

ALGORITMOS EM GRAFOS

ALGORITMO DE TARJAN (CONECTIVIDADE)

Prof. Alexei Machado

Algoritmo de Tarjan

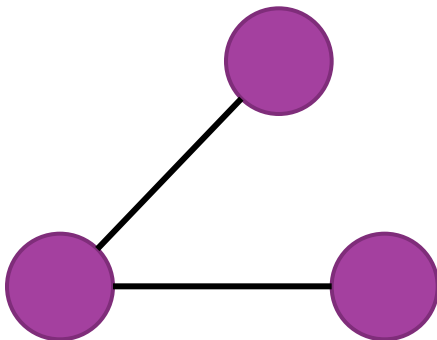
2

- Baseado na busca em profundidade, tenta localizar pontes (*cut-edges*) e pontos de articulação (*cut-vertices*)

Algoritmo de Tarjan

3

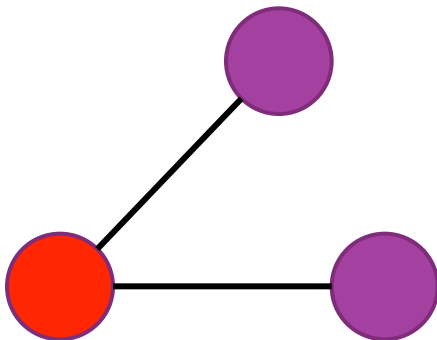
- Um vértice é ponto de articulação se:
 - ▣ Possuir dois ou mais sucessores (arestas de árvore)



Algoritmo de Tarjan

4

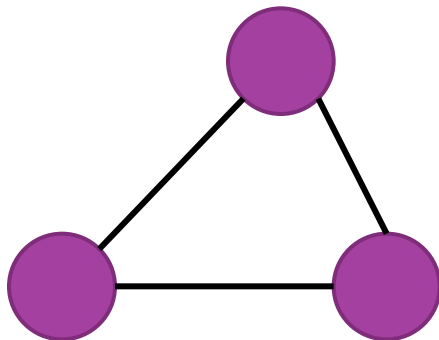
- Um vértice é ponto de articulação se:
 - ▣ Possuir dois ou mais sucessores (arestas de árvore)



Algoritmo de Tarjan

5

- Um vértice é ponto de articulação se e somente se:
 - ▣ Possuir um sucessor que não alcance um antecessor seu



Algoritmo de Tarjan

6

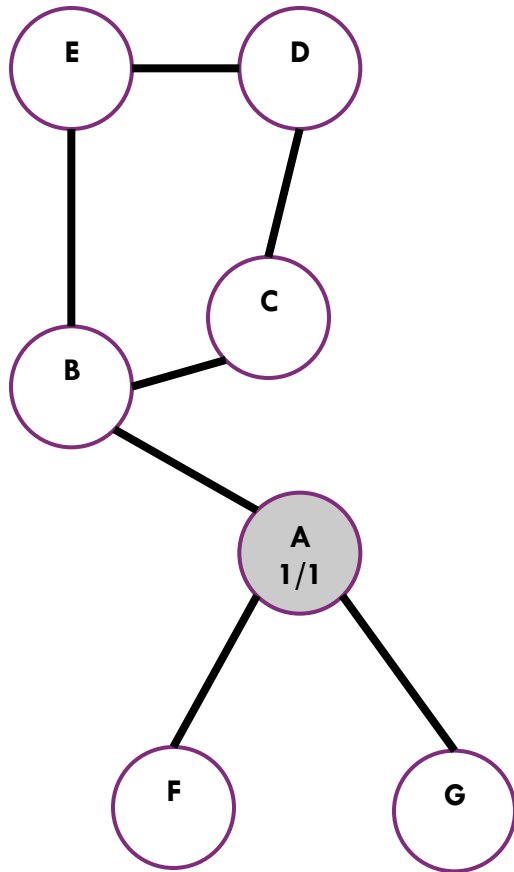
- ponto[u]: identifica se $u \in V$ é um ponto de articulação (booleano)
- ponte[u, v]: identifica se $(u, v) \in E$ é uma ponte (booleano)
- d[u]: tempo de descoberta do vértice u
- m[u]: menor valor de d[v] para um vértice v alcançável na busca em profundidade a partir de u

Algoritmo de Tarjan - principal

```
para cada  $u \in V$  faça
    ponto[u] = falso; cor[u] = branco;
para cada  $(u, v) \in E$  faça
    ponte[u, v] = falso;
tempo = 0;
para cada  $u \in V$  faça
    se cor[u] == branco então
        contador = 0;
        TARJAN-BP(G, u, u);
        se contador  $\geq 2$  então
            ponto[u] = verdadeiro;
        senão
            ponto[u] = falso;
```

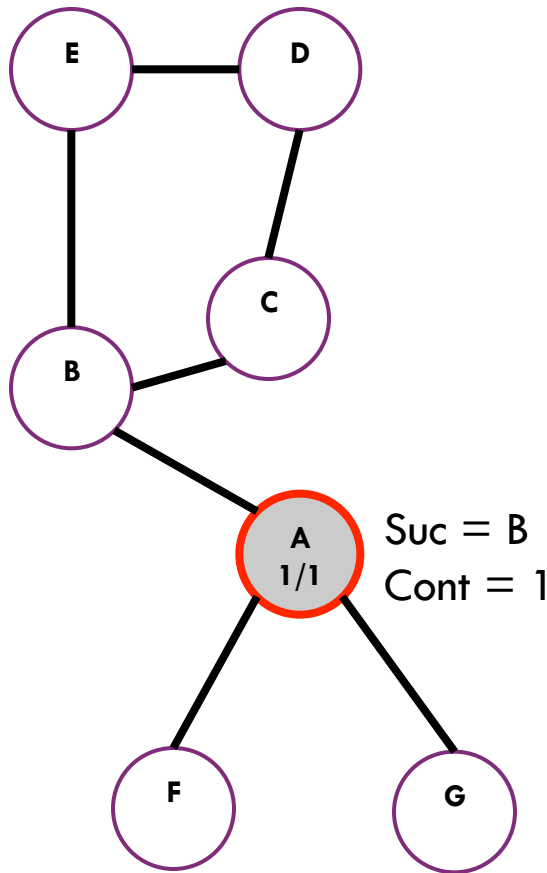
Algoritmo de Tarjan - BP

```
tempo = tempo + 1;  
d[u] = m[u] = tempo;  
cor[u] = cinza;  
para cada v ∈ Adj(u) faça  
    se cor[v] == branco então  
        sucessor[u] = v;  
        se u == r então  
            contador = contador + 1;  
        TARJAN-BP(G, v, r );  
        se m[v] ≥ d[u] então  
            ponto[u] = verdadeiro;  
        se m[v] > d[u] então  
            ponte[u, v] = verdadeiro;  
        m[u] = min(m[u], m[v]);  
    senão se v != sucessor[u] então  
        m[u] = min(m[u], d[v]);  
cor[u] = preto;
```



