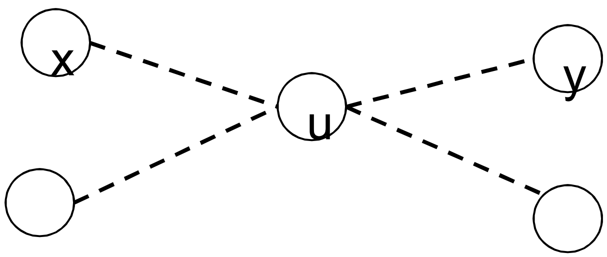
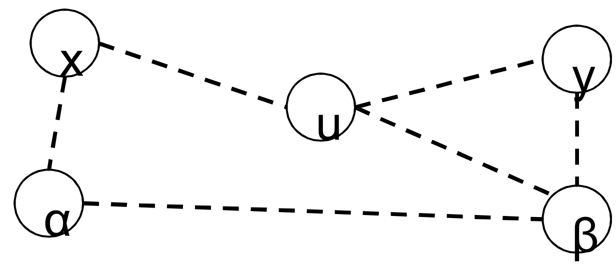
Puncte de articulație și punți

Puncte de articulație

- G=(N,A) graf neorientat, $u \in N$
- u este punct de articulație dacă $\exists$ $x,y \in N$, $x \neq y$, $x \neq u$, $y \neq u$, a.i. $\forall$ x- - ->y in G trece prin u



Orice drum x..y trece prin u=>u este punct de articulatie



Exista x..α..y care nu trece prin u; u nu mai este punct de articulatie

Teoremă

- G=(N,A), graf neorientat, $u \in N$; u este punct de articulație în G **<=>** în urma DFS în G una din proprietățile de mai jos este satisfacută
  1. u este rădăcină și u domină cel puțin 2 subarbori
  2. u nu este rădăcină și

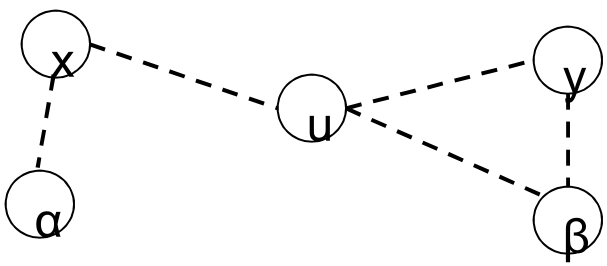
     $\exists$v descendent al lui u în Arb(u) a.i.

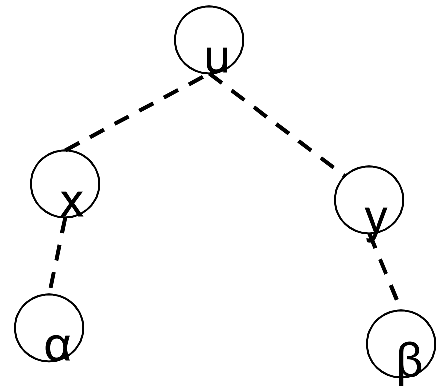
     $\forall x \in$Arb(v) si $\forall$(x,z) parcurs de DFS(G)

     debut(z)>=debut(u)

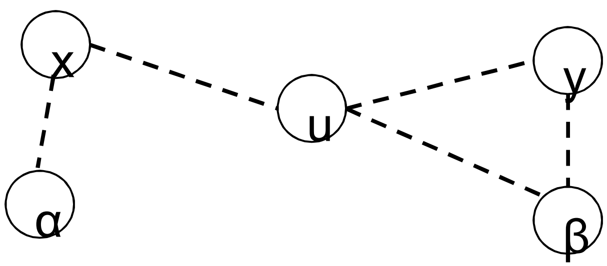
Exemplu

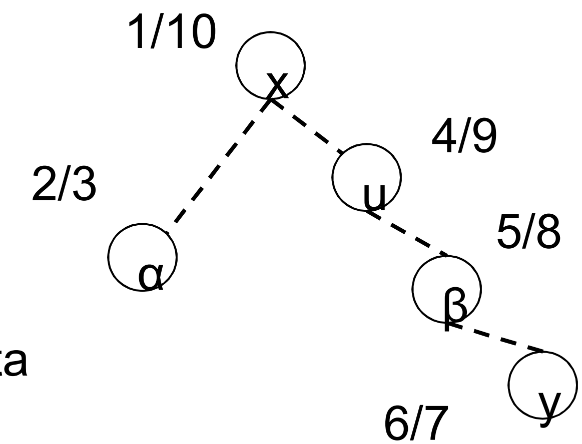
- p(u)=null si u domina cel putin 2 subarbori





- p(u)$\neq$null si $\exists$v descendent al lui u in Arb(u) a.i. $\forall x \in$Arb(v) si $\forall$(x,z) parcurs de DFS(G) debut(z)>=debut(u)



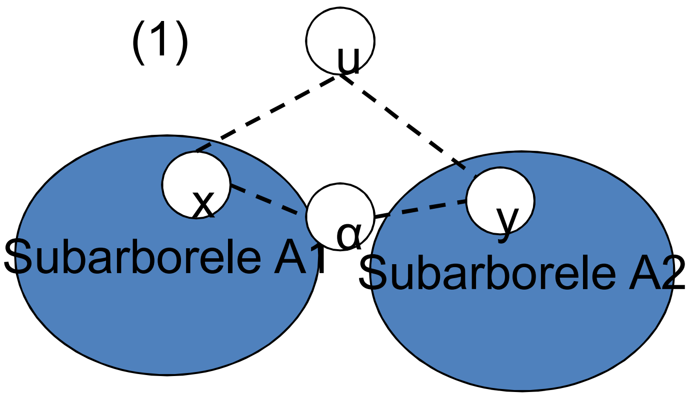


Pentru orice arc din subarborele lui β nu exista nici un arc inapoi spre un nod descoperit inaintea lui u

Puncte de articulatie. Demonstratie teorema

- p(u)=null si u domina cel putin 2 subarbori => u este punct de articulatie

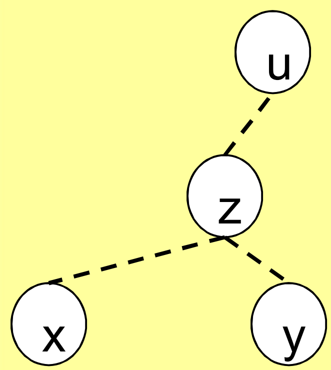
(1)



Pp ∃ x..α..y

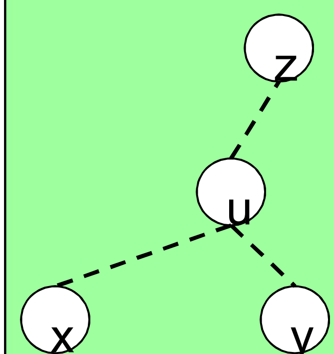
z= primul nod din x..α..y descoperit la DFS
Cf. T caii albe x,y∈Arb(z)

Caz 1. debut(u)<debut(z)



Contradictie (1) x,y nu sunt in subarbori diferiti ai lui Arb(u)

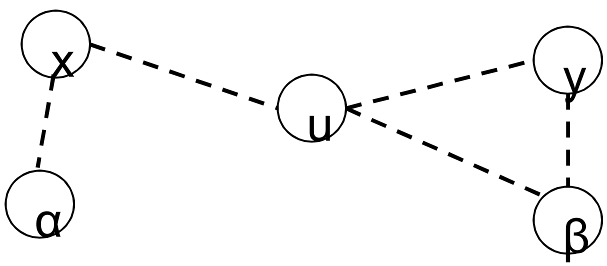
Caz 2. debut(z)<debut(u)



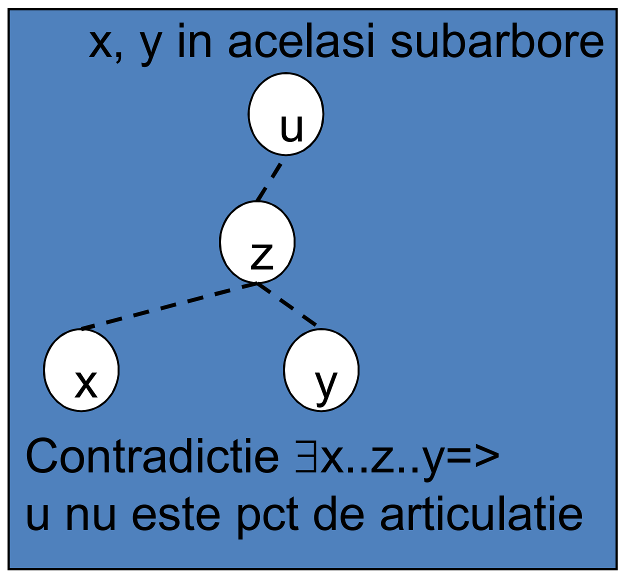
Contradictie (1), p(u)≠null

Puncte de articulatie. Demonstratie teorema

- u este punct de articulatie si este descoperit in ciclul principal al DFS =>p(u)=null si u domina cel putin 2 subarbori



u – primul descoperit din x..y
=>x,y $\in$Arb(u)



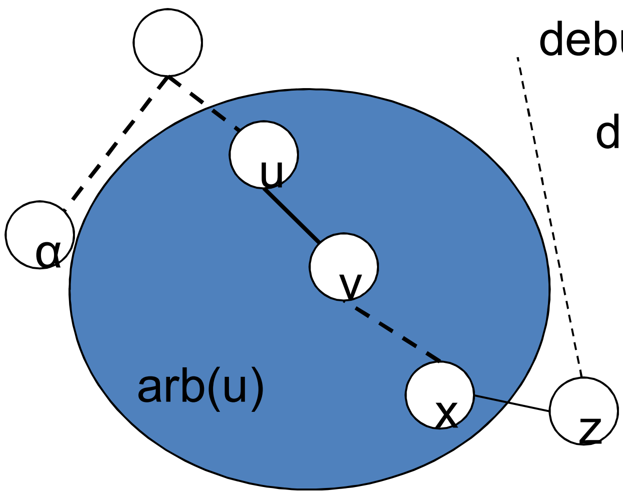
Puncte de articulatie. Demonstratie teorema

p(u)≠null si $\exists$v descendent al lui u in Arb(u) a.i.

$\forall$x$\in$Arb(v) si $\forall$(x,z) parcurs de DFS(G)

debut(z)>=debut(u) ➔ u este punct de articulatie

Dem. prin red. la absurd



debut(z)>debut(u)=>z$\in$Arb(u)=> contradictie (z$\notin$Arb(u))

debut(z)<debut(u)=>contradictie (ipoteza)

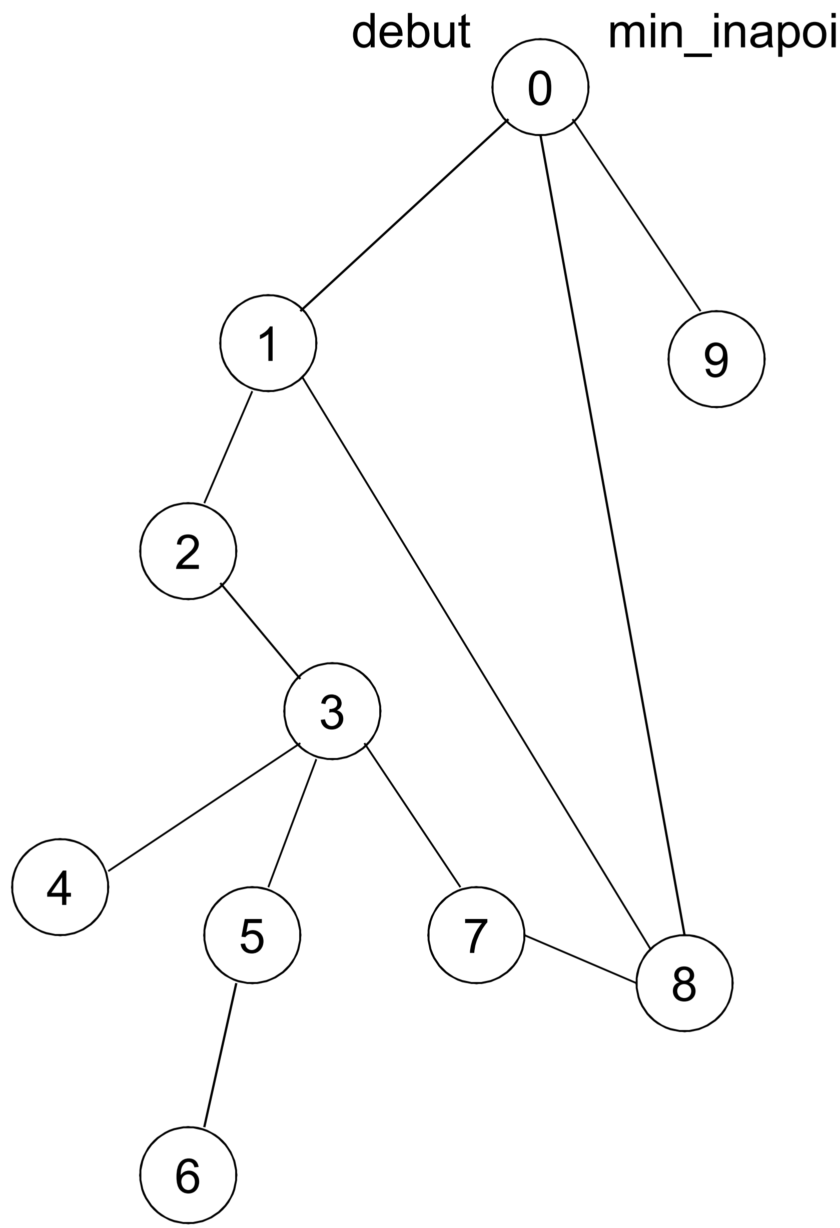
Algoritm

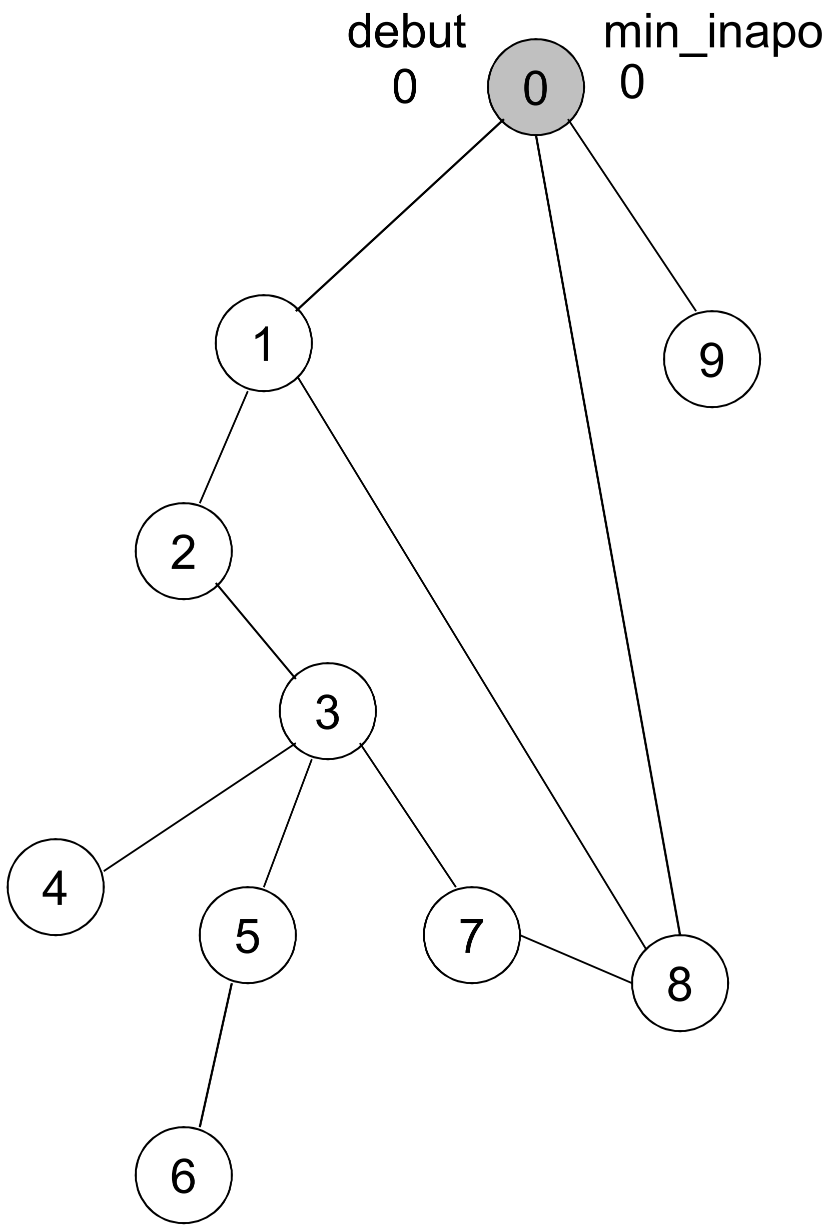
```
Algoritm Articulatii (G)
    Timp ← 0;
    Foreach (u∈V)
        debut[u] ← -1
        min_inapoi[u] ← 0
        p[u] ← null
        subarb[u] ← 0
        articulatii[u] ← 0
    Foreach (u∈V)
        If(debut[u]=-1)
            Exploreaza(u);
            If(subarb[u]>1)         // cazul in care u este radacina in arborele
             articulatii[u]=1       // DFS si are mai multi subarbori
```

