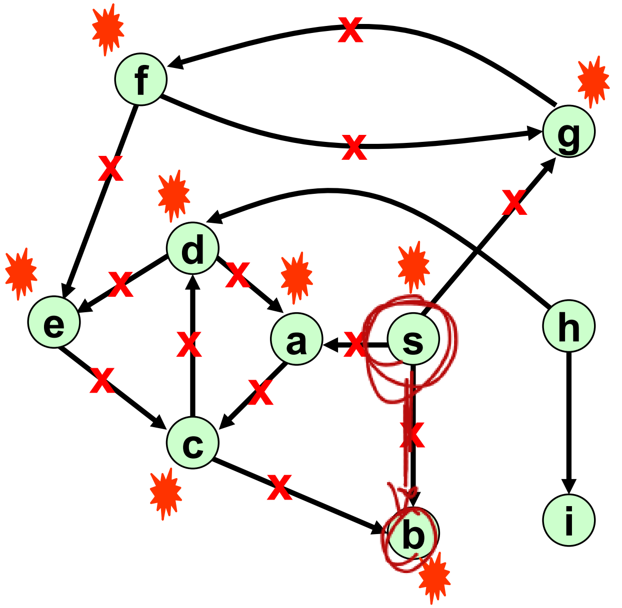
s

Q



g é removido da pilha

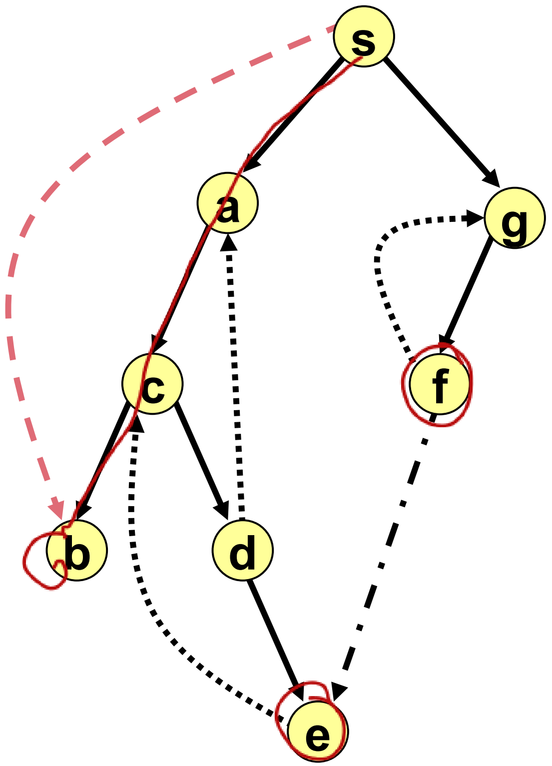
Busca Profundidade - Digrafos



A(s) = {a, g, b}
~~A(a) = {c}~~
~~A(b) = { }~~
~~A(c) = {b, d}~~
~~A(d) = {a, e}~~
~~A(e) = {c}~~
~~A(f) = {e, g}~~
~~A(g) = {f}~~
A(h) = {i, d}
A(i) = { }

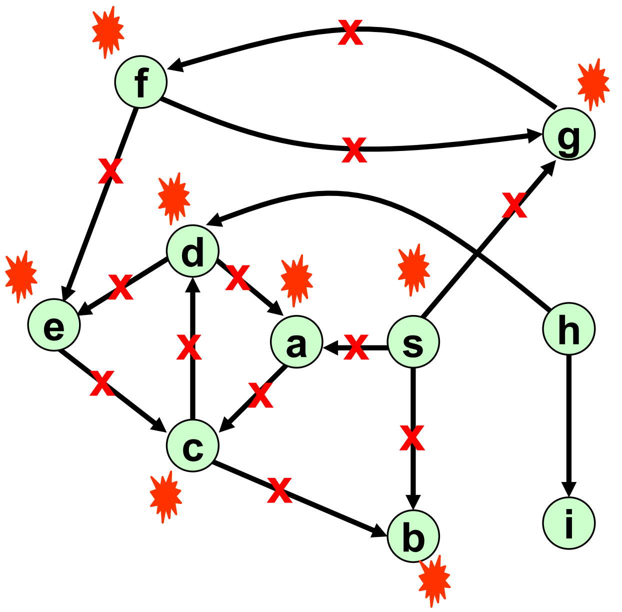
s

Q

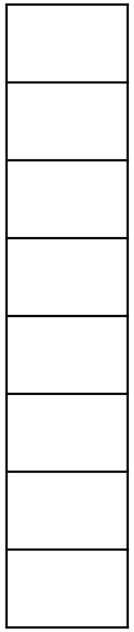


Considere a aresta (*v*,*w*). Se *w* estava marcado antes da visita, então (*v*,*w*) é **aresta de avanço** *(s é ancestral de b)*

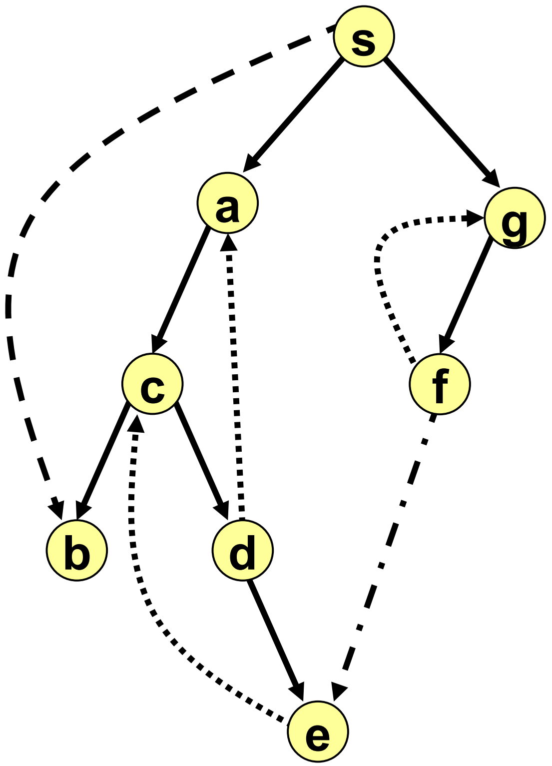
Busca Profundidade - Digrafos



~~A(s) = {a, g, b}~~
~~A(a) = {c}~~
~~A(b) = { }~~
~~A(c) = {b, d}~~
~~A(d) = {a, e}~~
~~A(e) = {c}~~
~~A(f) = {e, g}~~
~~A(g) = {f}~~
A(h) = {i, d}
A(i) = { }