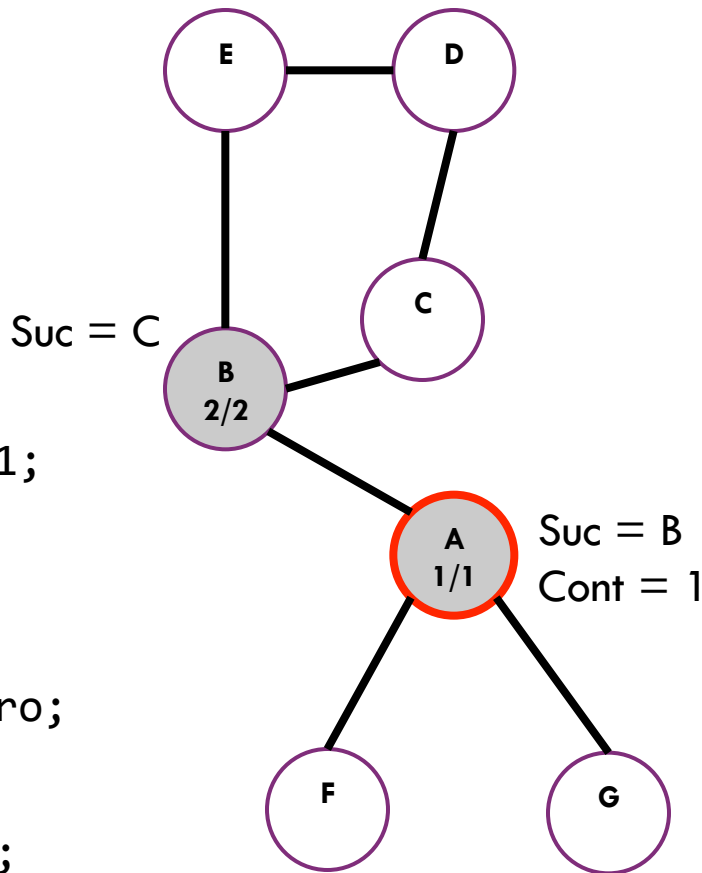
Algoritmo de Tarjan - BP

```
tempo = tempo + 1;  
d[u] = m[u] = tempo;  
cor[u] = cinza;  
para cada v ∈ Adj(u) faça  
    se cor[v] == branco então  
        sucessor[u] = v;  
        se u == r então  
            contador = contador + 1;  
        TARJAN-BP(G, v, r );  
        se m[v] ≥ d[u] então  
            ponto[u] = verdadeiro;  
        se m[v] > d[u] então  
            ponte[u, v] = verdadeiro;  
        m[u] = min(m[u], m[v]);  
    senão se v != sucessor[u] então  
        m[u] = min(m[u], d[v]);  
cor[u] = preto;
```



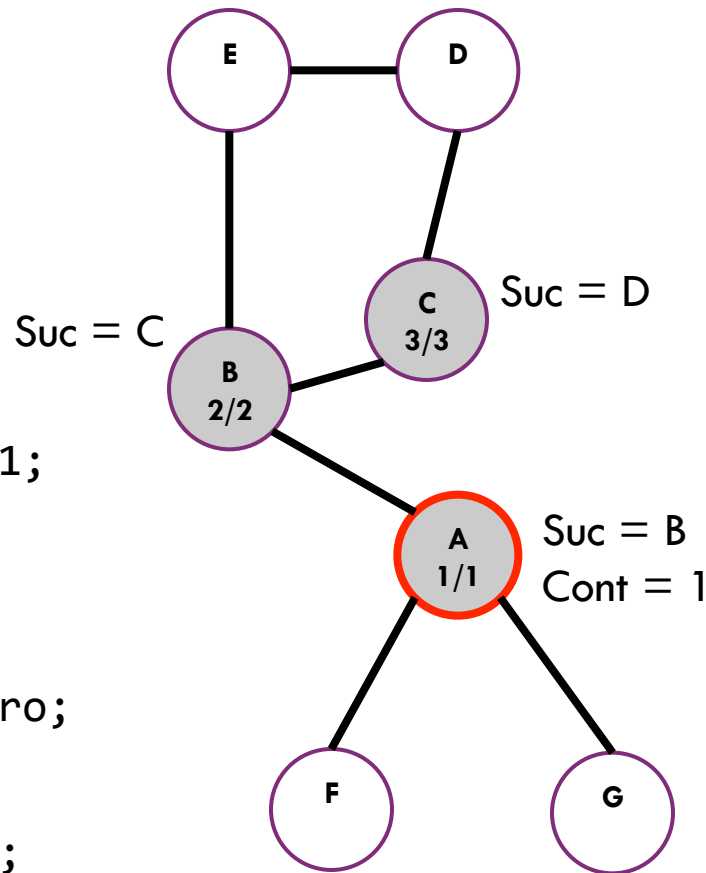
Algoritmo de Tarjan - BP

```
tempo = tempo + 1;  
d[u] = m[u] = tempo;  
cor[u] = cinza;  
para cada v ∈ Adj(u) faça  
    se cor[v] == branco então  
        sucessor[u] = v;  
        se u == r então  
            contador = contador + 1;  
            TARJAN-BP(G, v, r );  
        se m[v] ≥ d[u] então  
            ponto[u] = verdadeiro;  
        se m[v] > d[u] então  
            ponte[u, v] = verdadeiro;  
        m[u] = min(m[u], m[v]);  
    senão se v != sucessor[u] então  
        m[u] = min(m[u], d[v]);  
cor[u] = preto;
```



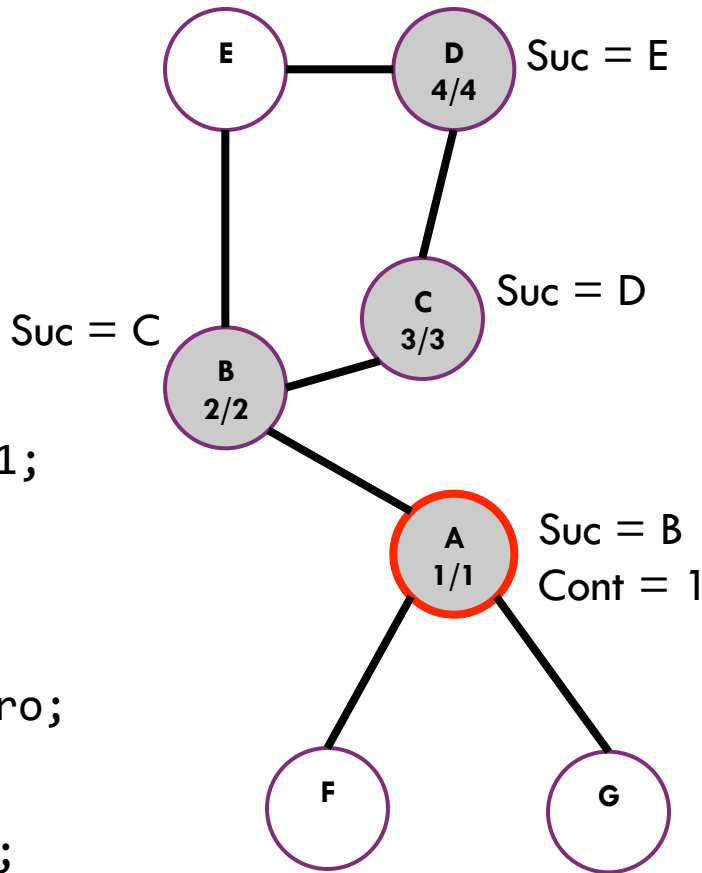
Algoritmo de Tarjan - BP

```
tempo = tempo + 1;  
d[u] = m[u] = tempo;  
cor[u] = cinza;  
para cada v ∈ Adj(u) faça  
    se cor[v] == branco então  
        sucessor[u] = v;  
        se u == r então  
            contador = contador + 1;  
            TARJAN-BP(G, v, r );  
        se m[v] ≥ d[u] então  
            ponto[u] = verdadeiro;  
        se m[v] > d[u] então  
            ponte[u, v] = verdadeiro;  
        m[u] = min(m[u], m[v]);  
    senão se v != sucessor[u] então  
        m[u] = min(m[u], d[v]);  
cor[u] = preto;
```