Algoritm (II)

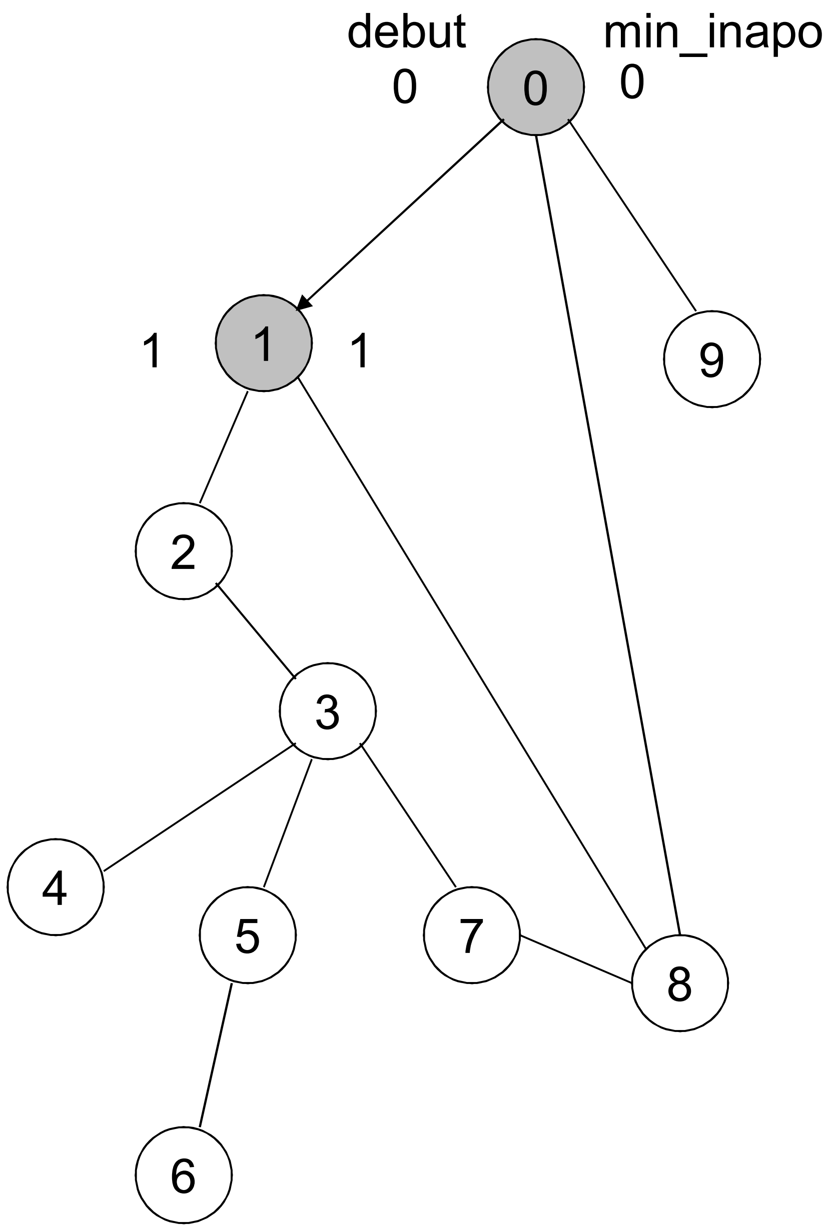
- Exploreaza(u)

```
debut[u] ← timp                                     // si devine nod gri
min_inapoi[u] ← timp
timp ← timp+1
foreach v succesor al lui u
    If (debut[v]=-1)                                // nod alb
        P[v] ← u
        Exploreaza(v)
        min_inapoi[u] ← min{min_inapoi[u],min_inapoi[v]}
        If(p[u]!=null and min_inapoi[v]>=debut[u])
            articulatii[u] ← 1                      // cazul 2 al teoremei
    else min_inapoi[u] ← min{min_inapoi[u], debut[v]}
```