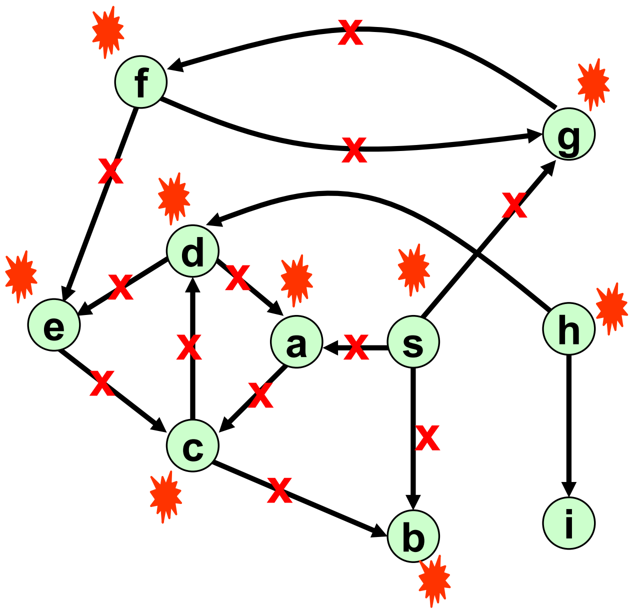


Q



s é removido da pilha

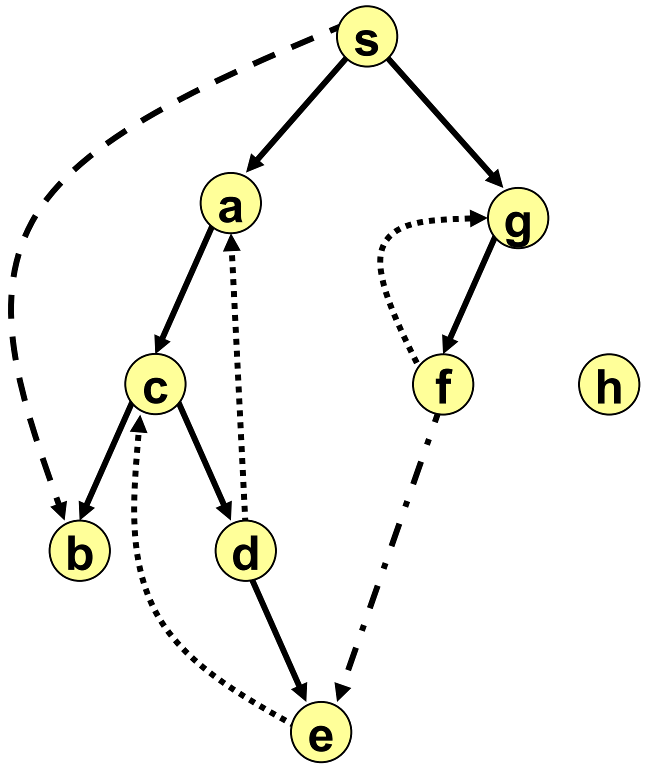
Busca Profundidade - Digrafos



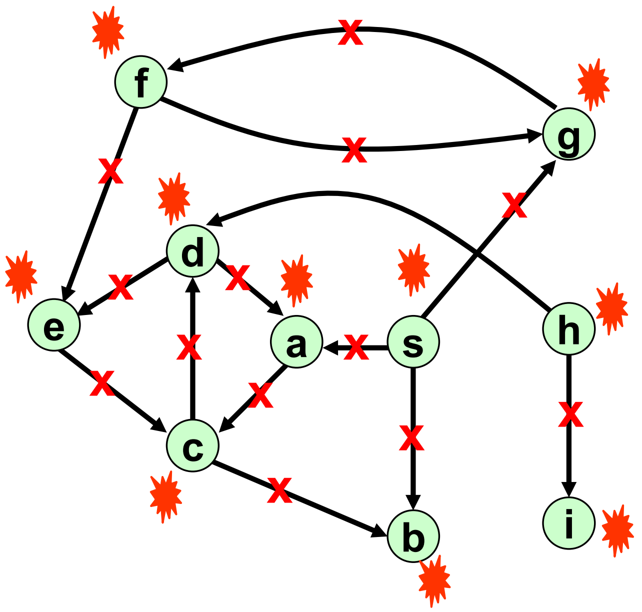
~~A(s) = {a, g, b}~~
~~A(a) = {c}~~
~~A(b) = { }~~
~~A(c) = {b, d}~~
~~A(d) = {a, e}~~
~~A(e) = {c}~~
~~A(f) = {e, g}~~
~~A(g) = {f}~~
A(h) = {i, d}
A(i) = { }

h

Q



Busca Profundidade - Digrafos



~~A(s) = {a, g, b}~~

~~A(a) = {c}~~

~~A(b) = { }~~

~~A(c) = {b, d}~~

~~A(d) = {a, e}~~

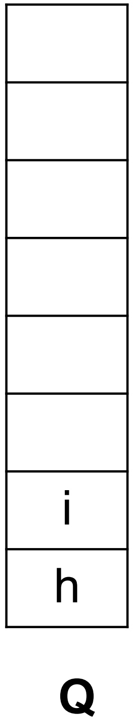
~~A(e) = {c}~~

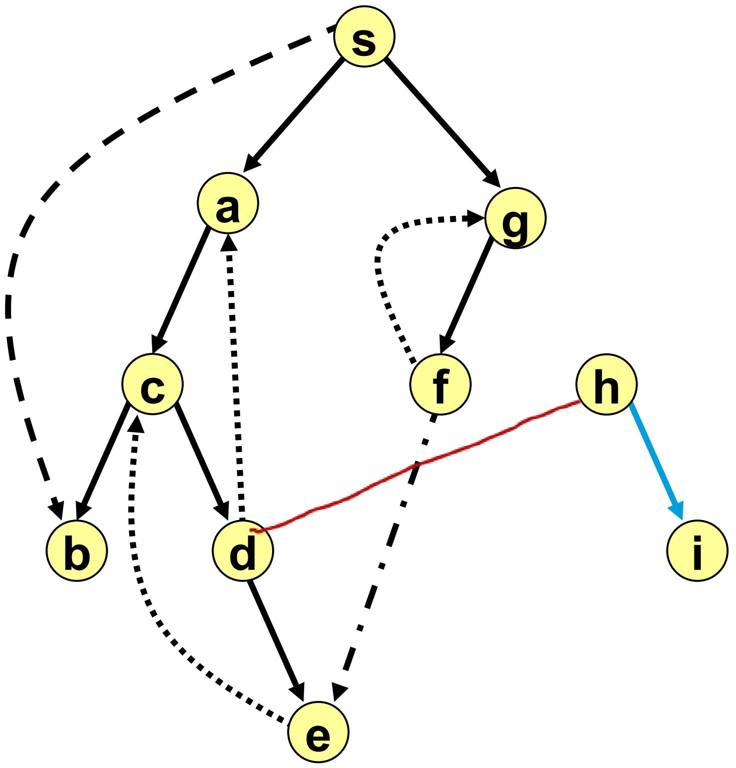
~~A(f) = {c, g}~~

~~A(g) = {f}~~

A(h) = {i, d}

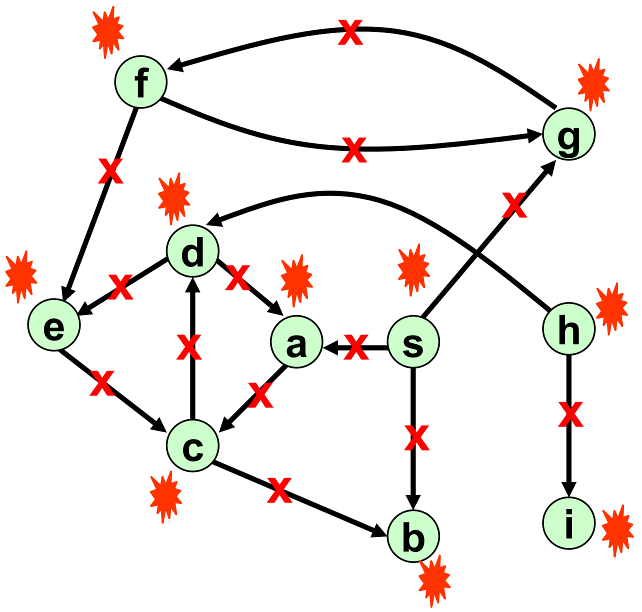
~~A(i) = { }~~





Considere a aresta (*v*,*w*). Se *w* estava desmarcado antes da visita, então (*v*,*w*) é **aresta da árvore** (floresta) de profundidade

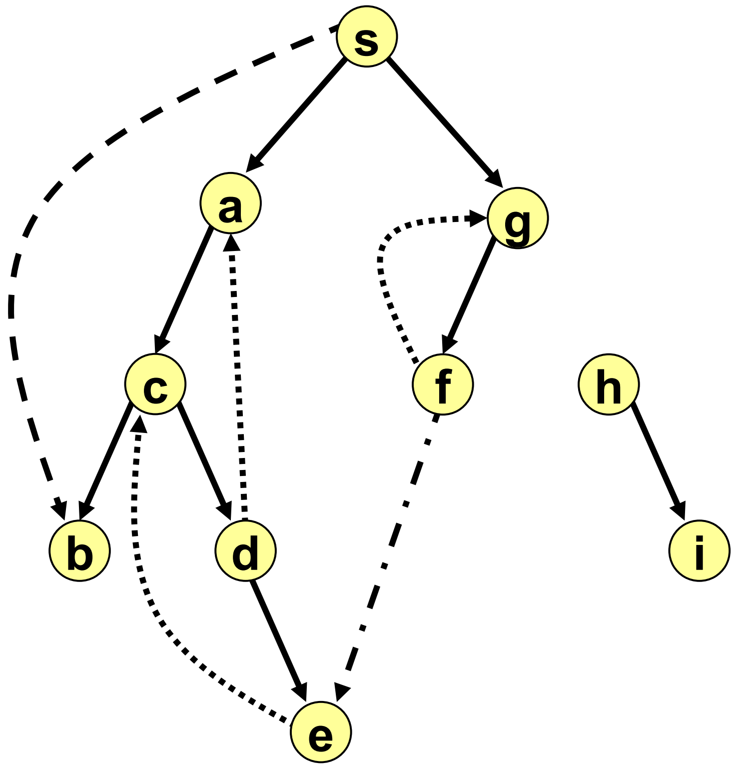
Busca Profundidade - Digrafos



~~A(s) = {a, g, b}~~
~~A(a) = {c}~~
~~A(b) = { }~~
~~A(c) = {b, d}~~
~~A(d) = {a, e}~~
~~A(e) = {c}~~
~~A(f) = {e, g}~~
~~A(g) = {f}~~
A(h) = {i, d}
~~A(i) = { }~~