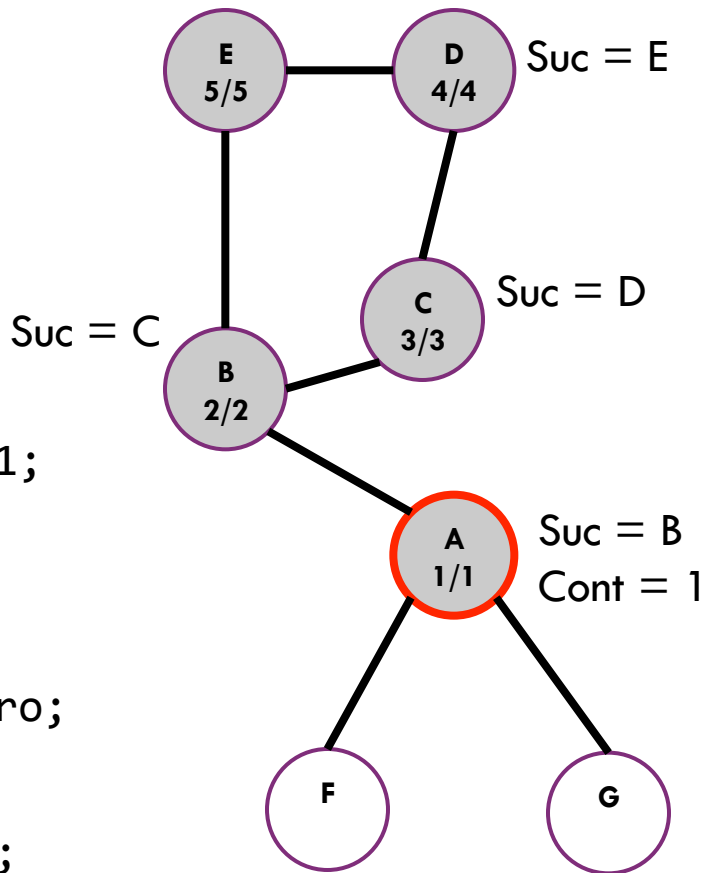
Algoritmo de Tarjan - BP

```
tempo = tempo + 1;  
d[u] = m[u] = tempo;  
cor[u] = cinza;  
para cada v ∈ Adj(u) faça  
    se cor[v] == branco então  
        sucessor[u] = v;  
        se u == r então  
            contador = contador + 1;  
            TARJAN-BP(G, v, r );  
        se m[v] ≥ d[u] então  
            ponto[u] = verdadeiro;  
        se m[v] > d[u] então  
            ponte[u, v] = verdadeiro;  
        m[u] = min(m[u], m[v]);  
    senão se v != sucessor[u] então  
        m[u] = min(m[u], d[v]);  
cor[u] = preto;
```



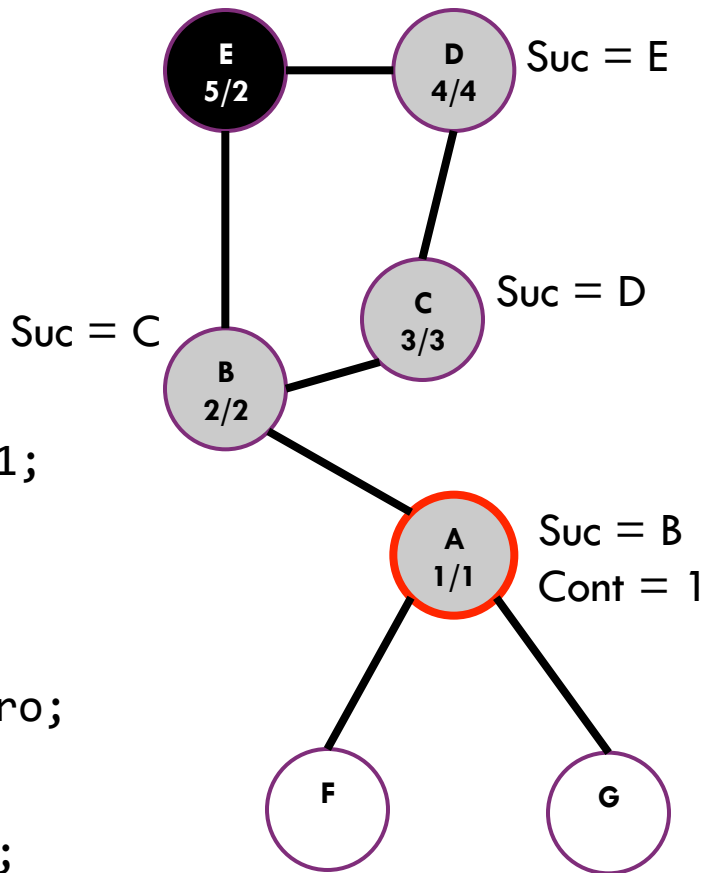
Algoritmo de Tarjan - BP

```
tempo = tempo + 1;  
d[u] = m[u] = tempo;  
cor[u] = cinza;  
para cada v ∈ Adj(u) faça  
    se cor[v] == branco então  
        sucessor[u] = v;  
        se u == r então  
            contador = contador + 1;  
        TARJAN-BP(G, v, r );  
        se m[v] ≥ d[u] então  
            ponto[u] = verdadeiro;  
        se m[v] > d[u] então  
            ponte[u, v] = verdadeiro;  
        m[u] = min(m[u], m[v]);  
    senão se v != sucessor[u] então  
        m[u] = min(m[u], d[v]);  
cor[u] = preto;
```



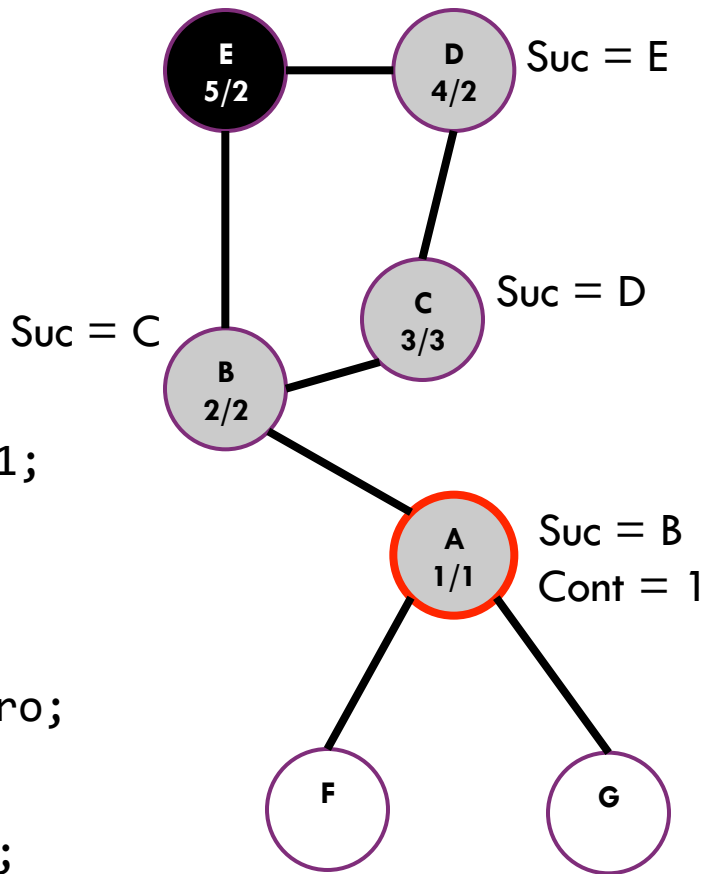
Algoritmo de Tarjan - BP

```
tempo = tempo + 1;  
d[u] = m[u] = tempo;  
cor[u] = cinza;  
para cada v ∈ Adj(u) faça  
    se cor[v] == branco então  
        sucessor[u] = v;  
        se u == r então  
            contador = contador + 1;  
        TARJAN-BP(G, v, r );  
        se m[v] ≥ d[u] então  
            ponto[u] = verdadeiro;  
        se m[v] > d[u] então  
            ponte[u, v] = verdadeiro;  
        m[u] = min(m[u], m[v]);  
    senão se v != sucessor[u] então  
        m[u] = min(m[u], d[v]);  
cor[u] = preto;
```



