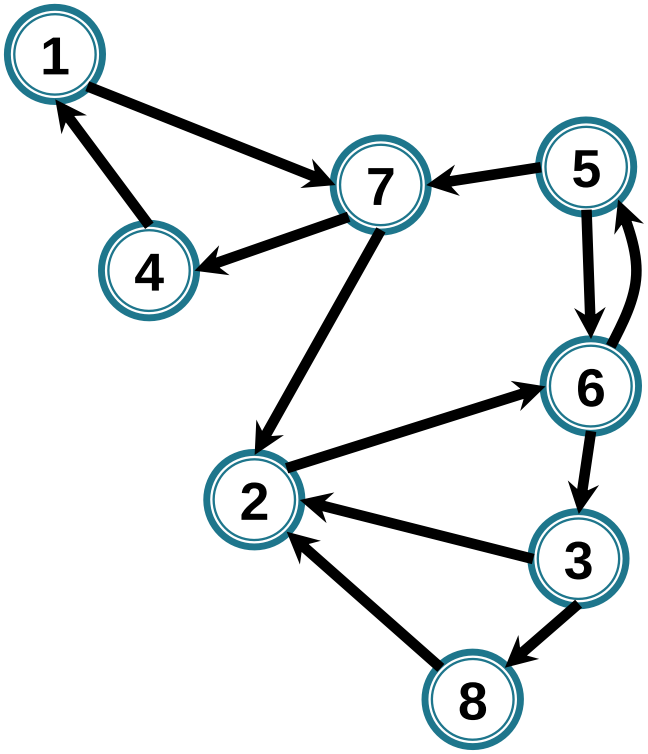
Determinarea componentelor tare conexe ale unui graf orientat

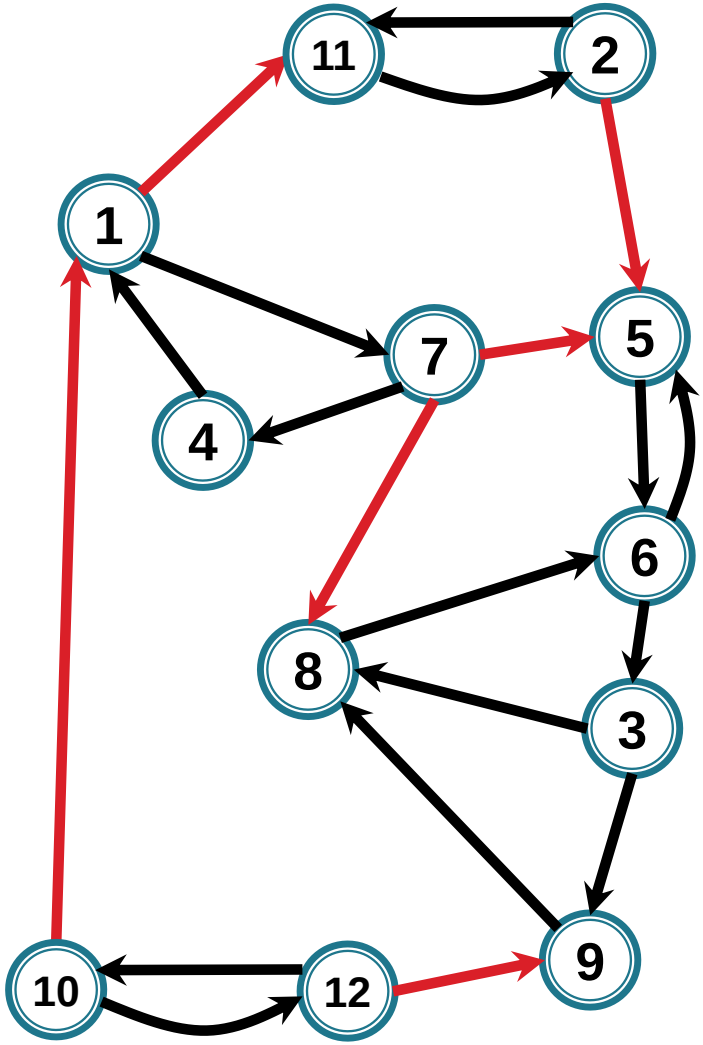
Fie G=(V,E) orientat

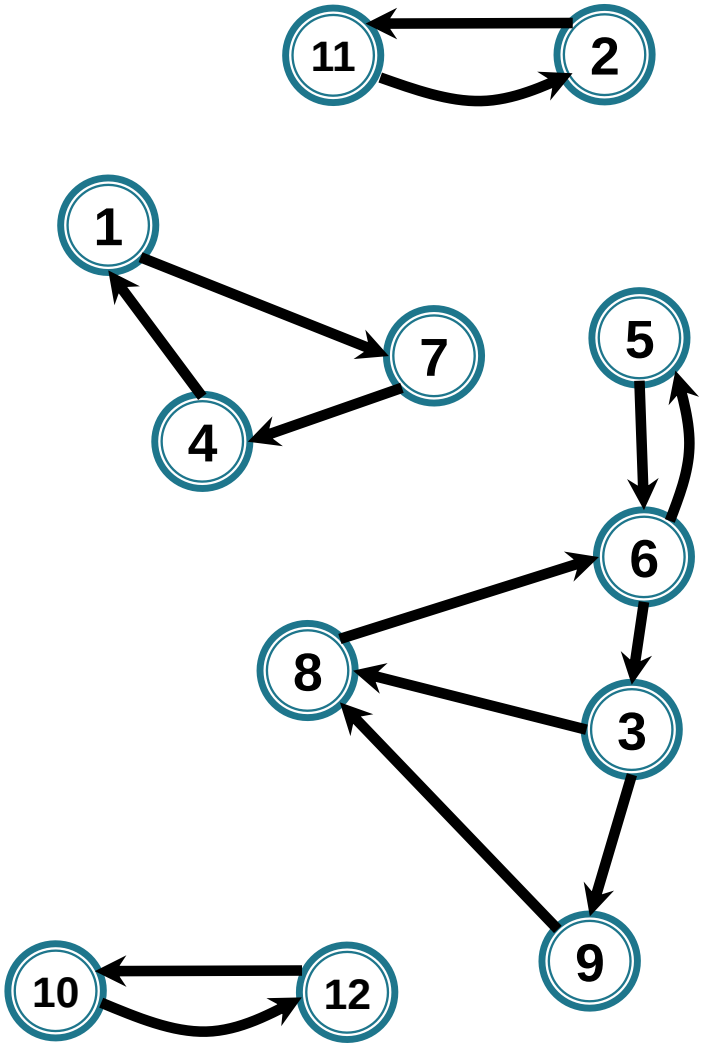
- G este **tare conex** dacă între oricare două vârfuri există un drum
- O **componentă tare conexă** a lui G = subgraf indus al lui G tare conex, maximal



Graf tare conex

Componente tare conexe

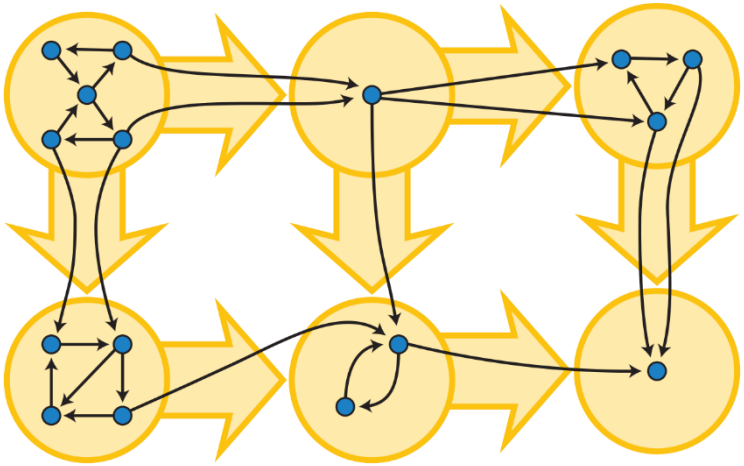




Componentele tare conexe

Componente tare conexe

- Detectarea de comunități în rețele (sociale, de colaborare/citări, economice)



- Etape în alți algoritmi, rezolvarea altor probleme, precum 2-SAT

Algoritmi componente tare conexe

Folosind mai multe parcurgeri?

Algoritmi componente tare conexe

- **Folosind mai multe parcurgeri?**

Posibilă idee:

componenta(x) =

multimea vârfurilor accesibile din x în G $\cap$

multimea vârfurilor accesibile din x în G^T

unde $G^T = (V, E^T)$, $E^T = \{yx \mid xy \in E\}$

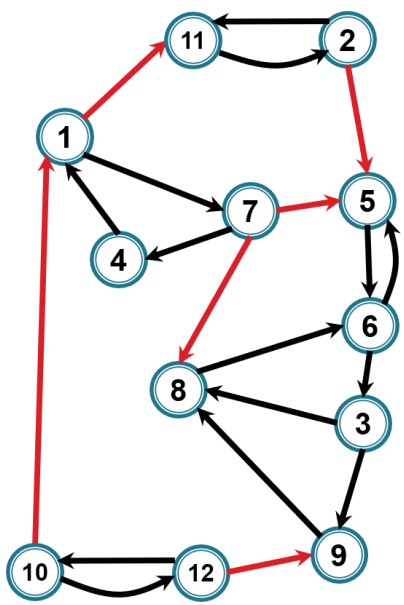
Algoritmi componente tare conexe

Componenta tare conexă a vârfului 1 =

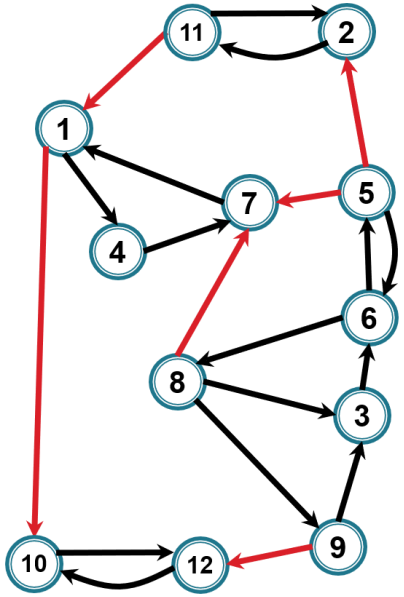
mulțimea vârfurilor accesibile din 1 în G (vizitate în DFS(1,G))

$\cap$ intersectată cu

mulțimea vârfurilor accesibile din 1 în G^T (vizitate în DFS(1,G^T))



G



G^T

DFS(1,G) => 1,7,4,5,6,3,9,8,11,2

DFS(1,G^T) => 1,4,7,10,12

Intersecția vârfurilor vizitate =>

Componenta(1) = 1,4,7

Algoritmi componente tare conexe

- **Observații**

  Într-un graf orientat:

  - Componentele tare conexe sunt vârf-disjuncte
  - Orice vârf aparține unei (unice) componente tare-conexe
  - Un arc poate să nu aparțină niciunei componente tare-conexe
  - Componente tare conexe din G = componente tare conexe din G^T

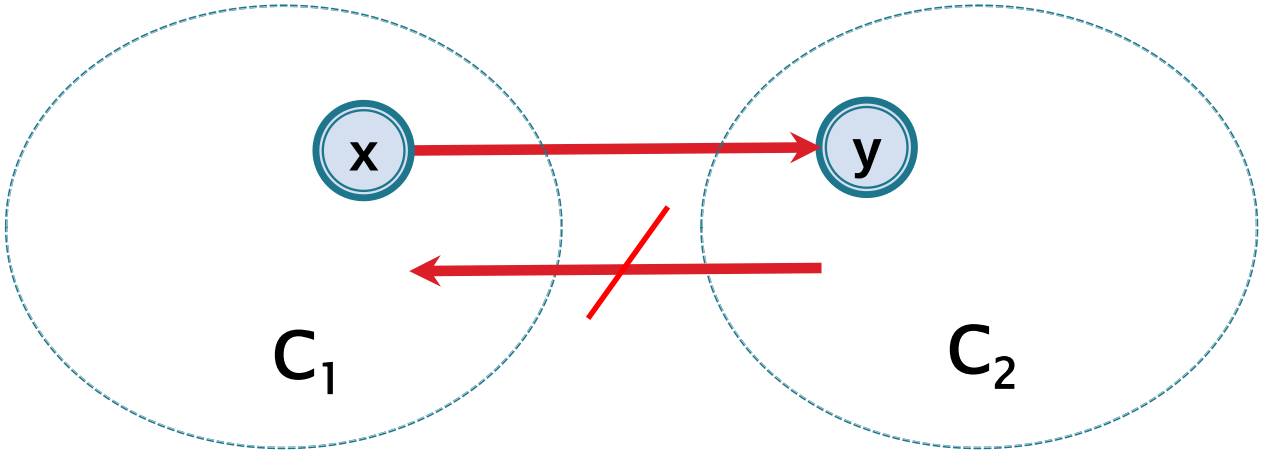
Algoritmi componente tare conexe

Folosind mai puține (un număr constant) de parcurgeri?

Algoritmi componente tare conexe

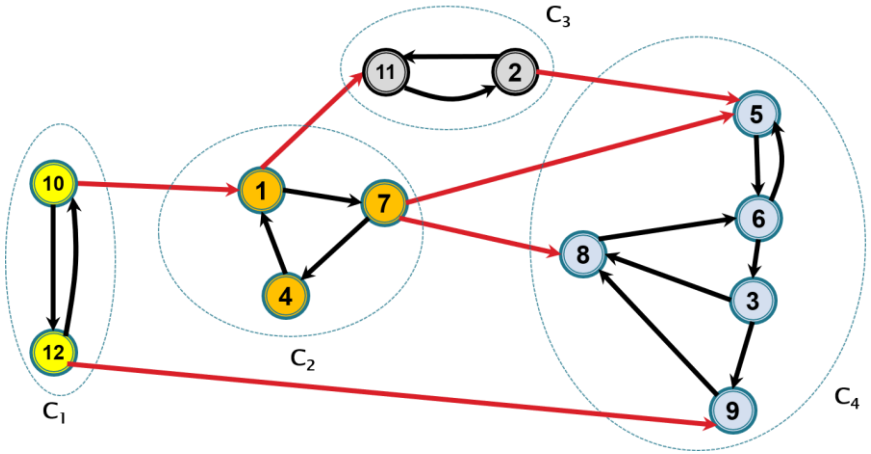
- **Observații**

Dacă C_1 și C_2 sunt componente tare conexe aî există arc de la C_1 la C_2, atunci nu există arc/drum de la C_2 la C_1

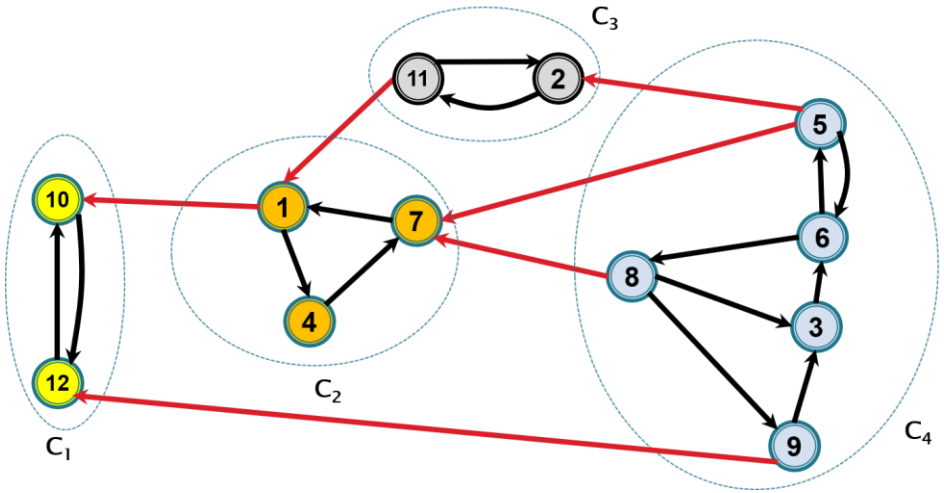


Componente tare conexe

G:

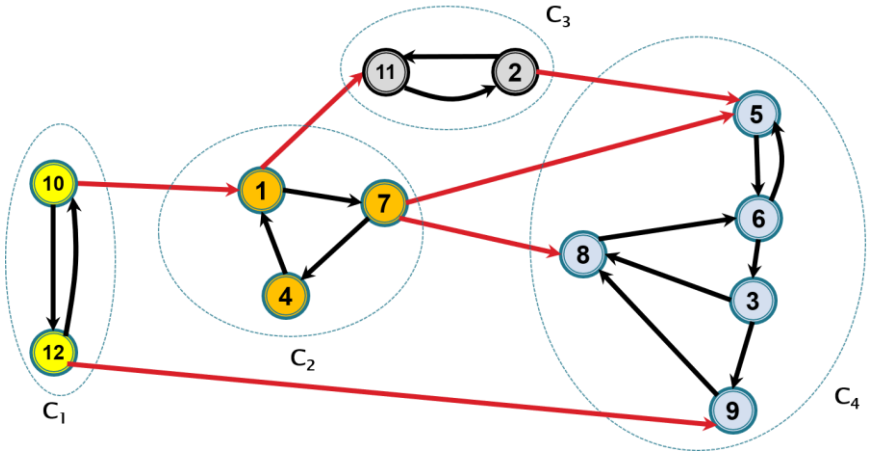


G^T:

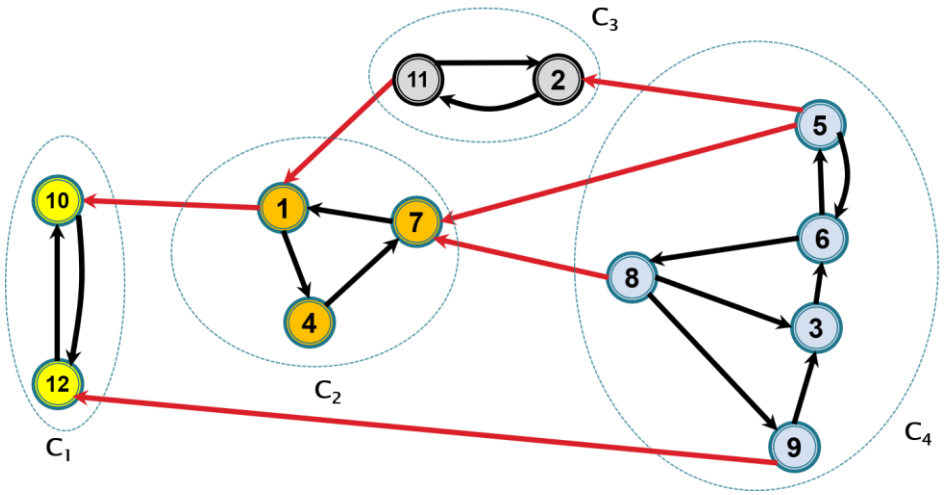


Componente tare conexe

G:



G^T:



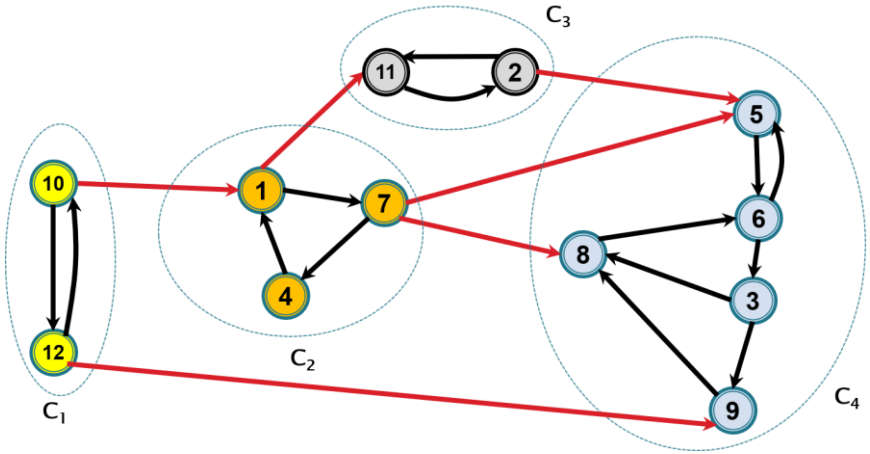
în G^T – ar fi bine sa determinăm întâi componenta lui 10: DF(G^T,10)

(să nu se “amestece” componentele

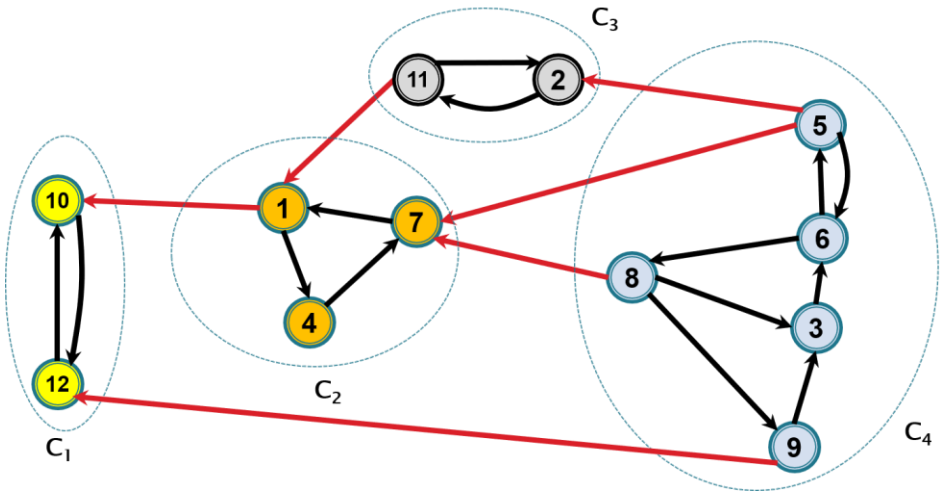
=> nu mai trebuie intersecție cu nodurile vizitate în DF(G,10))

Componente tare conexe

G:



G^T:

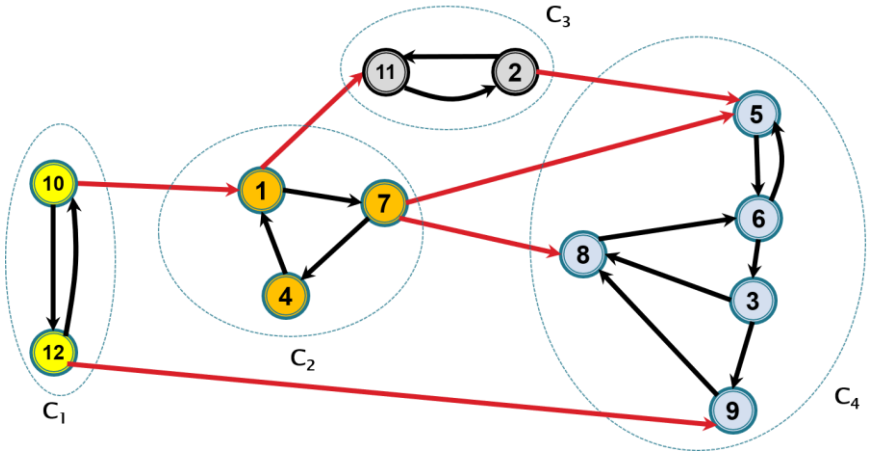


Ar fi bine sa determinăm întâi componenta lui 10 în G^T (“de la margine”).