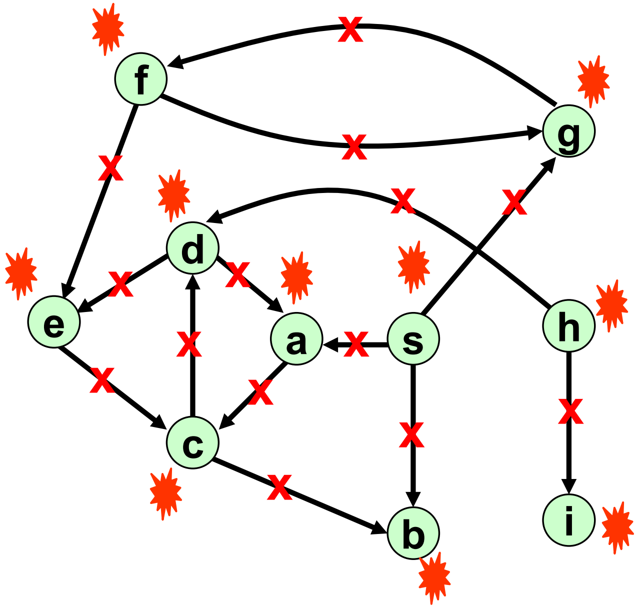
h

Q



i é removido da pilha

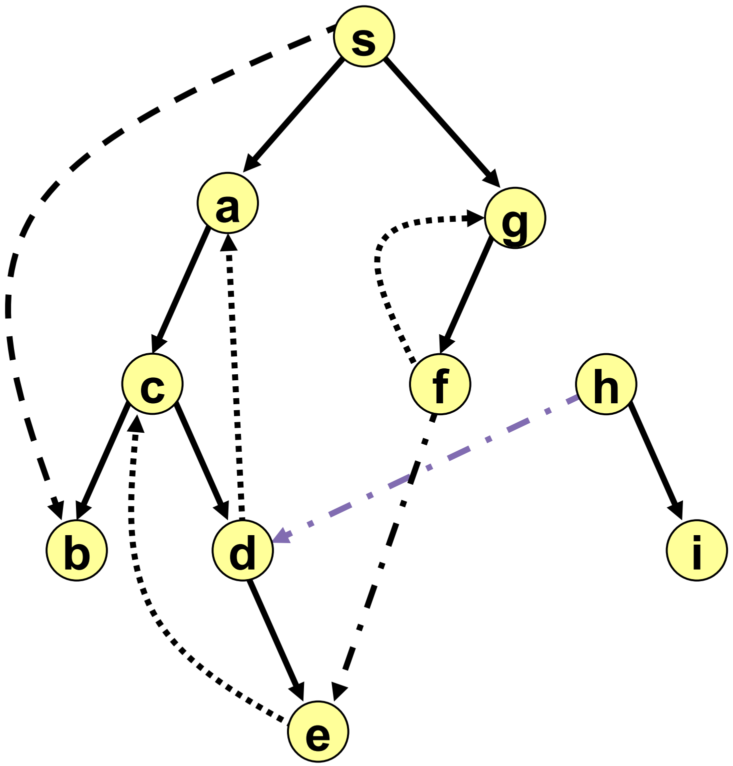
Busca Profundidade - Digrafos



~~A(s) = {a, g, b}~~
~~A(a) = {c}~~
~~A(b) = { }~~
~~A(c) = {b, d}~~
~~A(d) = {a, e}~~
~~A(e) = {c}~~
~~A(f) = {c, g}~~
~~A(g) = {f}~~
A(h) = {i, d}
~~A(i) = { }~~

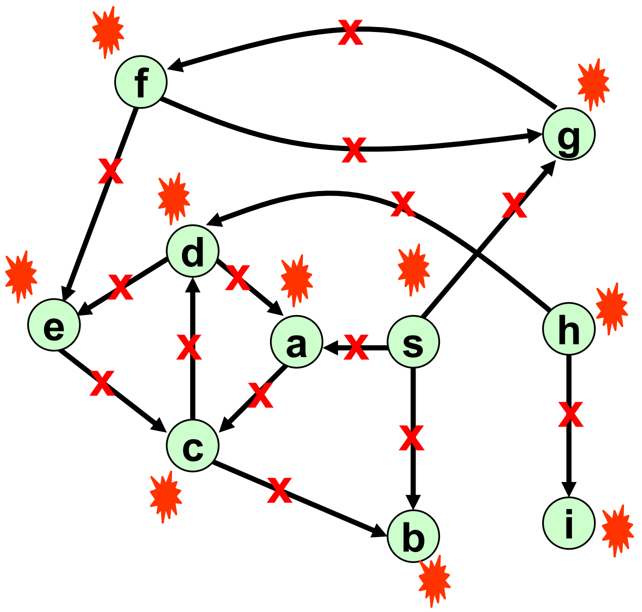
h

Q

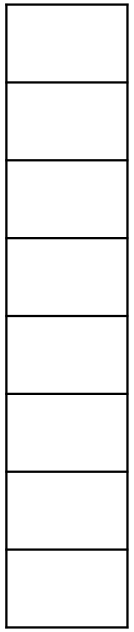


Considere a aresta (v,w). Se $w \notin Q$ no momento da visita, então (v,w) é denominada **aresta de cruzamento**

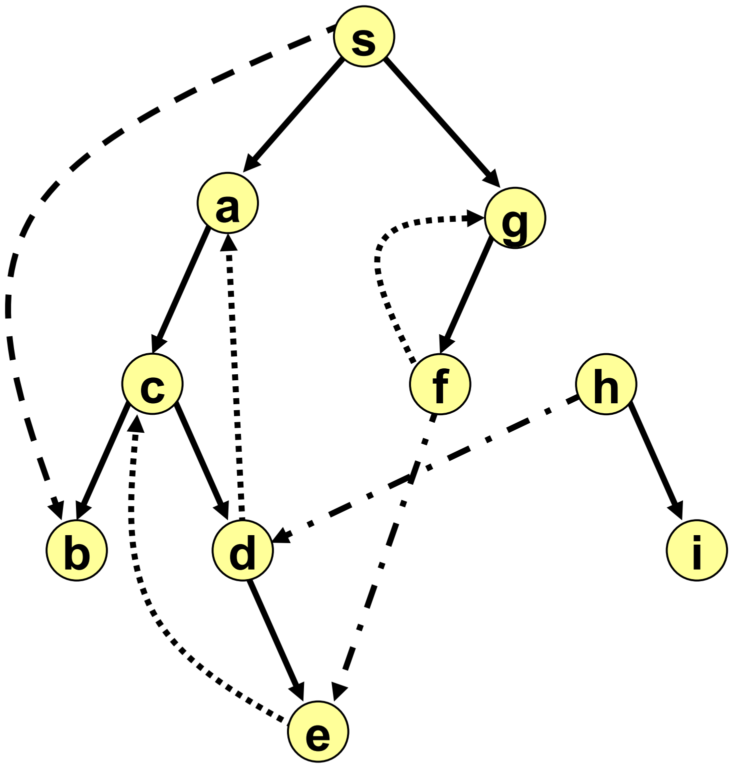
Busca Profundidade - Digrafos



~~A(s) = {a, g, b}~~
~~A(a) = {c}~~
~~A(b) = { }~~
~~A(c) = {b, d}~~
~~A(d) = {a, e}~~
~~A(e) = {c}~~
~~A(f) = {e, g}~~
~~A(g) = {f}~~
~~A(h) = {i, d}~~
~~A(i) = { }~~