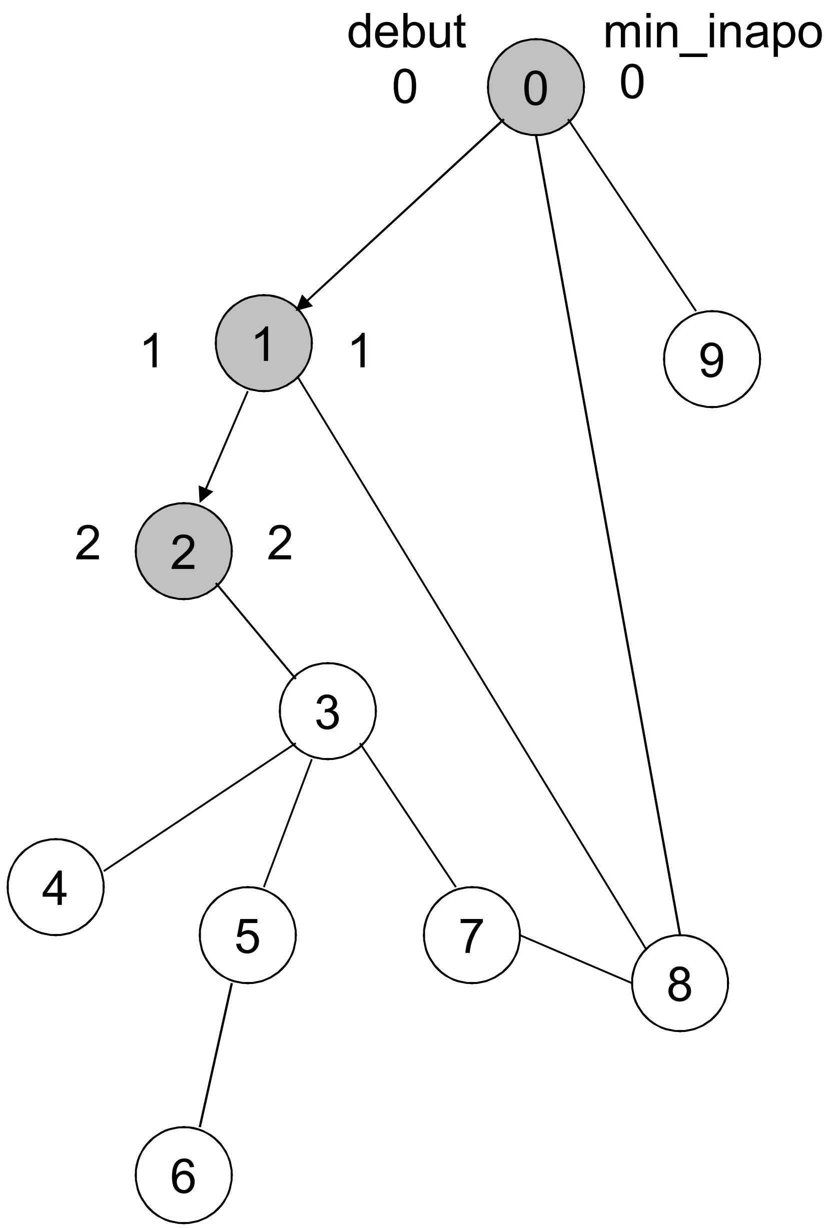

debut
min_inapoi
0
1
9
2
3
4
5
7
8
6



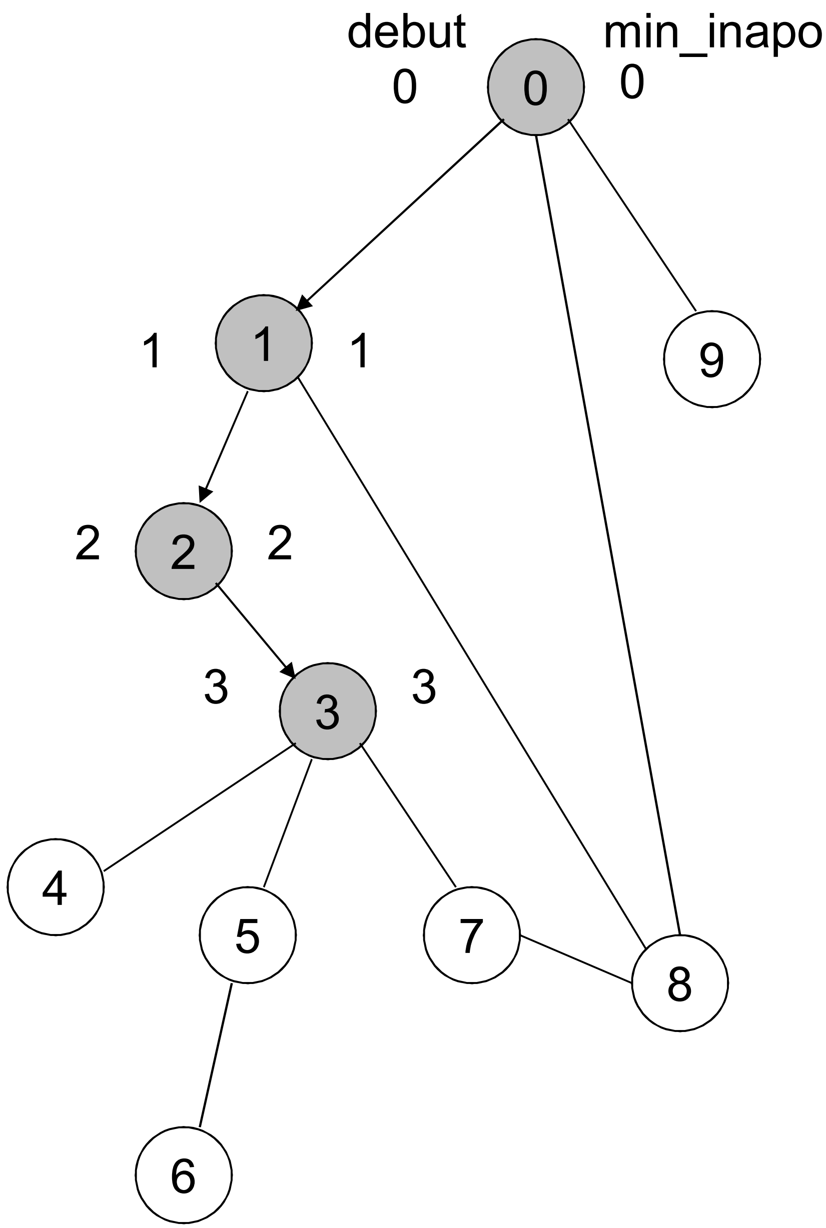
min_inapoi = debut



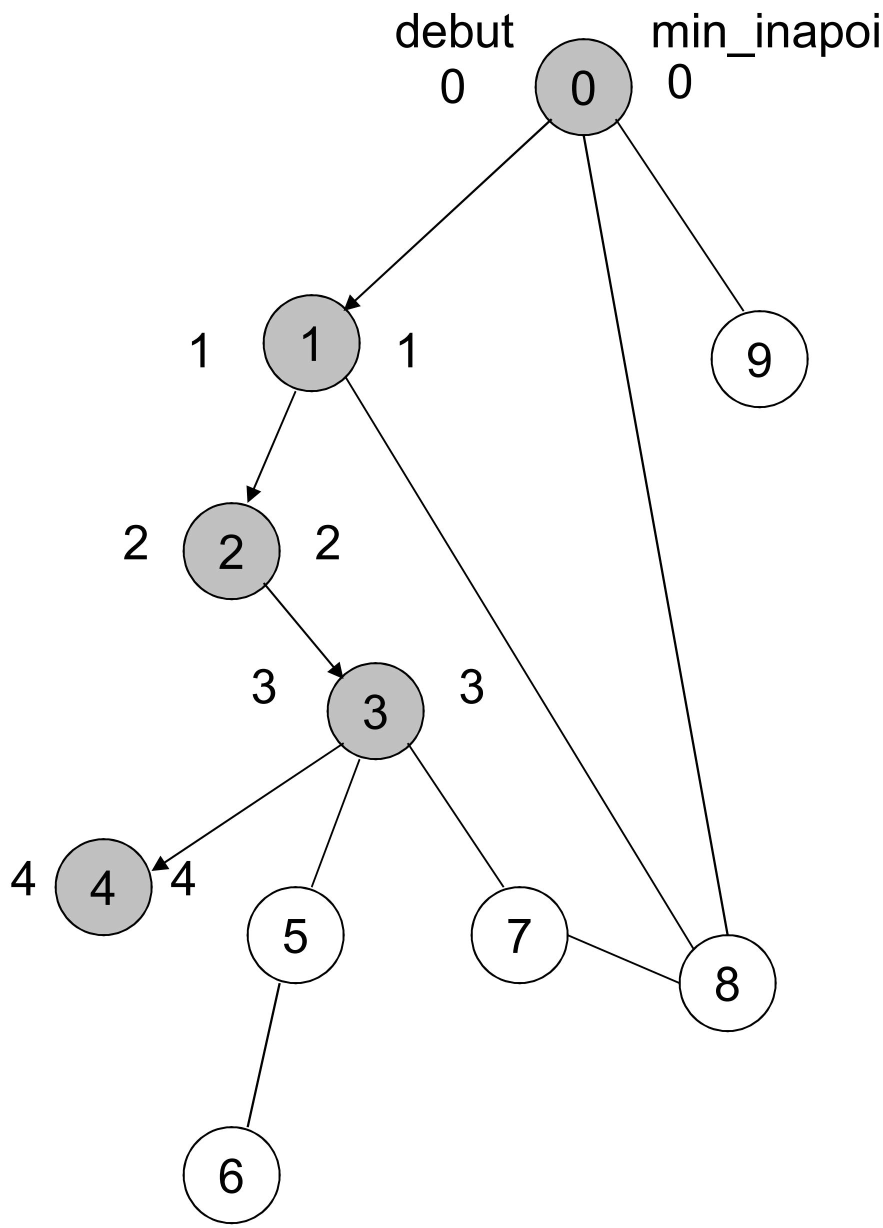
min_inapoi = debut



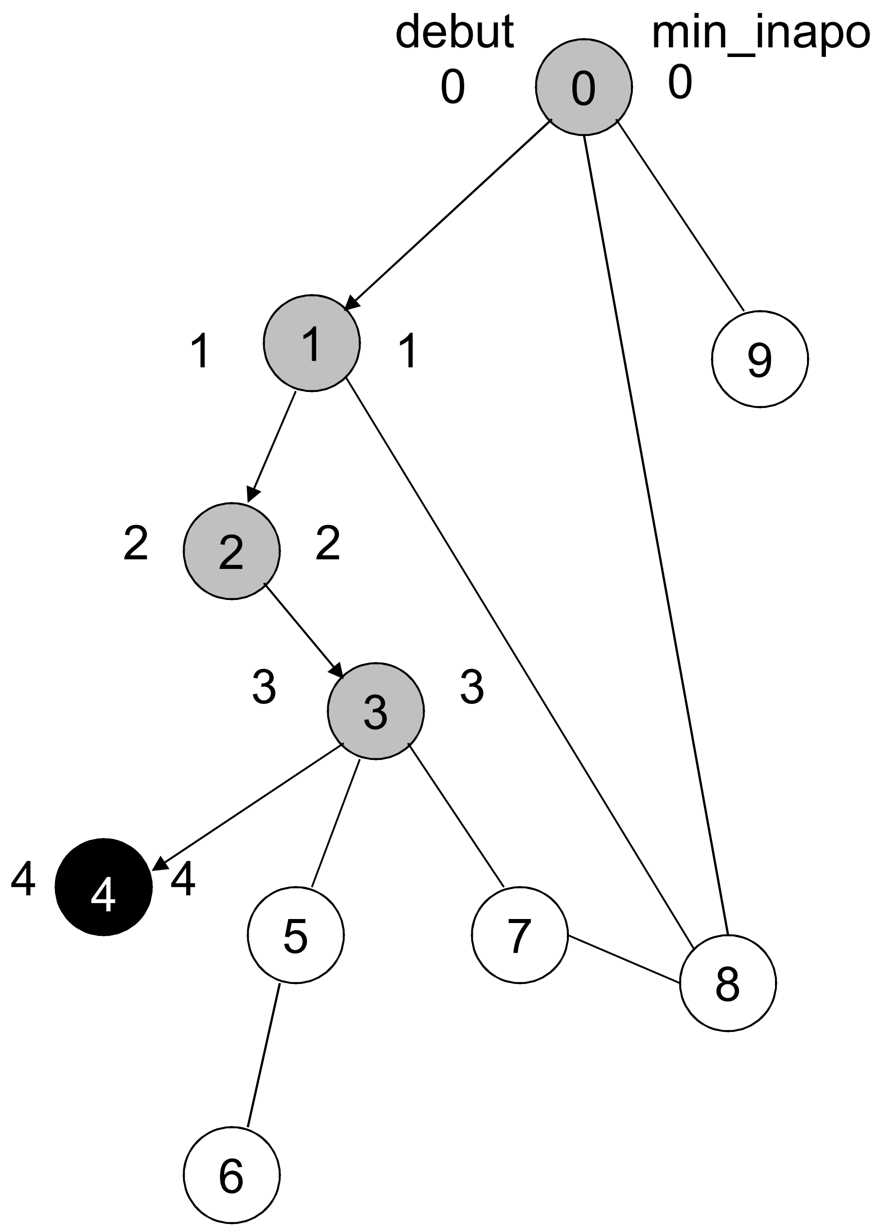
min_inapoi = debut

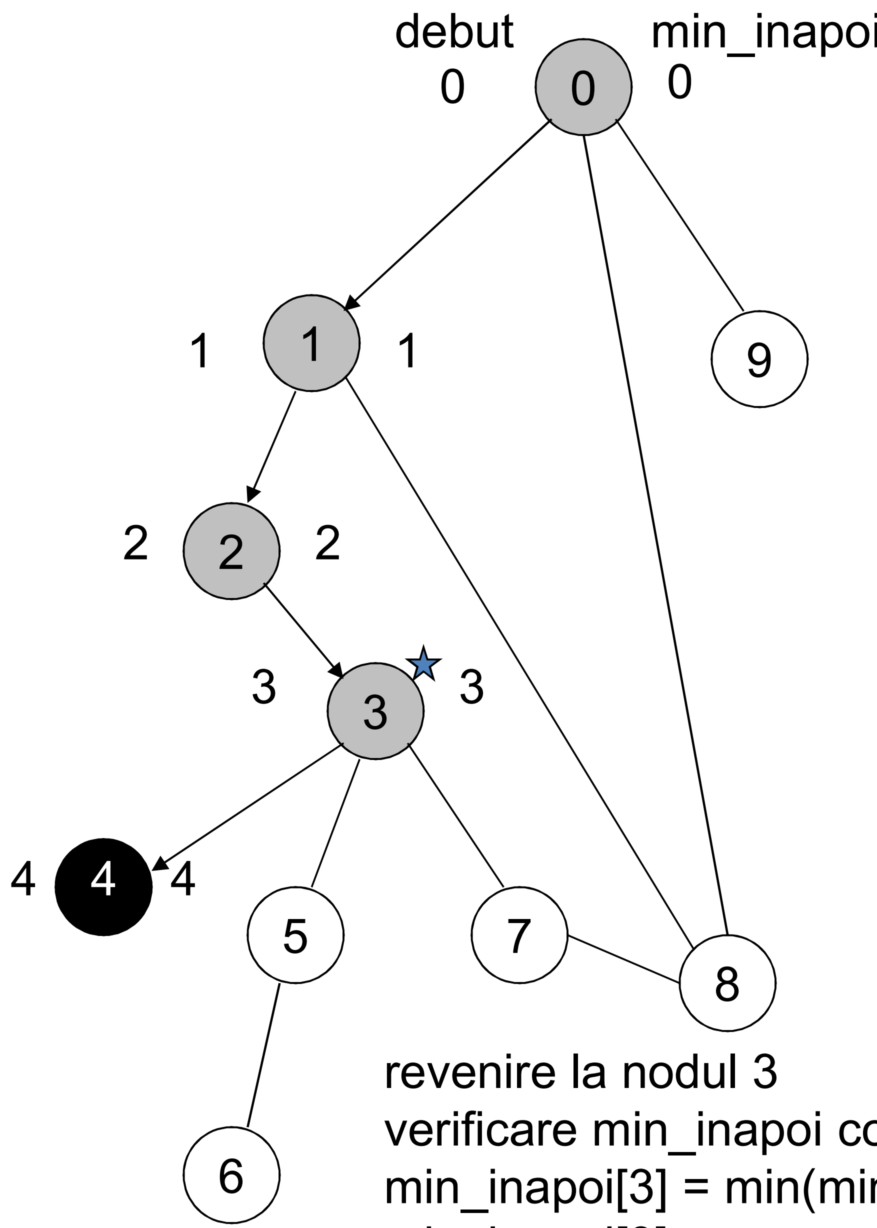


min_inapoi = debut



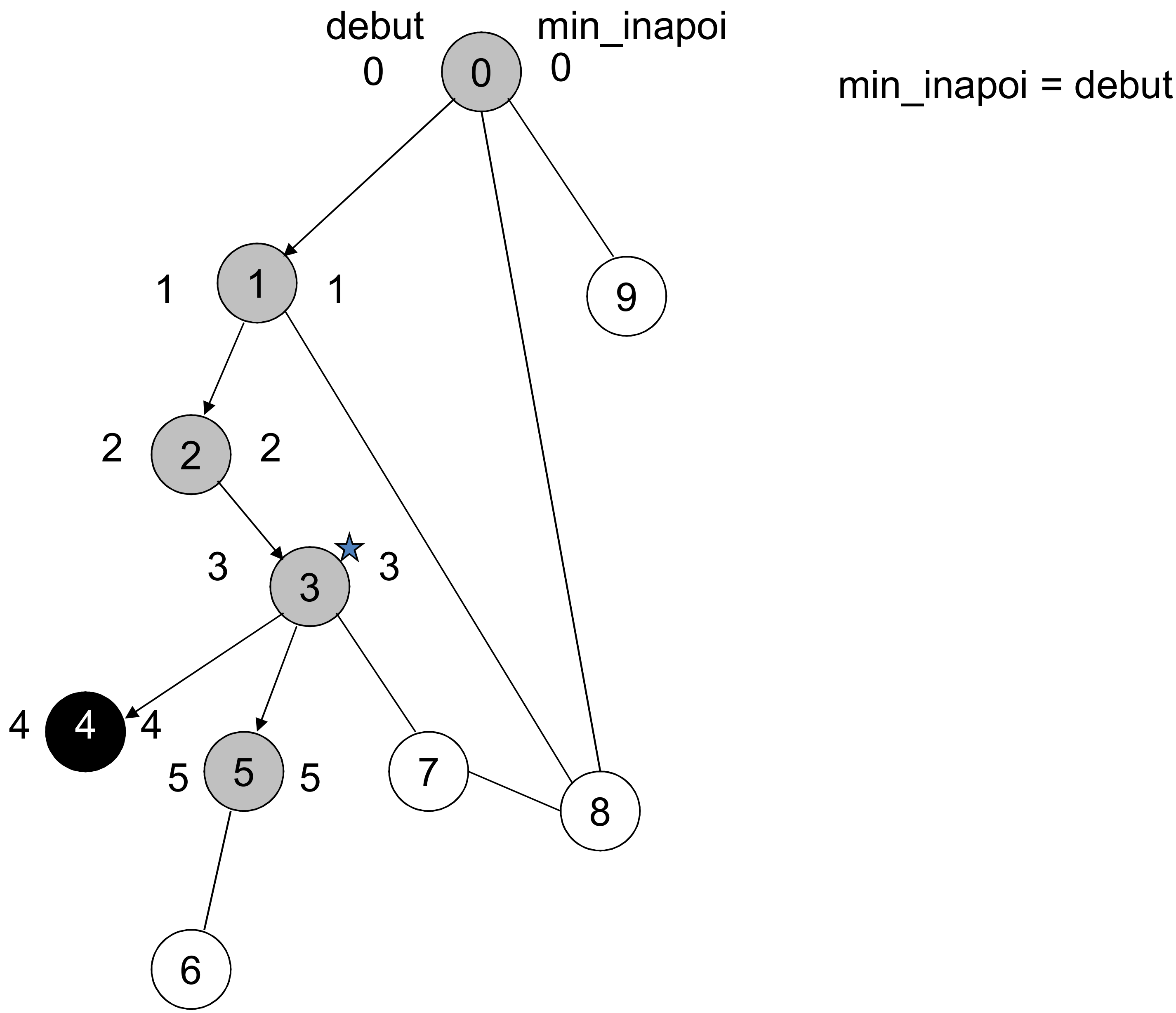
min_inapoi = debut

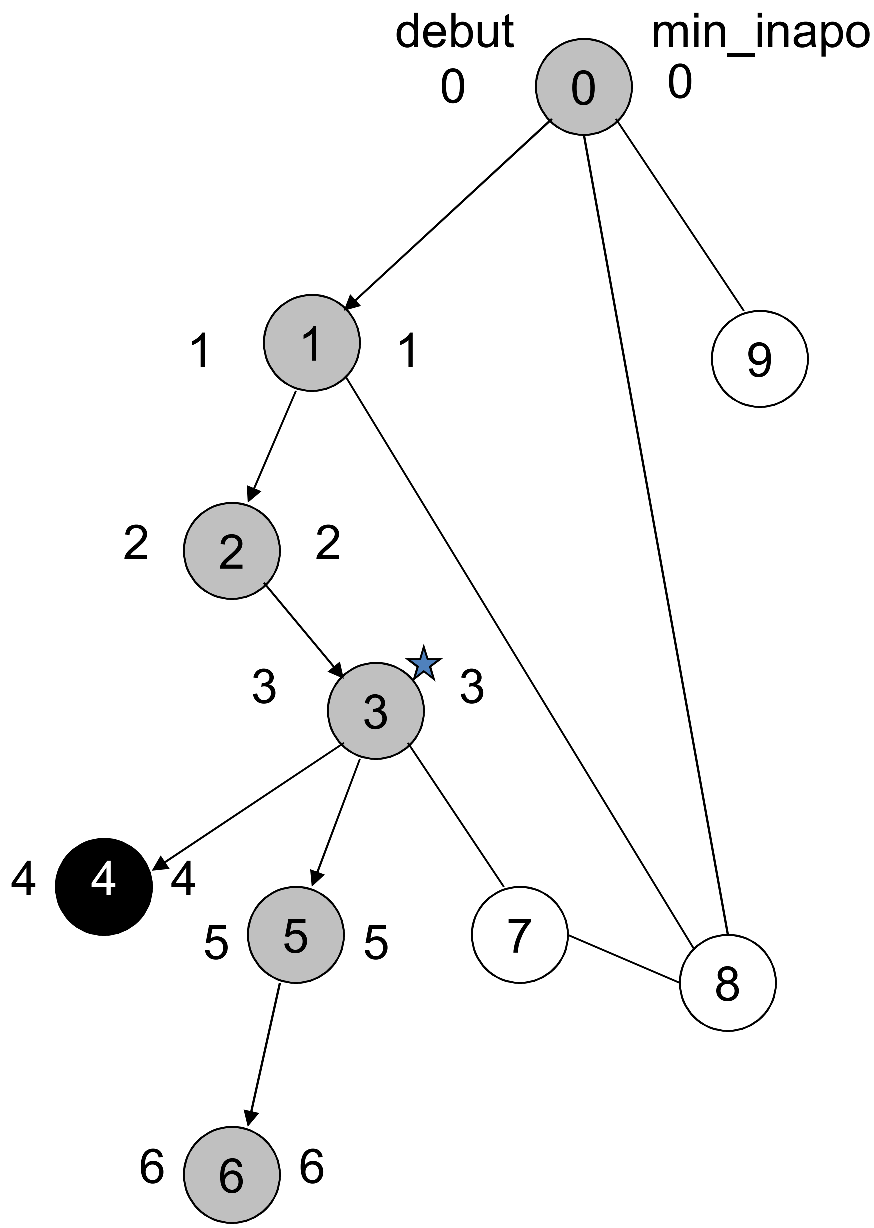

debut
min_inapoi
0
0
0
1
1
1
9
2
2
2
3
3
3
4
4
4
5
7
8
6



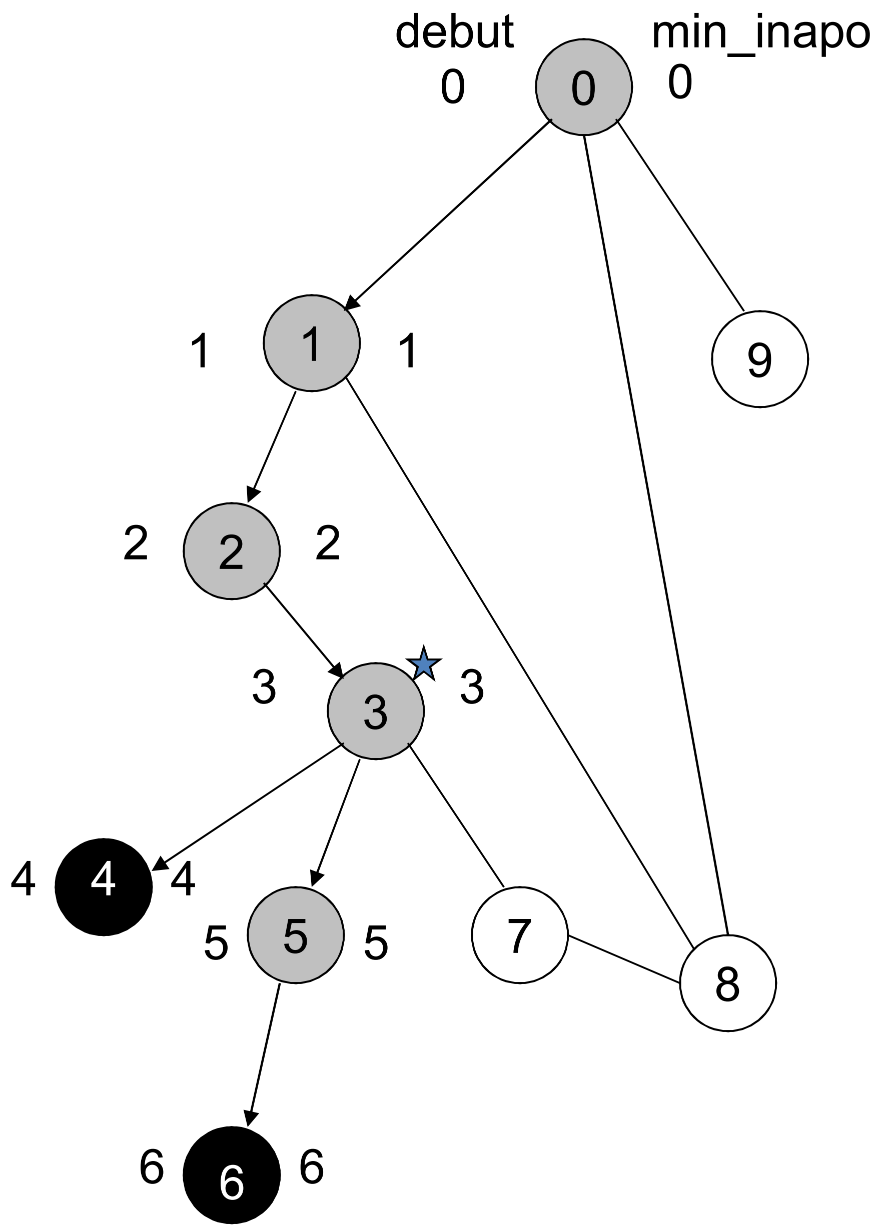
revenire la nodul 3
verificare min_inapoi copil:
min_inapoi[3] = min(min_inapoi[3], min_inapoi[4]) = min_inapoi[3]