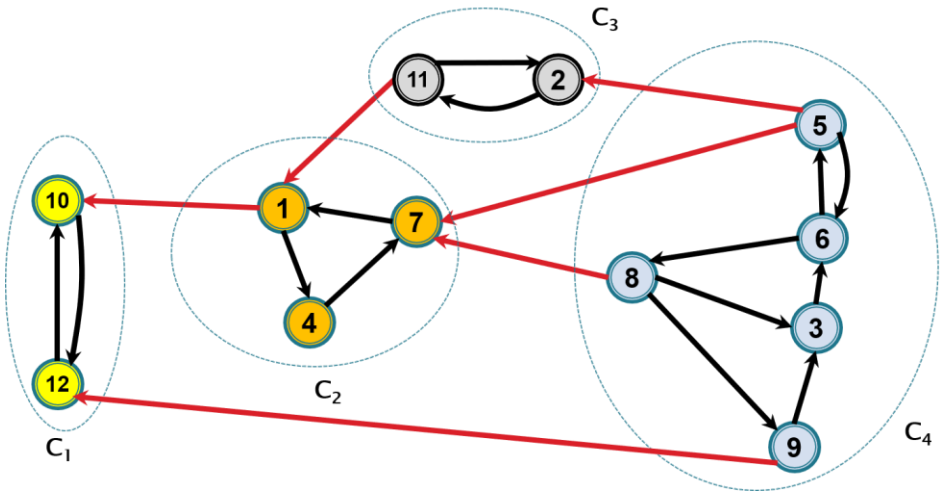
Dar ce proprietate are vârful 10?

Componente tare conexe

G:



G^T:



Ar fi bine sa determinăm întâi componenta lui 10 în G^T ("de la margine").

Dar ce proprietate are vârful 10? =>

Este ultimul finalizat în DF pentru G

Algoritmi componente tare conexe

Folosind mai puține (un număr constant) de parcurgeri?

Algoritmi componente tare conexe

- Folosind doar două parcurgeri, una în G și una în G^T?

DA, dar a doua într-o ordine particulară a vârfurilor (în funcție de ordinea în care au fost finalizate de parcurgerea DF în G)

$\Rightarrow$ Algoritmul lui Kosaraju

Algoritmi componente tare conexe

- Dar folosind o singură parcurgere?

DA, folosind o idee similară cu cea de la componente biconexe (vom discuta)

$\Rightarrow$ Algoritmul lui Tarjan (SUPLIMENTAR)

Algoritmul lui Kosaraju

Algoritmul lui Kosaraju

- **Pasul 0.** Constuim G^T

Algoritmul lui Kosaraju

- **Pasul 1.** Parcurgem DFS graful G +

  introducem într-o stivă S fiecare varf la momentul la care este finalizat

  (pentru a obține o ordonare descrescătoare a varfurilor după timpul de finalizare)

```
stack S

DFS(G, i)
     viz[i] = 1
     pentru ij ∈ E(G)
          daca viz[j]==0 atunci
              DFS(j)
     push(S, i) //i este finalizat

pentru x ∈ V executa
     daca viz[x]==0 atunci
            DFS(G, x)
```

Algoritmul lui Kosaraju

- **Pasul 2.** Parcurgem DFS graful G^T considerând vârfurile în ordinea în care sunt extrase din S (descrescătoare după timpul de finalizare de la Pasul 1):

```
marcăm toate vârfurile ca fiind nevizitate
cat timp S este nevida
    x = pop(S)
    daca x este nevizitat atunci
        DFS(G^T , x)
        afiseaza componenta tare conexă (formată cu
        varfurile vizitate in DFS(G^T,x))
```

Algoritmul lui Kosaraju

Complexitate:

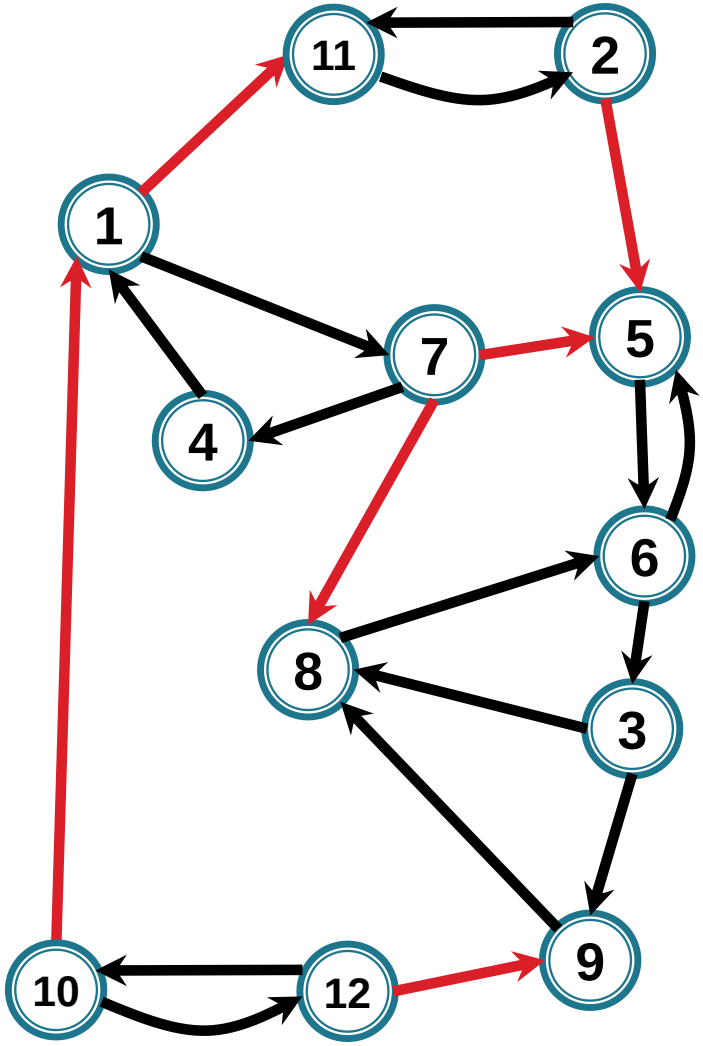
2 parcurgeri + construcția lui G^T => $O(n+m)$

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.

DFS(1)

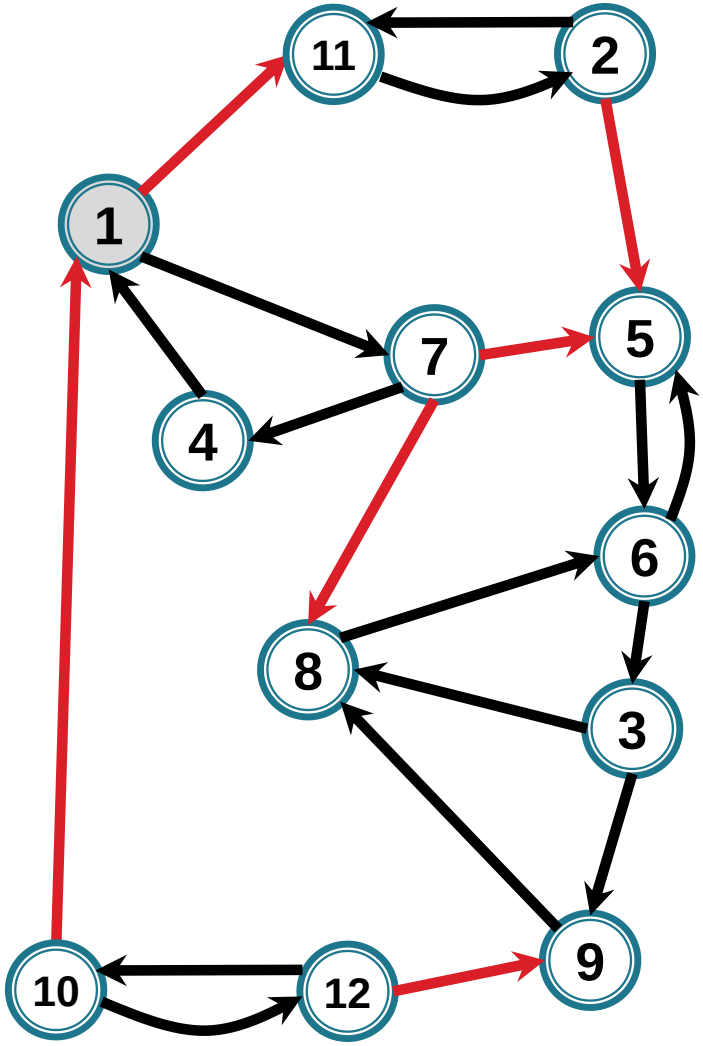


Ordinea descrescătoare finalizare:

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.

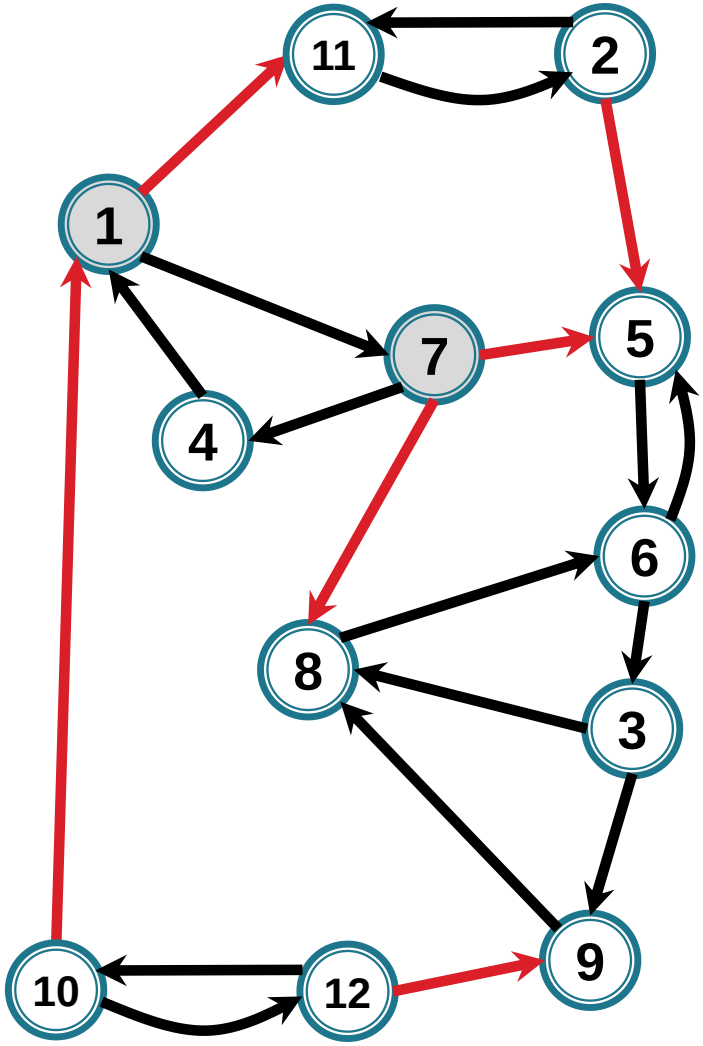


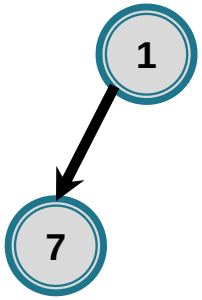
Ordinea descrescătoare finalizare:

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.



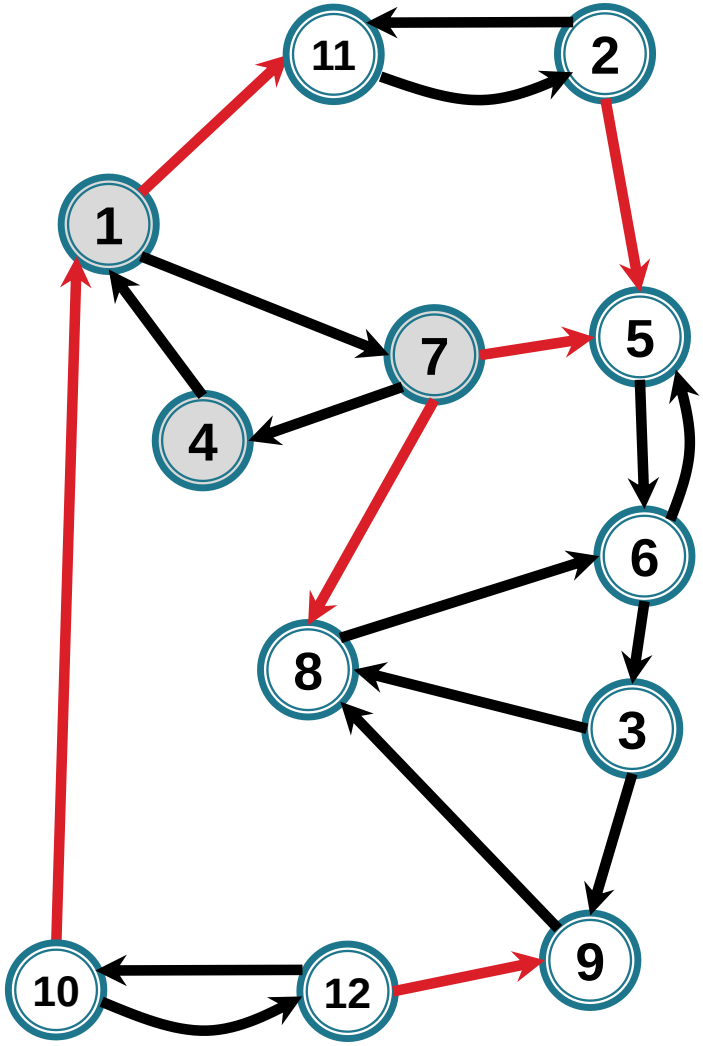


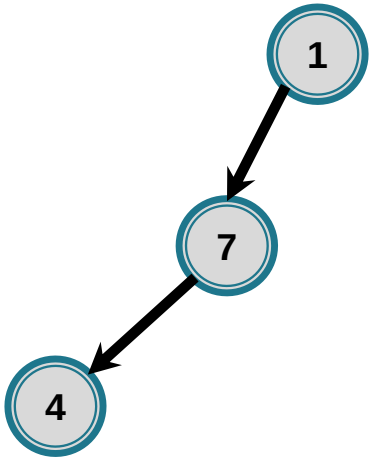
Ordinea descrescătoare finalizare:

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.



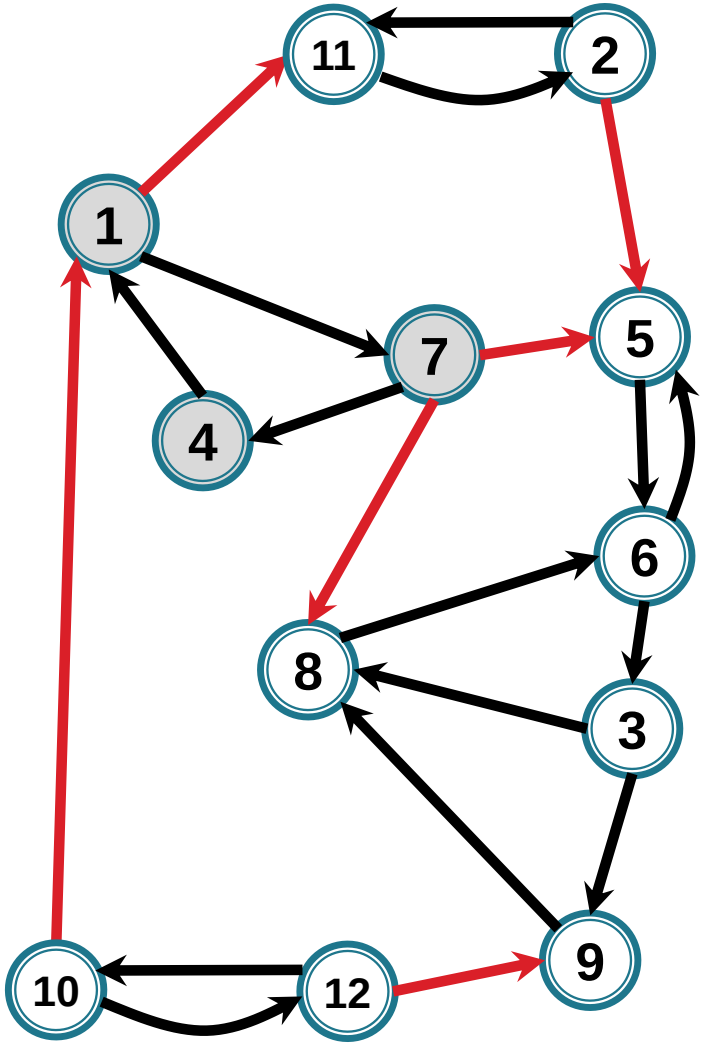


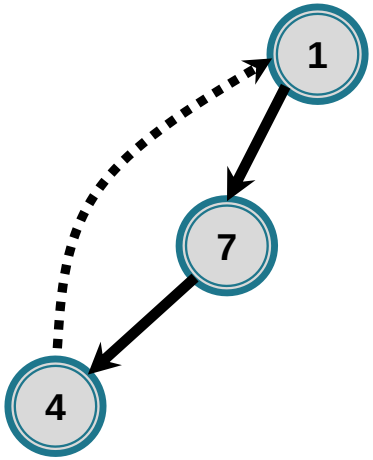
Ordinea descrescătoare finalizare:

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.



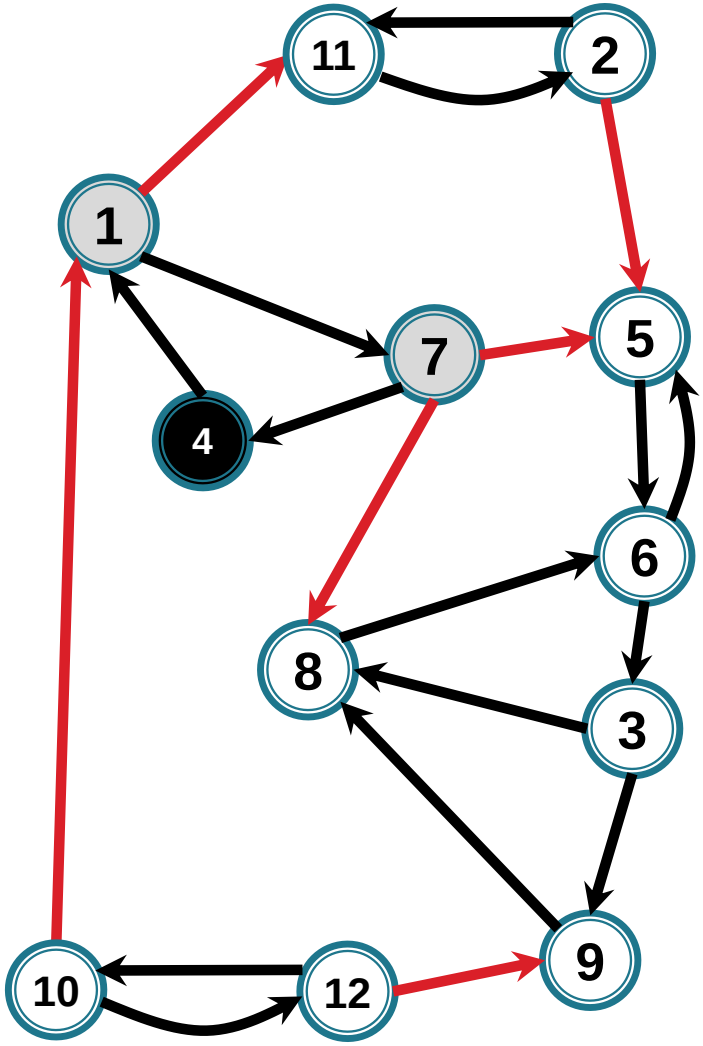


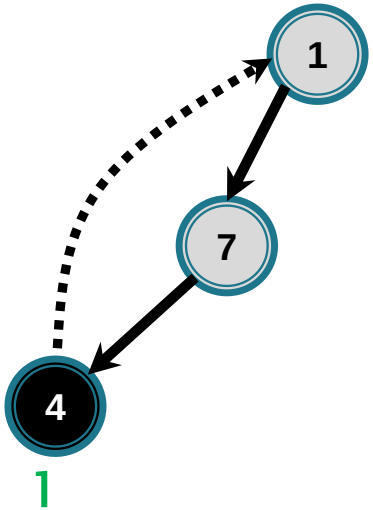
Ordinea descrescătoare finalizare:

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.





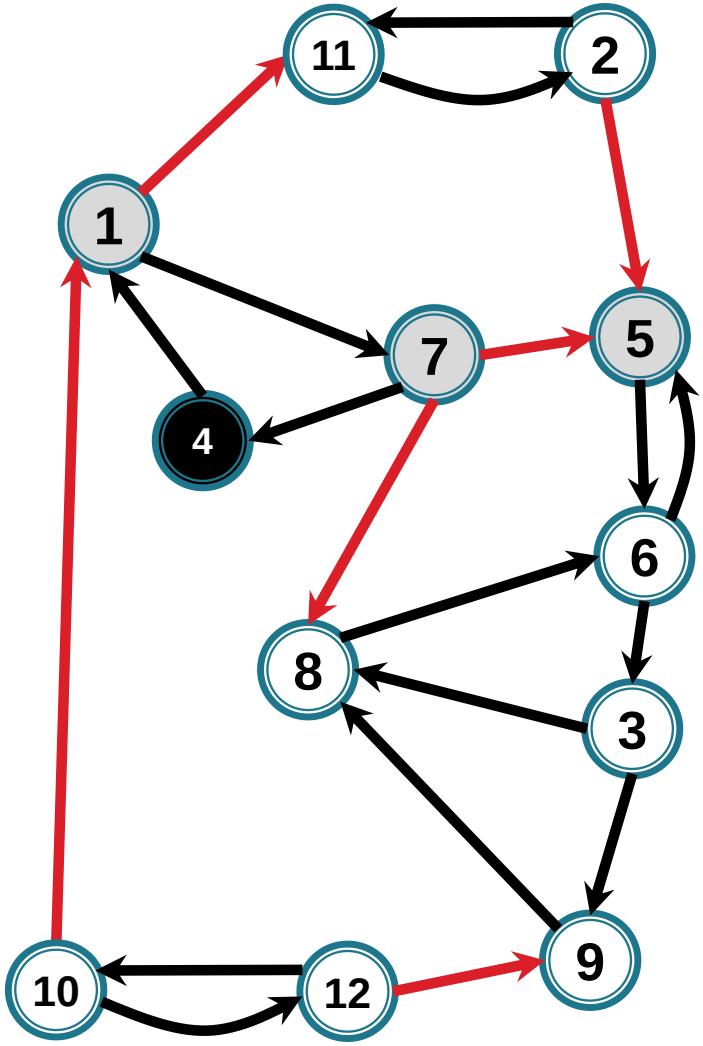
Ordinea descrescătoare finalizare:

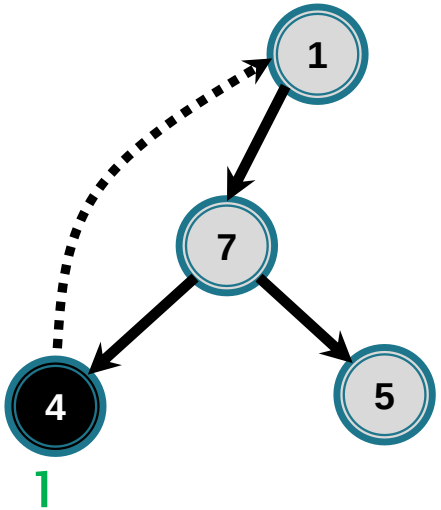
4

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.