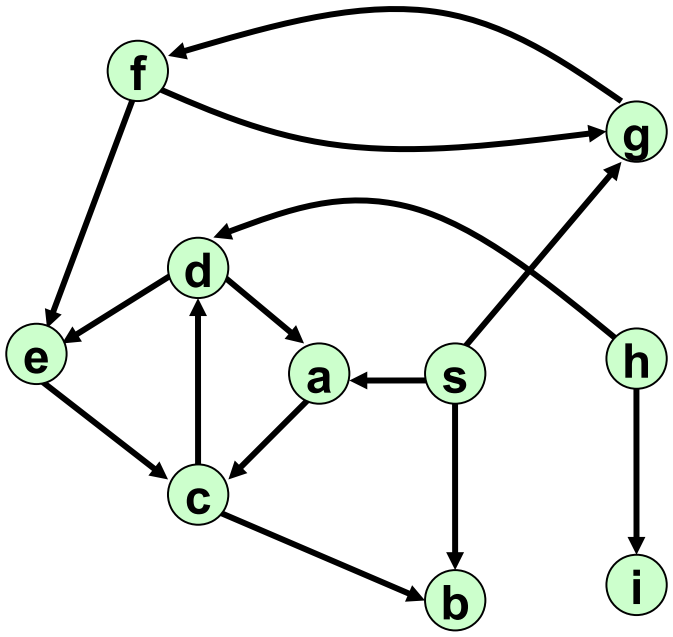


Q

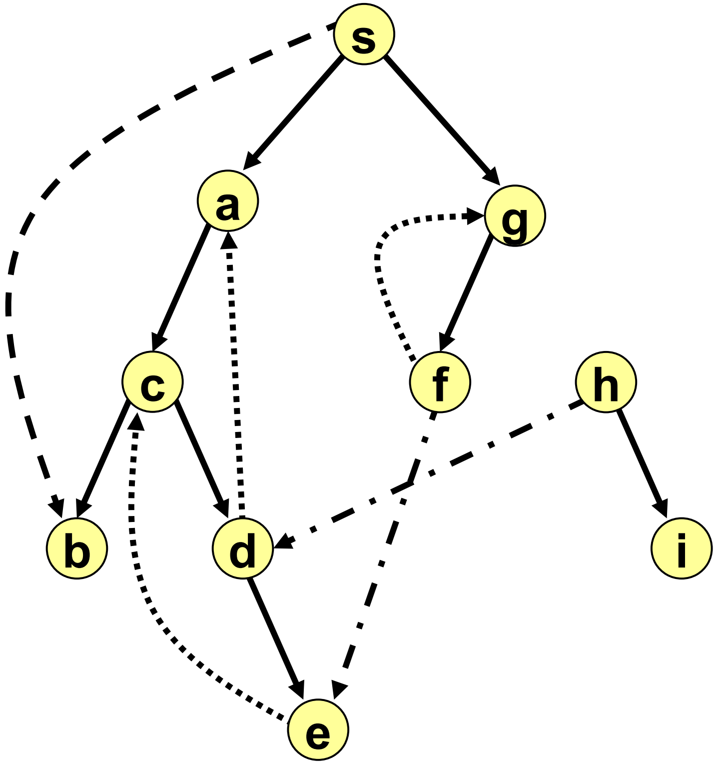


h é removido da pilha

Busca Profundidade - Digrafos



Digrafo



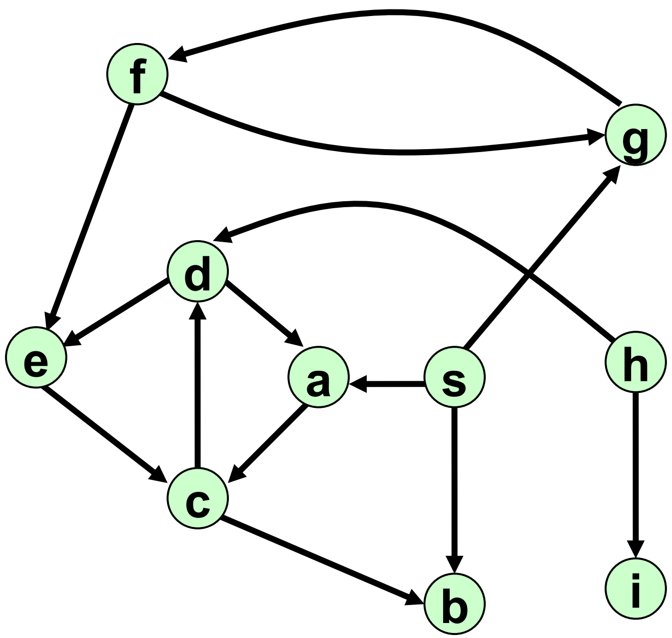
Aresta da árvore

Aresta de avanço

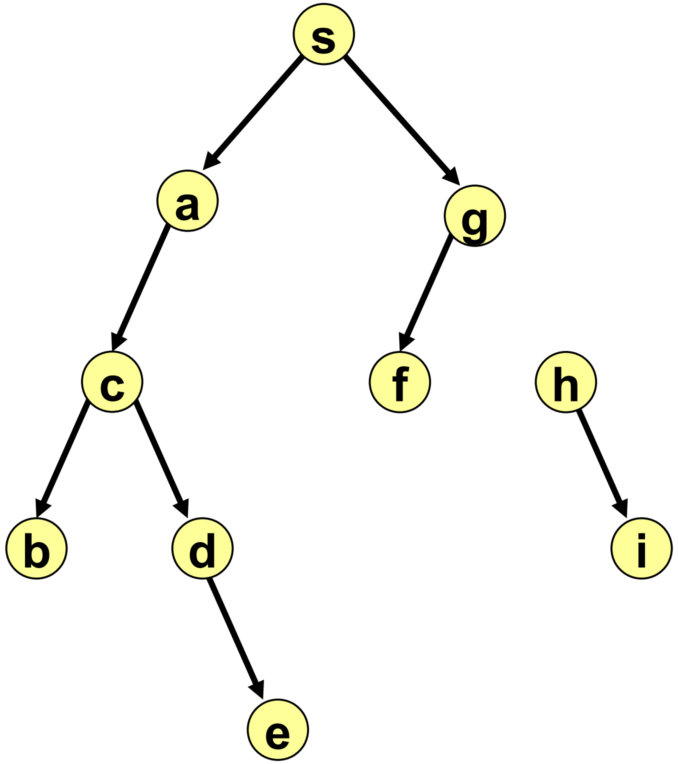
Aresta de retorno

Aresta de cruzamento

Busca Profundidade - Digrafos



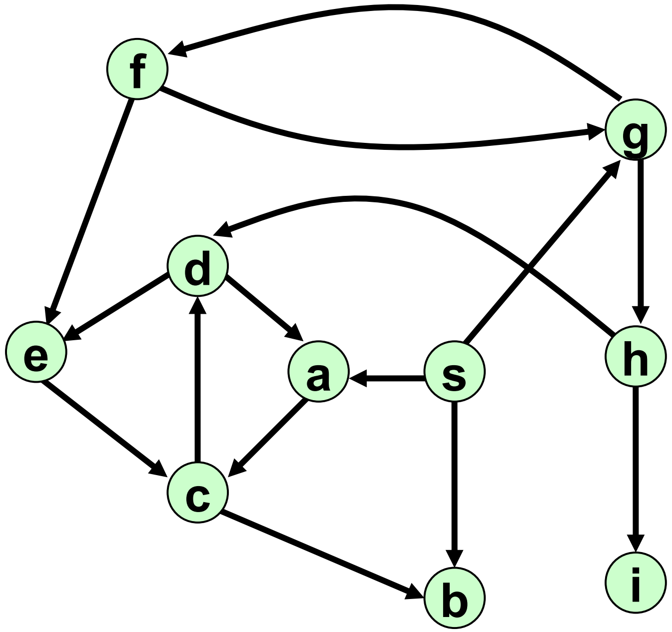
Digrafo



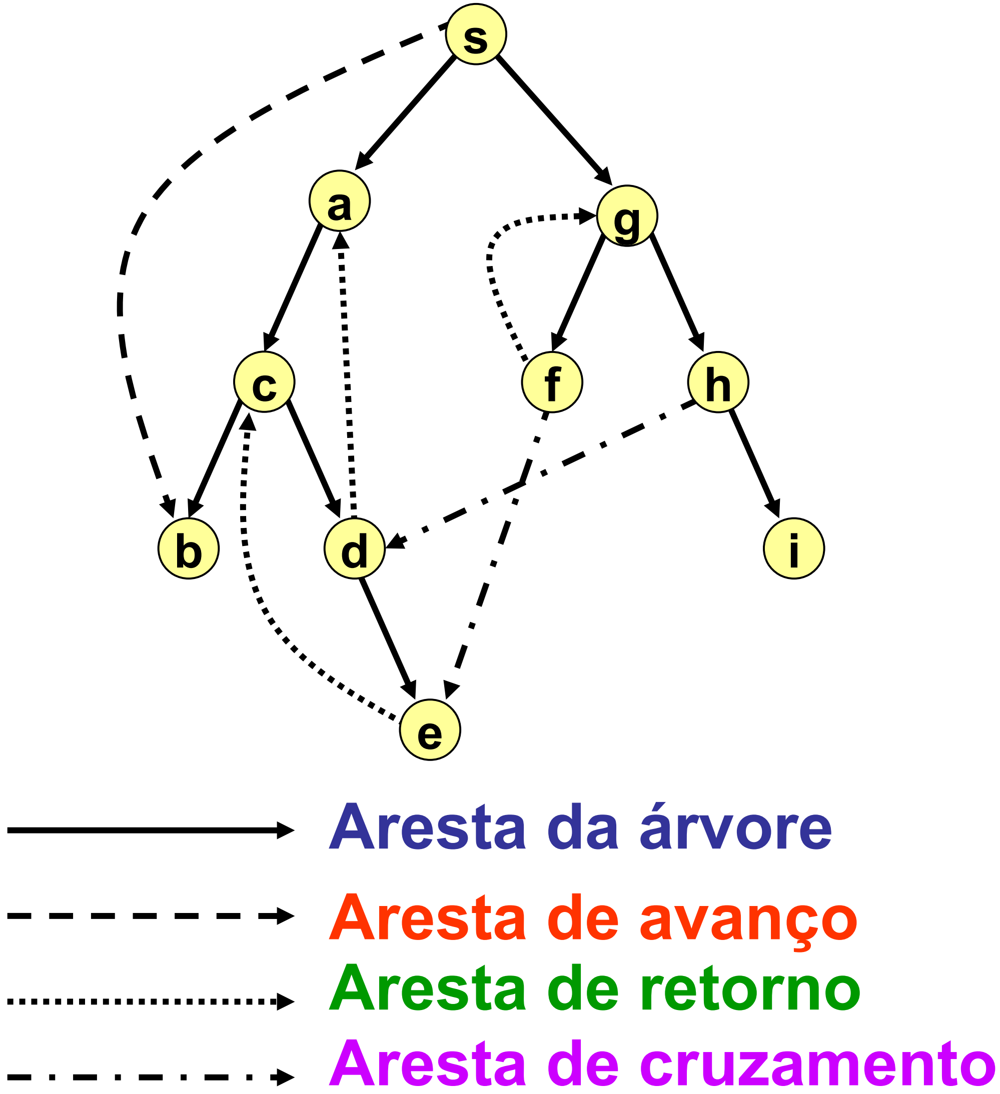
Floresta de Profundidade

Busca Profundidade - Digrafos

Outro Exemplo:

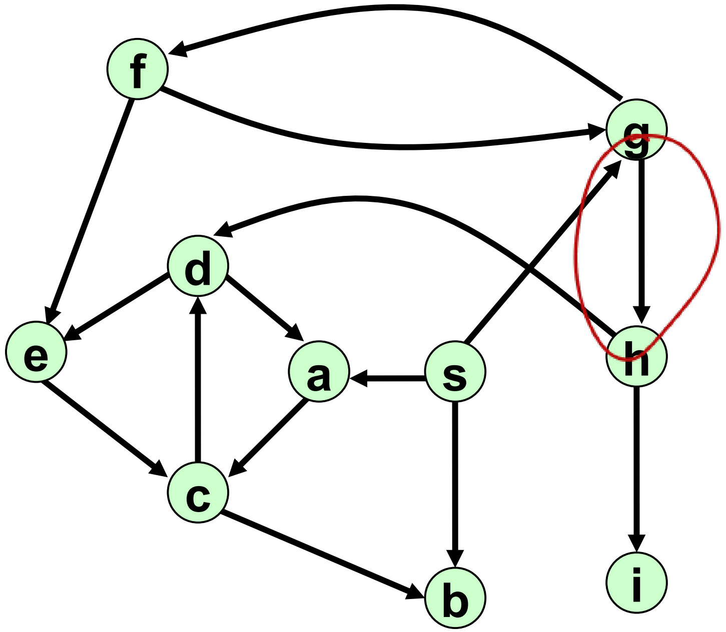


Digrafo

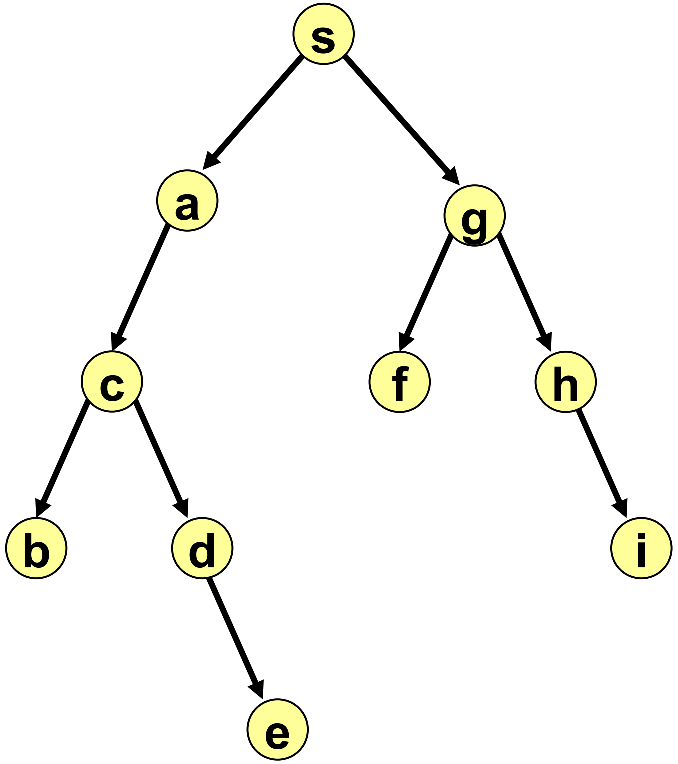


Busca Profundidade - Digrafos

Outro Exemplo:



Digrafo



Árvore de Profundidade

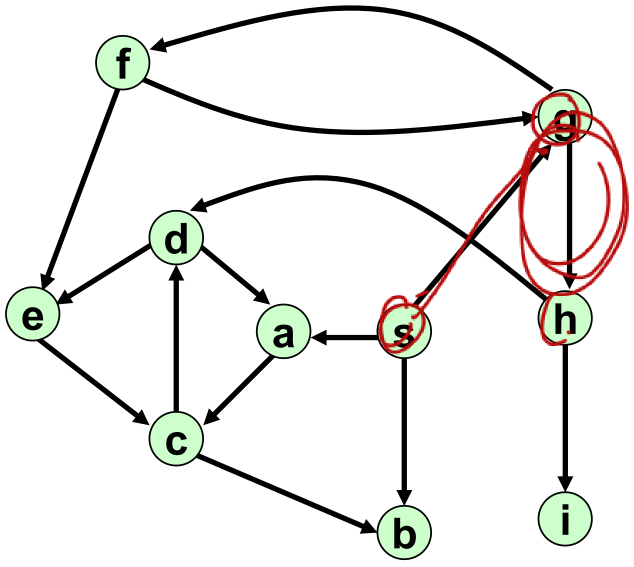
Busca Profundidade - Digrafos

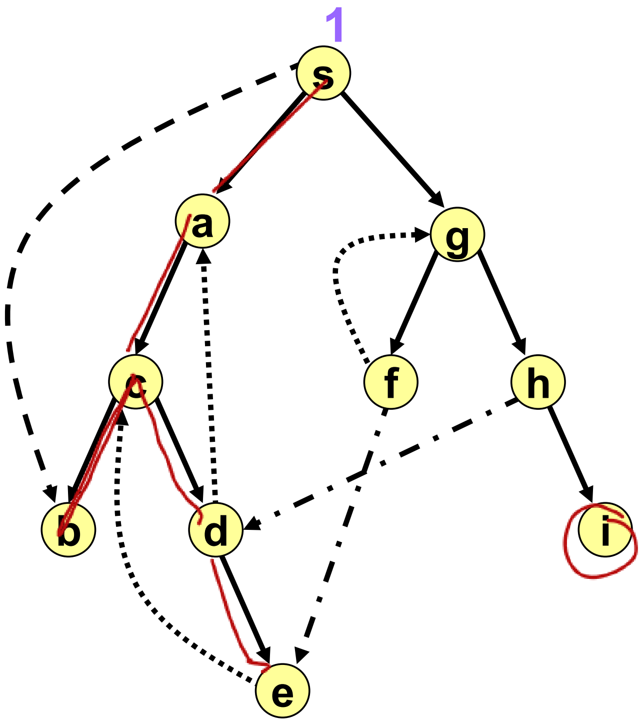
Assim como para grafos simples, a ordem de entrada e saída dos vértices na pilha *Q* são importantes para algumas aplicações.

Assim sendo, para cada vértice *v* define-se *profundidade de entrada de v*, PE(*v*), e *profundidade de saída de v*, PS(*v*), respectivamente, como sendo o número de ordem que *v* foi incluído e excluído da pilha *Q*.

Busca Profundidade - Digrafos

Para o exemplo anterior, temos:

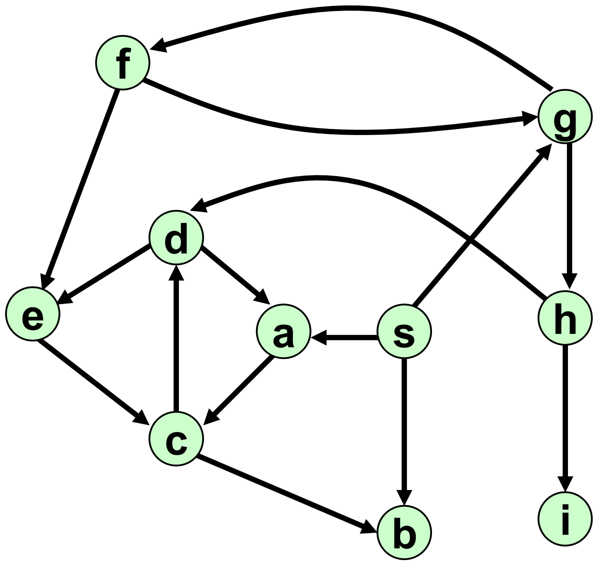


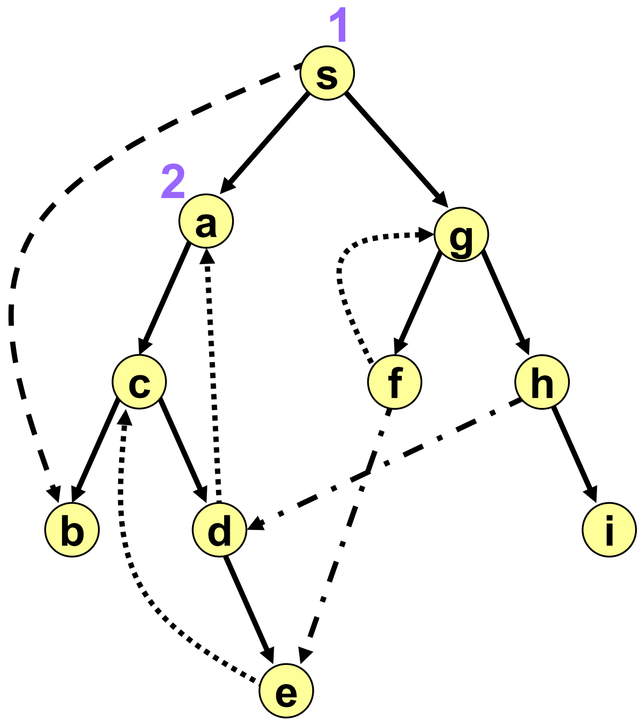


v	s	a	b	c	d	e	f	g	h	i
PE(v)	1	2	4	3	5	6	8	7	9	10
PS(v)	10	5	1	4	3	2	6	9	8	7

Busca Profundidade - Digrafos

Para o exemplo anterior, temos:

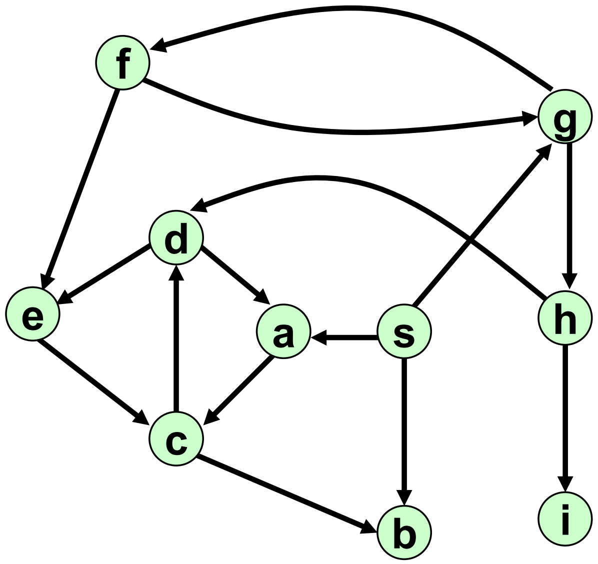


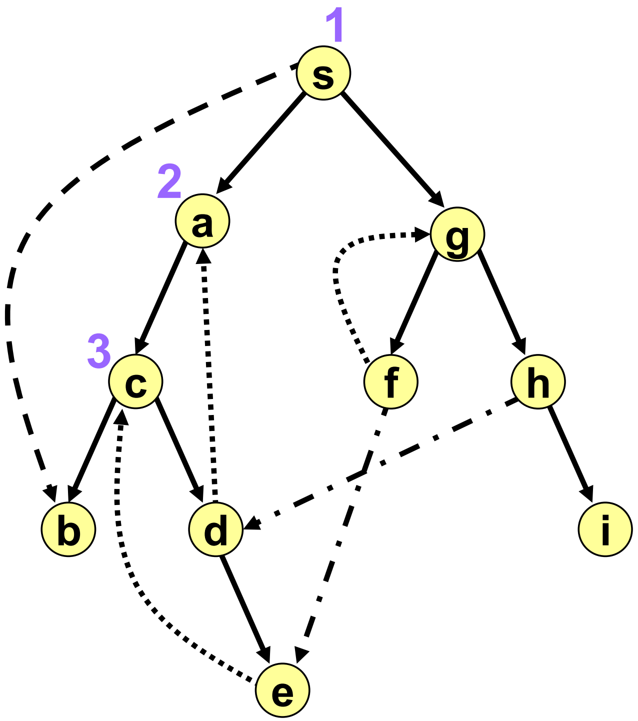


v	s	a	b	c	d	e	f	g	h	i
PE(v)	1	2								
PS(v)										

Busca Profundidade - Digrafos

Para o exemplo anterior, temos:

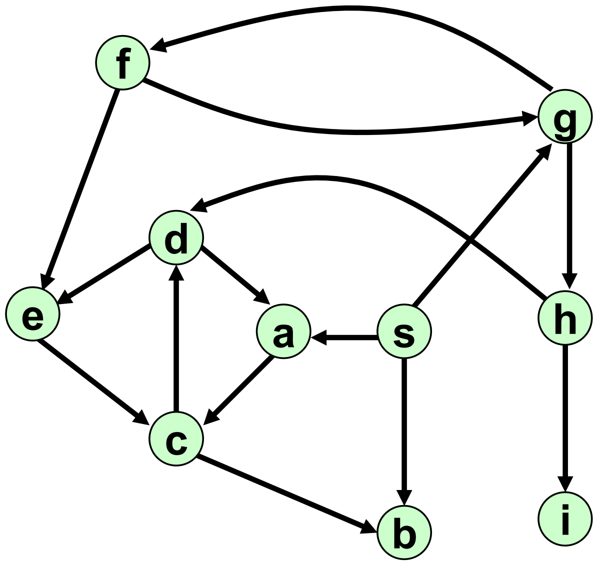


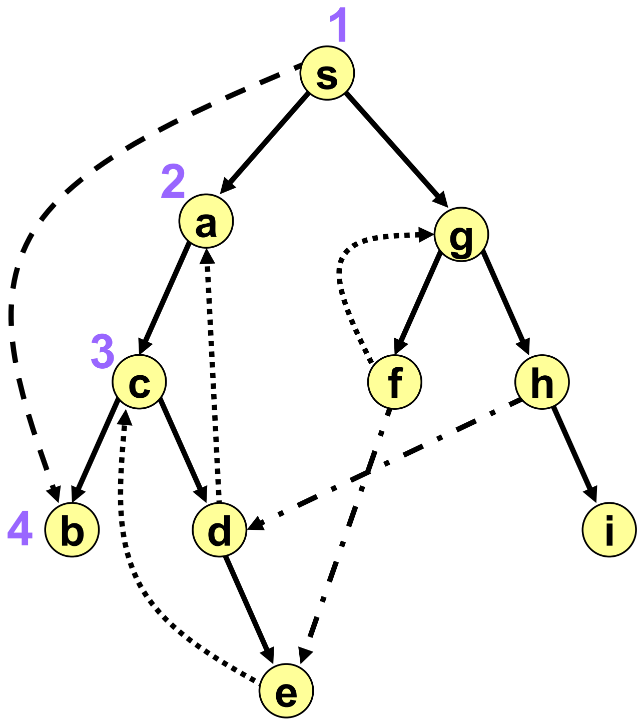


v	s	a	b	c	d	e	f	g	h	i
PE(v)	1	2		3						
PS(v)										

Busca Profundidade - Digrafos

Para o exemplo anterior, temos:

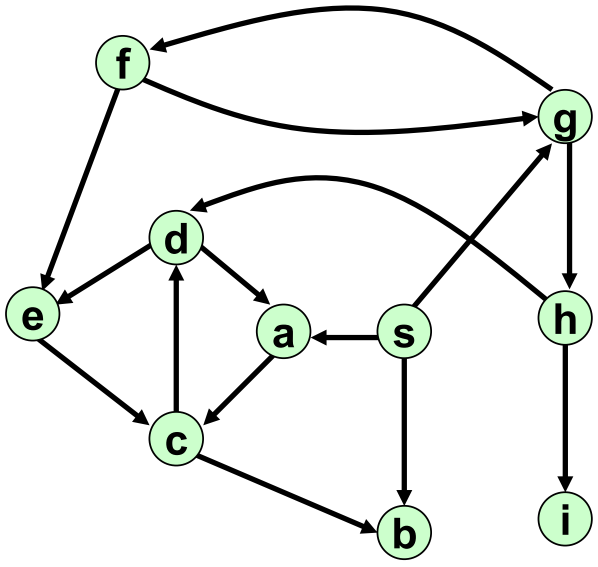


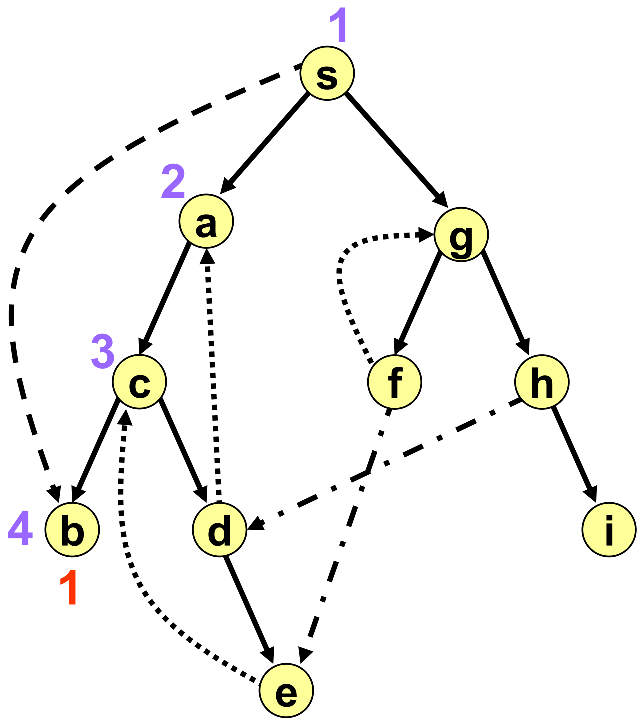


v	s	a	b	c	d	e	f	g	h	i
PE(v)	1	2	4	3						
PS(v)										

Busca Profundidade - Digrafos

Para o exemplo anterior, temos:

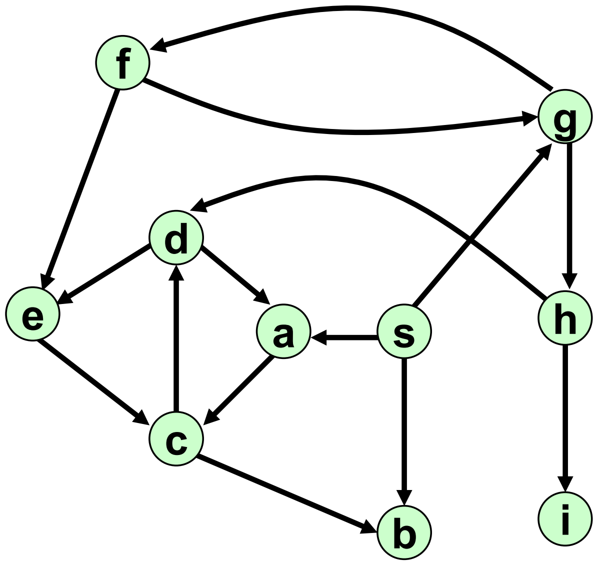


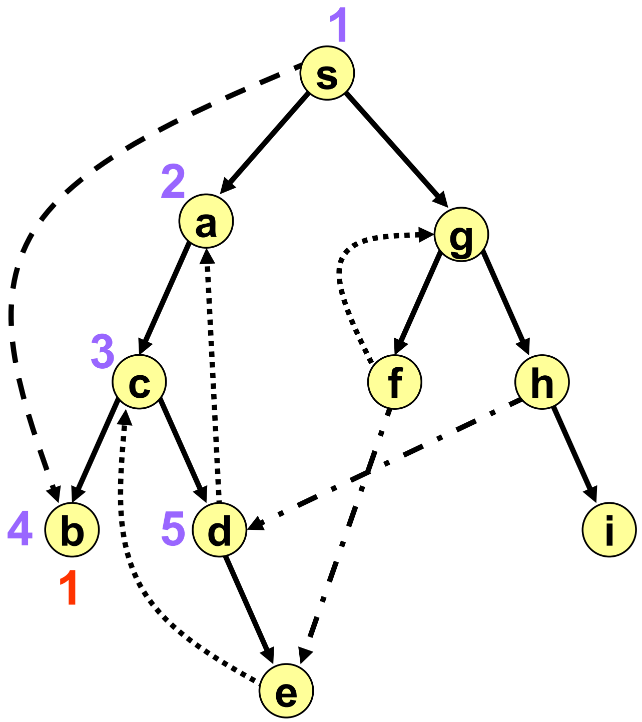


v	s	a	b	c	d	e	f	g	h	i
PE(v)	1	2	4	3						
PS(v)			1							

Busca Profundidade - Digrafos

Para o exemplo anterior, temos:

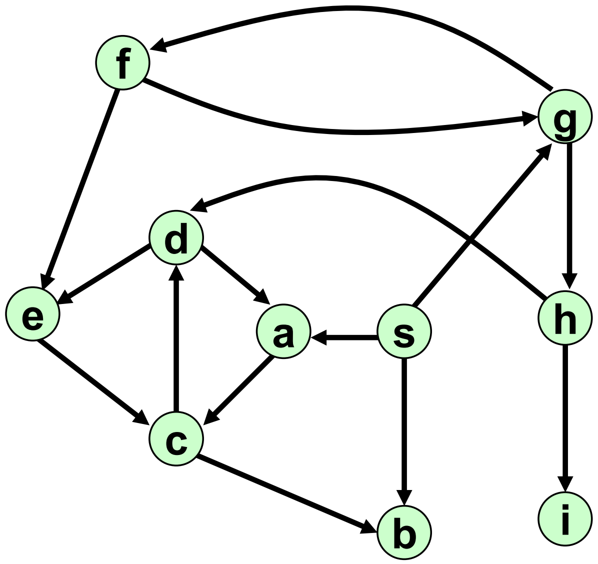


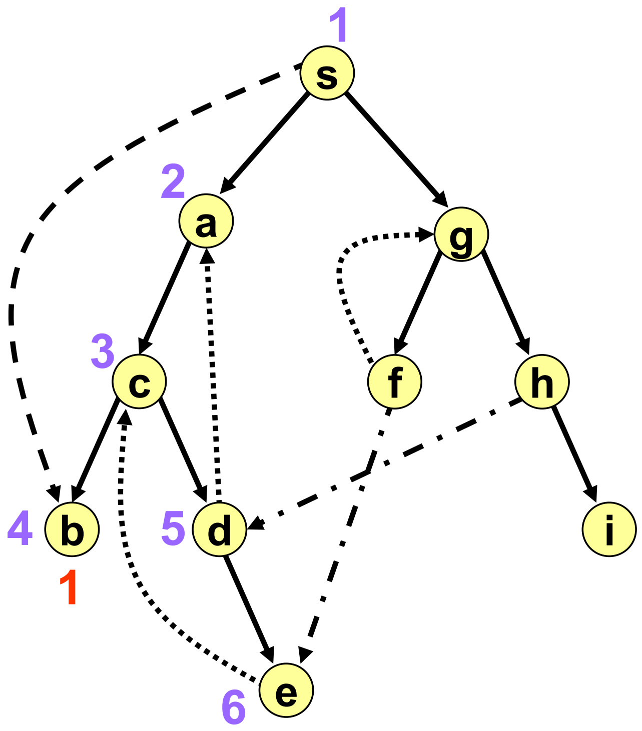


v	s	a	b	c	d	e	f	g	h	i
PE(v)	1	2	4	3	5					
PS(v)			1							

Busca Profundidade - Digrafos

Para o exemplo anterior, temos:

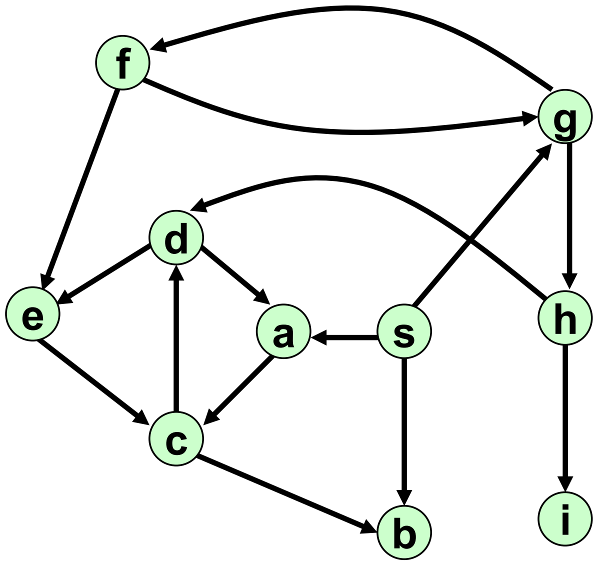


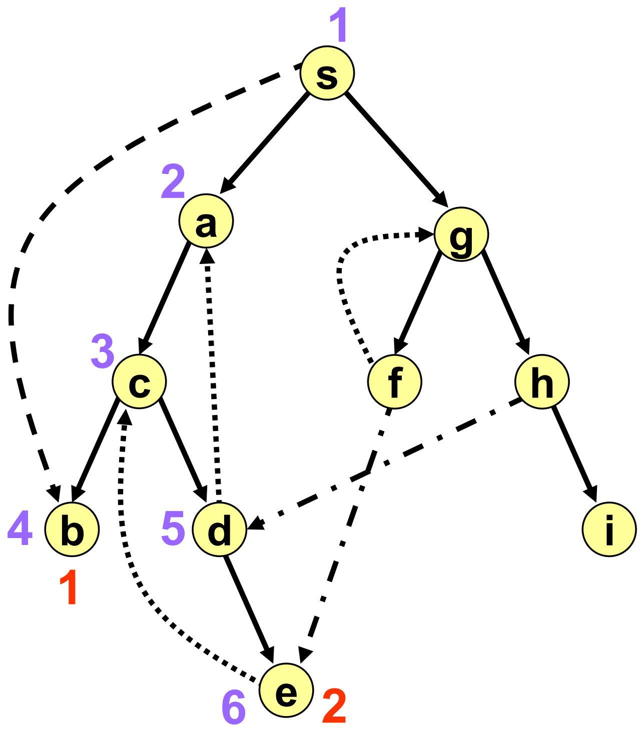


v	s	a	b	c	d	e	f	g	h	i
PE(v)	1	2	4	3	5	6				
PS(v)			1							

Busca Profundidade - Digrafos

Para o exemplo anterior, temos:



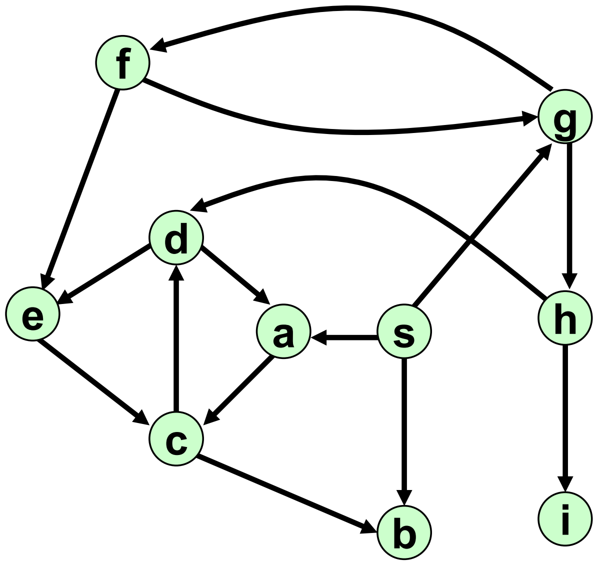


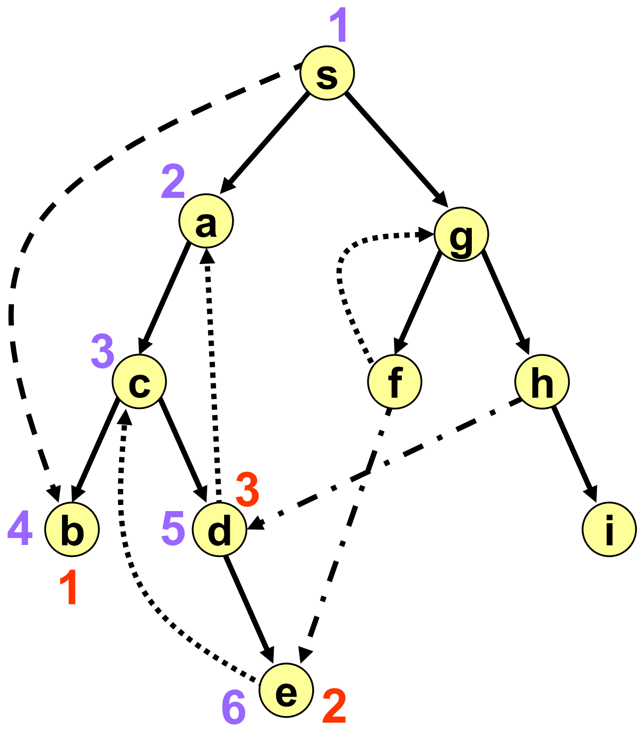
v	s	a	b	c	d	e	f	g	h	i
PE(v)	1	2	4	3	5	6				
PS(v)			1			2				