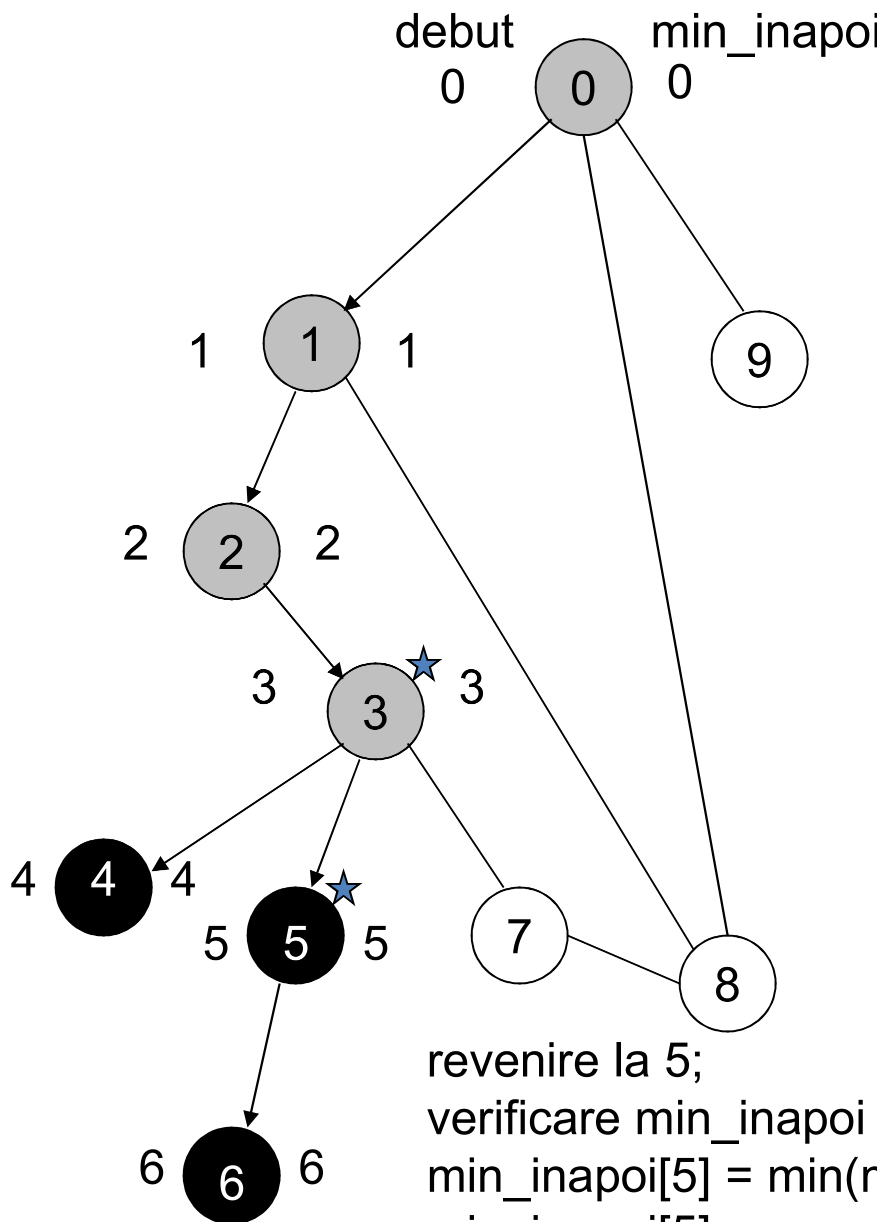
min_inapoi[4] >= debut[3] ➔ 3 este p. de art.


debut
0
min_inapoi
0
min_inapoi = debut
0
1
2
3
4
5
6
7
8
9
1 1
2 2
3 3
4 4
5 5



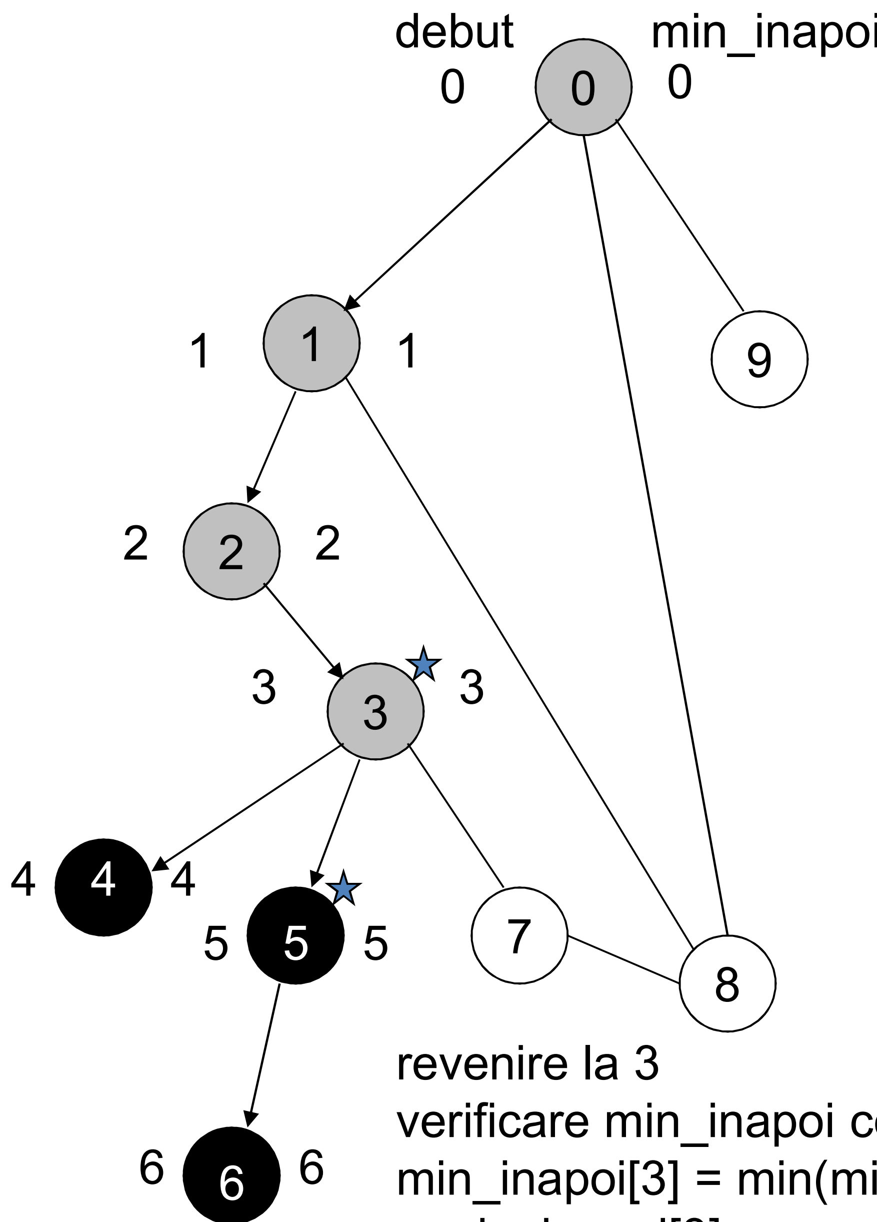
min_inapoi = debut


debut
0
0
min_inapoi
0
1
1
1
9
2
2
2
3
3
3
4
4
4
5
5
5
7
8
6
6
6



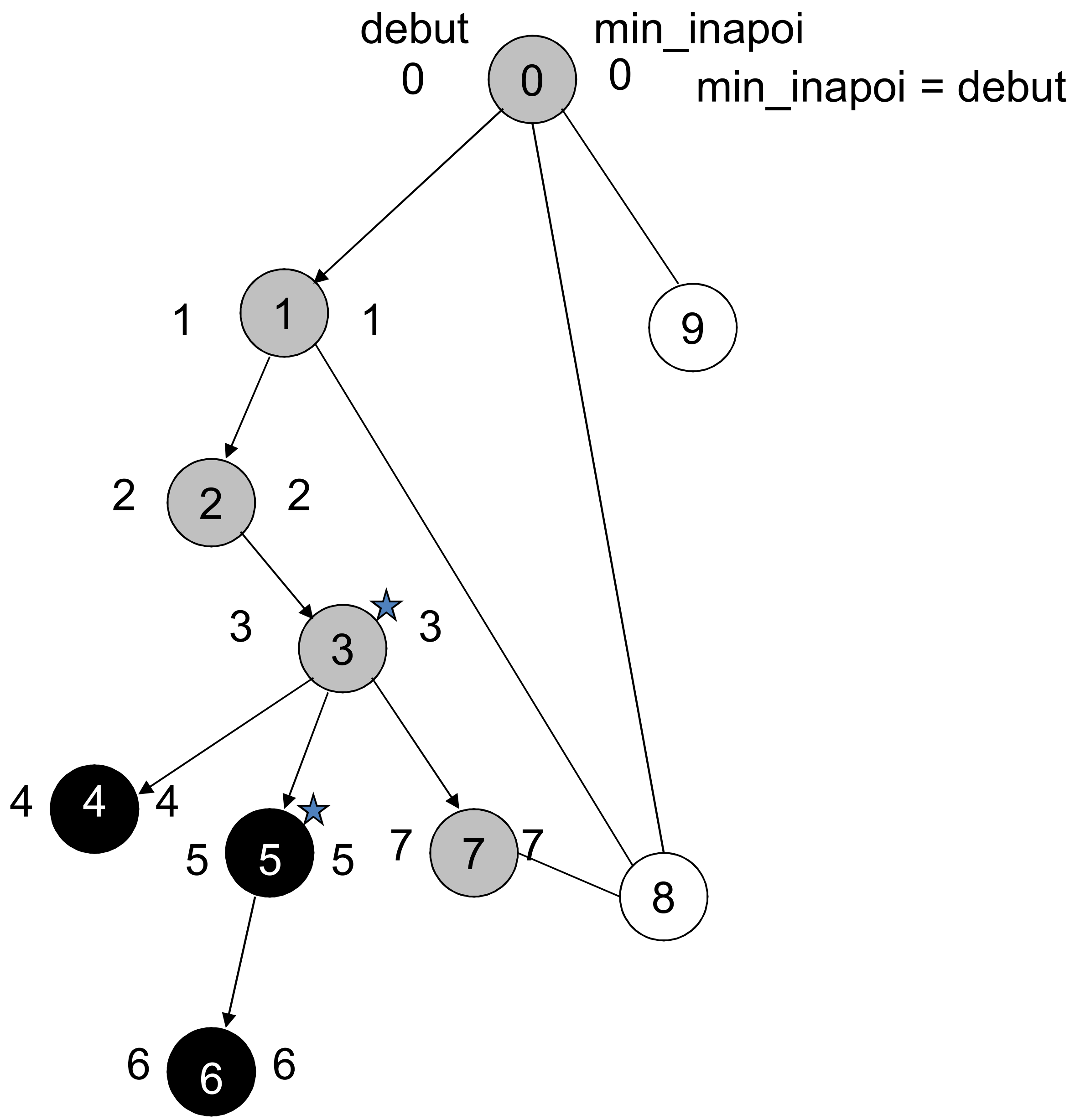
revenire la 5;
verificare min_inapoi copil:
min_inapoi[5] = min(min_inapoi[5], min_inapoi[6]) = min_inapoi[5]

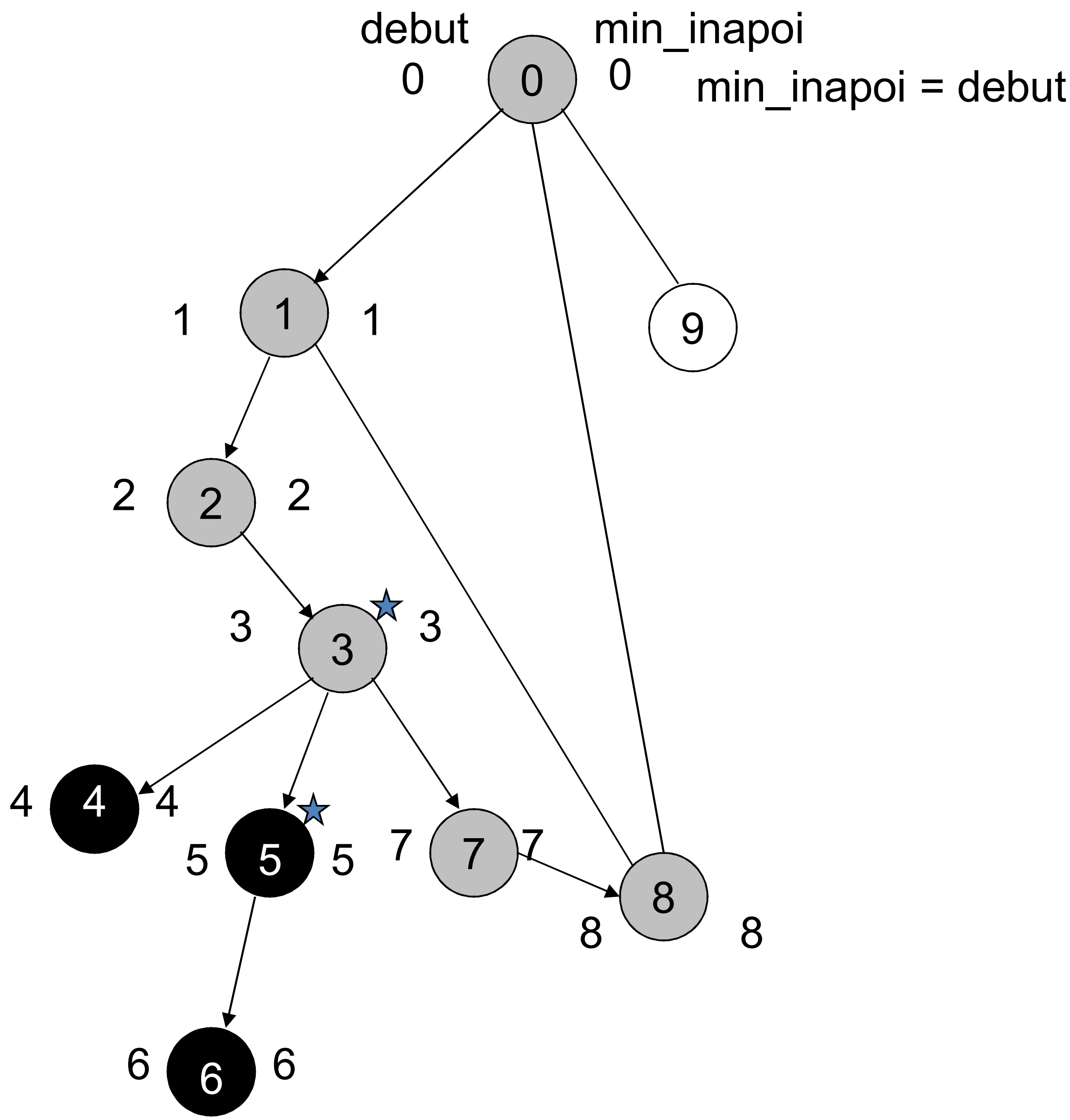
min_inapoi[6] >= debut[5] ➔ 5 este p. de art.



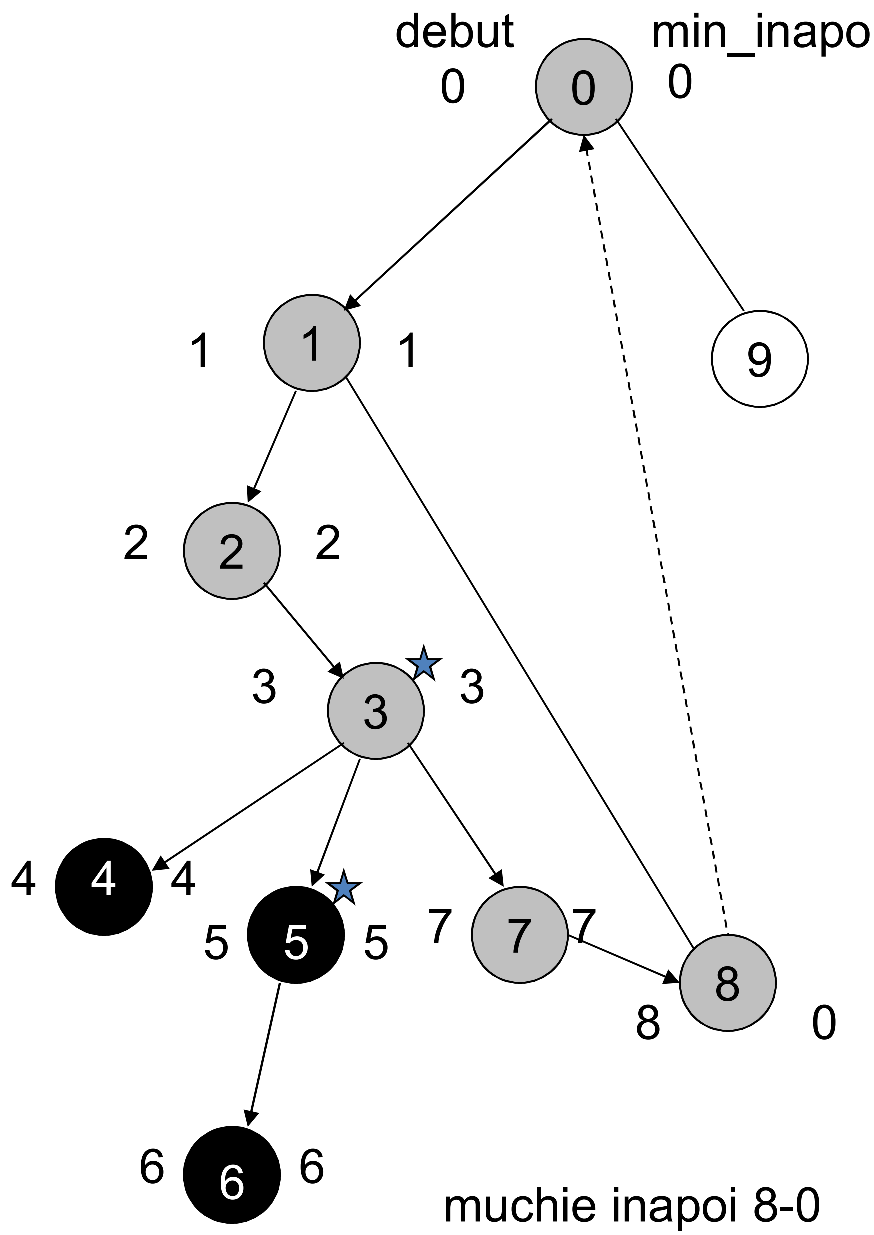
revenire la 3
verificare min_inapoi copil:
min_inapoi[3] = min(min_inapoi[3], min_inapoi[5])
= min_inapoi[3]

min_inapoi[5] >= debut[3] ➔ 3 este p. de art.