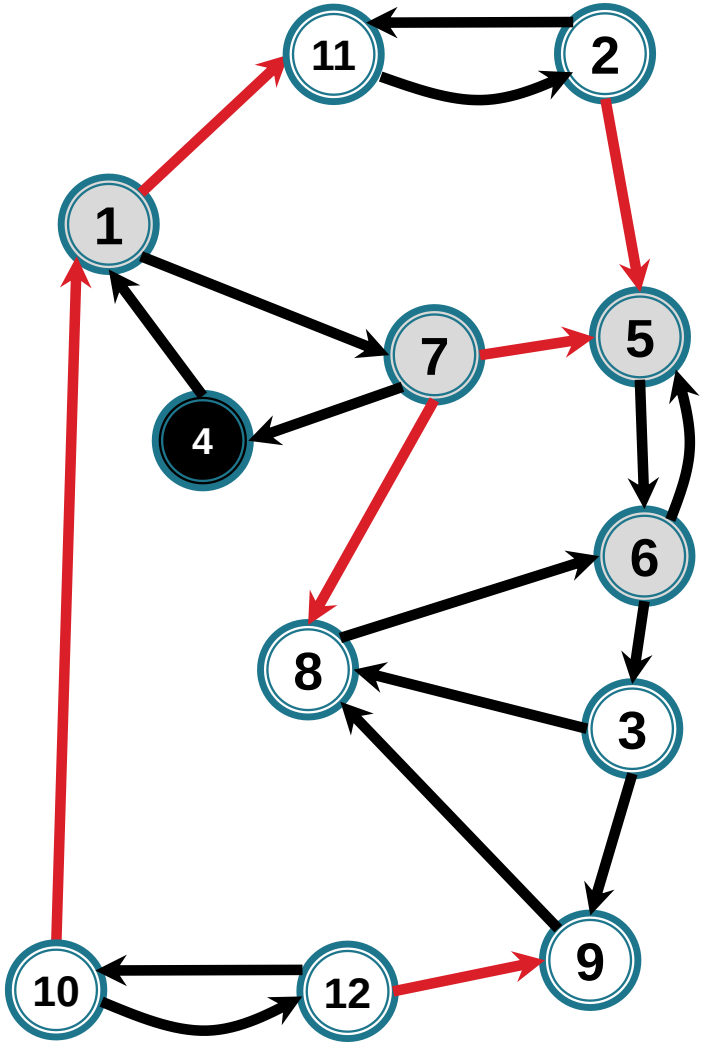
Ordinea descrescătoare finalizare:

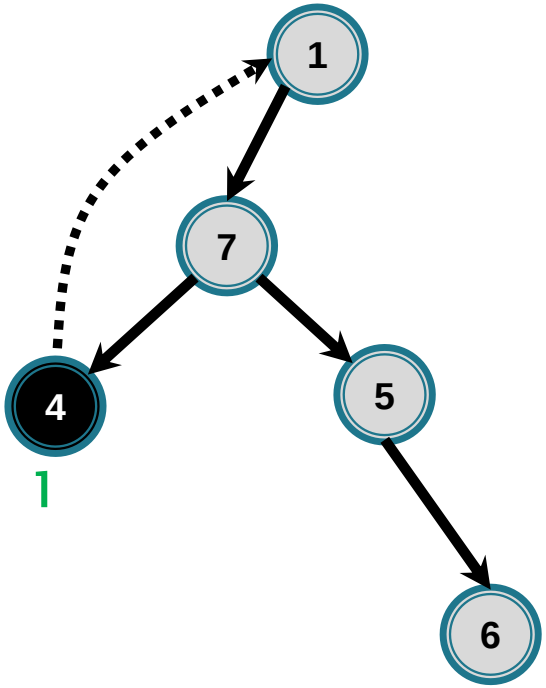
4

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.





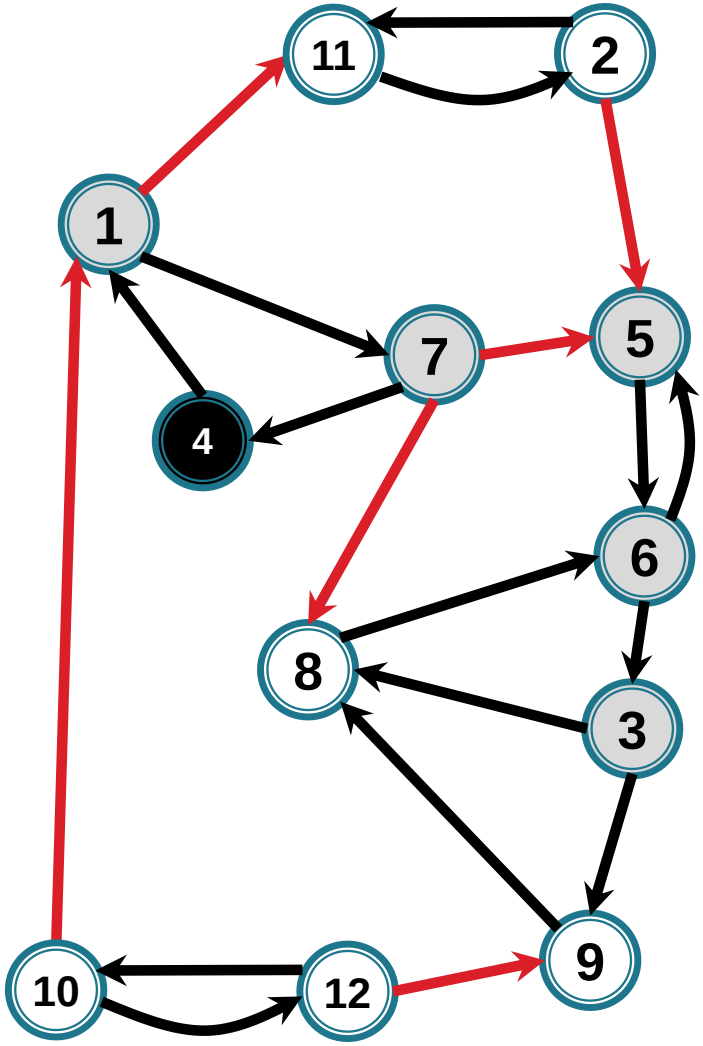
Ordinea descrescătoare finalizare:

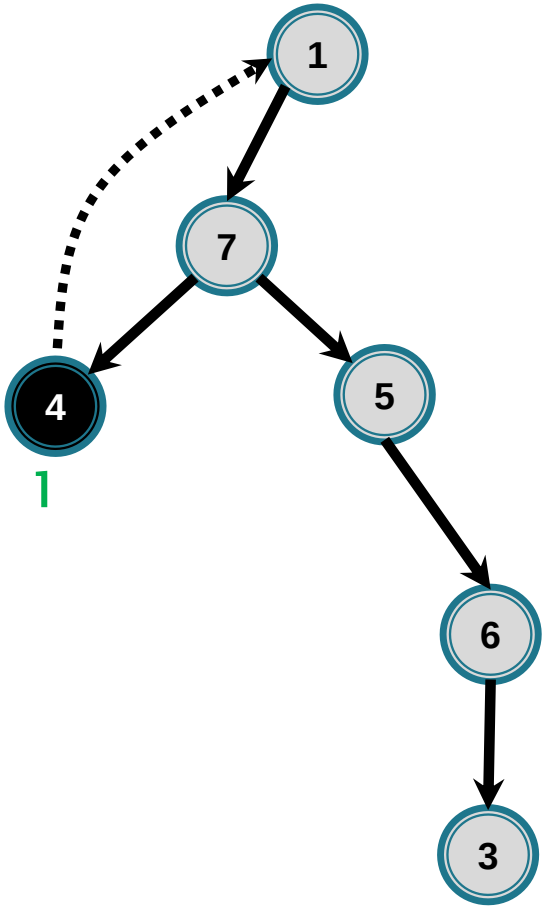
4

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.



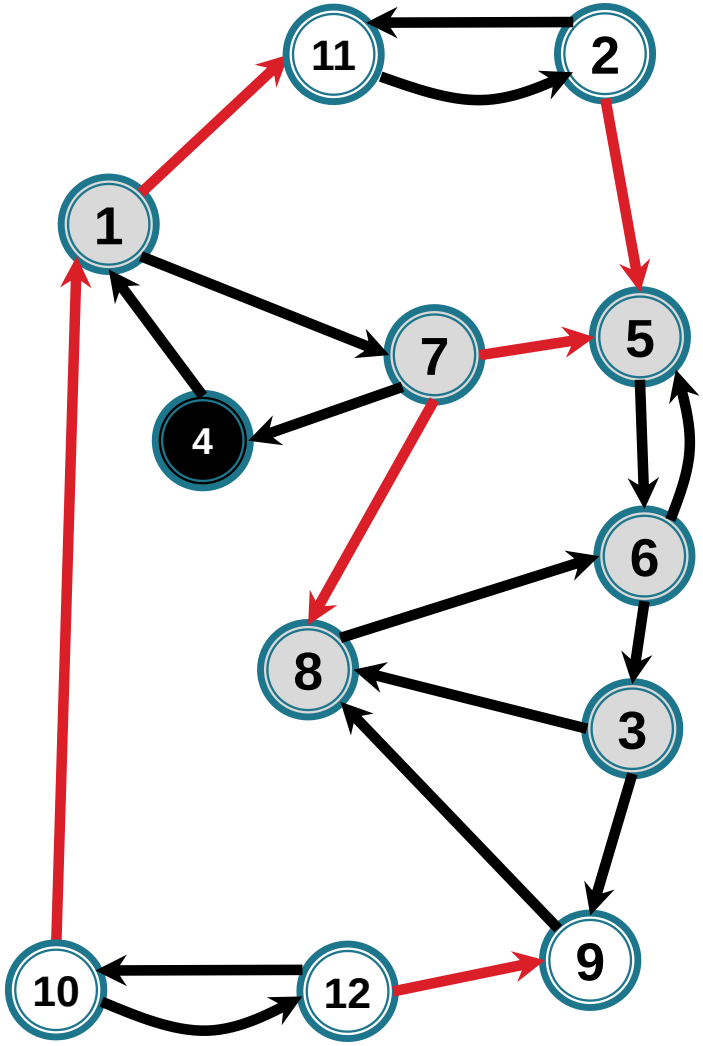


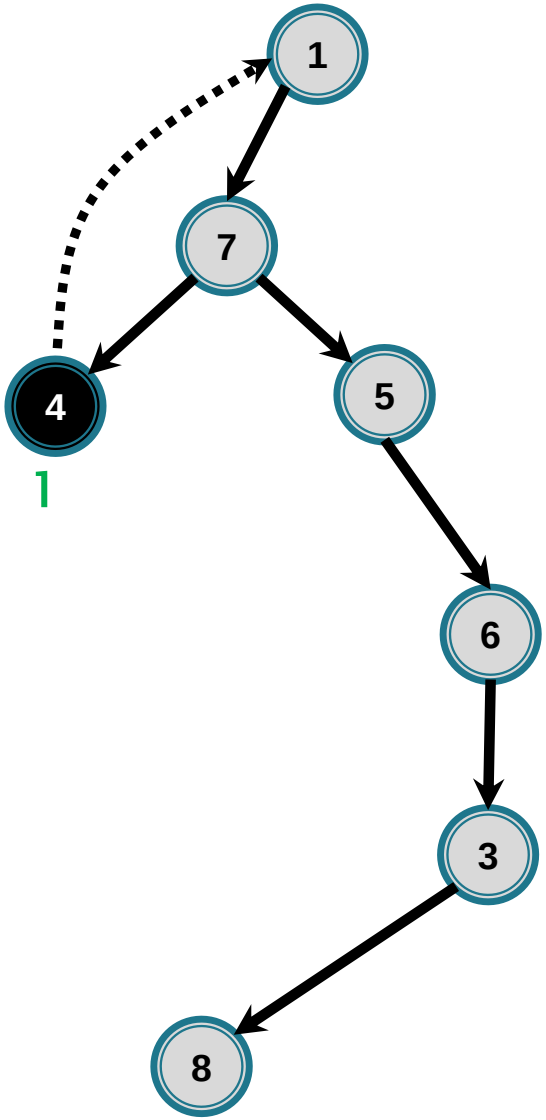
Ordinea descrescătoare finalizare:

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.



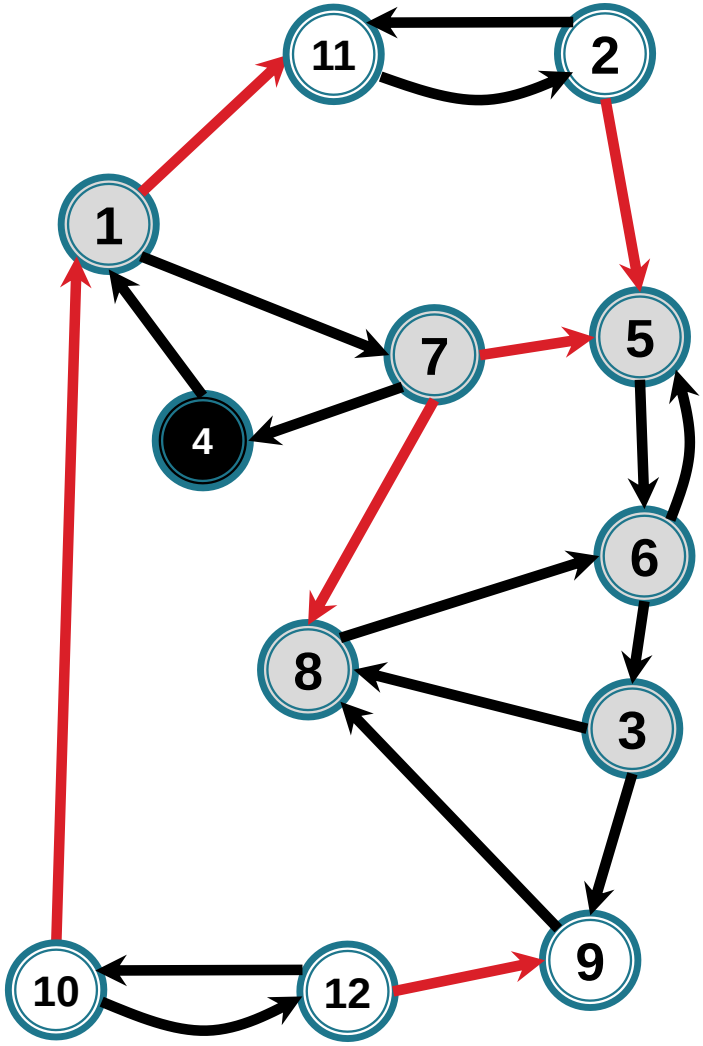


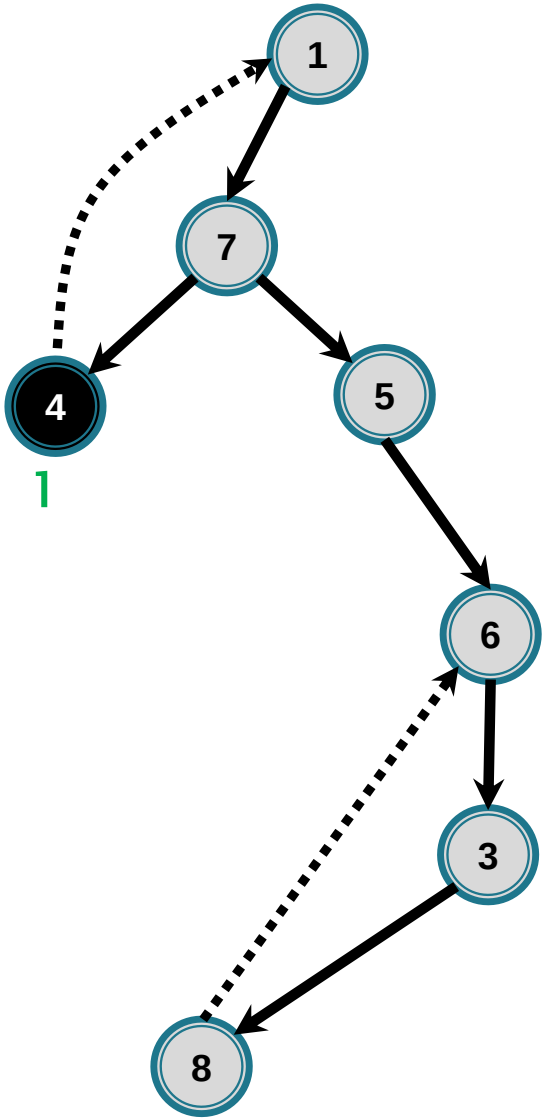
Ordinea descrescătoare finalizare:

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.



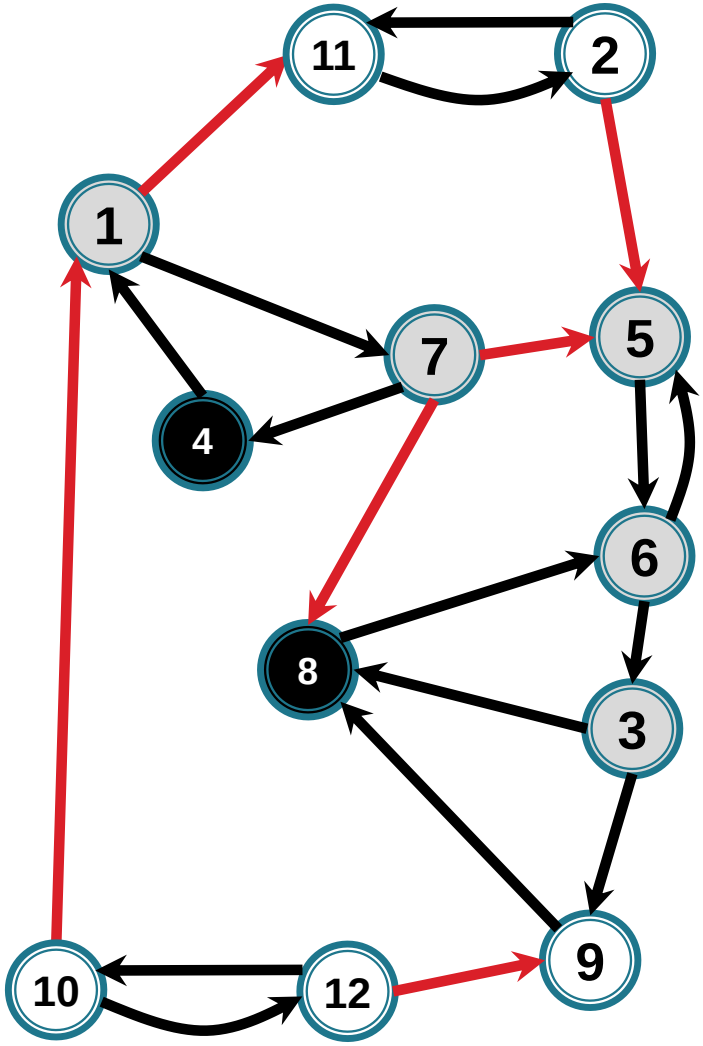


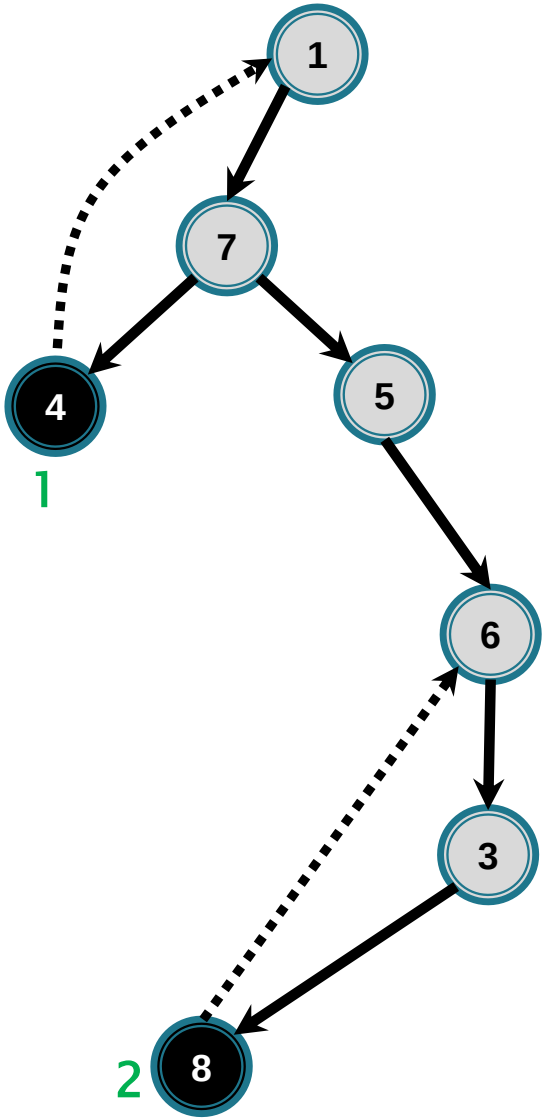
Ordinea descrescătoare finalizare:

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.