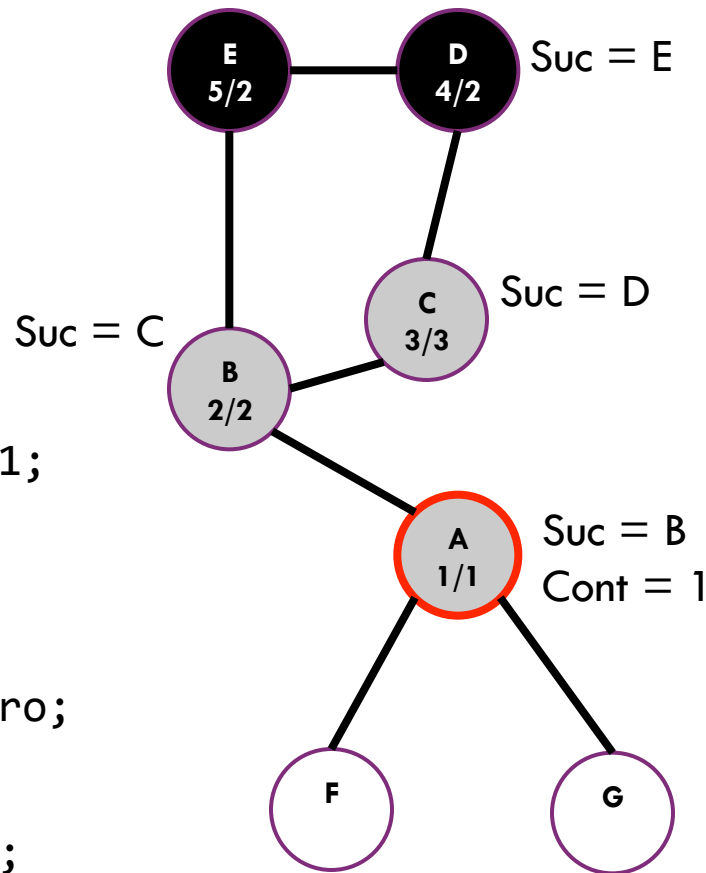
Algoritmo de Tarjan - BP

```
tempo = tempo + 1;  
d[u] = m[u] = tempo;  
cor[u] = cinza;  
para cada v ∈ Adj(u) faça  
    se cor[v] == branco então  
        sucessor[u] = v;  
        se u == r então  
            contador = contador + 1;  
        TARJAN-BP(G, v, r );  
        se m[v] ≥ d[u] então  
            ponto[u] = verdadeiro;  
        se m[v] > d[u] então  
            ponte[u, v] = verdadeiro;  
        m[u] = min(m[u], m[v]);  
    senão se v != sucessor[u] então  
        m[u] = min(m[u], d[v]);  
cor[u] = preto;
```



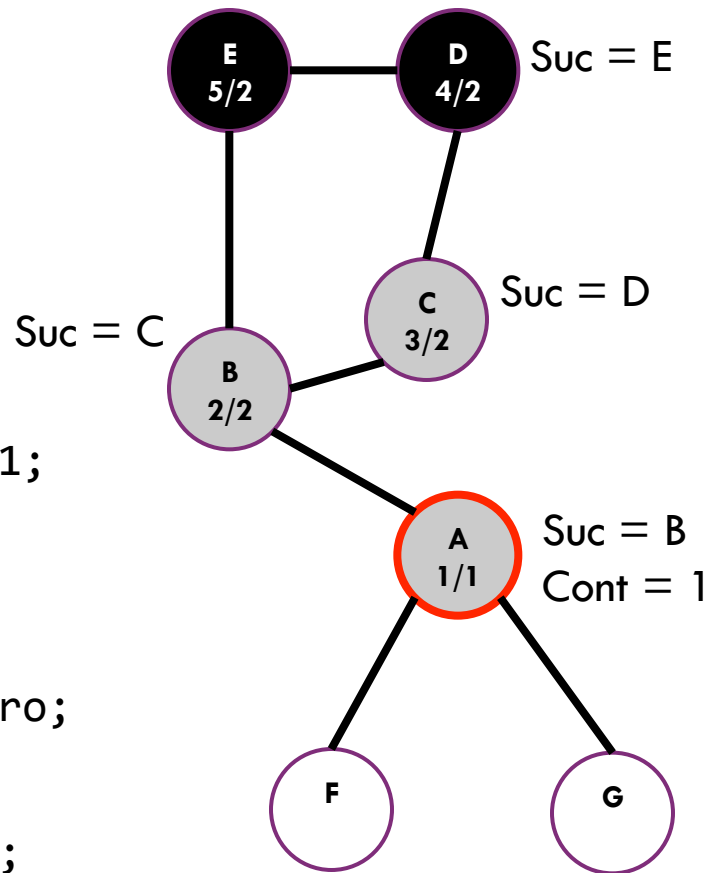
Algoritmo de Tarjan - BP

```
tempo = tempo + 1;  
d[u] = m[u] = tempo;  
cor[u] = cinza;  
para cada v ∈ Adj(u) faça  
    se cor[v] == branco então  
        sucessor[u] = v;  
        se u == r então  
            contador = contador + 1;  
            TARJAN-BP(G, v, r );  
        se m[v] ≥ d[u] então  
            ponto[u] = verdadeiro;  
        se m[v] > d[u] então  
            ponte[u, v] = verdadeiro;  
        m[u] = min(m[u], m[v]);  
    senão se v != sucessor[u] então  
        m[u] = min(m[u], d[v]);  
cor[u] = preto;
```



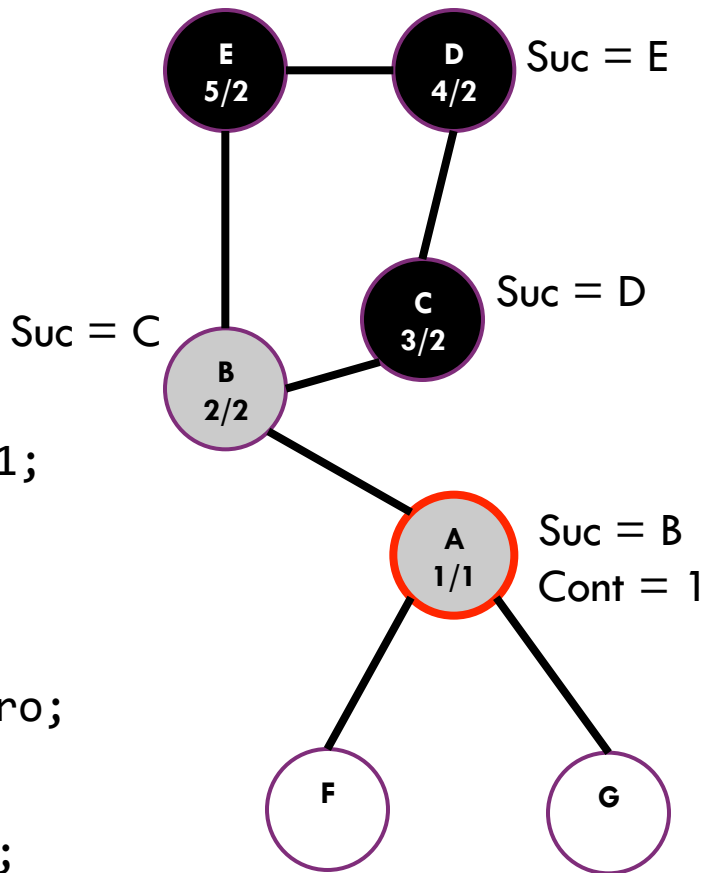
Algoritmo de Tarjan - BP

```
tempo = tempo + 1;  
d[u] = m[u] = tempo;  
cor[u] = cinza;  
para cada v ∈ Adj(u) faça  
    se cor[v] == branco então  
        sucessor[u] = v;  
        se u == r então  
            contador = contador + 1;  
            TARJAN-BP(G, v, r );  
        se m[v] ≥ d[u] então  
            ponto[u] = verdadeiro;  
        se m[v] > d[u] então  
            ponte[u, v] = verdadeiro;  
        m[u] = min(m[u], m[v]);  
    senão se v != sucessor[u] então  
        m[u] = min(m[u], d[v]);  
cor[u] = preto;
```