debut
min_inapoi
0
0
min_inapoi = debut
1
1
1
9
2
2
2
3
3
3
4
4
4
5
5
5
7
7
7
8
6
6
6

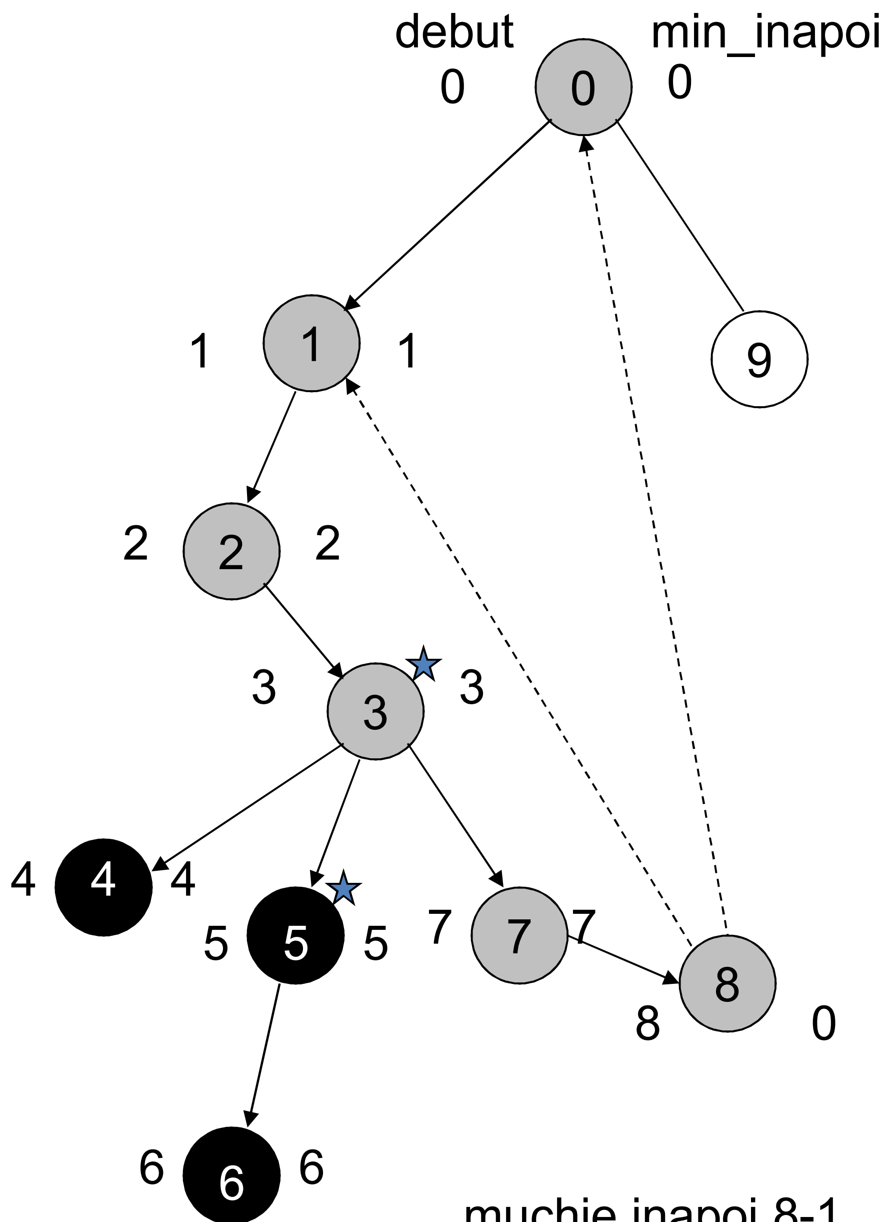


debut
0
0
min_inapoi
0
min_inapoi = debut
1
1
1
9
2
2
2
3
3
3
4
4
4
5
5
5
7
7
7
8
8
8
6
6
6



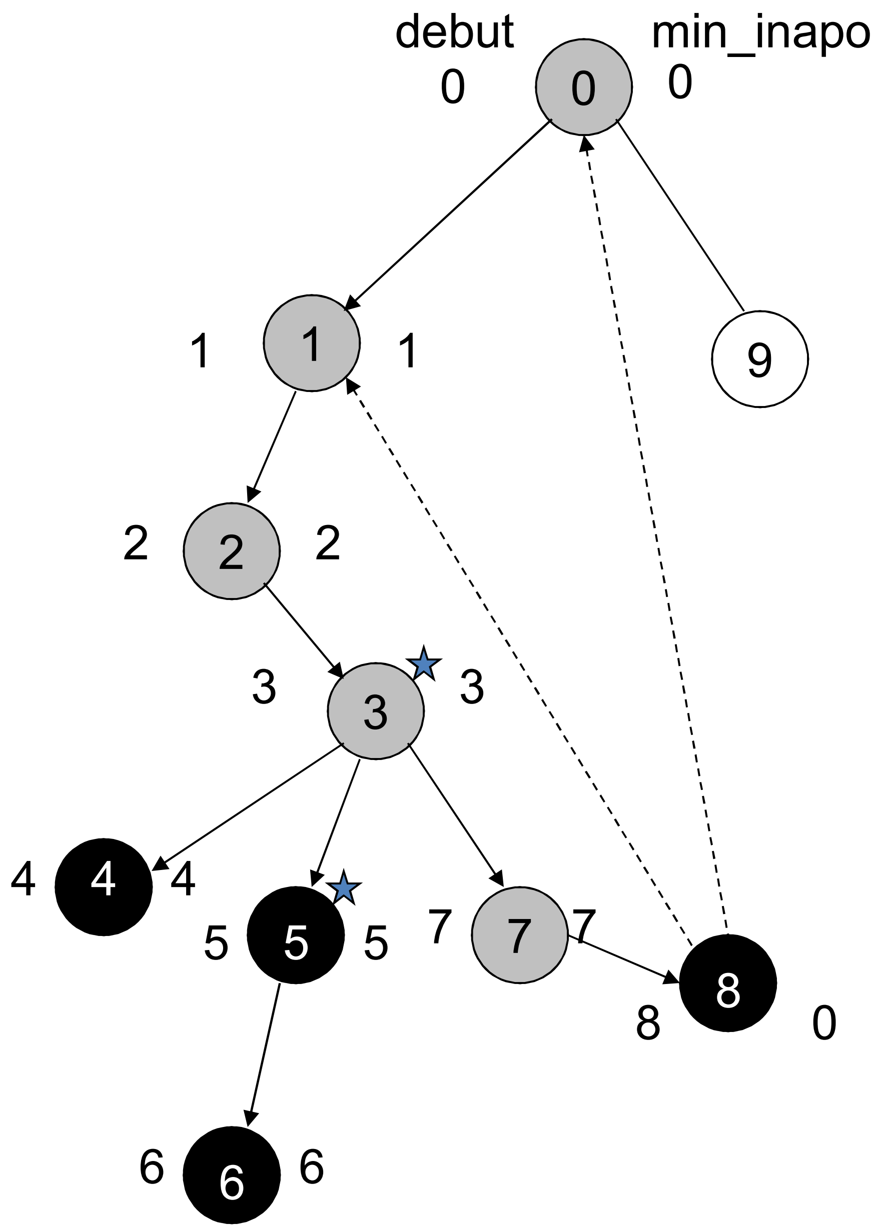
muchie inapoi 8-0

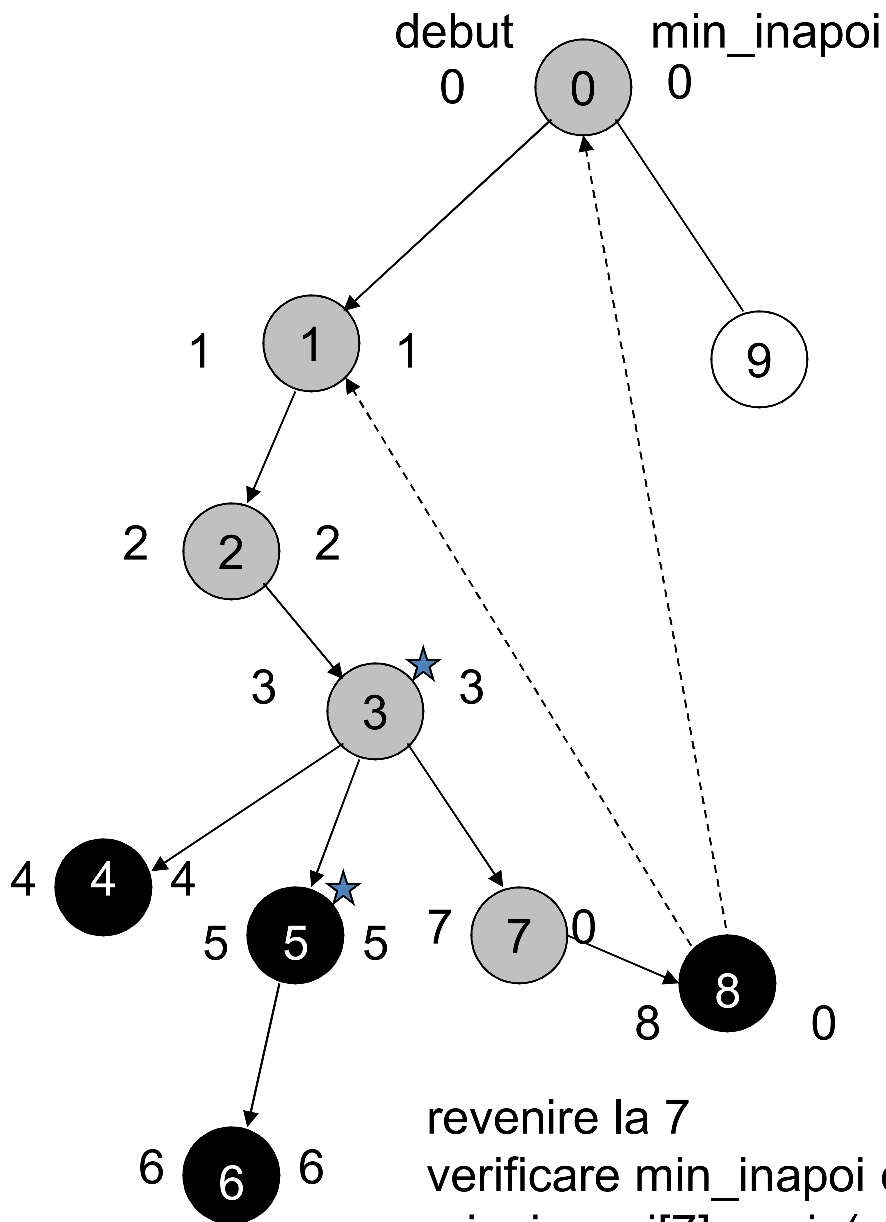
min_inapoi[8] = min(min_inapoi[8], debut[0]) = 0



muchie inapoi 8-1

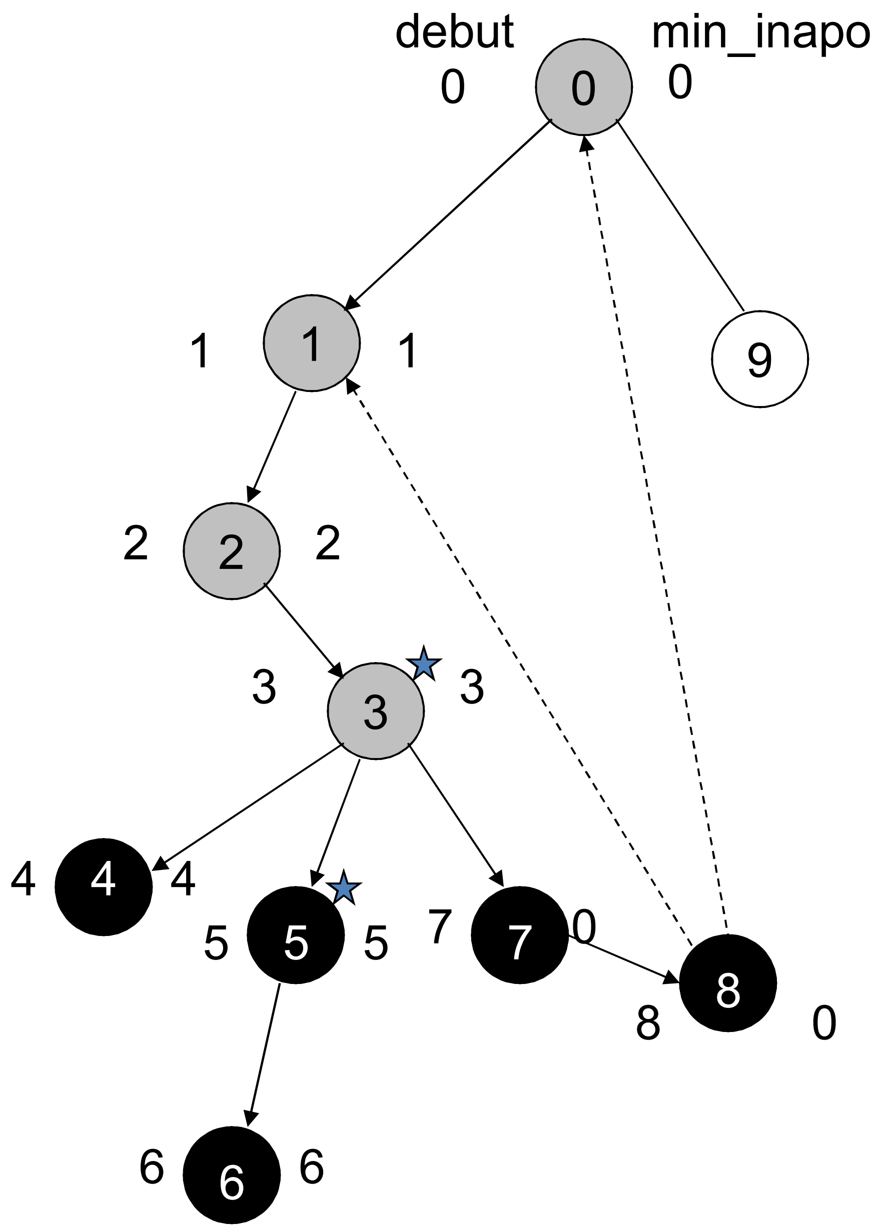
min_inapoi[8] = min(min_inapoi[8], debut[1]) = 0


debut
min_inapoi
0
0
0
1
1
1
9
2
2
2
3
3
3
4
4
4
5
5
5
7
7
7
8
8
0
6
6
6

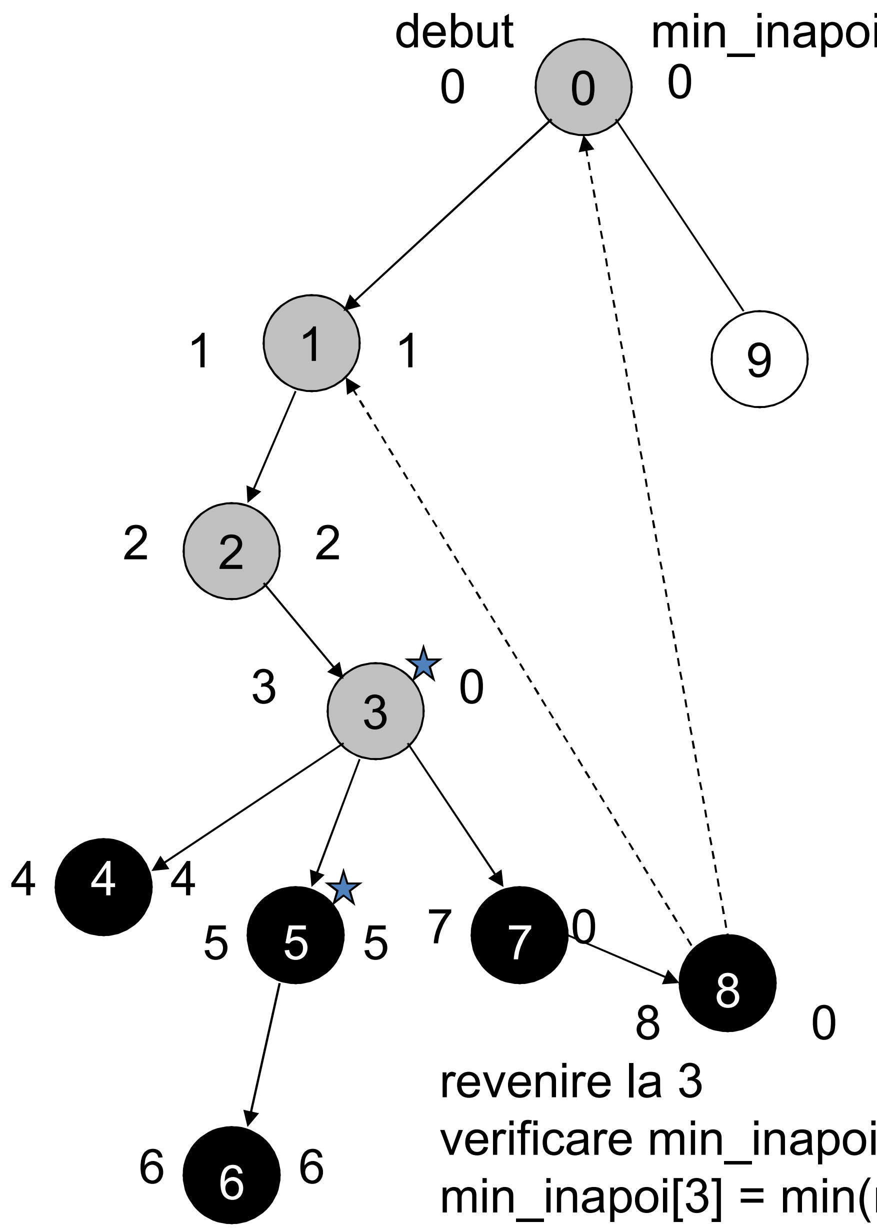


revenire la 7
verificare min_inapoi copil:
min_inapoi[7] = min(min_inapoi[7], min_inapoi[8]) = min_inapoi[8]