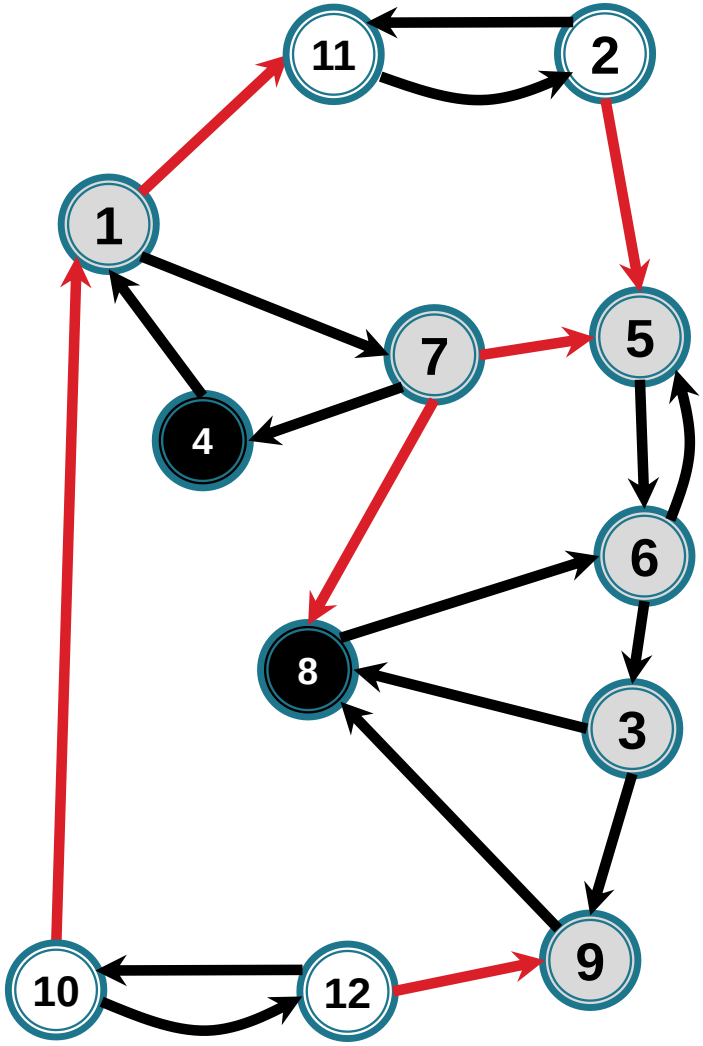


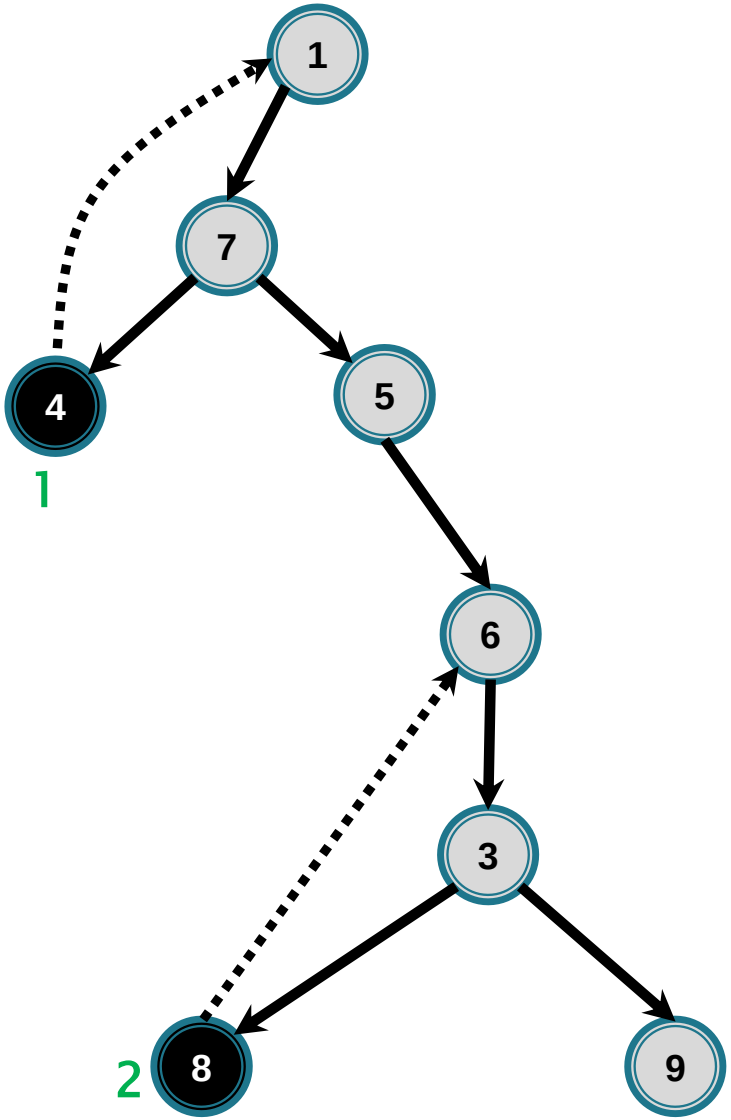
Ordinea descrescătoare finalizare: 8, 4

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.



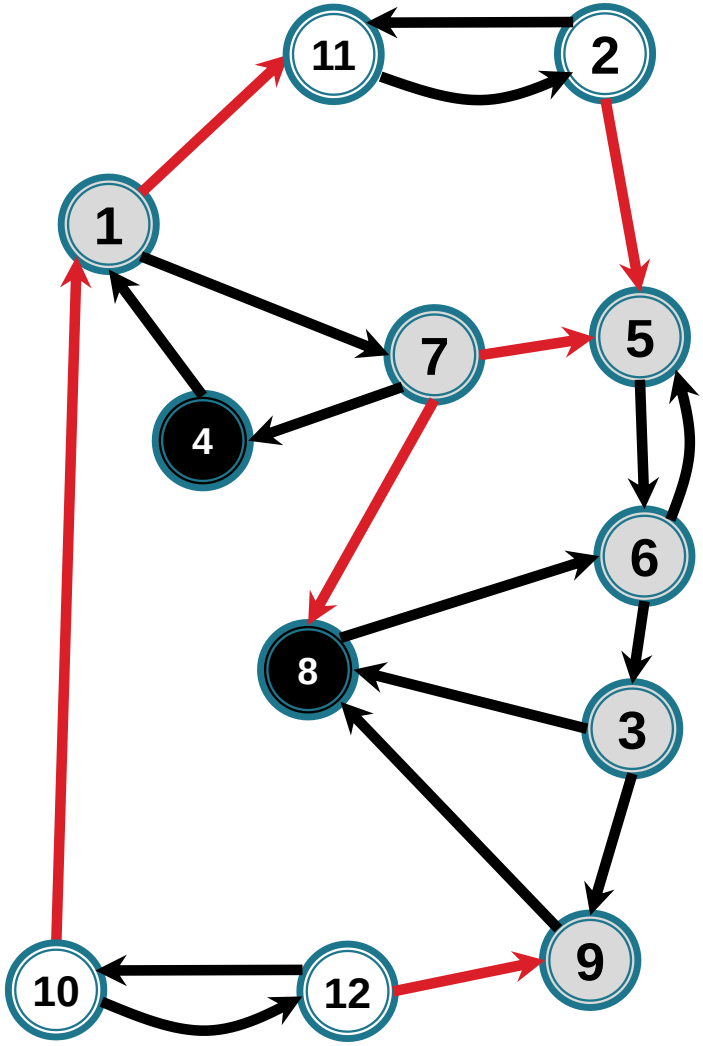


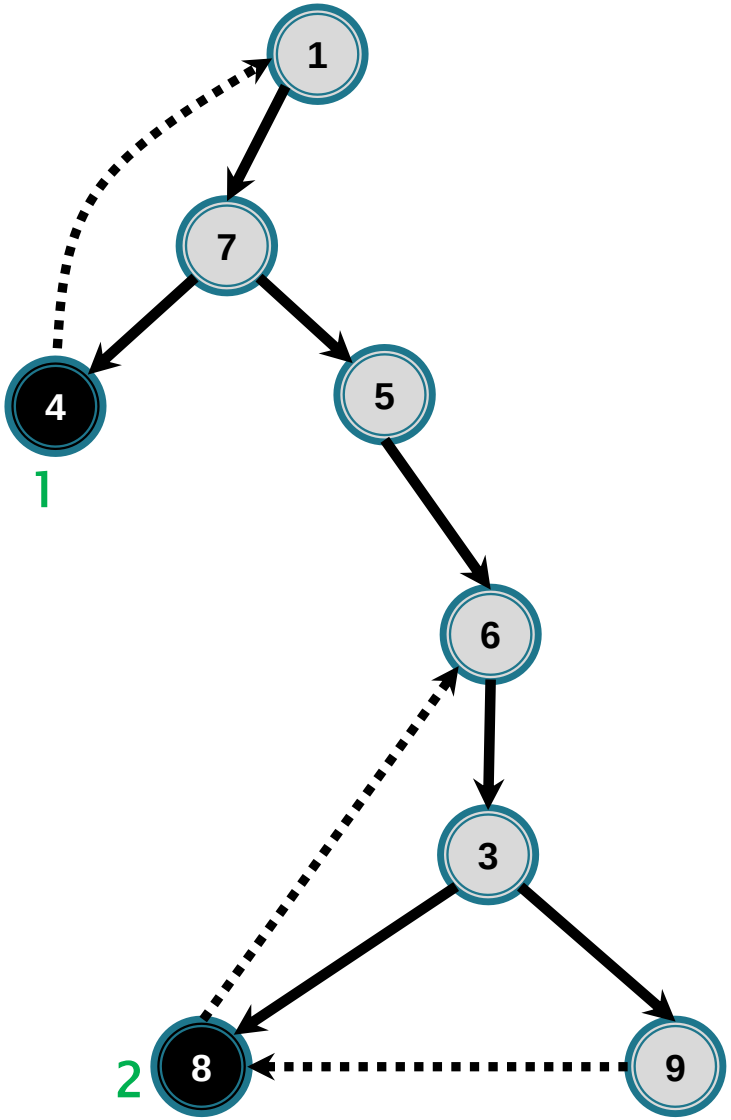
Ordinea descrescătoare finalizare: 8, 4

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.



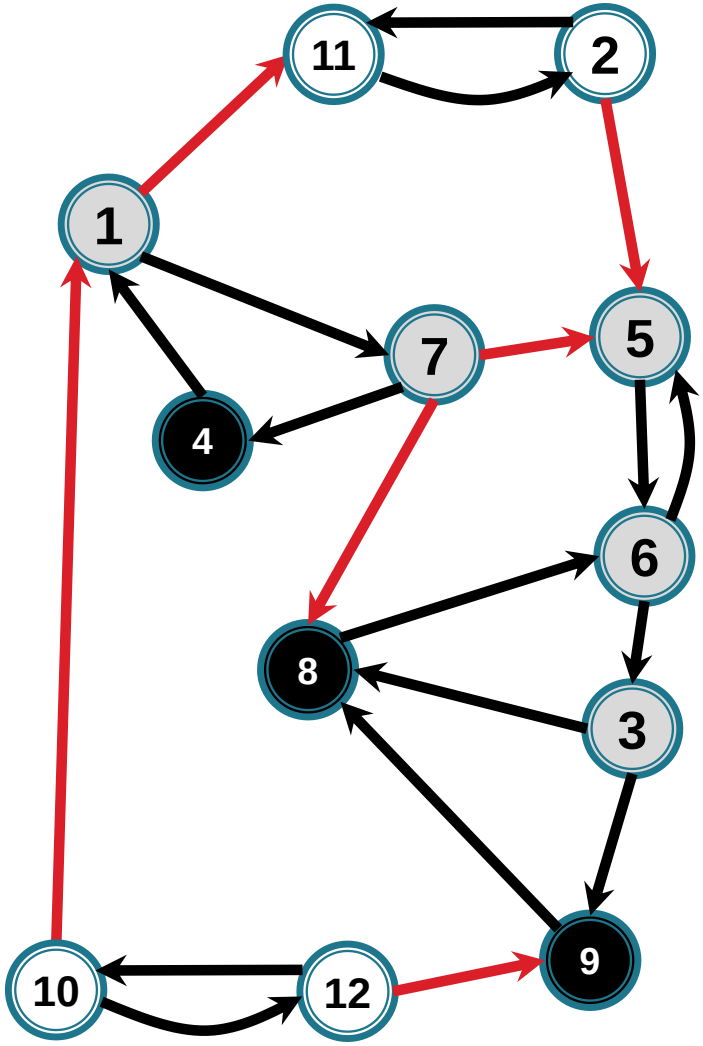


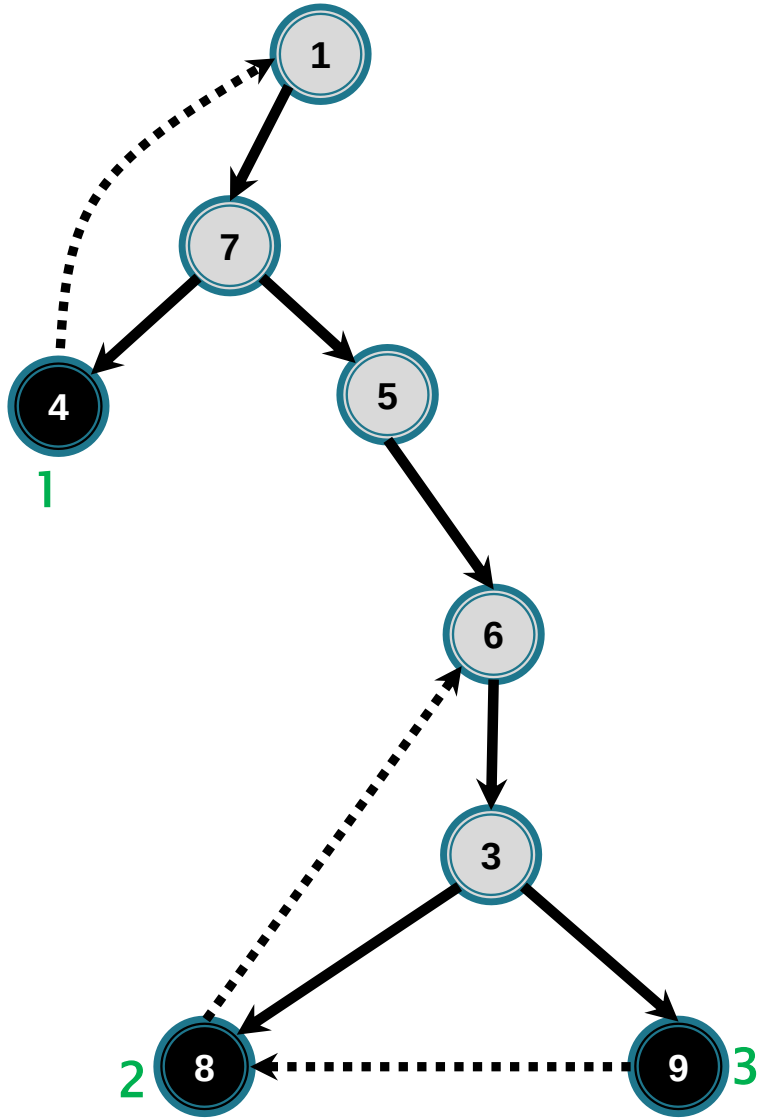
Ordinea descrescătoare finalizare: 8, 4

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.



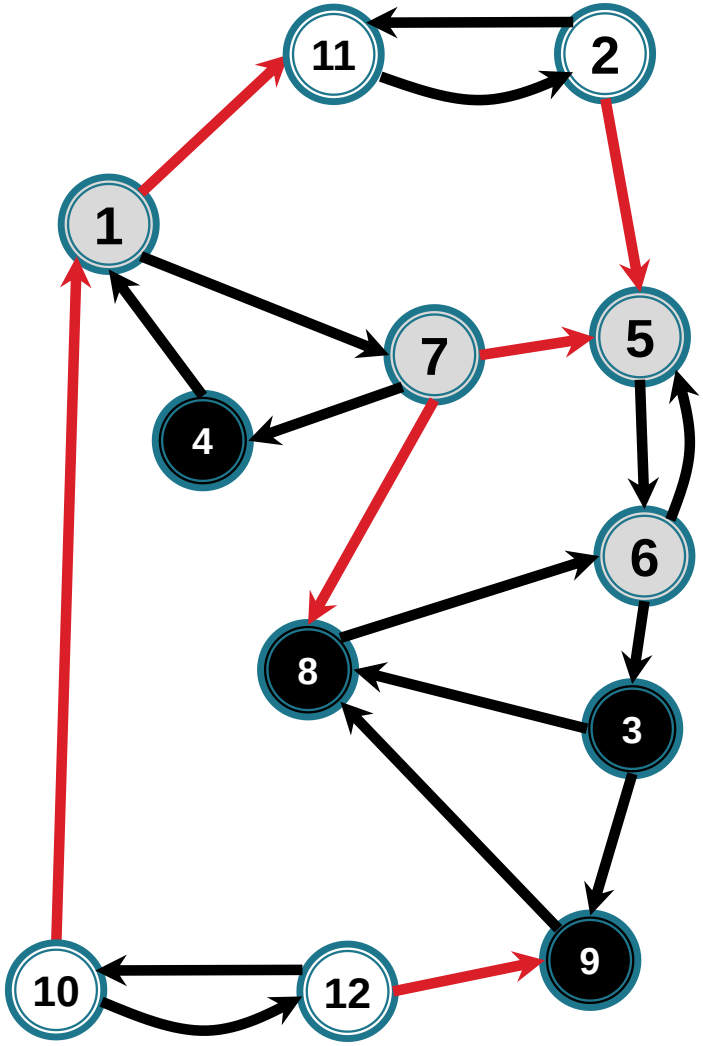


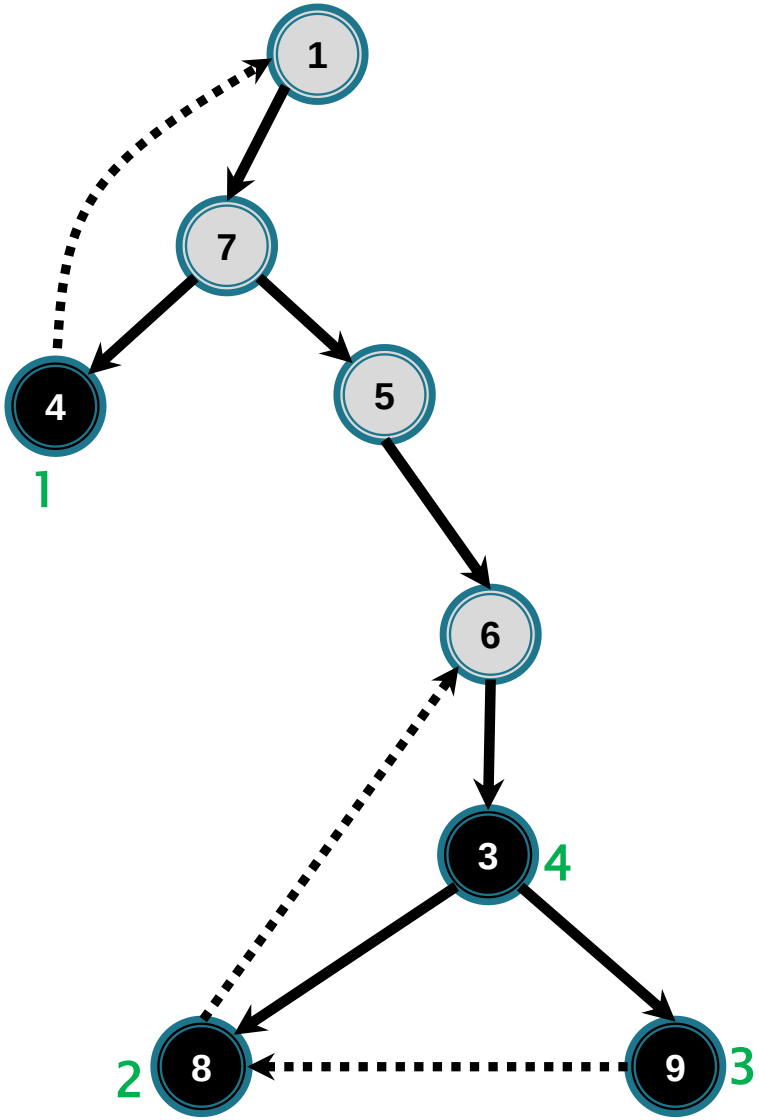
Ordinea descrescătoare finalizare: 9, 8, 4

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.