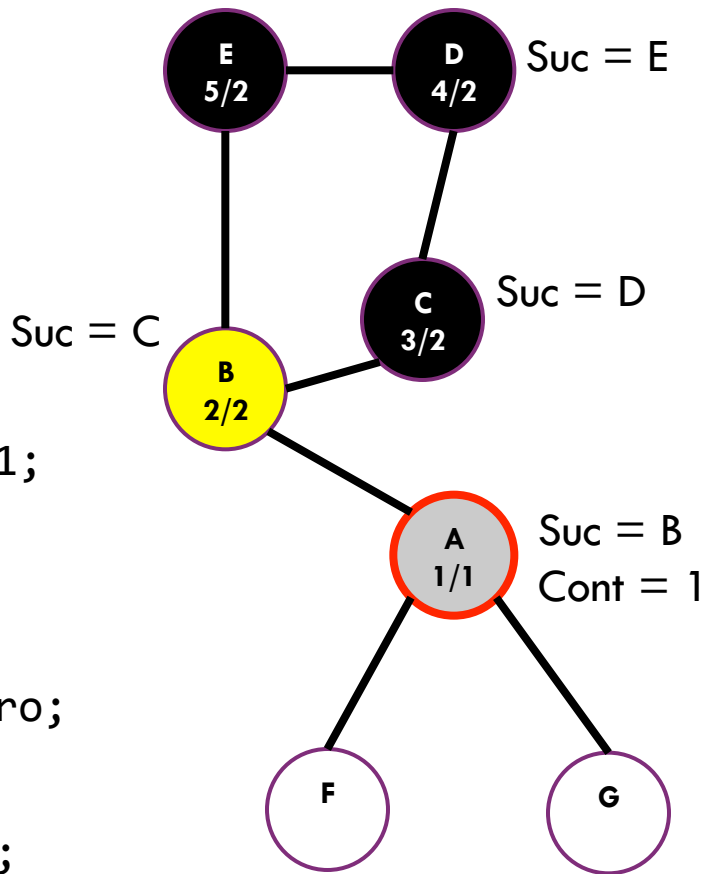
Algoritmo de Tarjan - BP

```
tempo = tempo + 1;  
d[u] = m[u] = tempo;  
cor[u] = cinza;  
para cada v ∈ Adj(u) faça  
    se cor[v] == branco então  
        sucessor[u] = v;  
        se u == r então  
            contador = contador + 1;  
        TARJAN-BP(G, v, r );  
        se m[v] ≥ d[u] então  
            ponto[u] = verdadeiro;  
        se m[v] > d[u] então  
            ponte[u, v] = verdadeiro;  
        m[u] = min(m[u], m[v]);  
    senão se v != sucessor[u] então  
        m[u] = min(m[u], d[v]);  
cor[u] = preto;
```



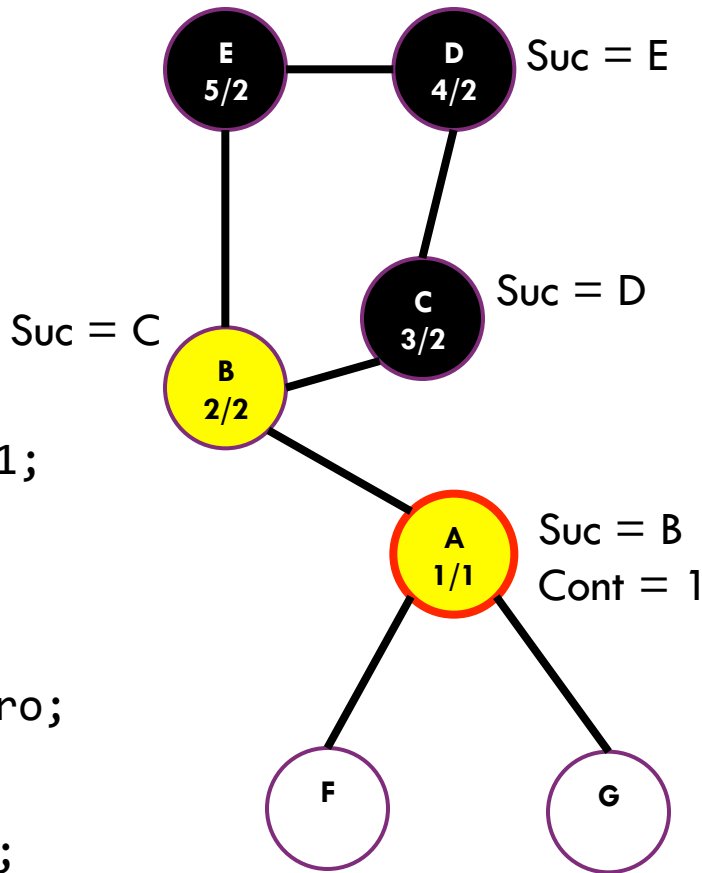
Algoritmo de Tarjan - BP

```
tempo = tempo + 1;  
d[u] = m[u] = tempo;  
cor[u] = cinza;  
para cada v ∈ Adj(u) faça  
    se cor[v] == branco então  
        sucessor[u] = v;  
        se u == r então  
            contador = contador + 1;  
            TARJAN-BP(G, v, r );  
        se m[v] ≥ d[u] então  
            ponto[u] = verdadeiro;  
        se m[v] > d[u] então  
            ponte[u, v] = verdadeiro;  
        m[u] = min(m[u], m[v]);  
    senão se v != sucessor[u] então  
        m[u] = min(m[u], d[v]);  
cor[u] = preto;
```



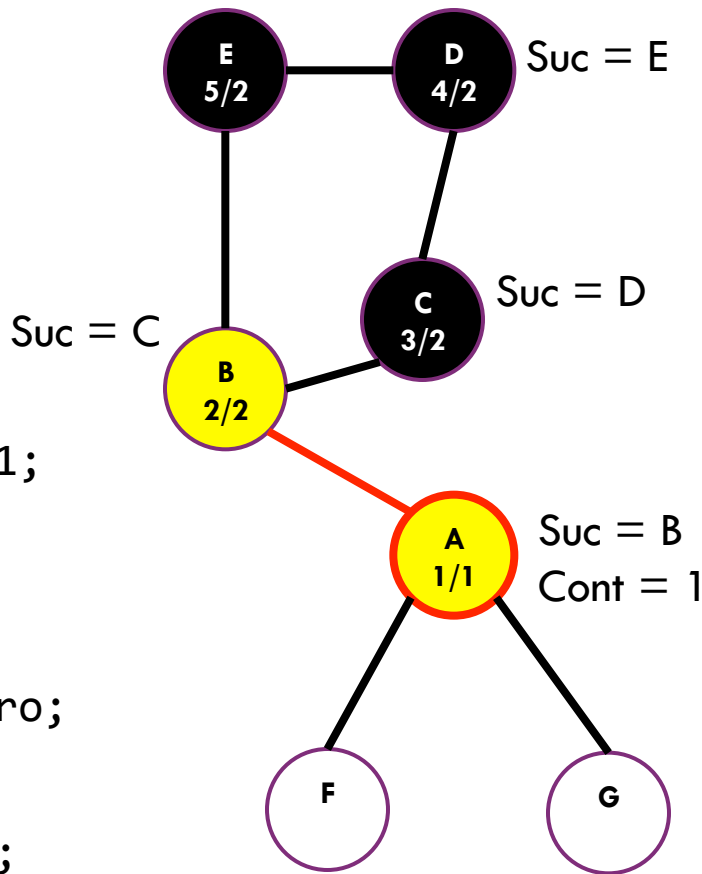
Algoritmo de Tarjan - BP

```
tempo = tempo + 1;  
d[u] = m[u] = tempo;  
cor[u] = cinza;  
para cada v ∈ Adj(u) faça  
    se cor[v] == branco então  
        sucessor[u] = v;  
        se u == r então  
            contador = contador + 1;  
            TARJAN-BP(G, v, r );  
        se m[v] ≥ d[u] então  
            ponto[u] = verdadeiro;  
        se m[v] > d[u] então  
            ponte[u, v] = verdadeiro;  
        m[u] = min(m[u], m[v]);  
    senão se v != sucessor[u] então  
        m[u] = min(m[u], d[v]);  
cor[u] = preto;
```