min_inapoi[8] >= debut[7] ? NU

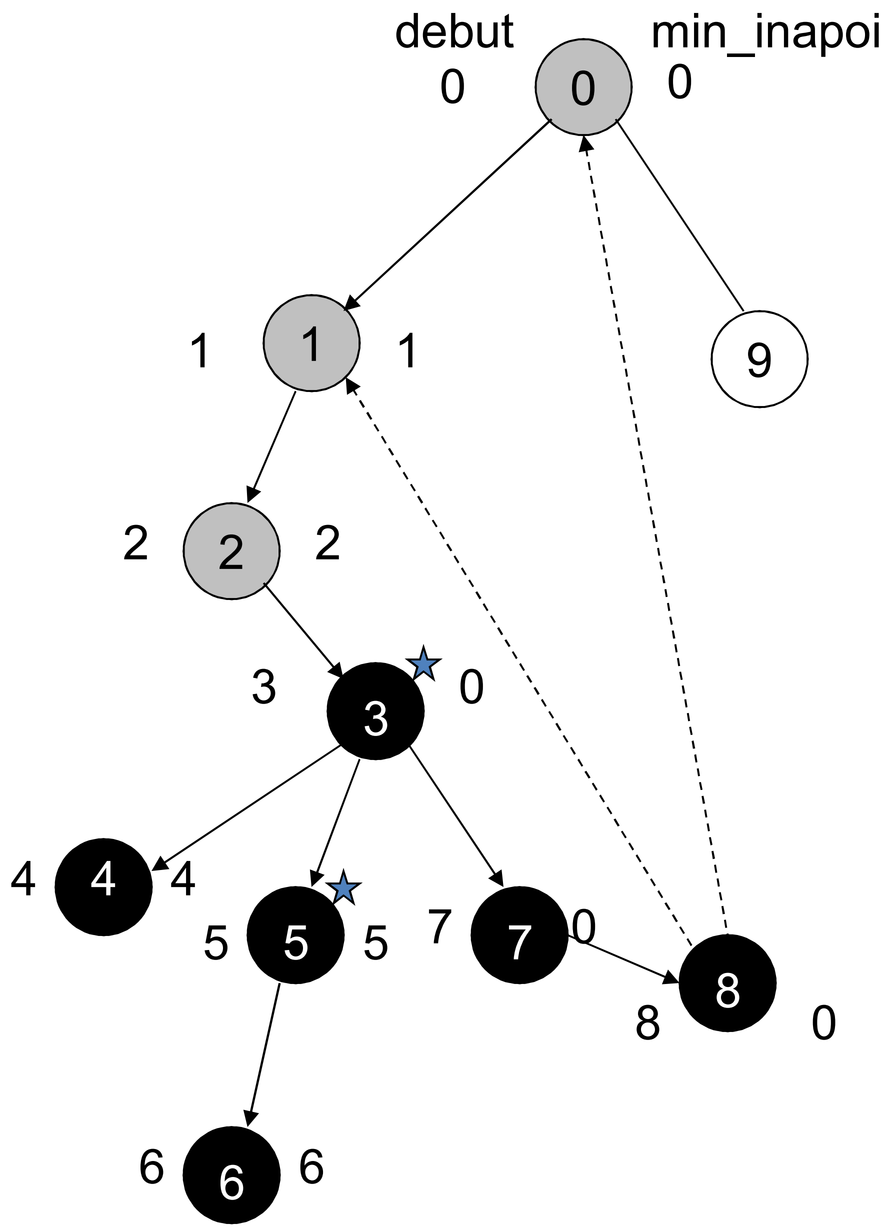


debut
min_inapoi
0
0
0
1
1
1
9
2
2
2
3
3
3
4
4
4
5
5
5
7
7
0
8
8
0
6
6
6

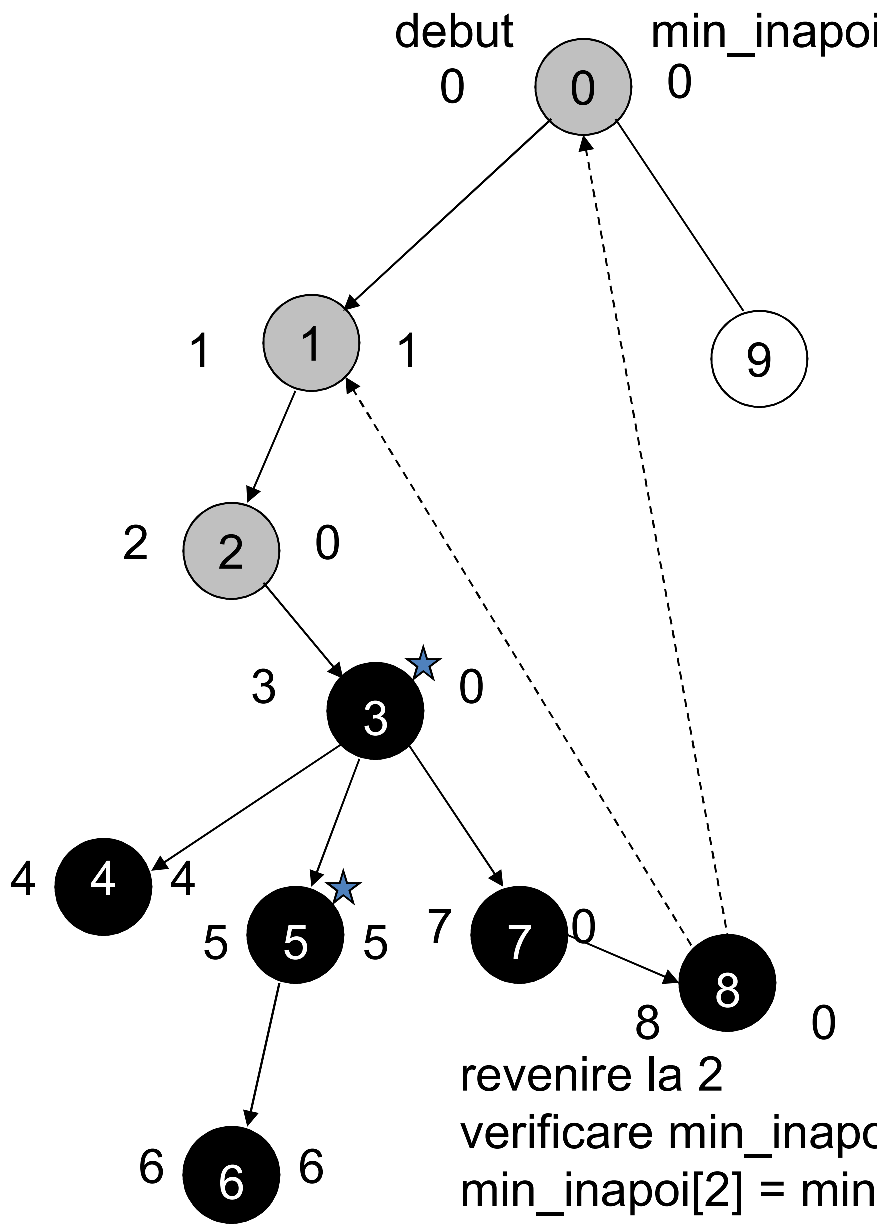


revenire la 3
verificare min_inapoi copil:
min_inapoi[3] = min(min_inapoi[3], min_inapoi[7])
= min_inapoi[7]

min_inapoi[7] >= debut[3] ? NU

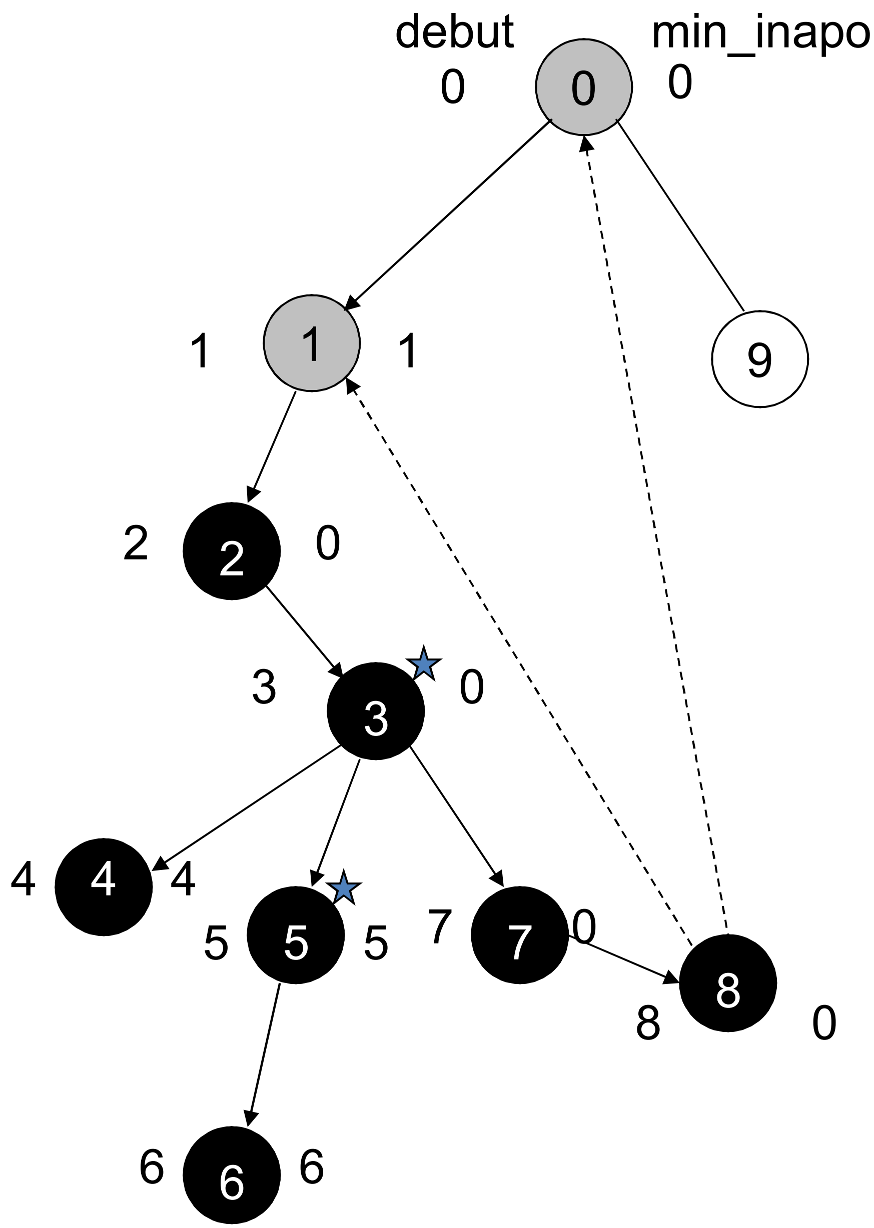


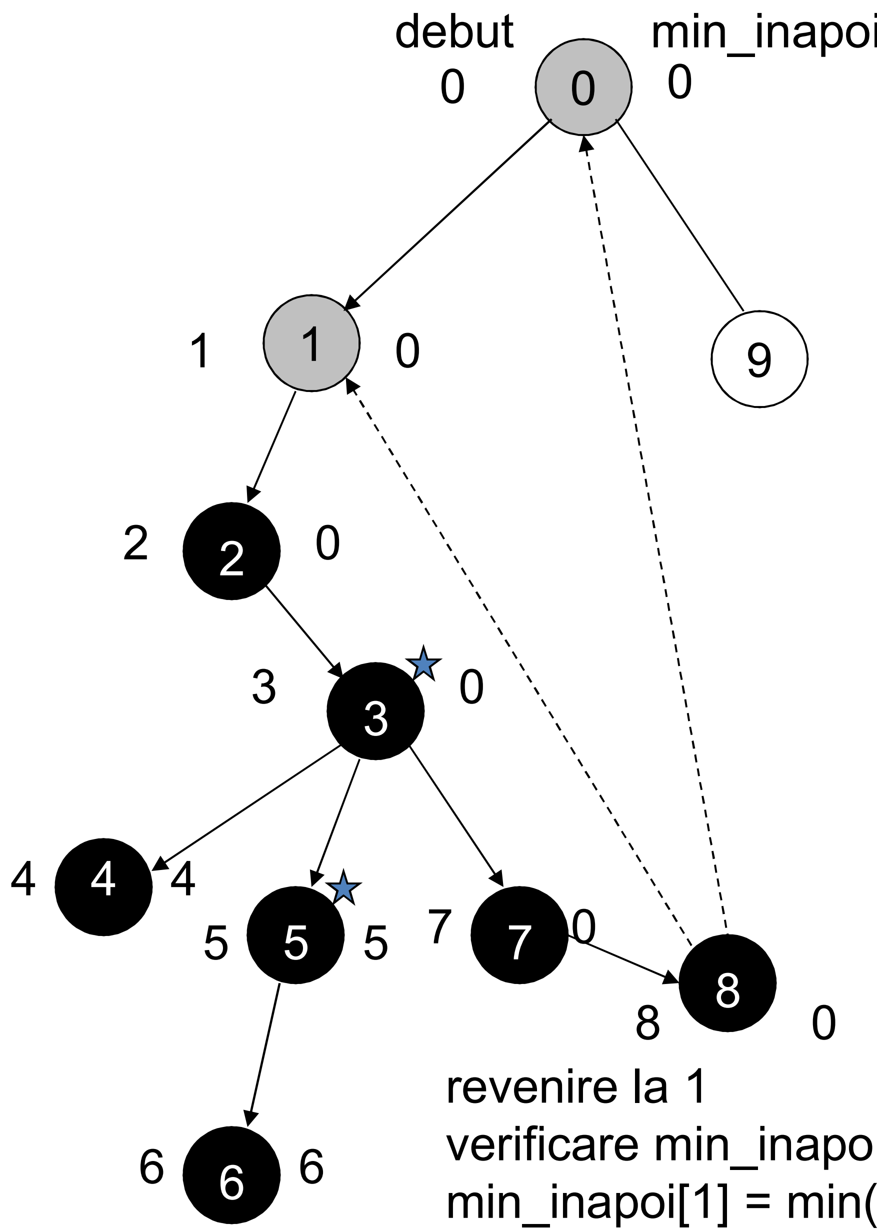
debut
min_inapoi
0
0
0
1
1
1
9
2
2
2
3
3
0
4
4
4
5
5
5
7
7
0
8
8
0
6
6
6



revenire la 2
verificare min_inapoi copil:
min_inapoi[2] = min(min_inapoi[2], min_inapoi[3])
= min_inapoi[3]