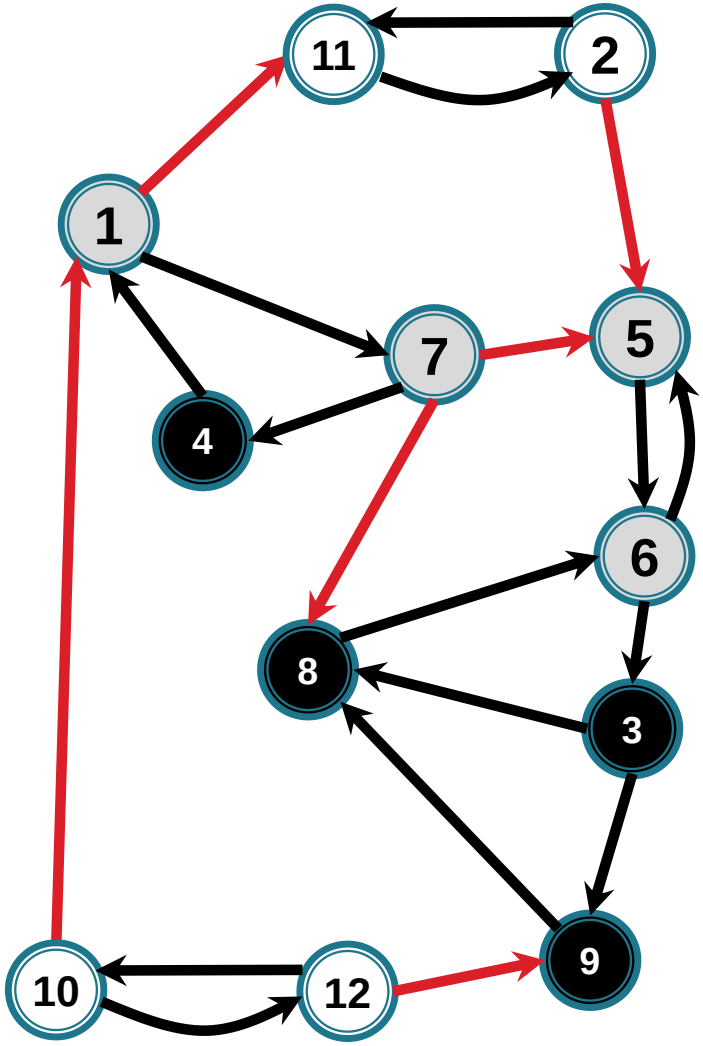


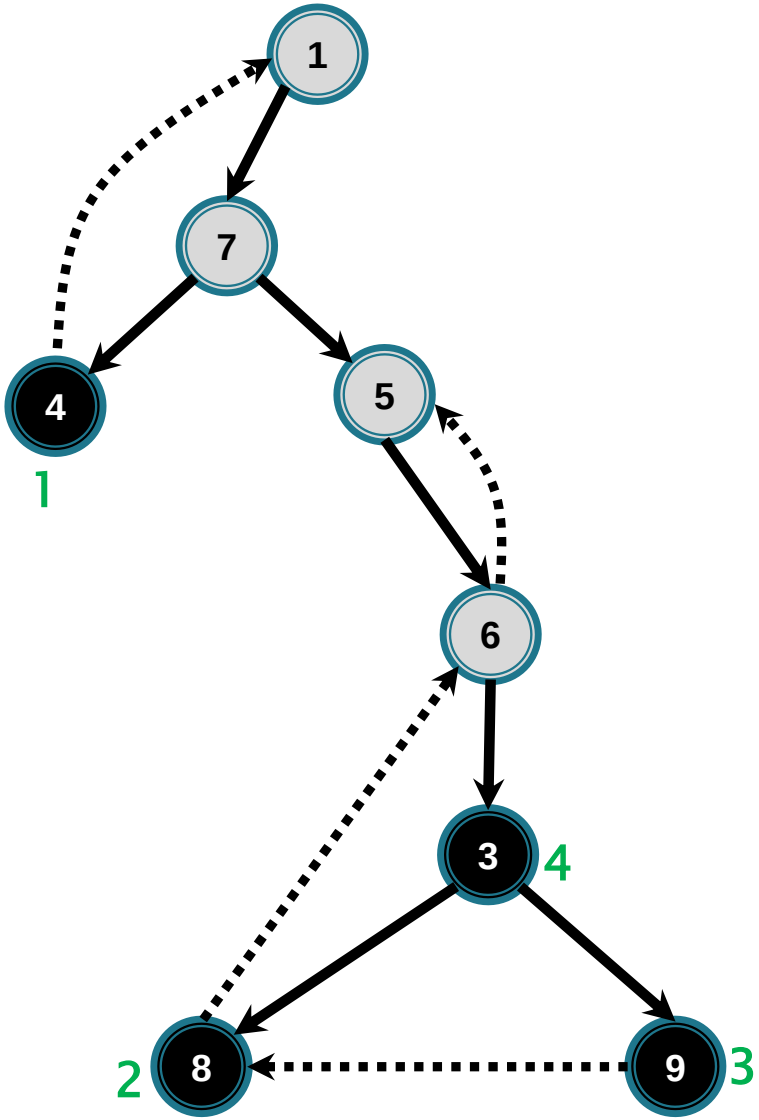
Ordinea descrescătoare finalizare: 3, 9, 8, 4

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.



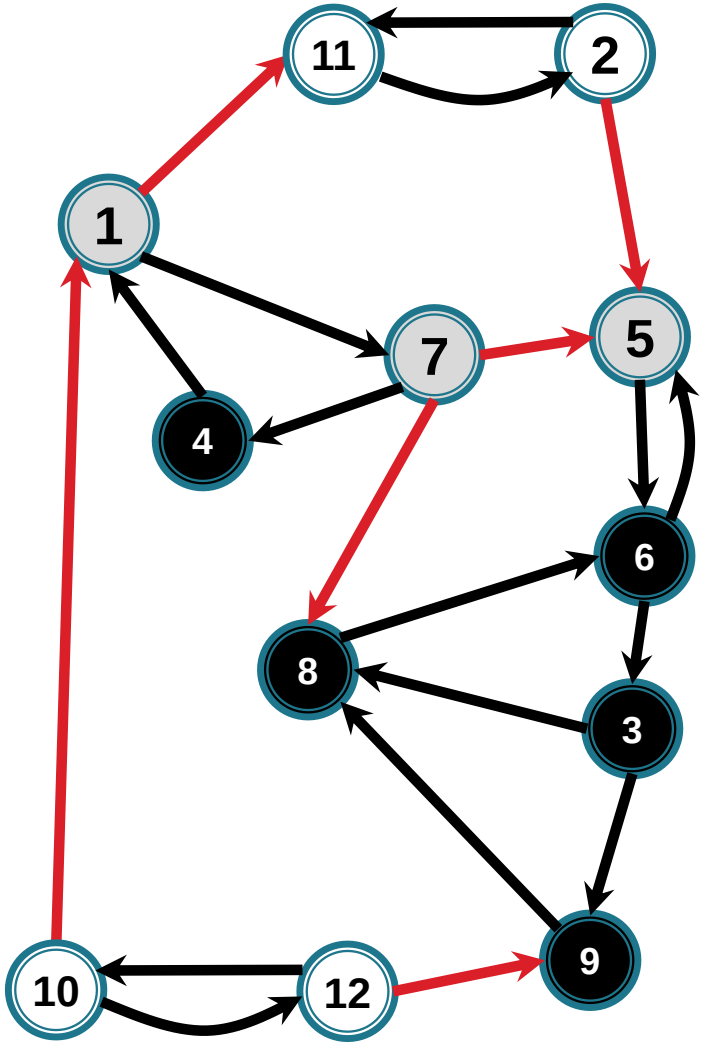


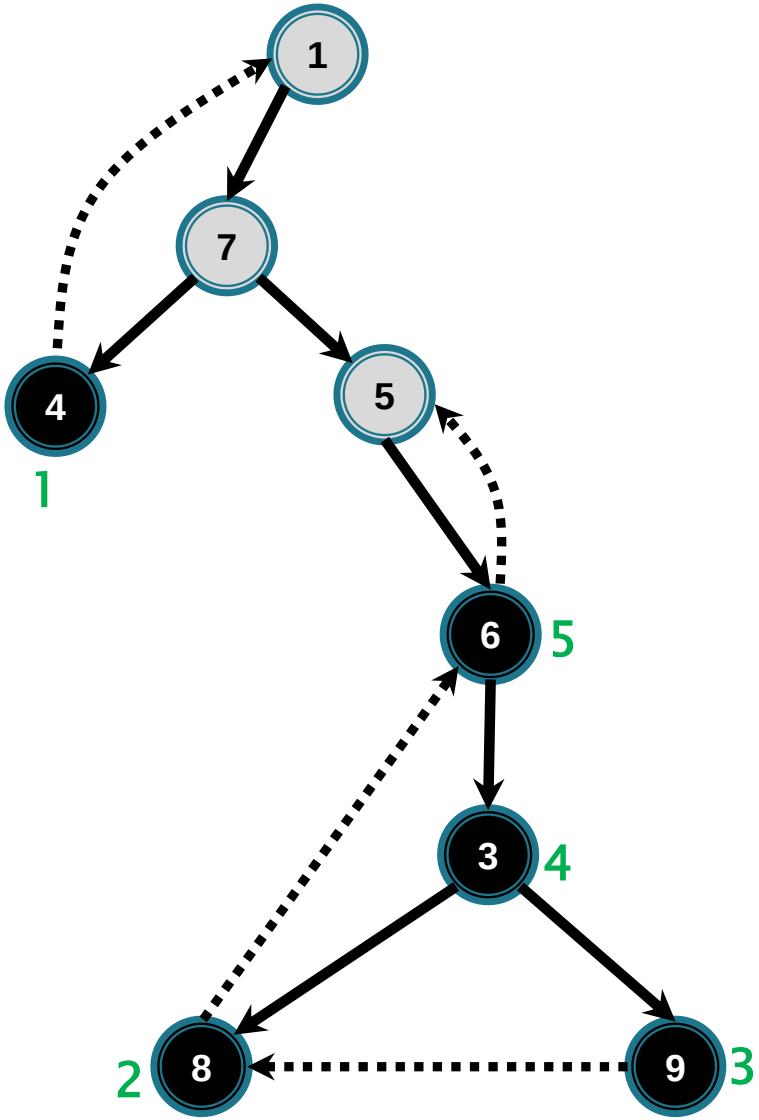
Ordinea descrescătoare finalizare: 3, 9, 8, 4

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.



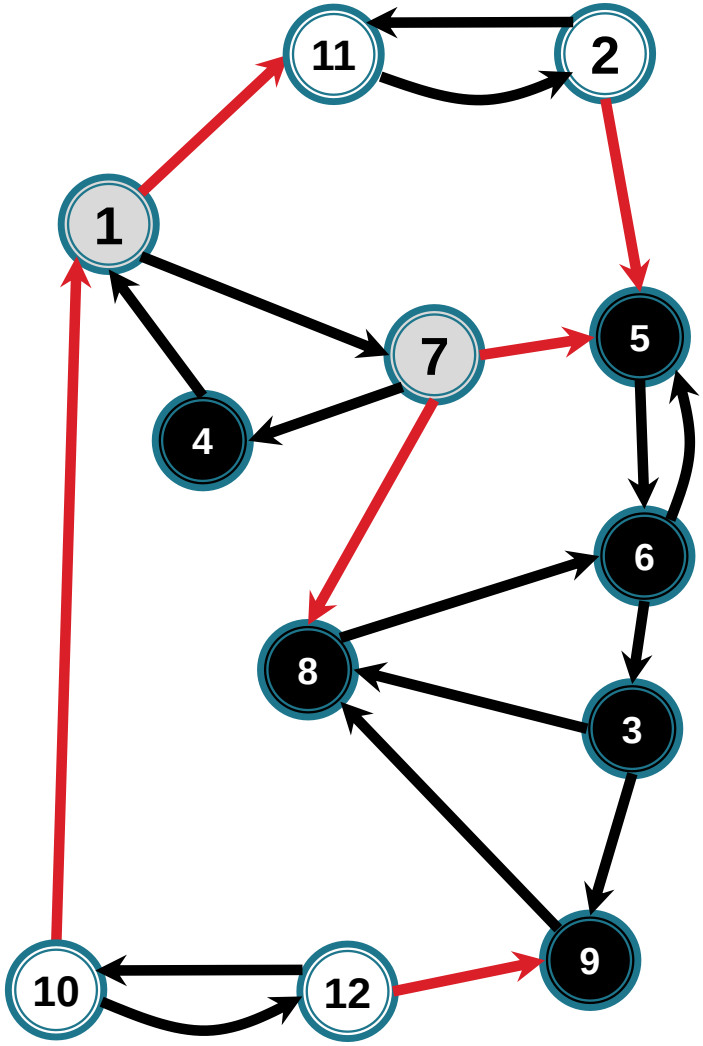


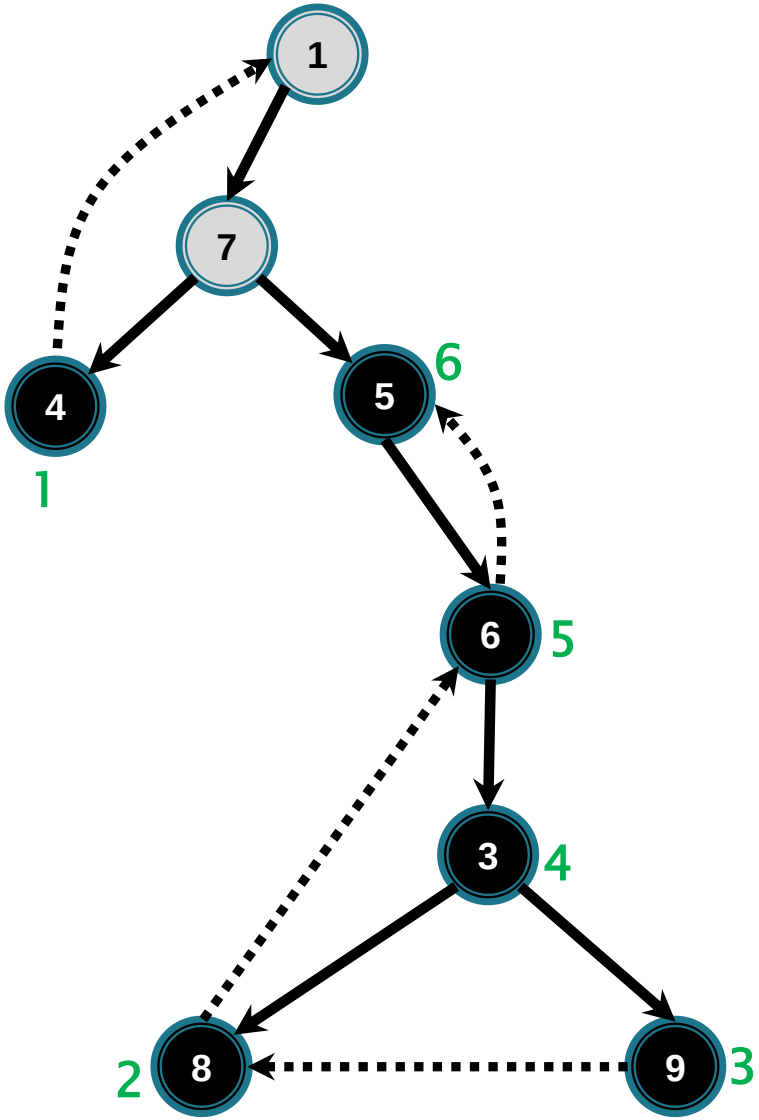
Ordinea descrescătoare finalizare: 6, 3, 9, 8, 4

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.



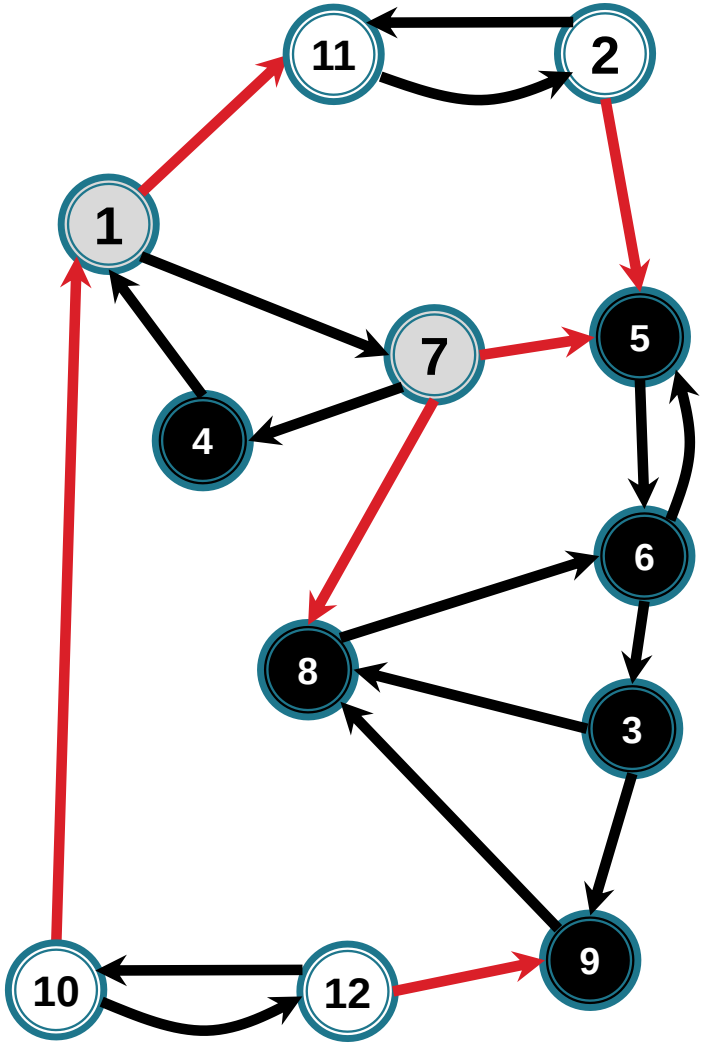


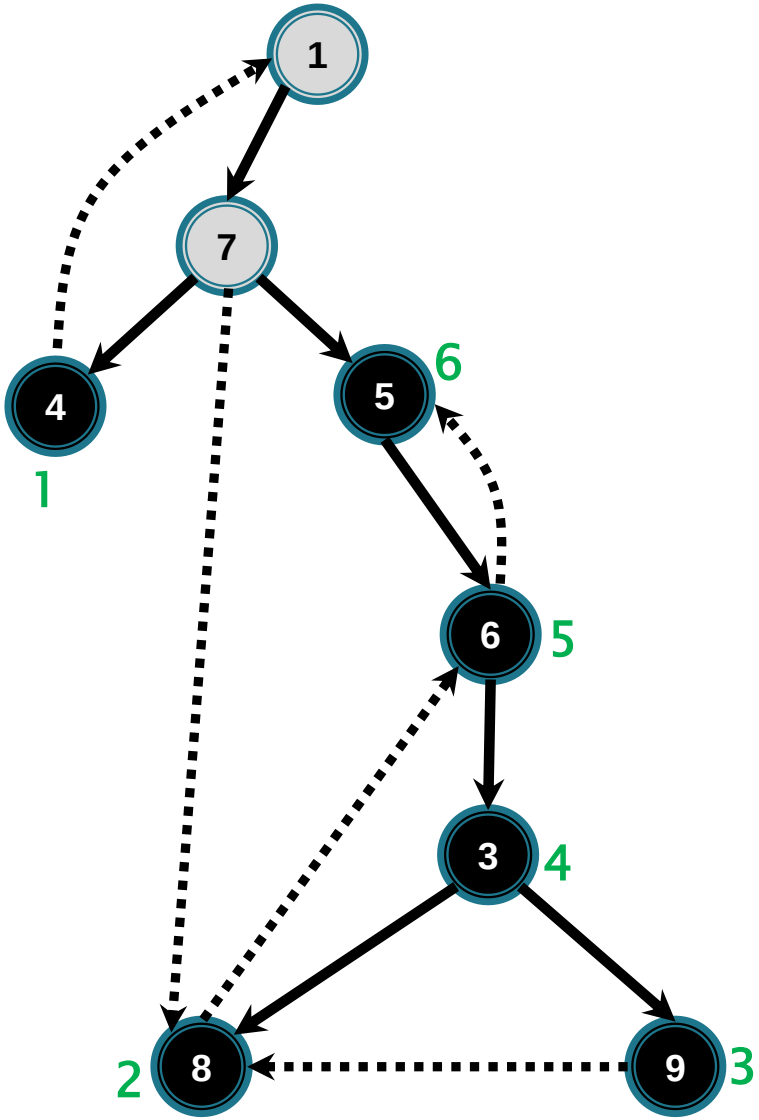
Ordinea descrescătoare finalizare: 5, 6, 3, 9, 8, 4

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.



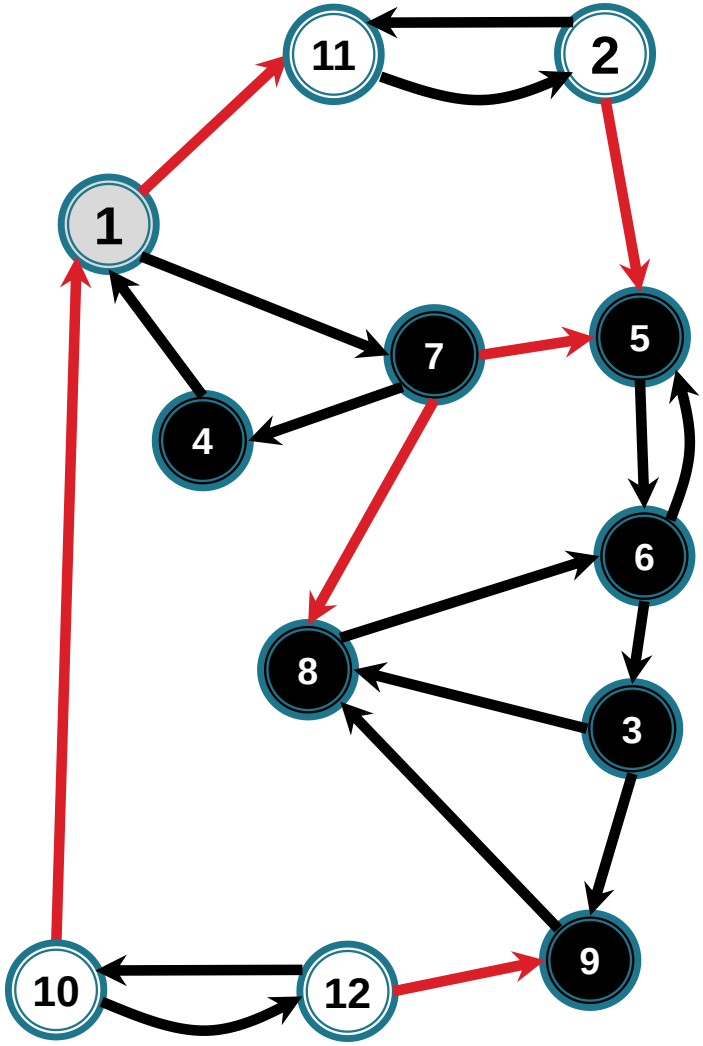


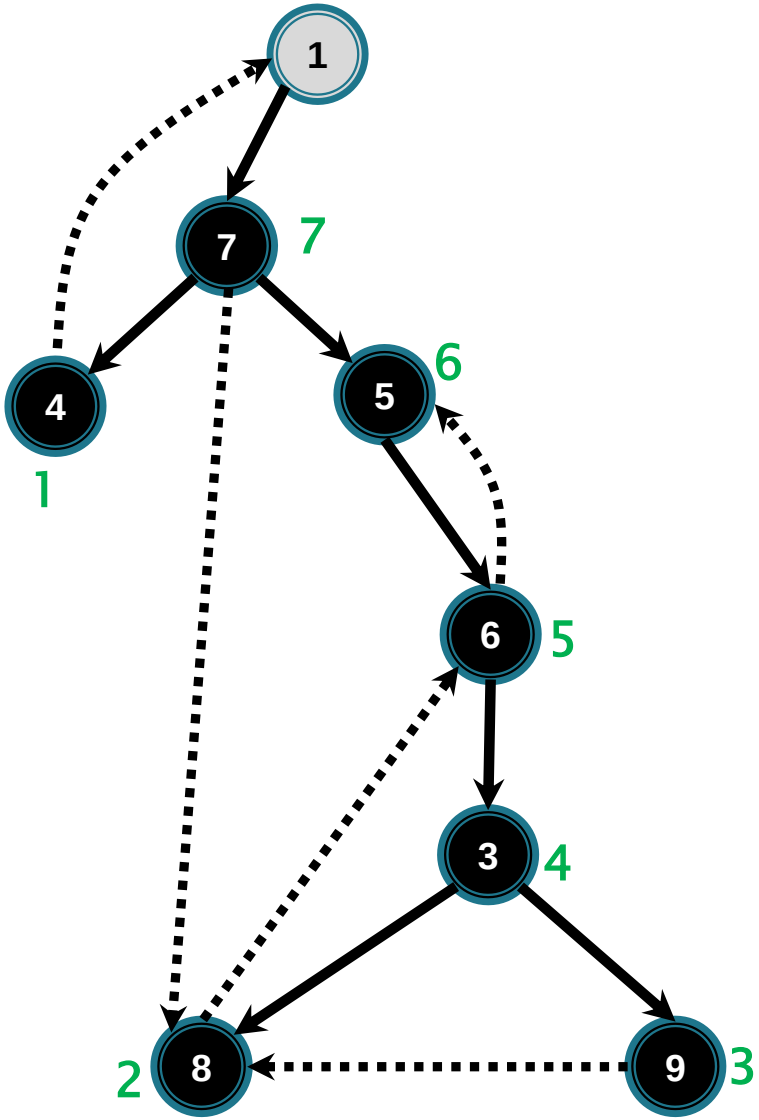
Ordinea descrescătoare finalizare: 5, 6, 3, 9, 8, 4

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.