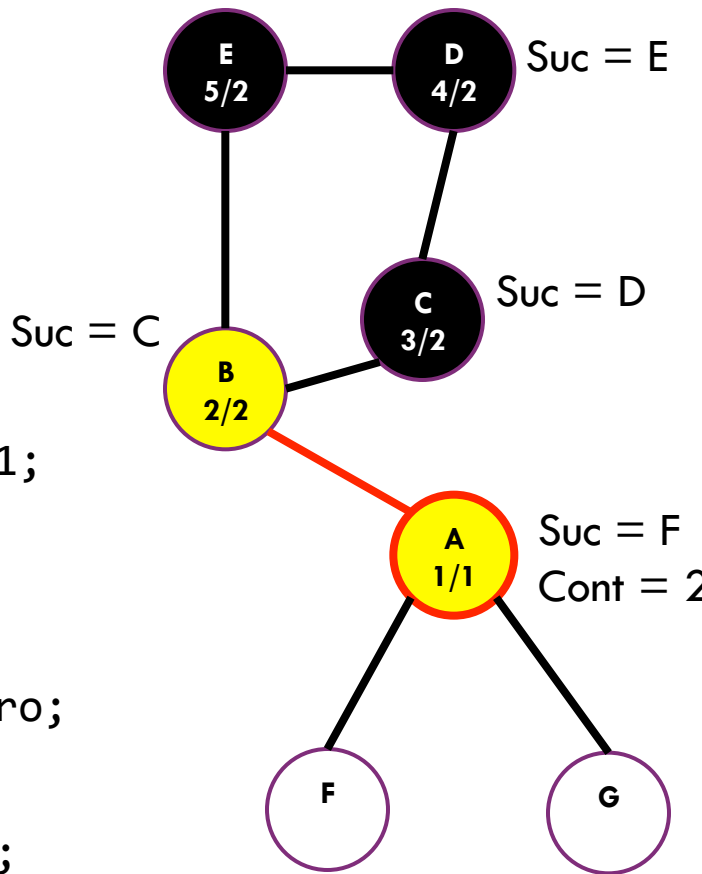
Algoritmo de Tarjan - BP

```
tempo = tempo + 1;  
d[u] = m[u] = tempo;  
cor[u] = cinza;  
para cada v ∈ Adj(u) faça  
    se cor[v] == branco então  
        sucessor[u] = v;  
        se u == r então  
            contador = contador + 1;  
            TARJAN-BP(G, v, r );  
        se m[v] ≥ d[u] então  
            ponto[u] = verdadeiro;  
        se m[v] > d[u] então  
            ponte[u, v] = verdadeiro;  
        m[u] = min(m[u], m[v]);  
    senão se v != sucessor[u] então  
        m[u] = min(m[u], d[v]);  
cor[u] = preto;
```



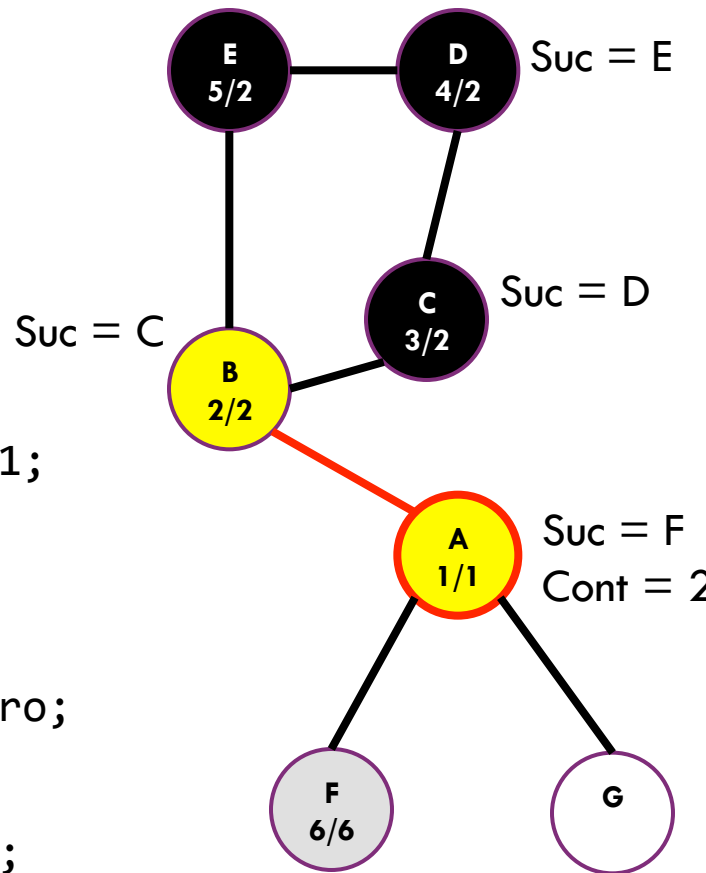
Algoritmo de Tarjan - BP

```
tempo = tempo + 1;  
d[u] = m[u] = tempo;  
cor[u] = cinza;  
para cada v ∈ Adj(u) faça  
    se cor[v] == branco então  
        sucessor[u] = v;  
        se u == r então  
            contador = contador + 1;  
            TARJAN-BP(G, v, r );  
        se m[v] ≥ d[u] então  
            ponto[u] = verdadeiro;  
        se m[v] > d[u] então  
            ponte[u, v] = verdadeiro;  
        m[u] = min(m[u], m[v]);  
    senão se v != sucessor[u] então  
        m[u] = min(m[u], d[v]);  
cor[u] = preto;
```



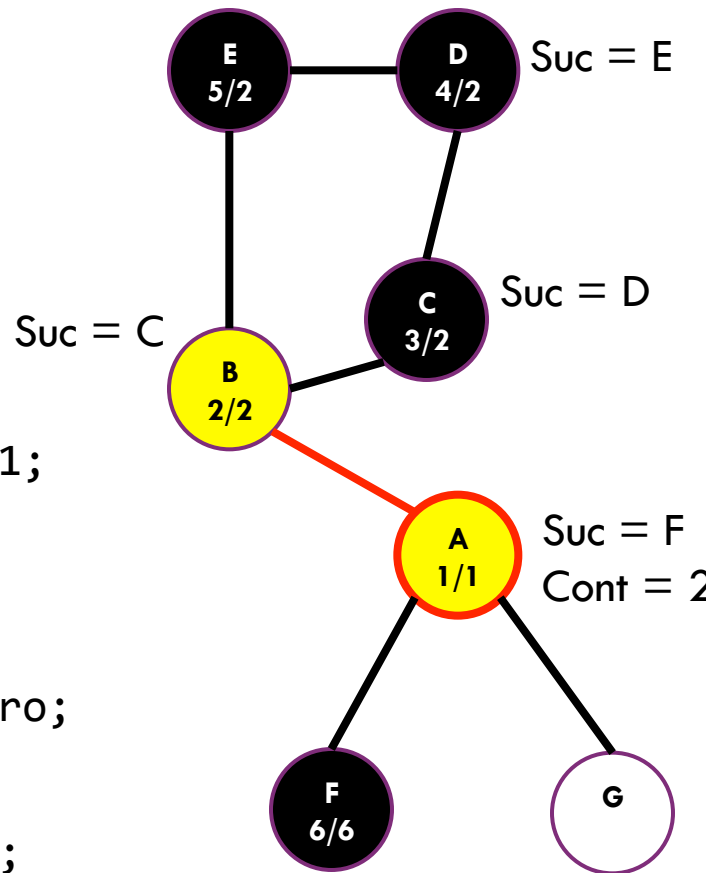
Algoritmo de Tarjan - BP

```
tempo = tempo + 1;  
d[u] = m[u] = tempo;  
cor[u] = cinza;  
para cada v ∈ Adj(u) faça  
    se cor[v] == branco então  
        sucessor[u] = v;  
        se u == r então  
            contador = contador + 1;  
            TARJAN-BP(G, v, r );  
        se m[v] ≥ d[u] então  
            ponto[u] = verdadeiro;  
        se m[v] > d[u] então  
            ponte[u, v] = verdadeiro;  
        m[u] = min(m[u], m[v]);  
    senão se v != sucessor[u] então  
        m[u] = min(m[u], d[v]);  
cor[u] = preto;
```