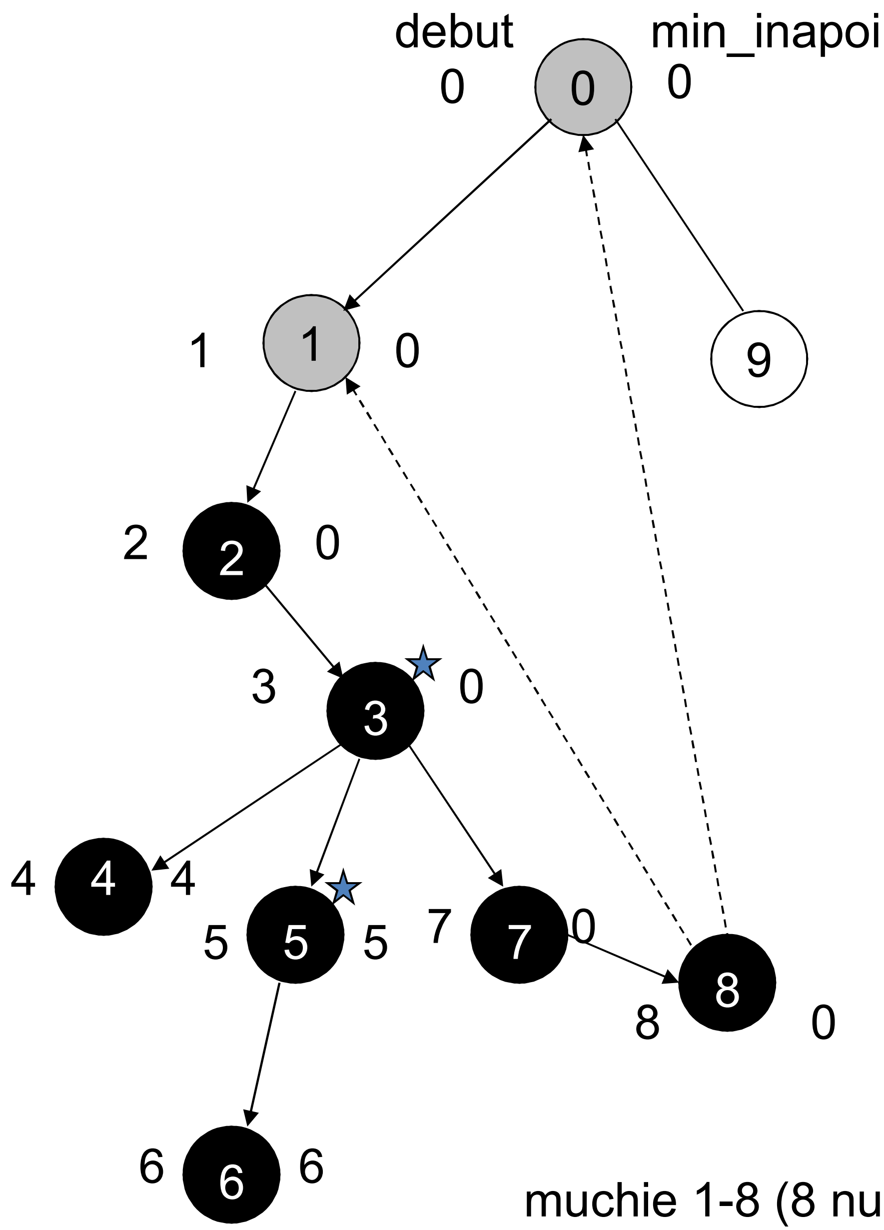
min_inapoi[3] >= debut[2] ? NU


debut
min_inapoi
0
0
0
1
1
1
9
2
2
0
3
3
0
4
4
4
5
5
5
7
7
0
8
8
0
6
6
6

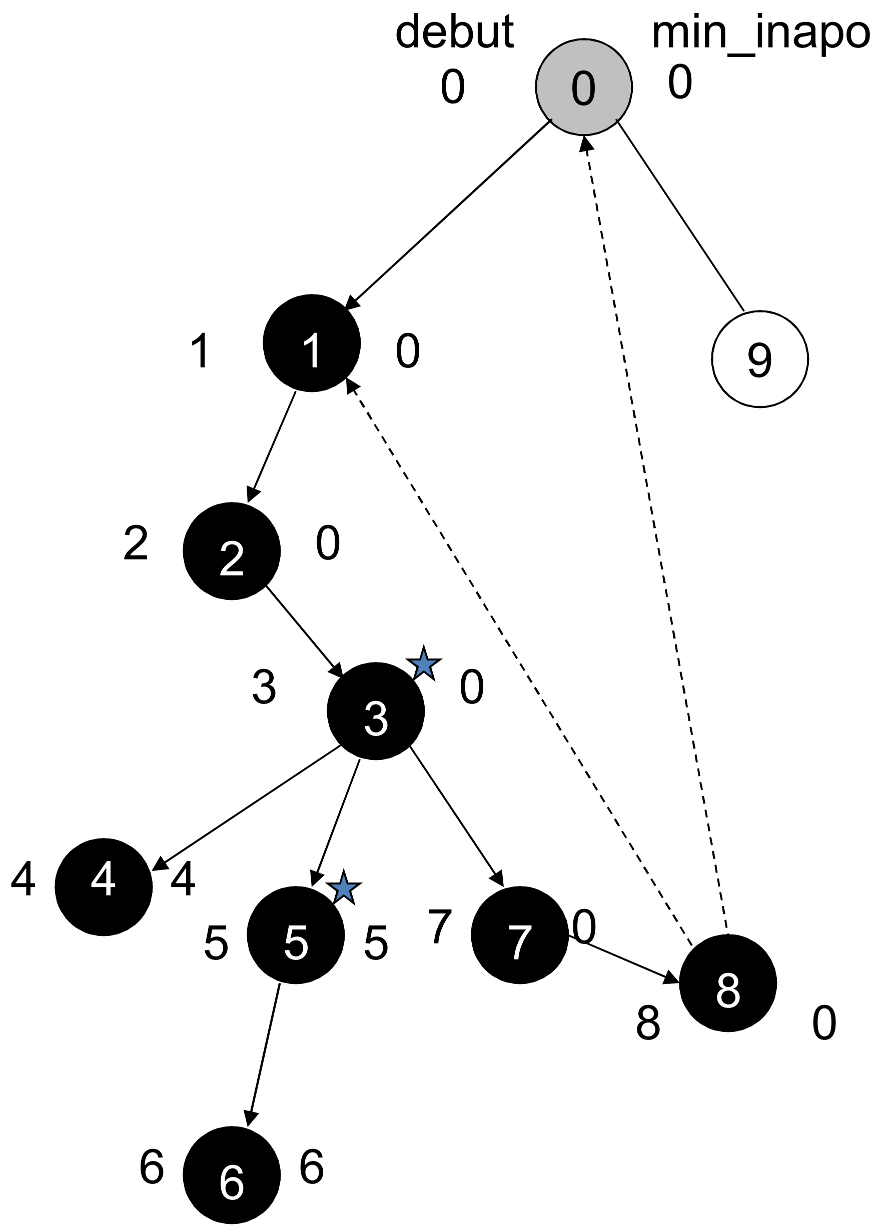


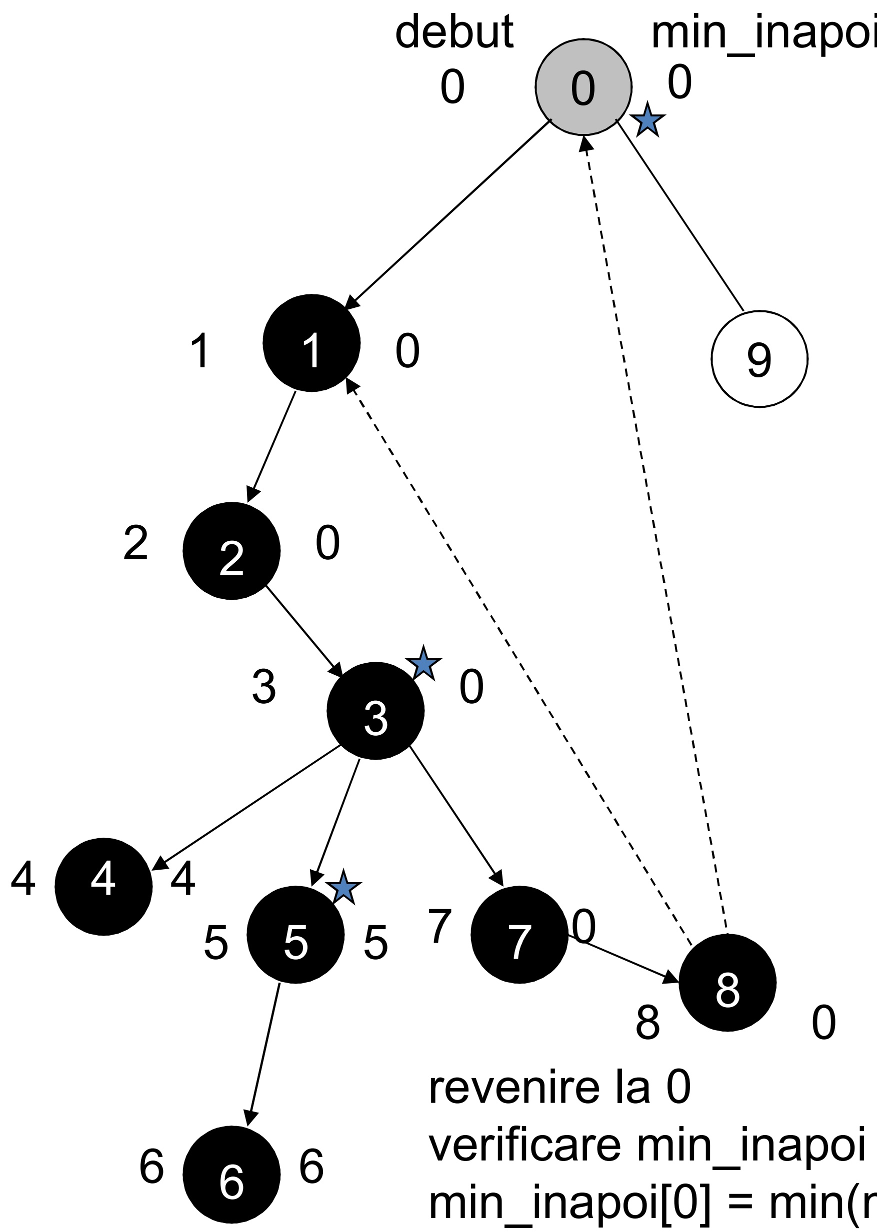
revenire la 1
verificare min_inapoi copil:
min_inapoi[1] = min(min_inapoi[1], min_inapoi[2])
= min_inapoi[2]

min_inapoi[2] >= debut[1] ? NU



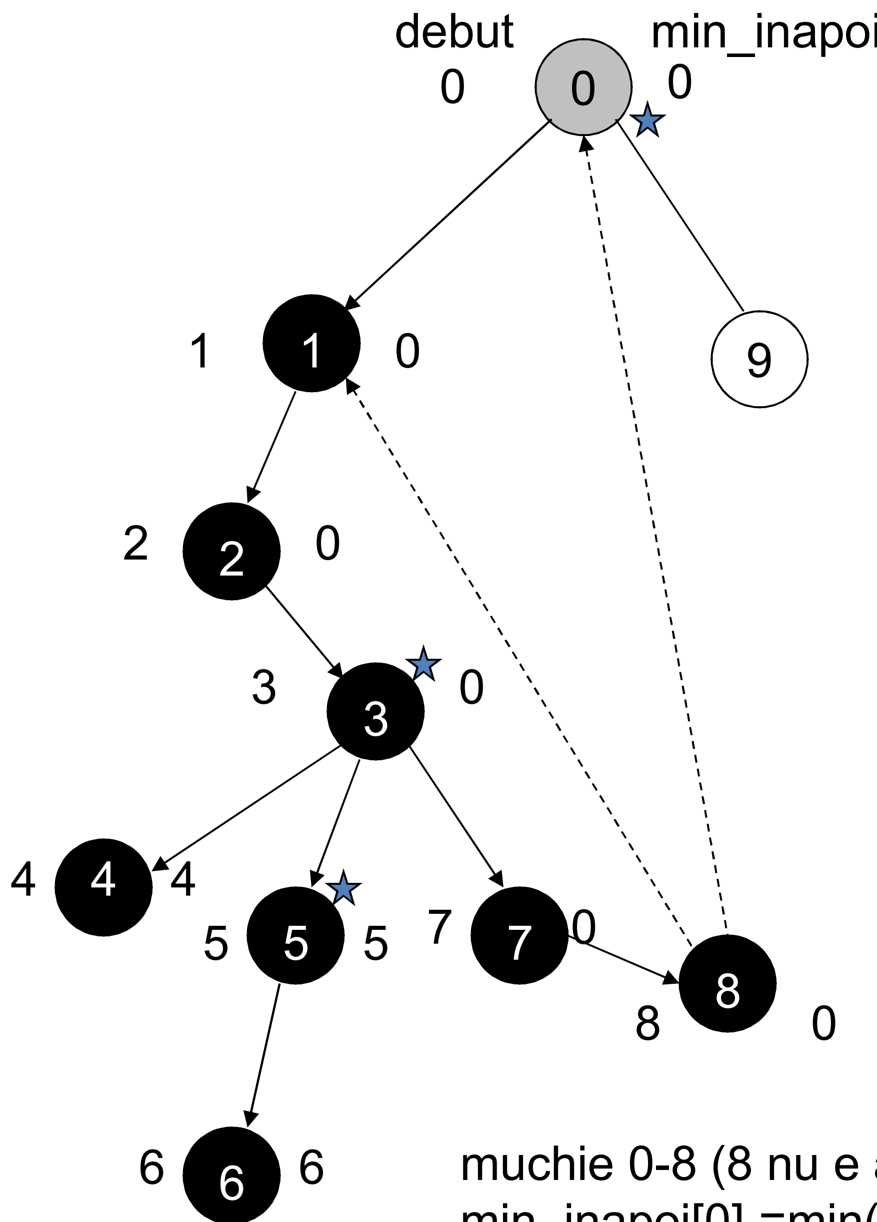
muchie 1-8 (8 nu e alb):
min_inapoi[1] =min(min_inapoi[1],debut[8])=0