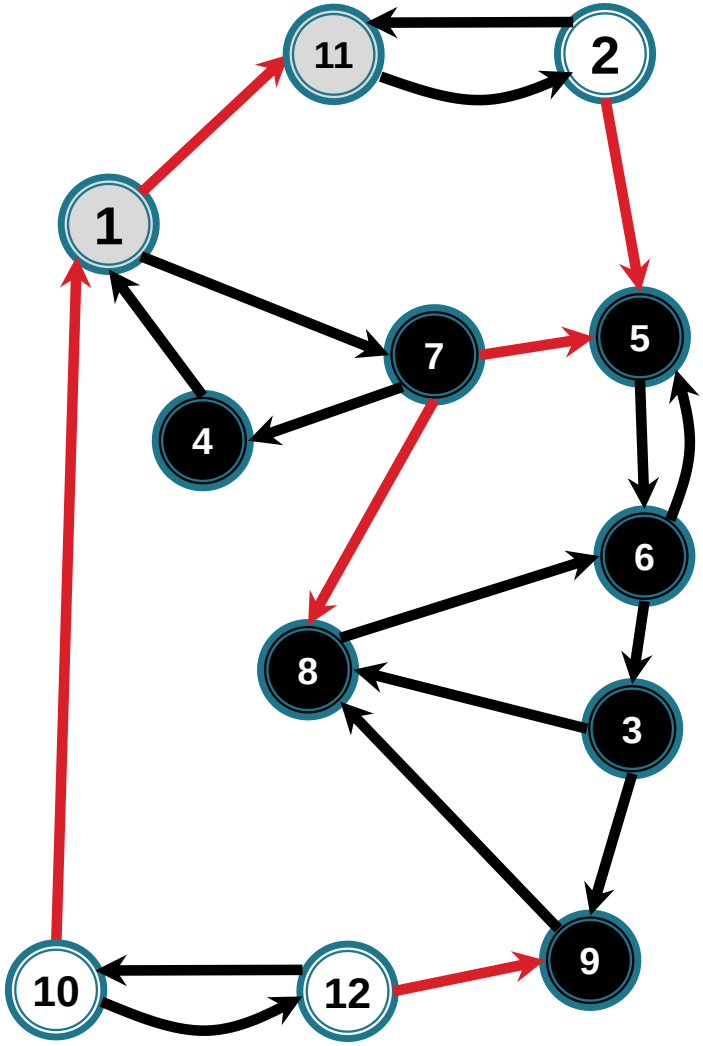


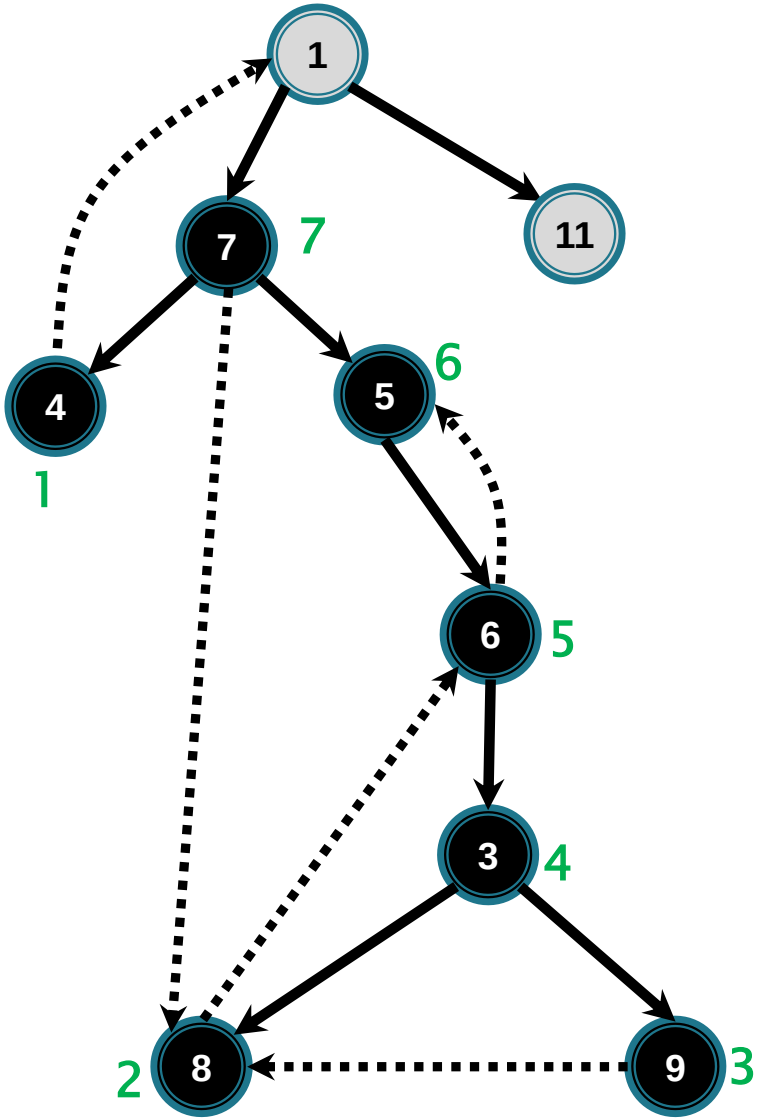
Ordinea descrescătoare finalizare: 7, 5, 6, 3, 9, 8, 4

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.



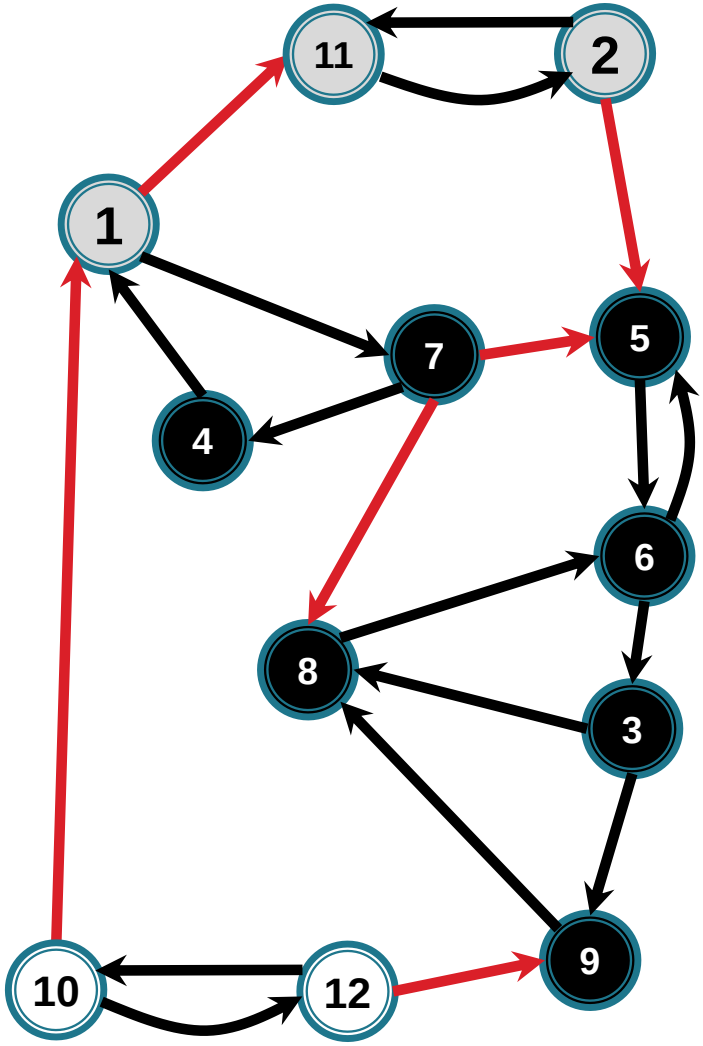


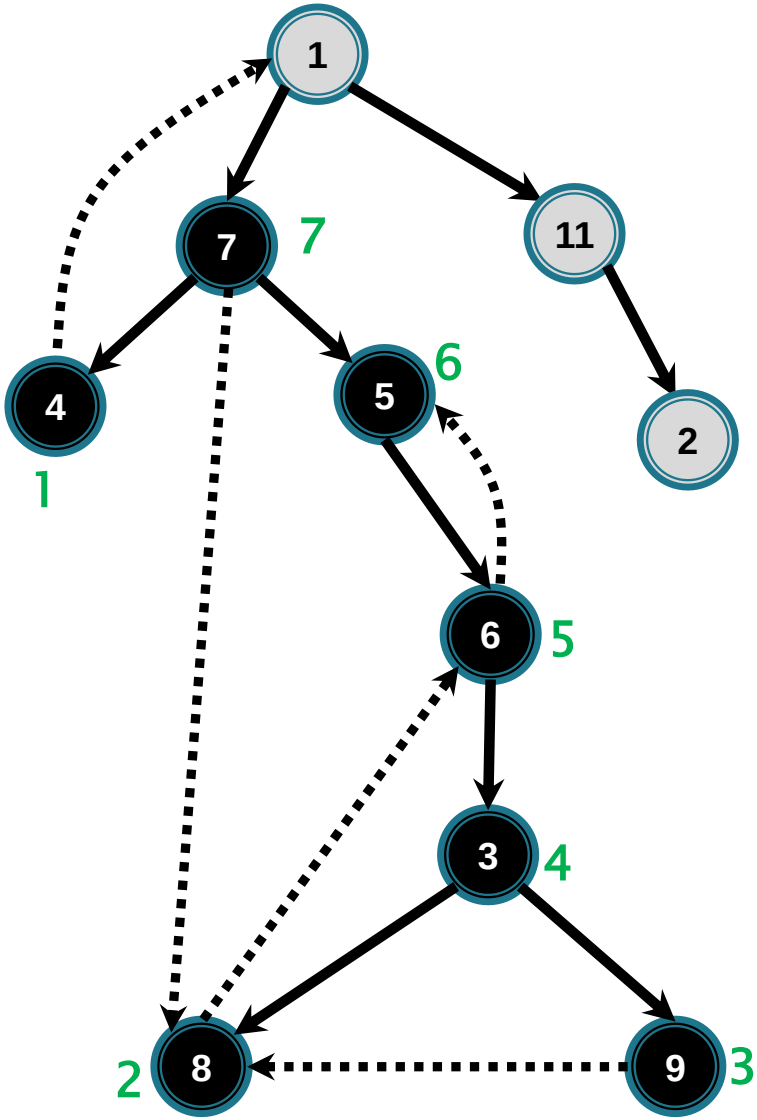
Ordinea descrescătoare finalizare: 7, 5, 6, 3, 9, 8, 4

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.



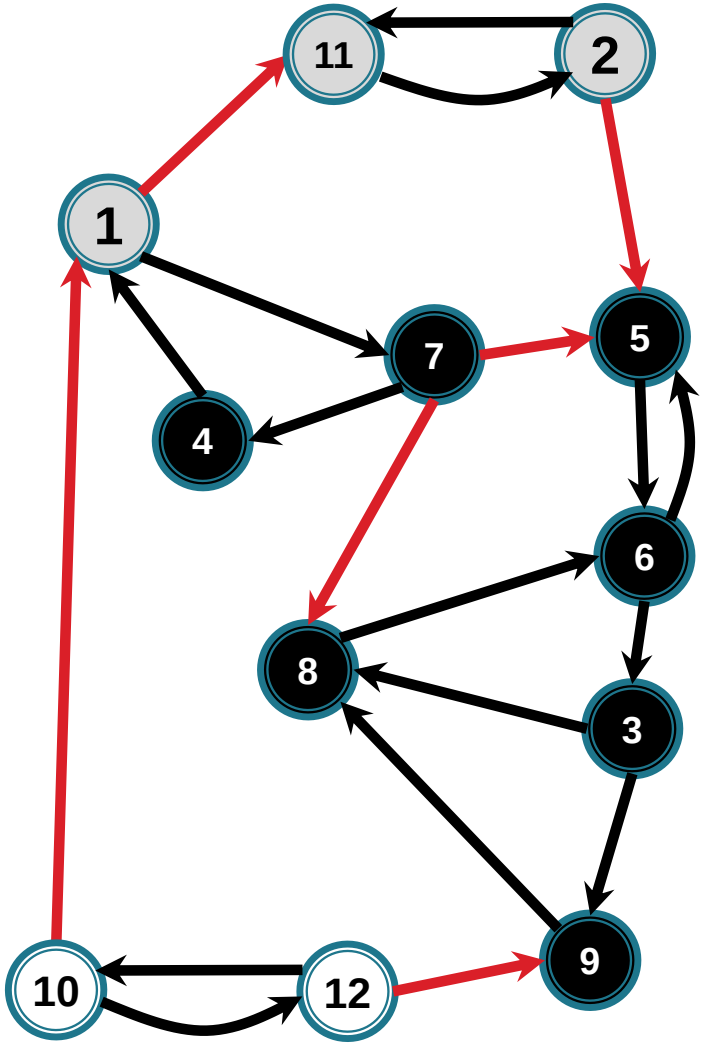


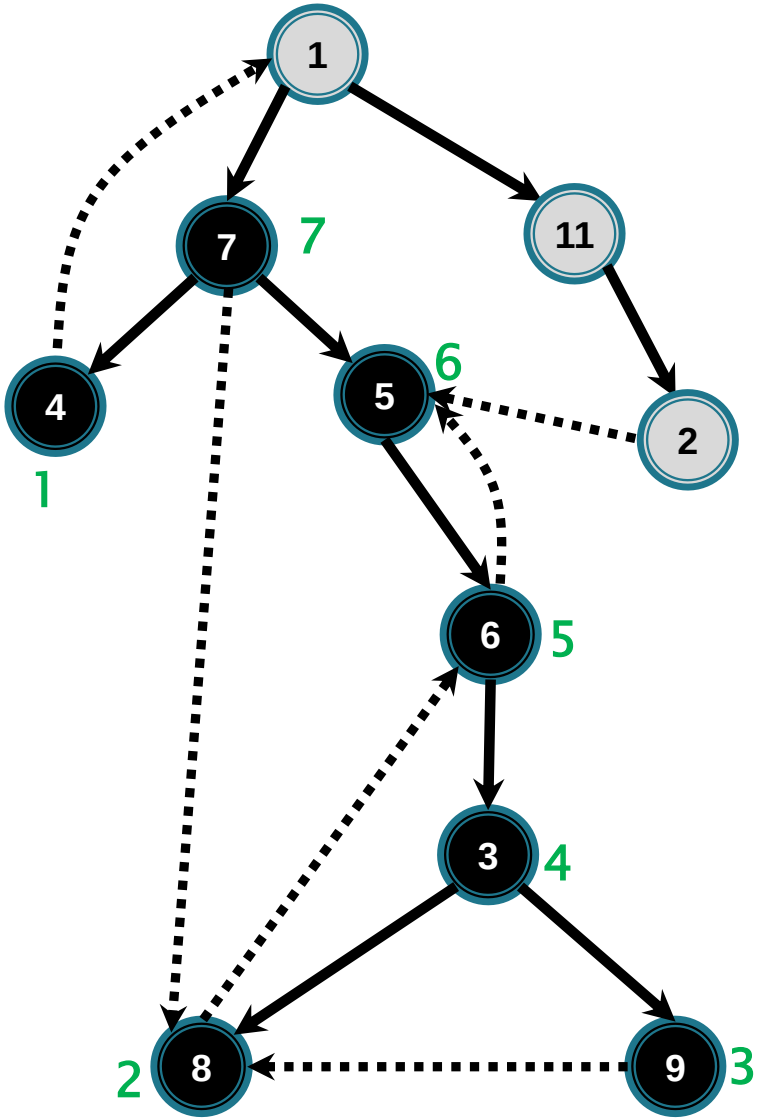
Ordinea descrescătoare finalizare: 7, 5, 6, 3, 9, 8, 4

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.



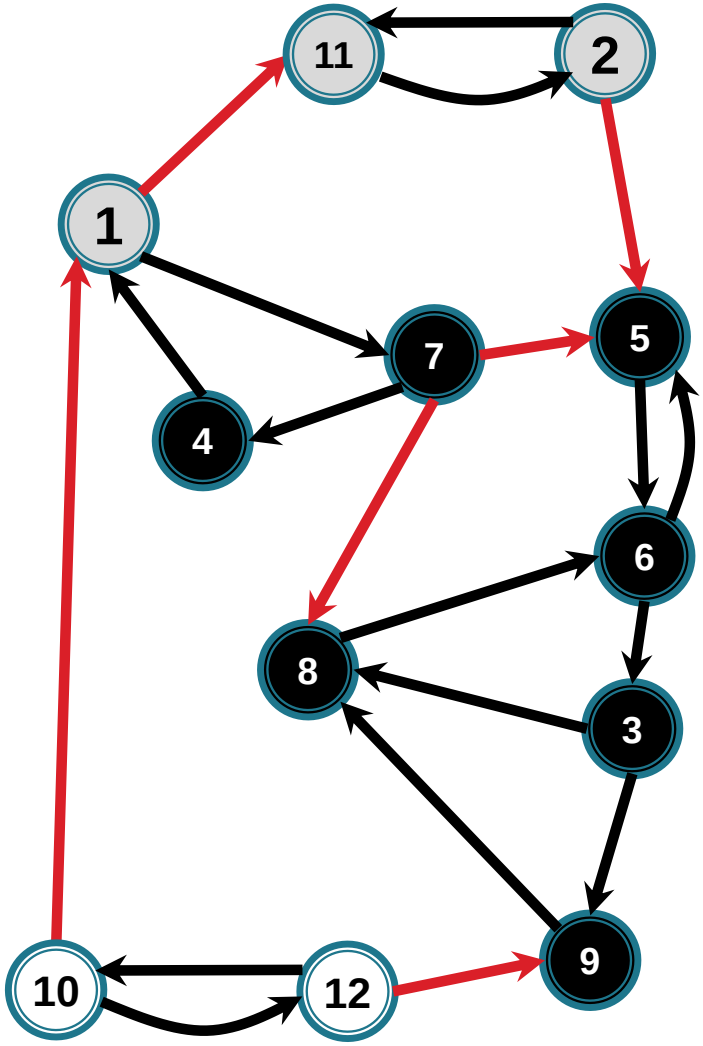


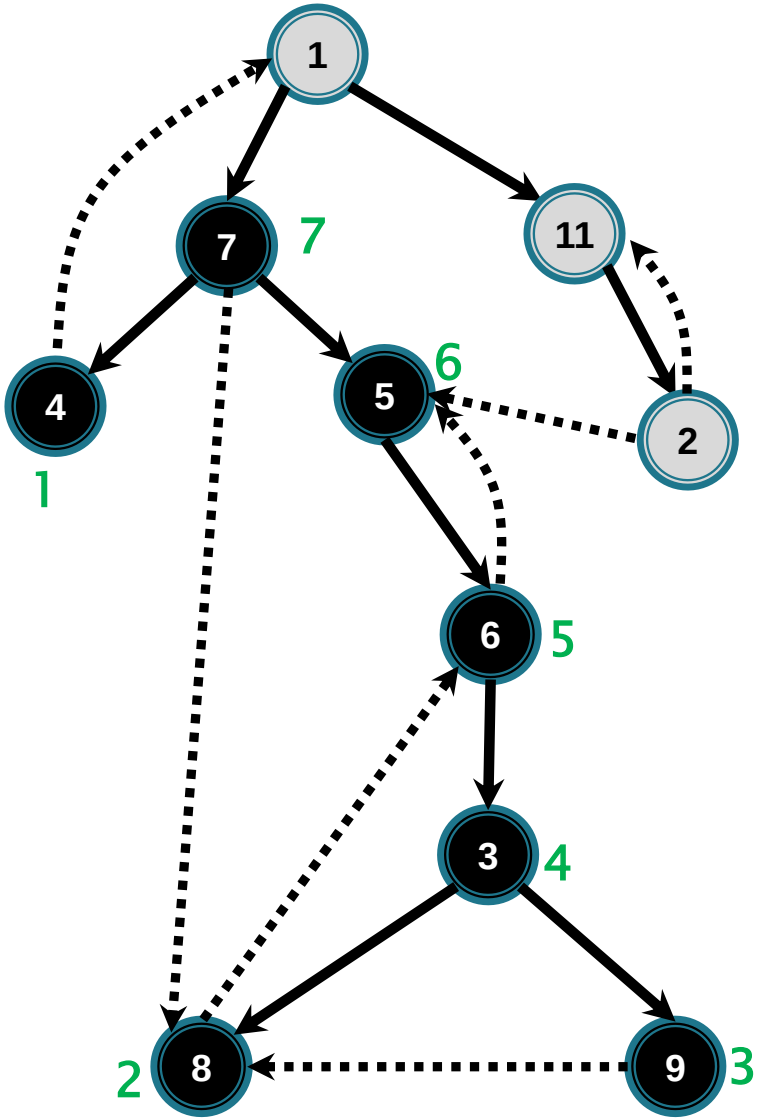
Ordinea descrescătoare finalizare: 7, 5, 6, 3, 9, 8, 4

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.