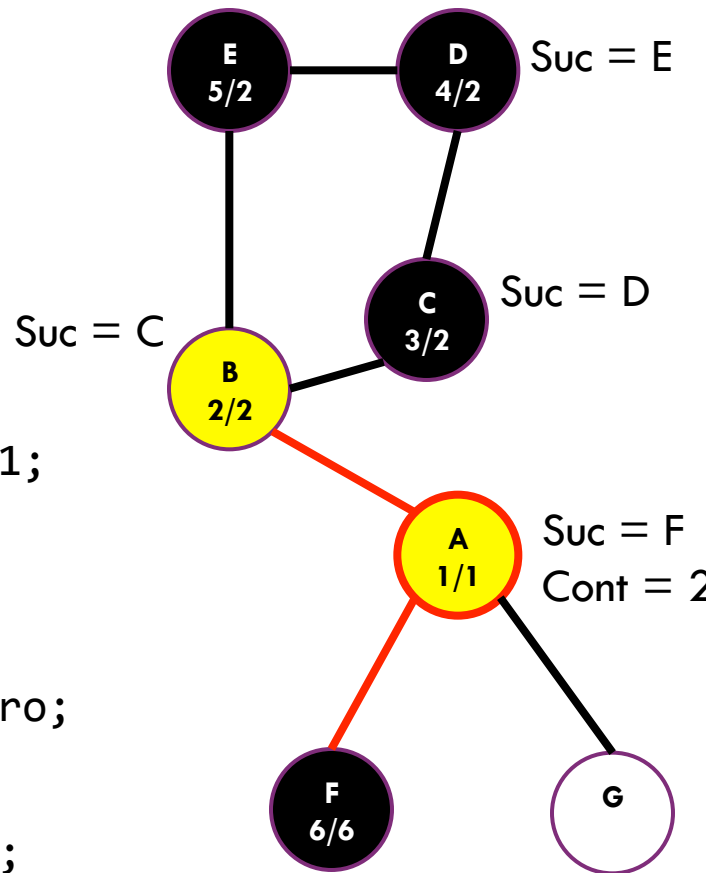
Algoritmo de Tarjan - BP

```
tempo = tempo + 1;  
d[u] = m[u] = tempo;  
cor[u] = cinza;  
para cada v ∈ Adj(u) faça  
    se cor[v] == branco então  
        sucessor[u] = v;  
        se u == r então  
            contador = contador + 1;  
        TARJAN-BP(G, v, r );  
        se m[v] ≥ d[u] então  
            ponto[u] = verdadeiro;  
        se m[v] > d[u] então  
            ponte[u, v] = verdadeiro;  
        m[u] = min(m[u], m[v]);  
    senão se v != sucessor[u] então  
        m[u] = min(m[u], d[v]);  
cor[u] = preto;
```



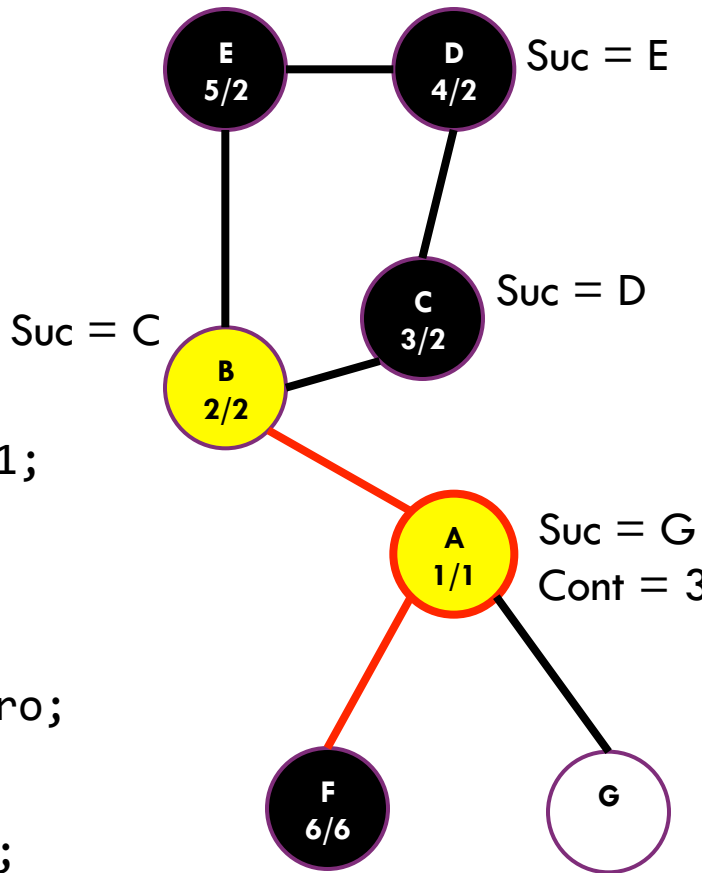
Algoritmo de Tarjan - BP

```
tempo = tempo + 1;  
d[u] = m[u] = tempo;  
cor[u] = cinza;  
para cada v ∈ Adj(u) faça  
    se cor[v] == branco então  
        sucessor[u] = v;  
        se u == r então  
            contador = contador + 1;  
            TARJAN-BP(G, v, r );  
        se m[v] ≥ d[u] então  
            ponto[u] = verdadeiro;  
        se m[v] > d[u] então  
            ponte[u, v] = verdadeiro;  
        m[u] = min(m[u], m[v]);  
    senão se v != sucessor[u] então  
        m[u] = min(m[u], d[v]);  
cor[u] = preto;
```



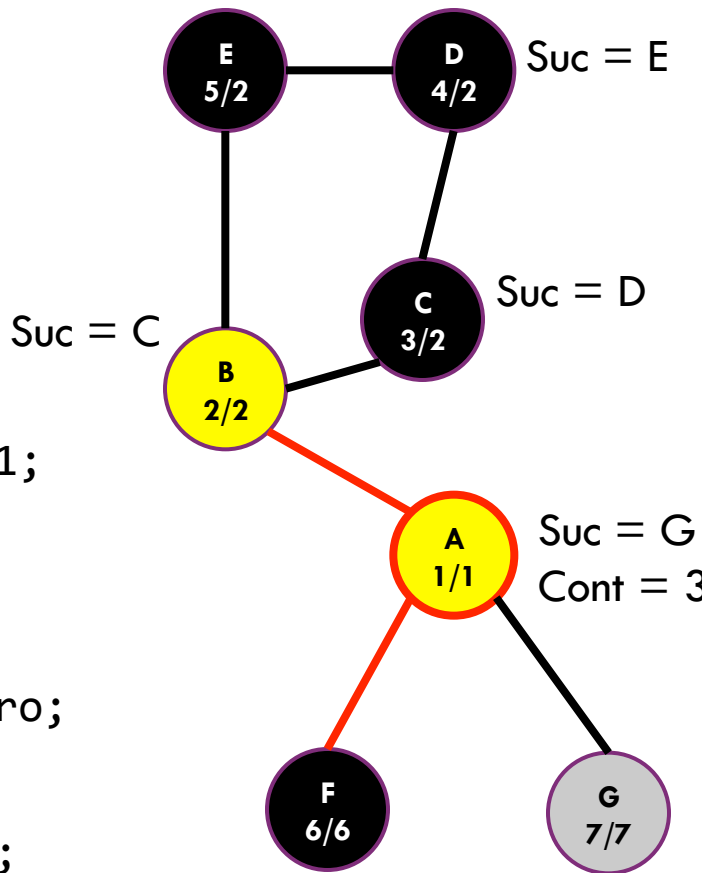
Algoritmo de Tarjan - BP

```
tempo = tempo + 1;  
d[u] = m[u] = tempo;  
cor[u] = cinza;  
para cada v ∈ Adj(u) faça  
    se cor[v] == branco então  
        sucessor[u] = v;  
        se u == r então  
            contador = contador + 1;  
        TARJAN-BP(G, v, r );  
        se m[v] ≥ d[u] então  
            ponto[u] = verdadeiro;  
        se m[v] > d[u] então  
            ponte[u, v] = verdadeiro;  
        m[u] = min(m[u], m[v]);  
    senão se v != sucessor[u] então  
        m[u] = min(m[u], d[v]);  
cor[u] = preto;
```