debut
min_inapoi
0
0
0
1
1
0
9
2
2
0
3
3
0
4
4
4
5
5
5
7
7
0
8
8
0
6
6
6

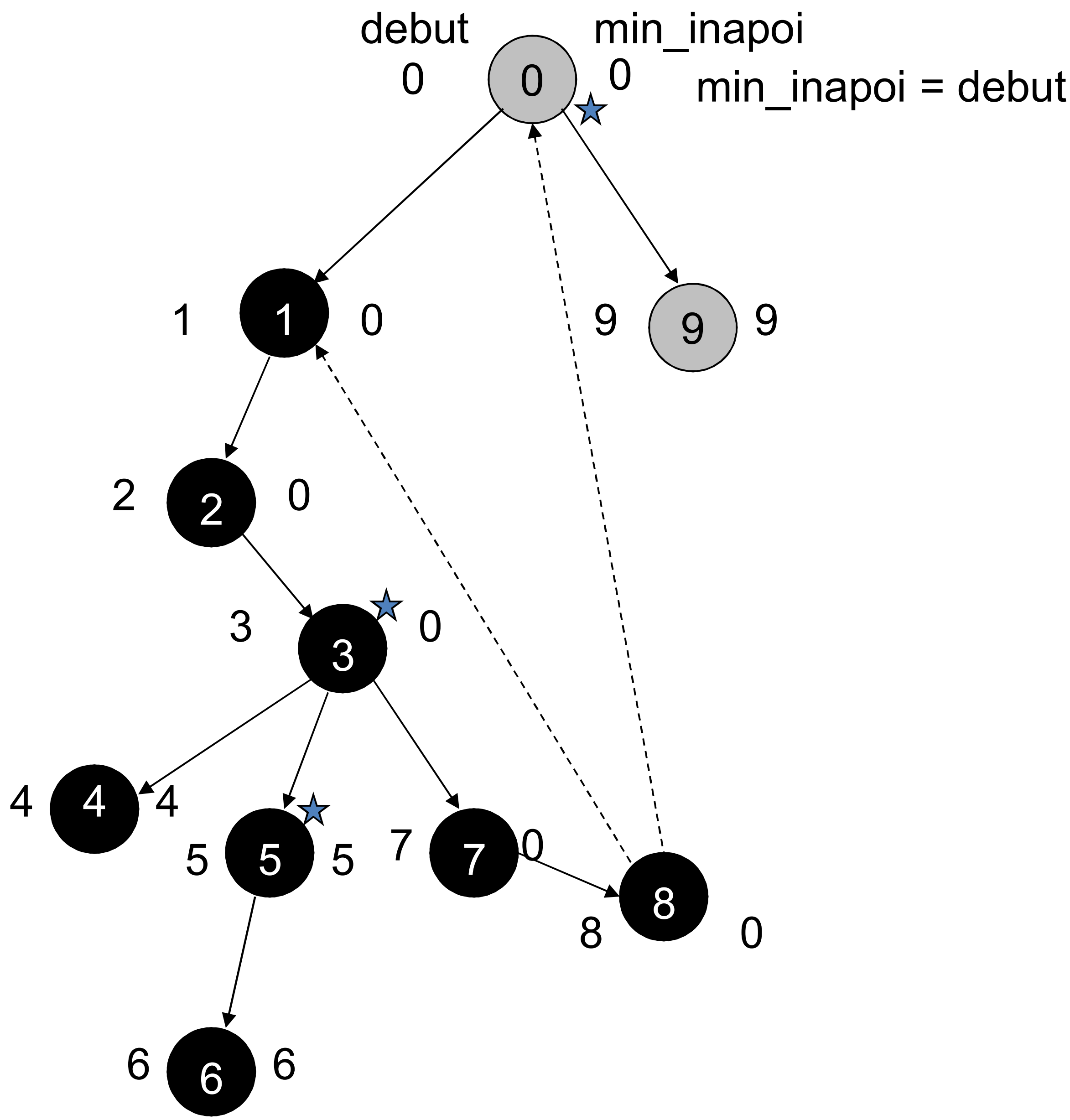


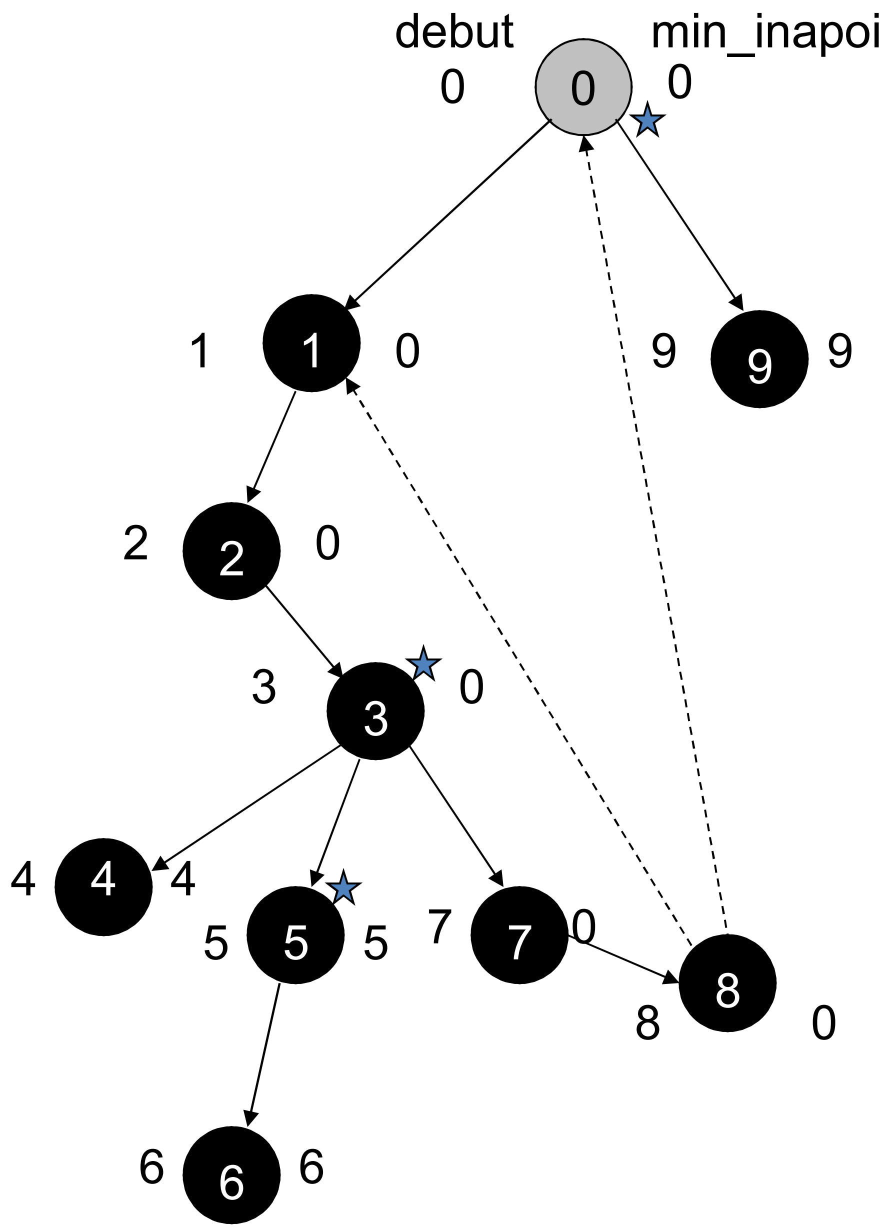
revenire la 0
verificare min_inapoi copil:
min_inapoi[0] = min(min_inapoi[0], min_inapoi[1])
= 0

min_inapoi[1] >= debut[0] ➔ 0 este p. de art.

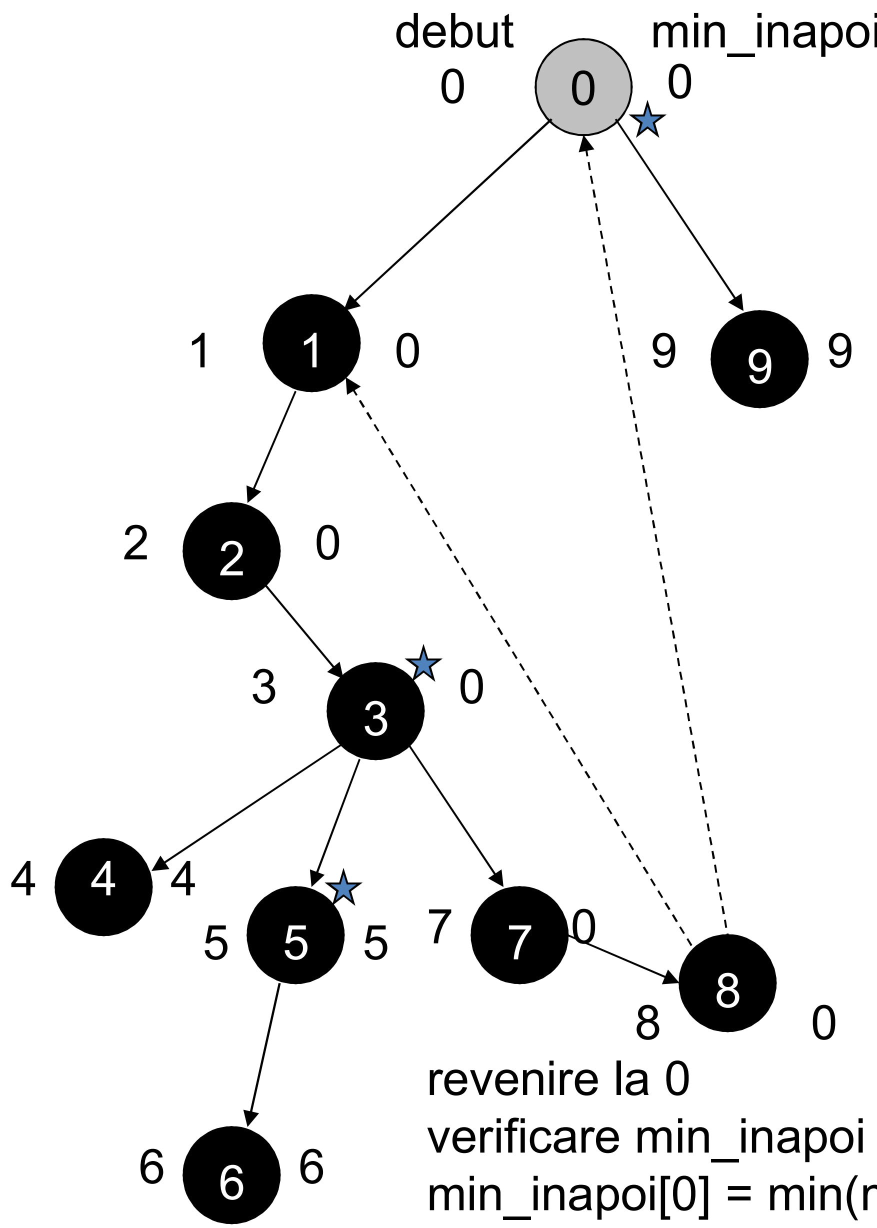


muchie 0-8 (8 nu e alb):
min_inapoi[0] =min(min_inapoi[0],debut[8])=0


debut
min_inapoi
min_inapoi = debut
0
0
0
1
1
0
9
9
9
2
2
0
3
3
0
4
4
4
5
5
5
7
7
0
8
8
0
6
6
6

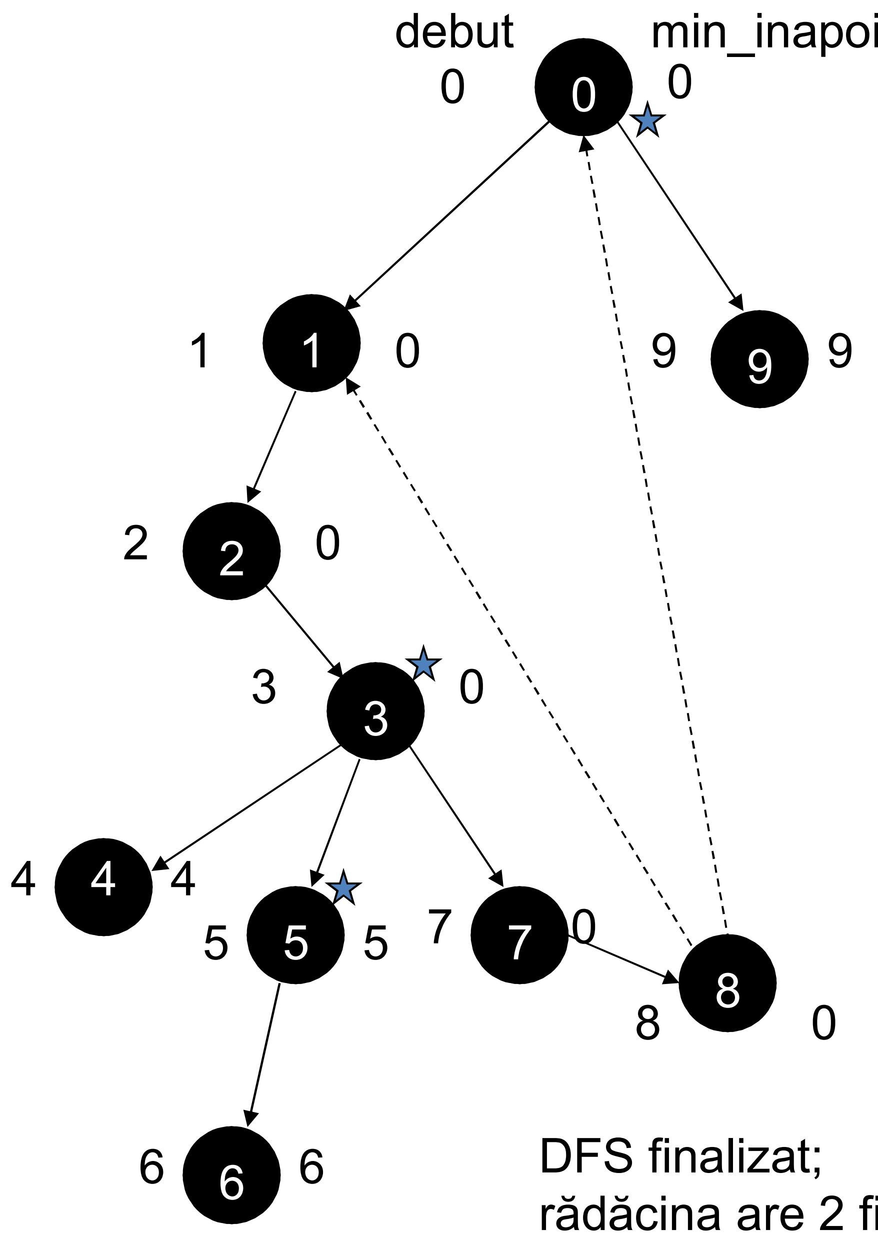


debut
min_inapoi
0
0
0
1
1
0
9
9
9
2
2
0
3
3
0
4
4
4
5
5
5
7
7
0
8
8
0
6
6
6



revenire la 0
verificare min_inapoi copil:
min_inapoi[0] = min(min_inapoi[0],
min_inapoi[9]) = 0

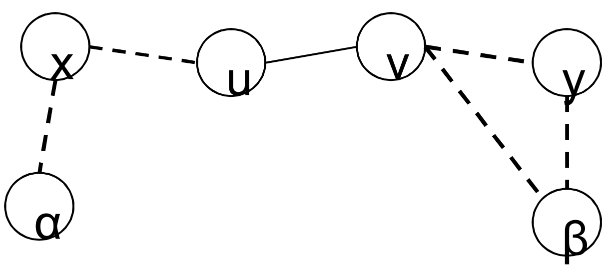
min_inapoi[9] >= debut[0] ➔ 0 este p. de art.



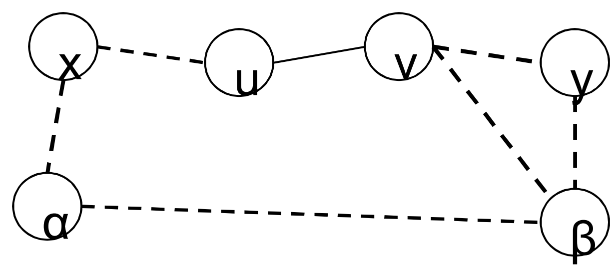
DFS finalizat;
rădăcina are 2 fii ➔ 0 punct de articulaţie
(altfel ar fi fost şters marcajul)

Punti

- G=(V,E), graf neorientat si (u,v)$\in$E
- (u,v) este punte in G **<=>** $\exists$x,y$\in$V, x≠y, a.i. $\forall$ x..y contine muchia (u,v)



Orice drum x..y trece prin (u,v)
=>(u,v) este punte

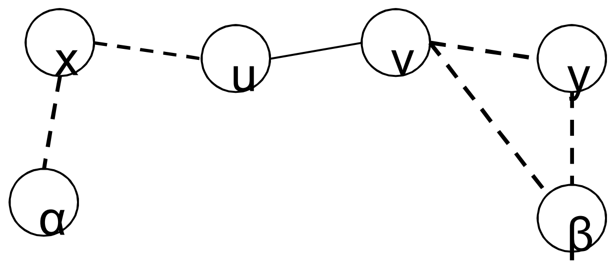


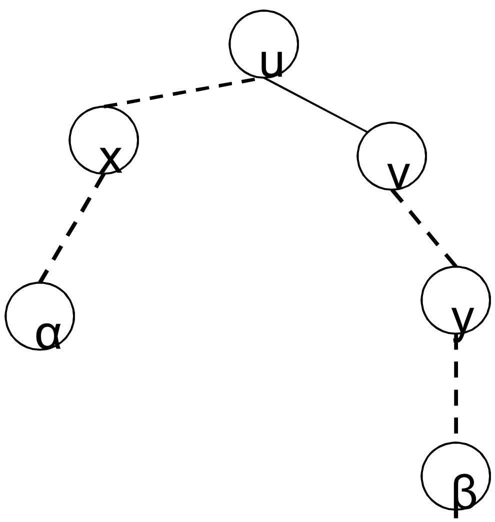
(u,v) nu este punte

Algoritm

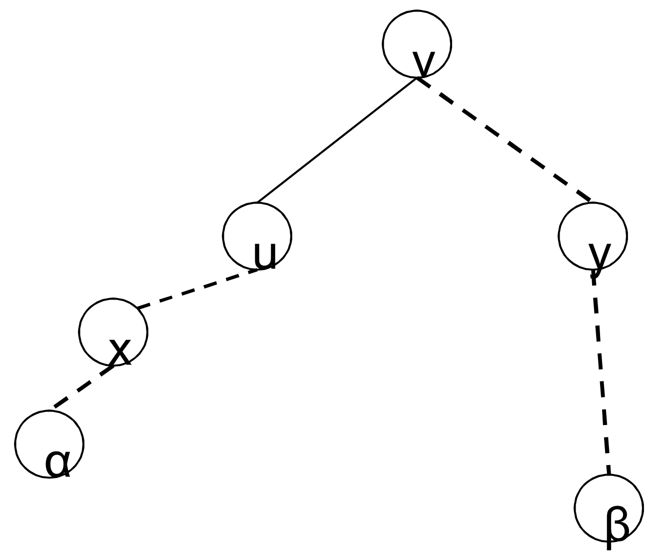
```
Exploreaza(u)
    debut[u] ← timp                          // si devine nod gri
    min_inapoi[u] ← timp
    timp ← timp+1
    foreach v succesor al lui u
        If (debut[v]=-1)                     // nod alb
            P[v] ← u
            Exploreaza(v)
            min_inapoi[u] ← min{min_inapoi[u],min_inapoi[v]}
            If(min_inapoi[v]>=debut[u]) punte[v]=1;
                articulatii[u] ← 1
            else min_inapoi[u] ← min{min_inapoi[u], debut[v]}
```

Exemplu





DFS din u; puntea este detectata in v



DFS din v; puntea este detectata in u