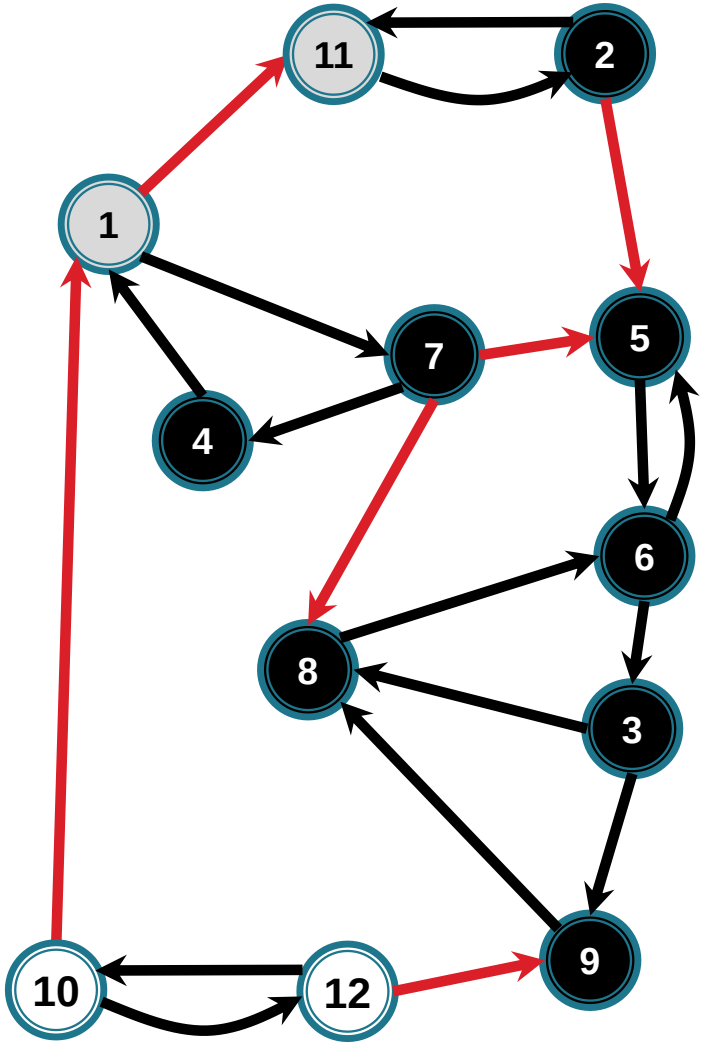


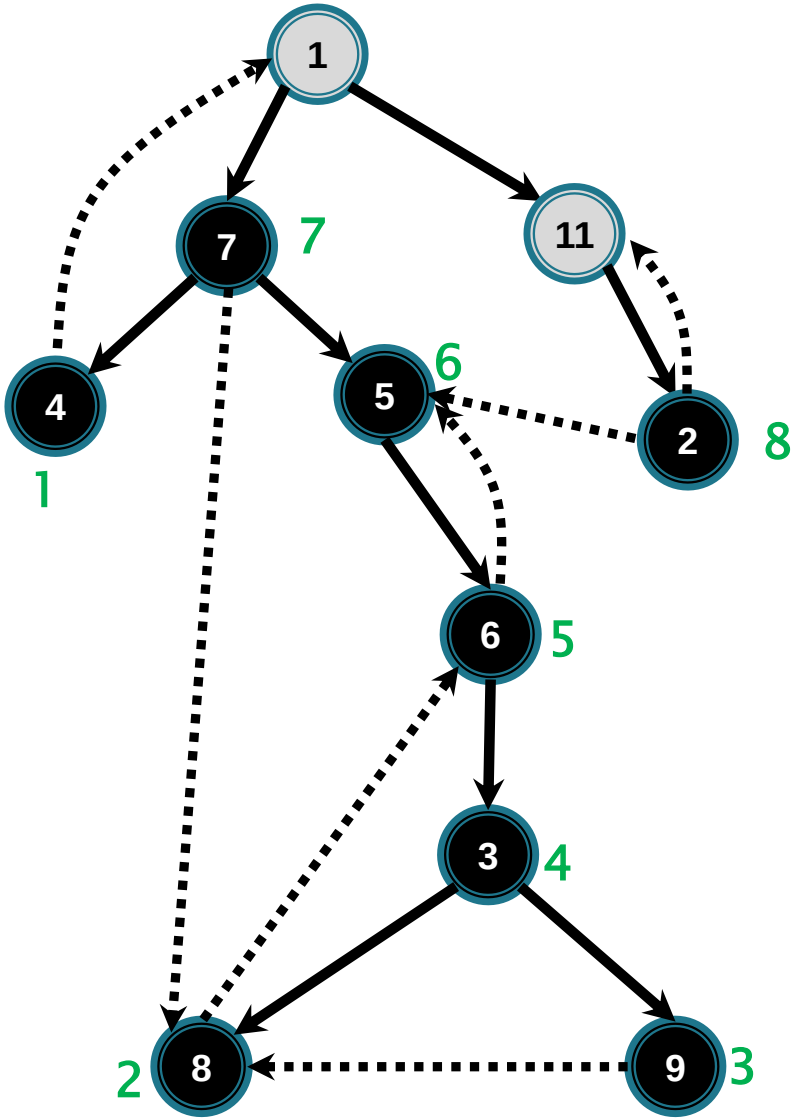
Ordinea descrescătoare finalizare: 7, 5, 6, 3, 9, 8, 4

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.



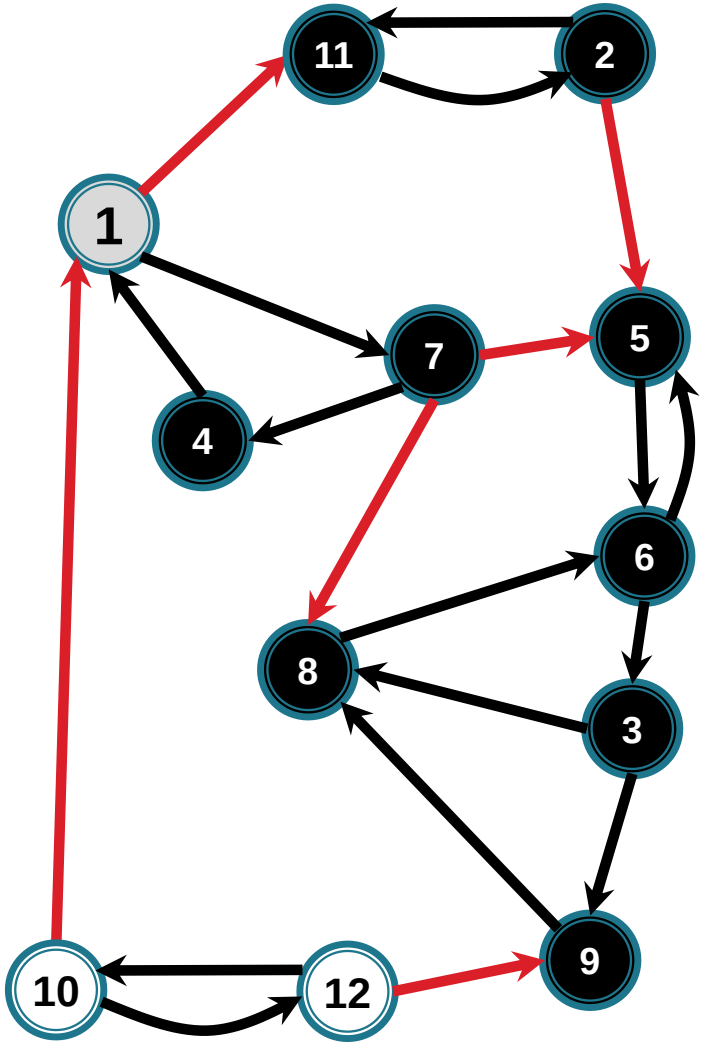


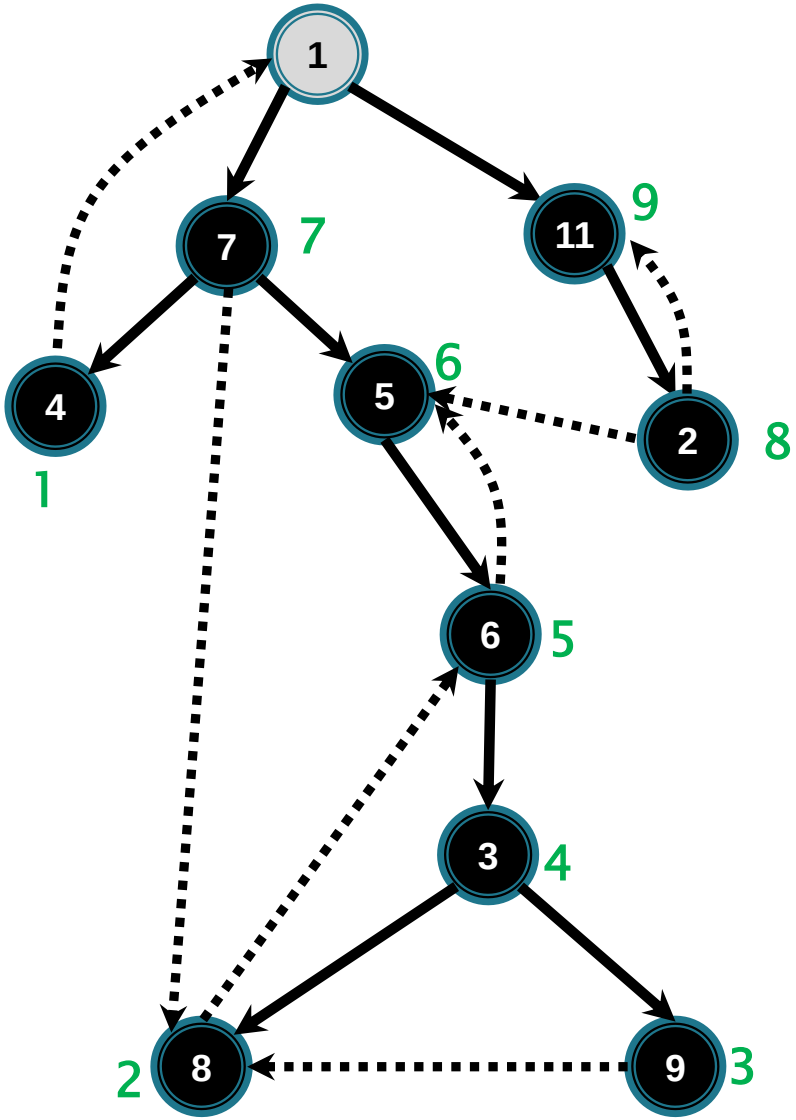
Ordinea descrescătoare finalizare: 2, 7, 5, 6, 3, 9, 8, 4

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.



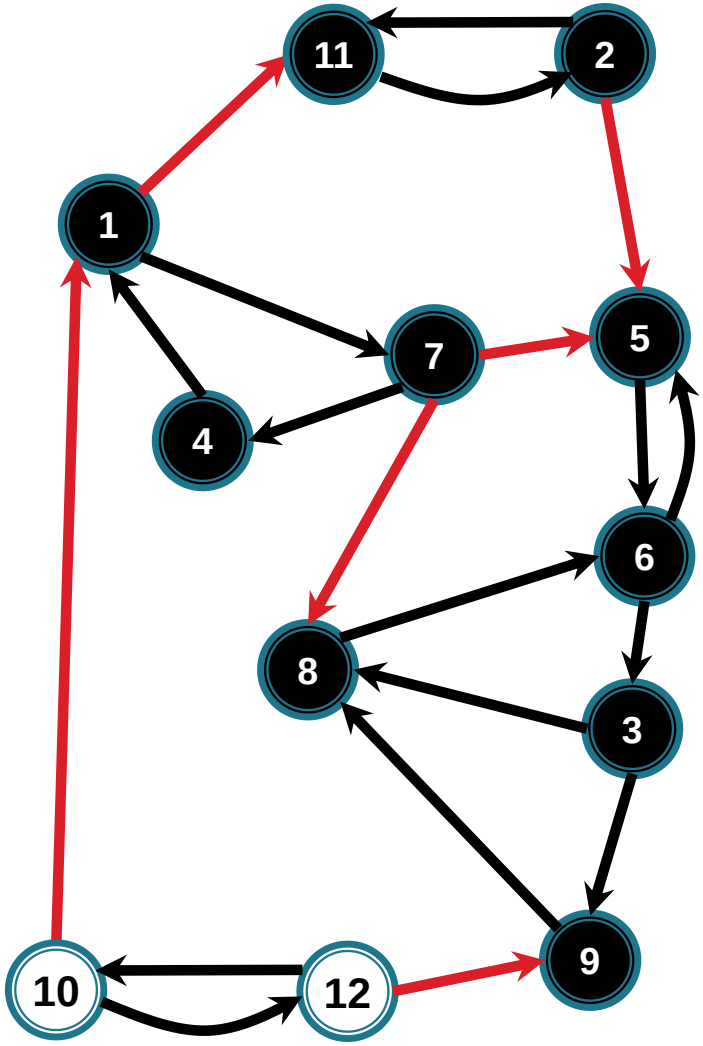


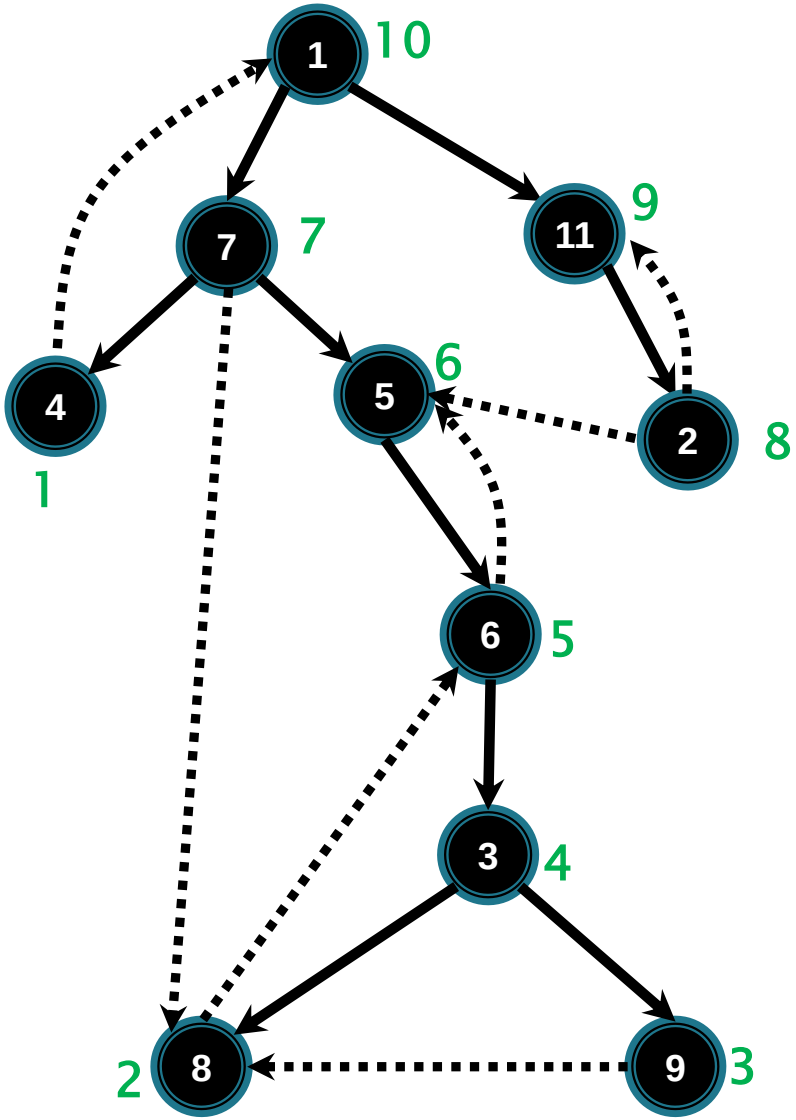
Ordinea descrescătoare finalizare: 11, 2, 7, 5, 6, 3, 9, 8, 4

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.



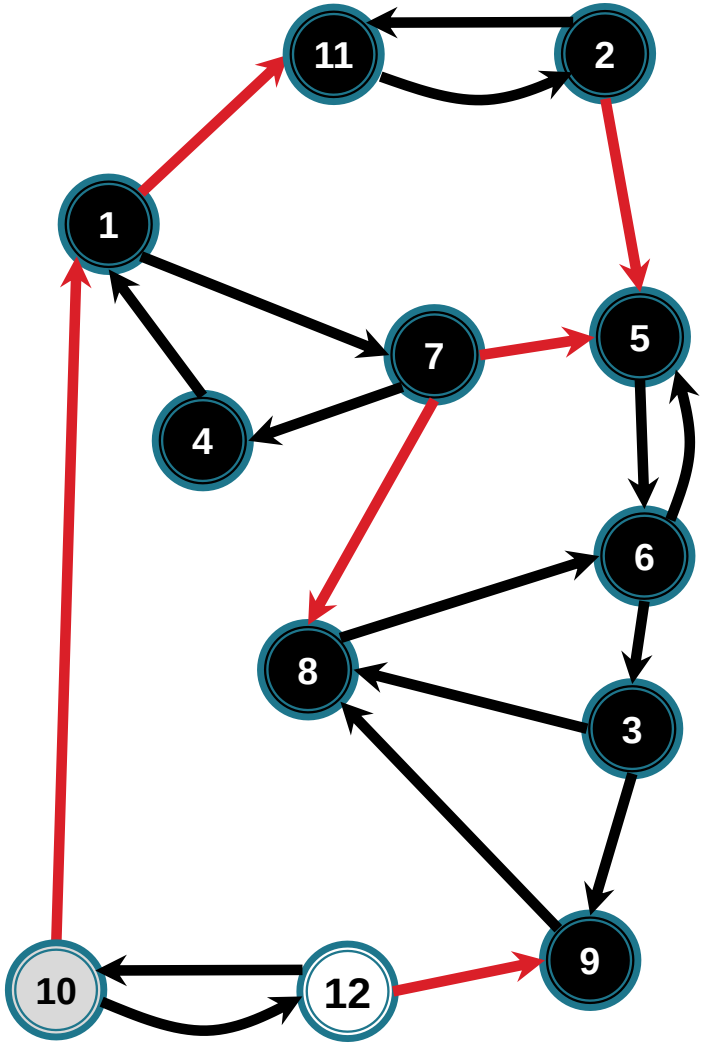


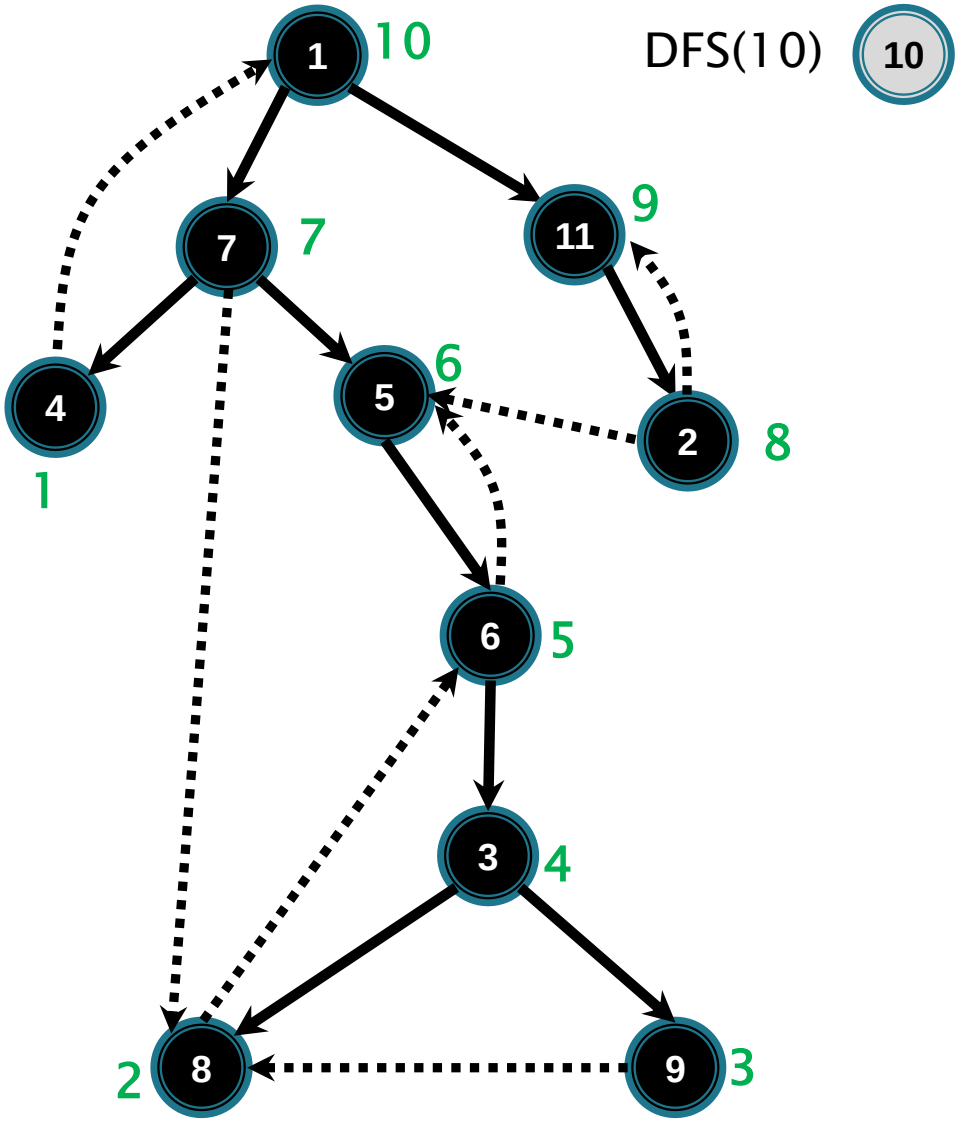
Ordinea descrescătoare finalizare: 1, 11, 2, 7, 5, 6, 3, 9, 8, 4

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.