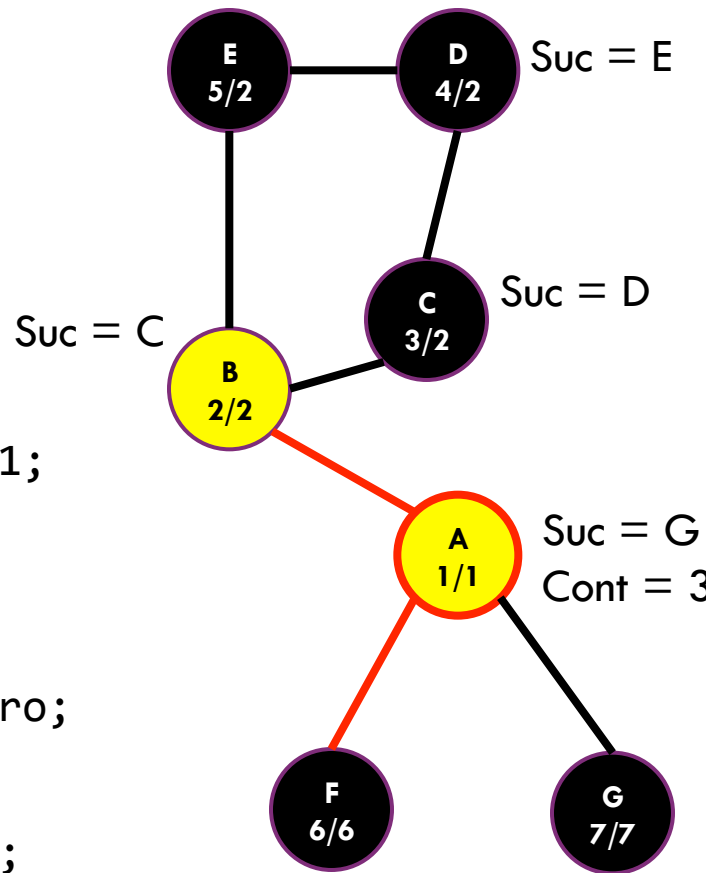
Algoritmo de Tarjan - BP

```
tempo = tempo + 1;  
d[u] = m[u] = tempo;  
cor[u] = cinza;  
para cada  $v \in \text{Adj}(u)$  faça  
    se  $\text{cor}[v] == \text{branco}$  então  
        sucessor[u] = v;  
        se  $u == r$  então  
            contador = contador + 1;  
            TARJAN-BP(G, v, r );  
        se  $m[v] \geq d[u]$  então  
            ponto[u] = verdadeiro;  
        se  $m[v] > d[u]$  então  
            ponte[u, v] = verdadeiro;  
         $m[u] = \min(m[u], m[v]);$   
    senão se  $v \neq \text{sucessor}[u]$  então  
         $m[u] = \min(m[u], d[v]);$   
  
 $\text{cor}[u] = \text{preto};$ 
```



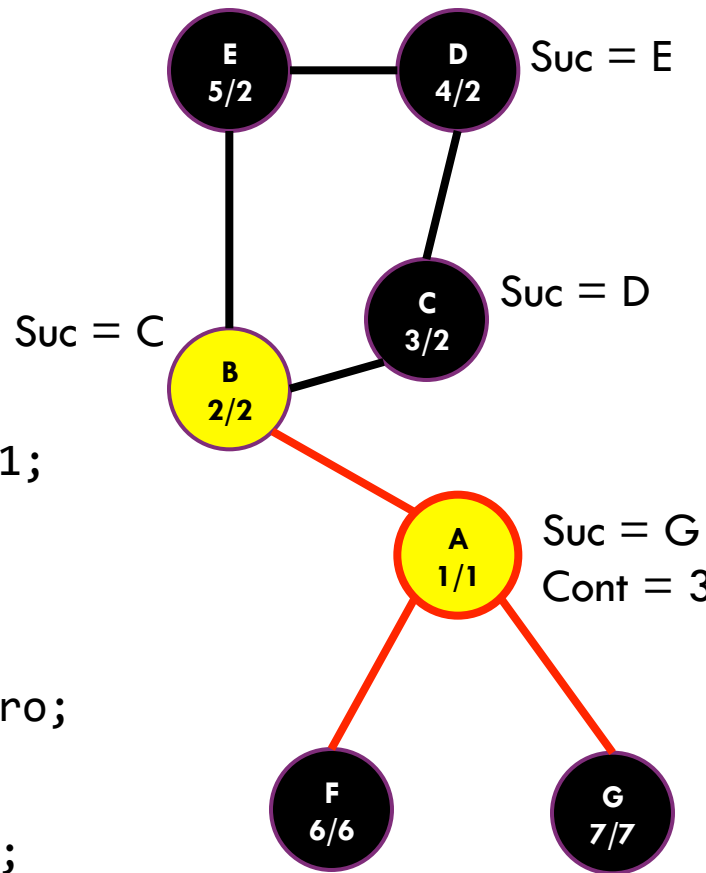
Algoritmo de Tarjan - BP

```
tempo = tempo + 1;  
d[u] = m[u] = tempo;  
cor[u] = cinza;  
para cada v ∈ Adj(u) faça  
    se cor[v] == branco então  
        sucessor[u] = v;  
        se u == r então  
            contador = contador + 1;  
            TARJAN-BP(G, v, r );  
        se m[v] ≥ d[u] então  
            ponto[u] = verdadeiro;  
        se m[v] > d[u] então  
            ponte[u, v] = verdadeiro;  
        m[u] = min(m[u], m[v]);  
    senão se v != sucessor[u] então  
        m[u] = min(m[u], d[v]);  
cor[u] = preto;
```



Algoritmo de Tarjan - BP

```
tempo = tempo + 1;  
d[u] = m[u] = tempo;  
cor[u] = cinza;  
para cada v ∈ Adj(u) faça  
    se cor[v] == branco então  
        sucessor[u] = v;  
        se u == r então  
            contador = contador + 1;  
        TARJAN-BP(G, v, r );  
        se m[v] ≥ d[u] então  
            ponto[u] = verdadeiro;  
        se m[v] > d[u] então  
            ponte[u, v] = verdadeiro;  
        m[u] = min(m[u], m[v]);  
    senão se v != sucessor[u] então  
        m[u] = min(m[u], d[v]);  
cor[u] = preto;
```



Algoritmo de Tarjan - principal

```
para cada  $u \in V$  faça  
    ponto[u] = falso; cor[u] = branco;  
para cada  $(u, v) \in E$  faça  
    ponte[u, v] = falso;  
tempo = 0;  
para cada  $u \in V$  faça  
    se cor[u] == branco então  
        contador = 0;  
        TARJAN-BP(G, u, u);  
    se contador  $\geq 2$  então  
        ponto[u] = verdadeiro;  
    senão  
        ponto[u] = falso;
```