Busca Profundidade - Digrafos

Para o exemplo anterior, temos:

4
1, 2
1, 3
1, 4
2, 3

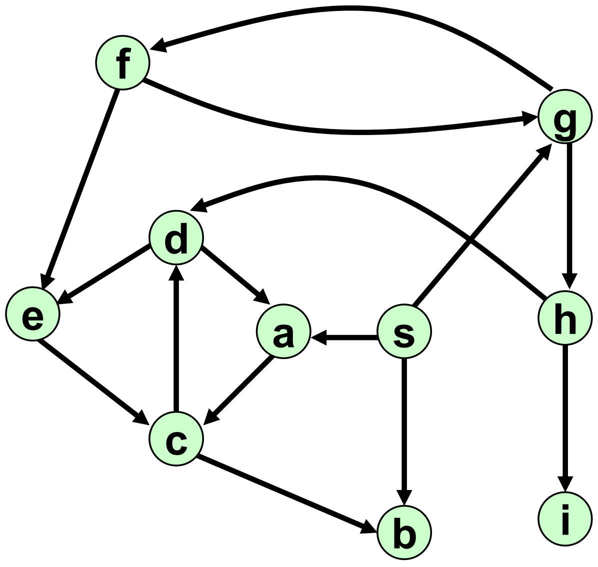


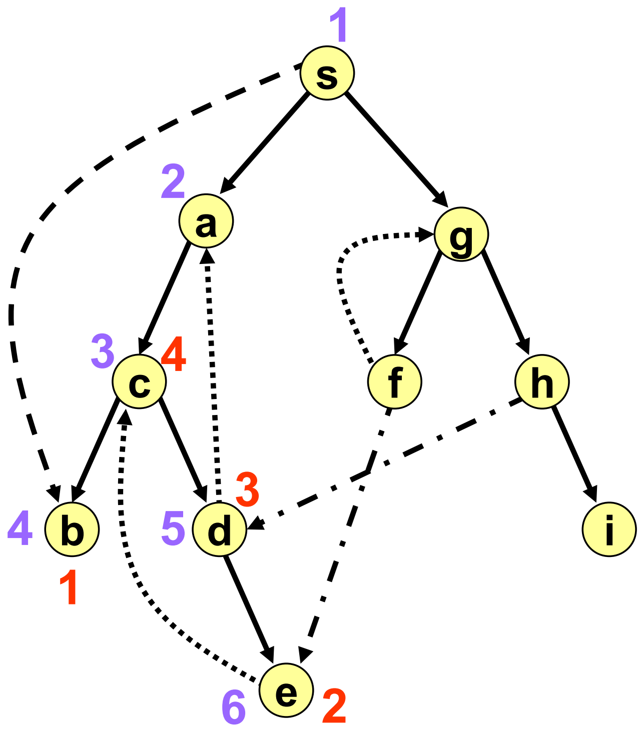


v	s	a	b	c	d	e	f	g	h	i
PE(v)	1	2	4	3	5	6				
PS(v)			1		3	2				

Busca Profundidade - Digrafos

Para o exemplo anterior, temos:

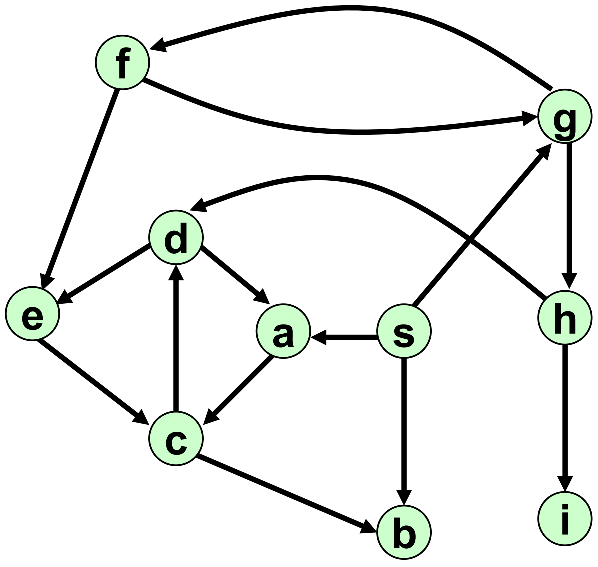


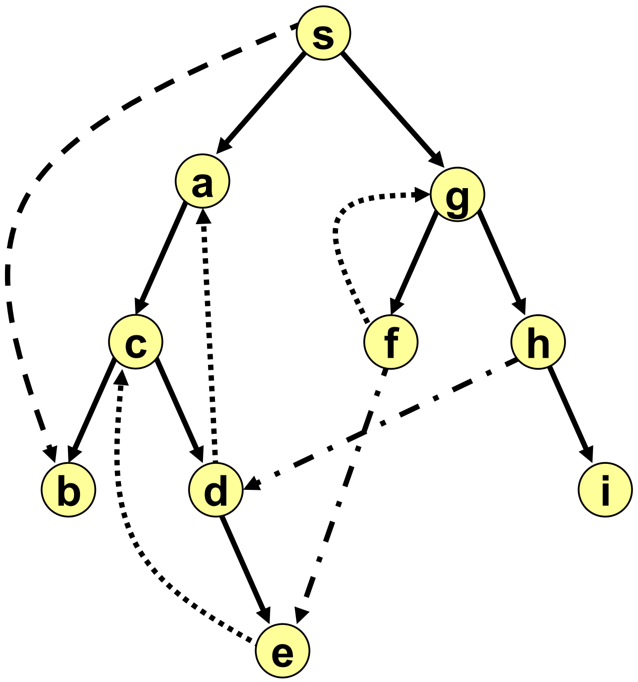


v	s	a	b	c	d	e	f	g	h	i
PE(v)	1	2	4	3	5	6				
PS(v)			1	4	3	2				

Busca Profundidade - Digrafos

Para o exemplo anterior, temos:

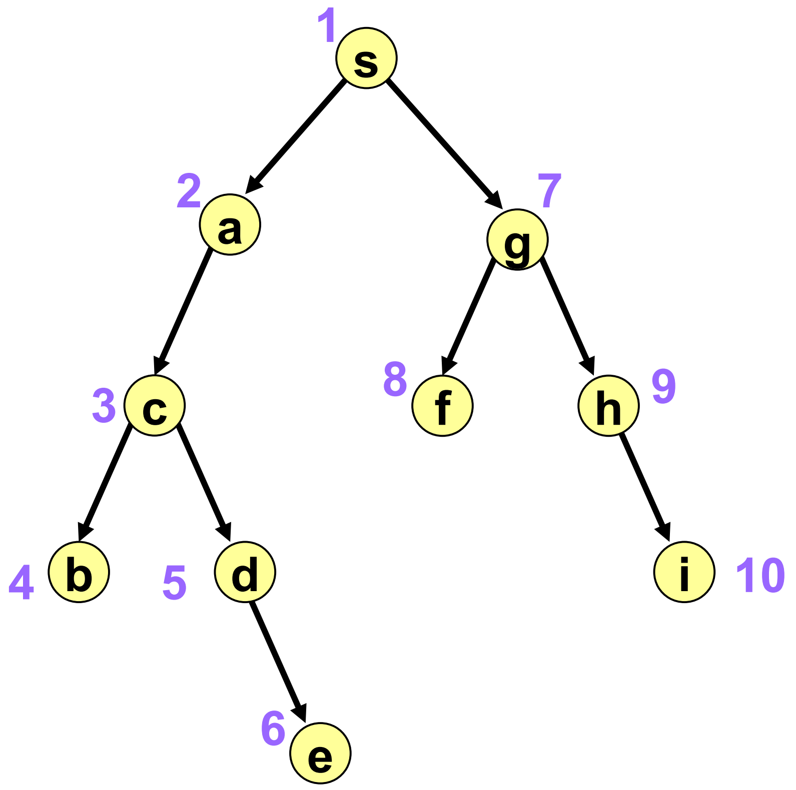




v	s	a	b	c	d	e	f	g	h	i
PE(v)	1	2	4	3	5	6	8	7	9	10
PS(v)	10	5	1	4	3	2	6	9	8	7

Busca Profundidade - Digrafos

Observa-se que a sequência dos vértices *v* de um digrafo *D* em ordem crescente de PE(*v*) corresponde a um caminhamento pré-ordem na árvore de profundidade produzida pela busca correspondente.



Busca Profundidade - Digrafos

Os valores de entrada e saída podem ser utilizados para classificar as arestas de um digrafo através de uma busca em profundidade com o seguinte lema:

Lema: Considere $D(V,E)$ um digrafo de raiz s, (v,w) uma aresta de D, e B um percurso em profundidade com s a raiz da busca, então:

1. (v,w) é **aresta da árvore ou avanço** se e somente se $PE(v) < PE(w)$

2. (v,w) é **aresta de retorno** se e somente se $PE(v) > PE(w)$ e $PS(v) < PS(w)$

3. (v,w) é **aresta de cruzamento** se e somente se $PE(v) > PE(w)$ e $PS(v) > PS(w)$