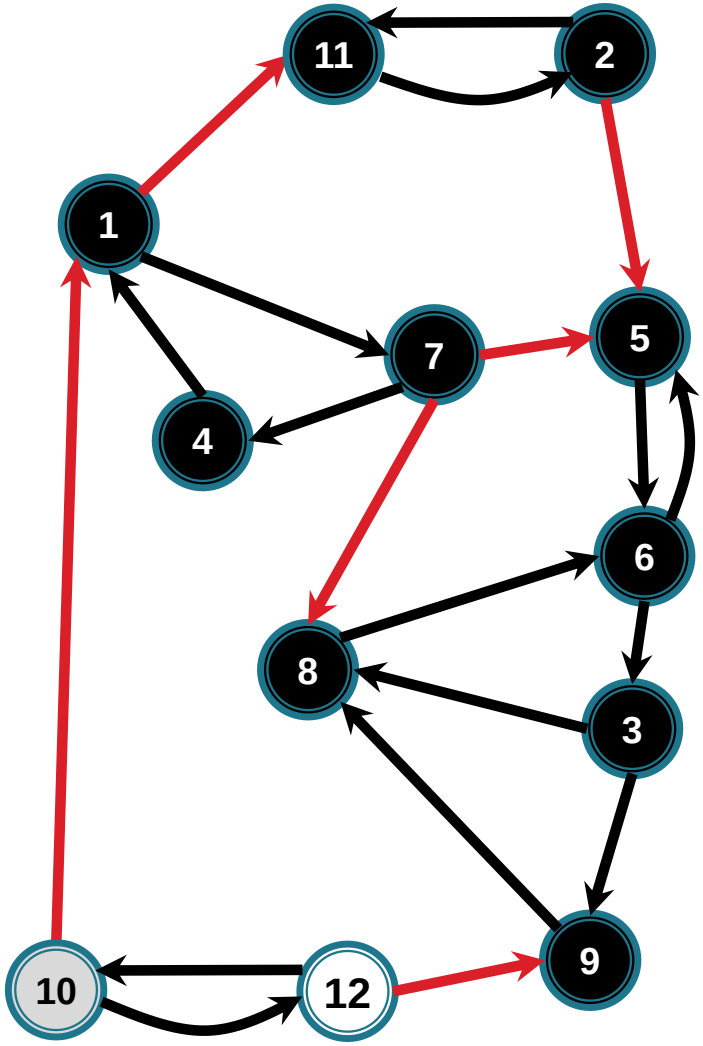


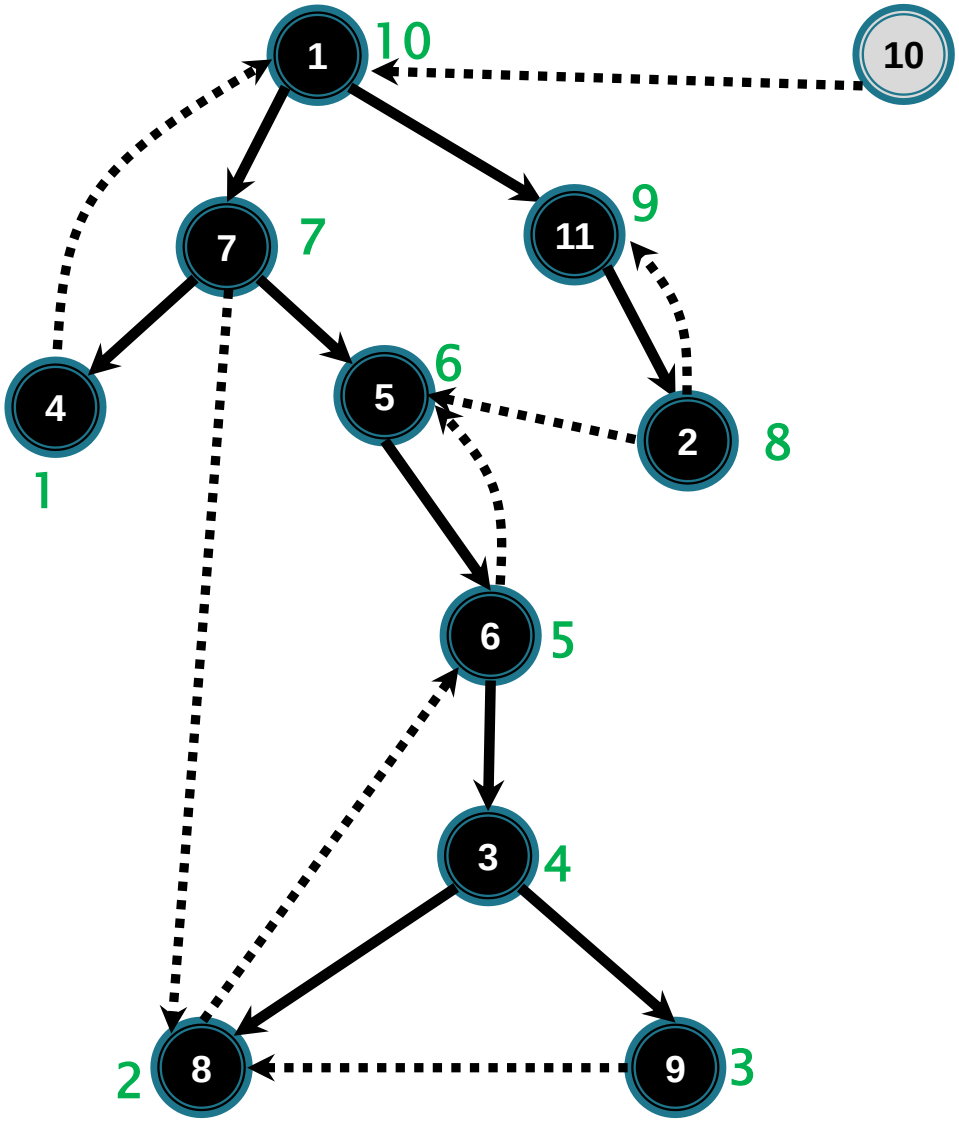
Ordinea descrescătoare finalizare: 1, 11, 2, 7, 5, 6, 3, 9, 8, 4

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.



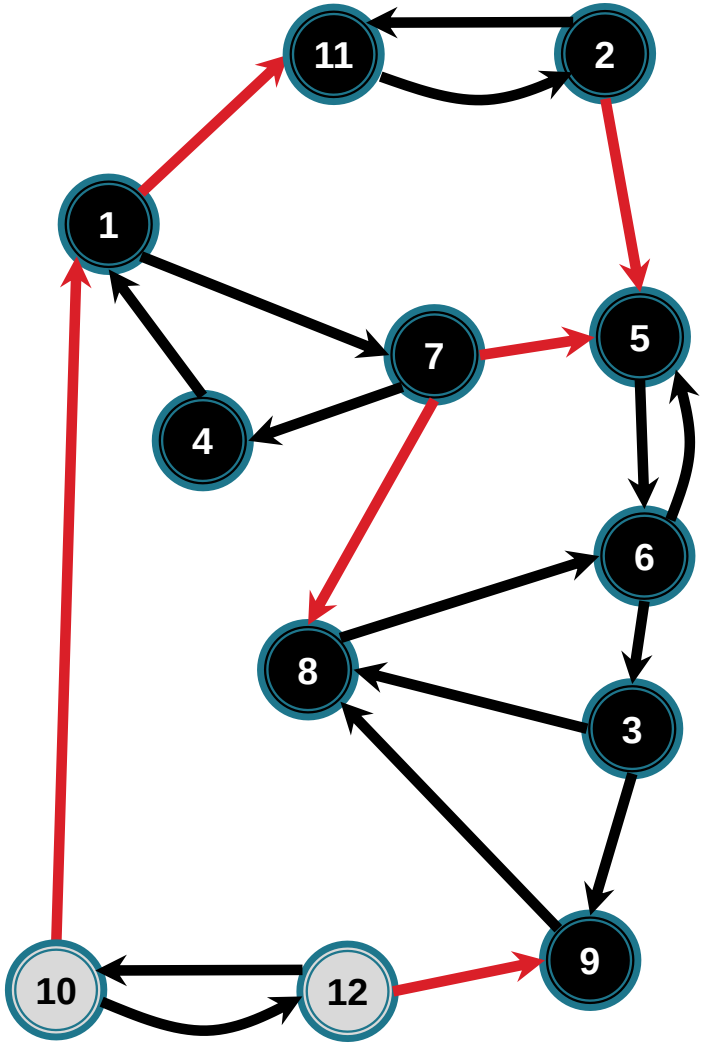


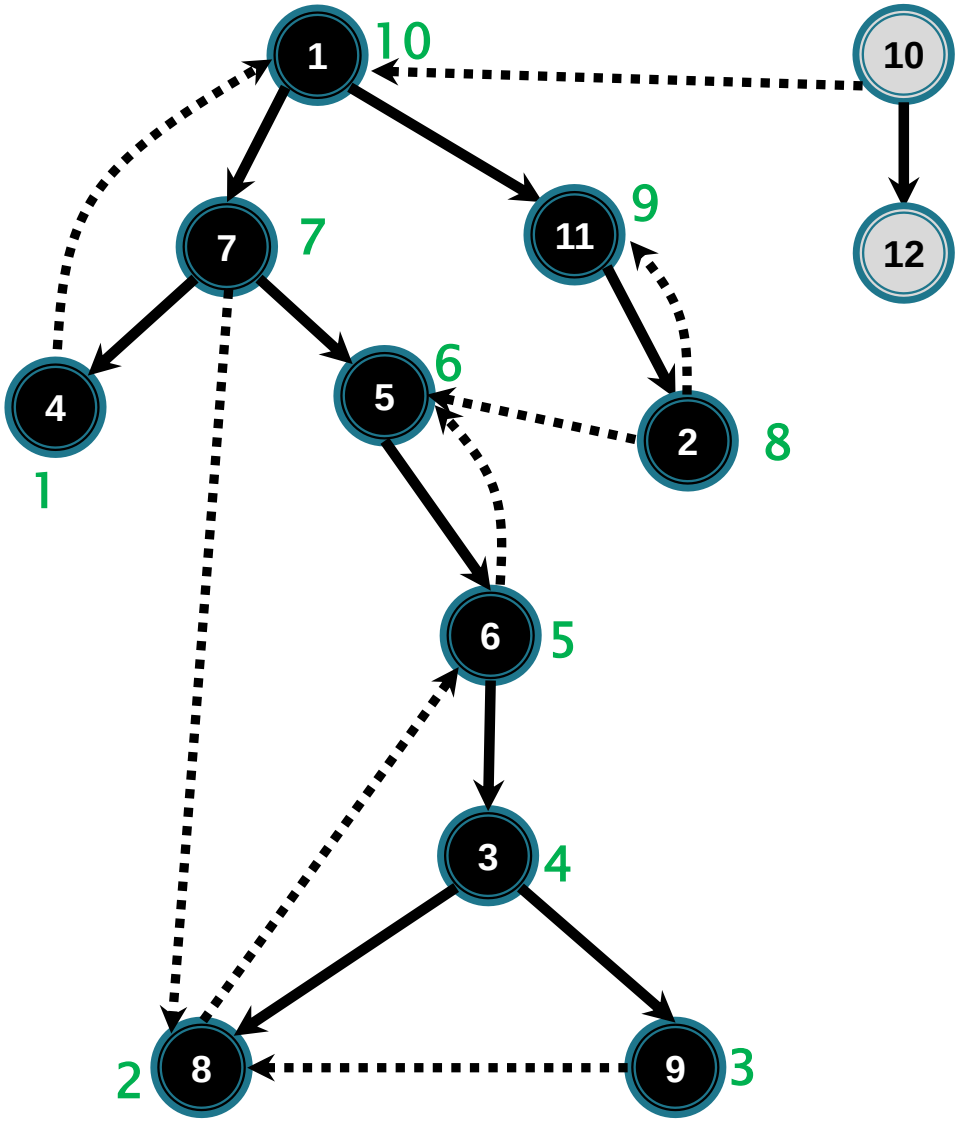
Ordinea descrescătoare finalizare: 1, 11, 2, 7, 5, 6, 3, 9, 8, 4

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.



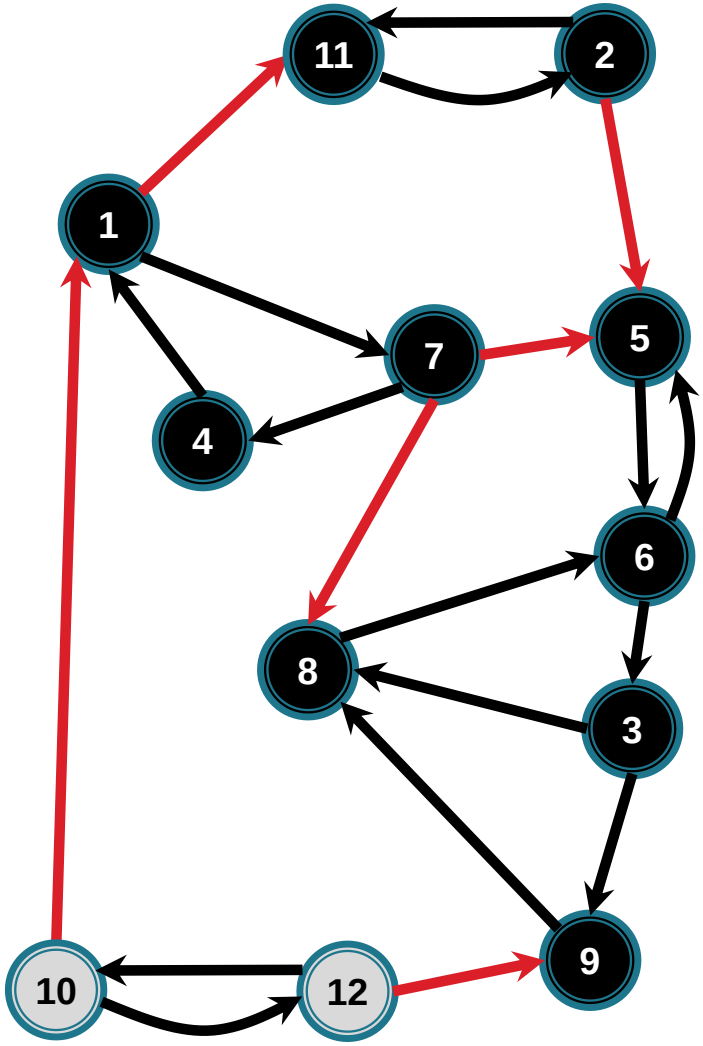


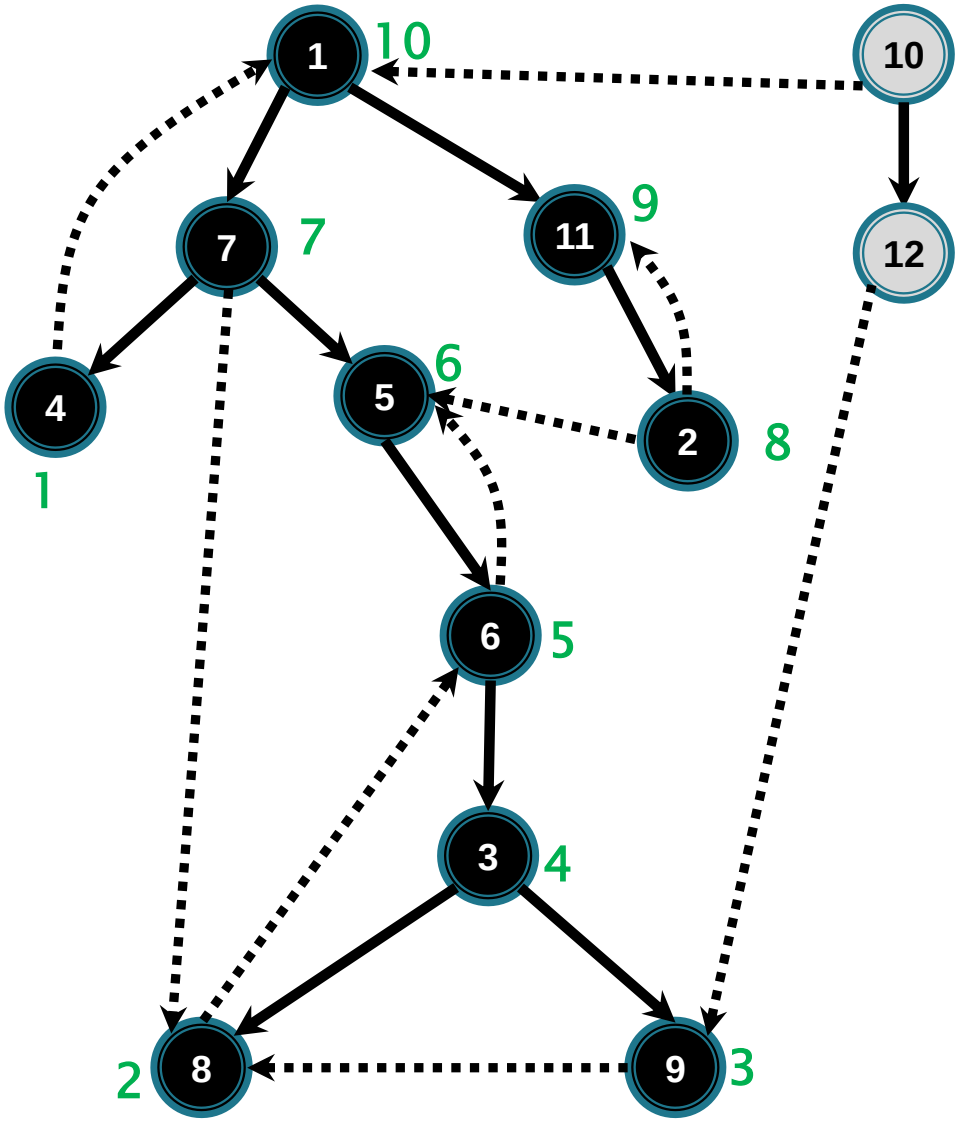
Ordinea descrescătoare finalizare: 1, 11, 2, 7, 5, 6, 3, 9, 8, 4

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.



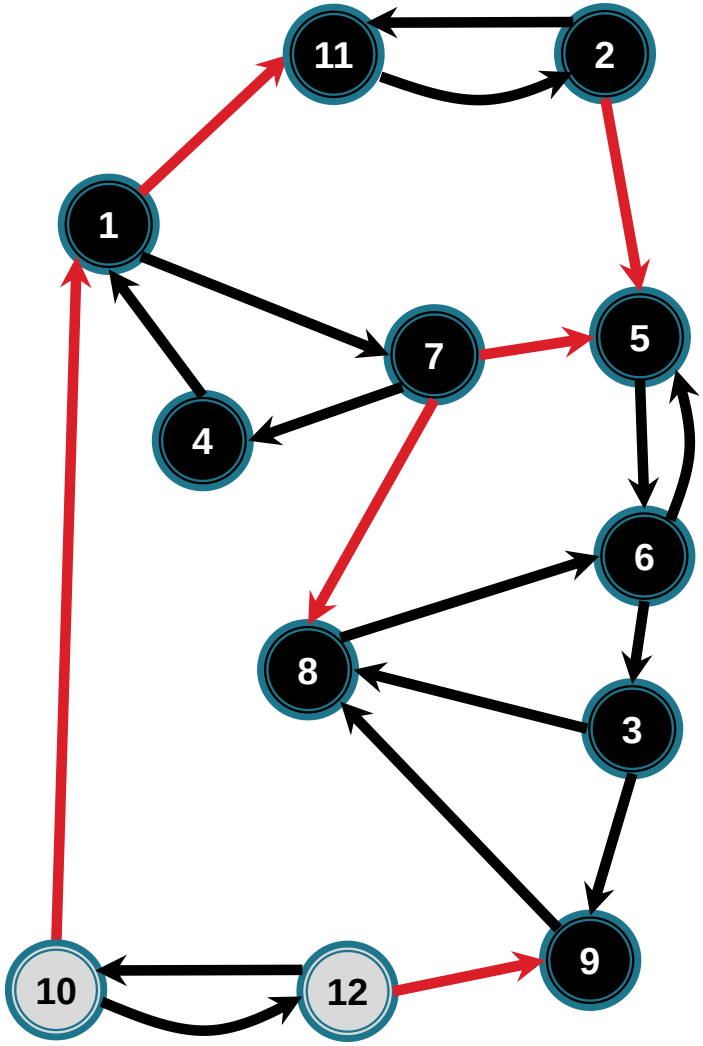


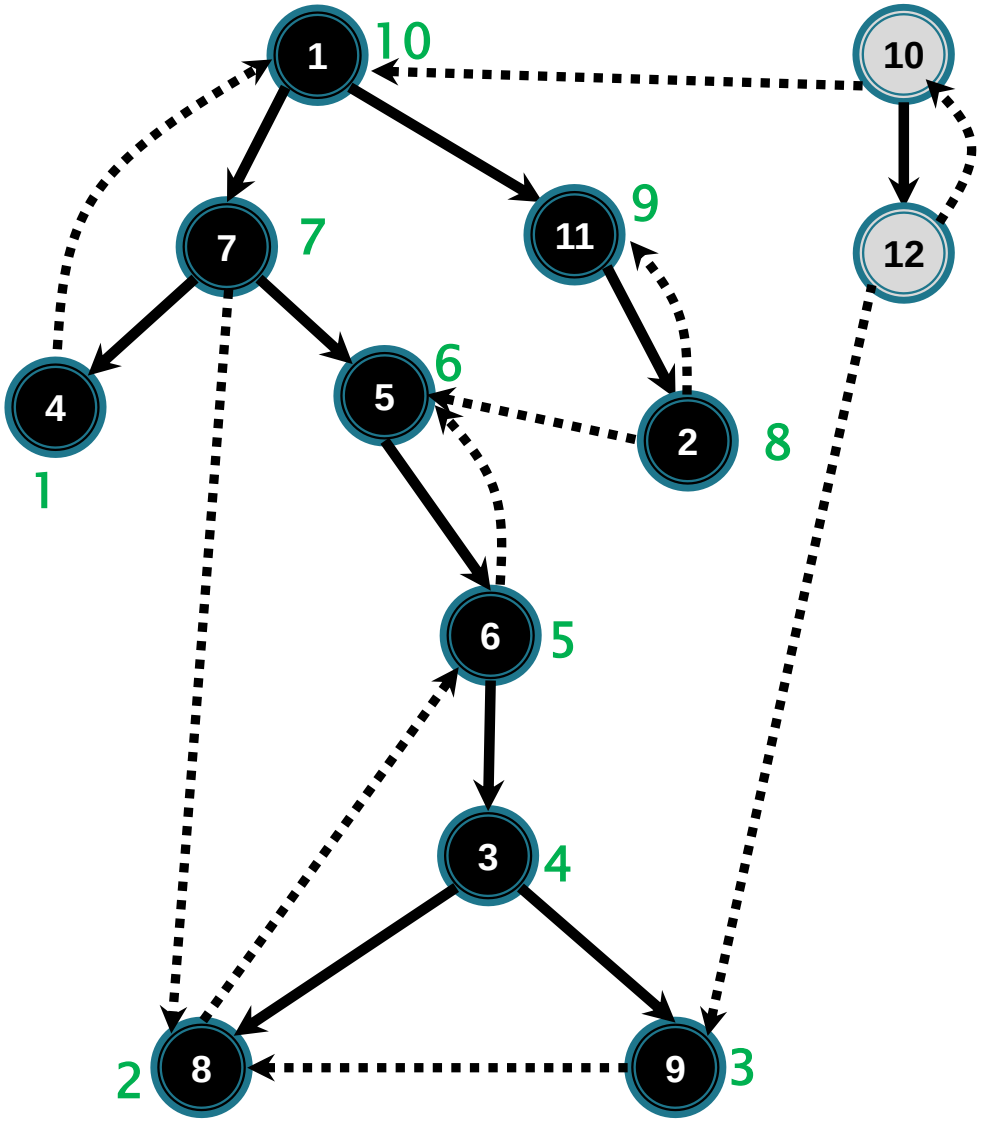
Ordinea descrescătoare finalizare: 1, 11, 2, 7, 5, 6, 3, 9, 8, 4

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.



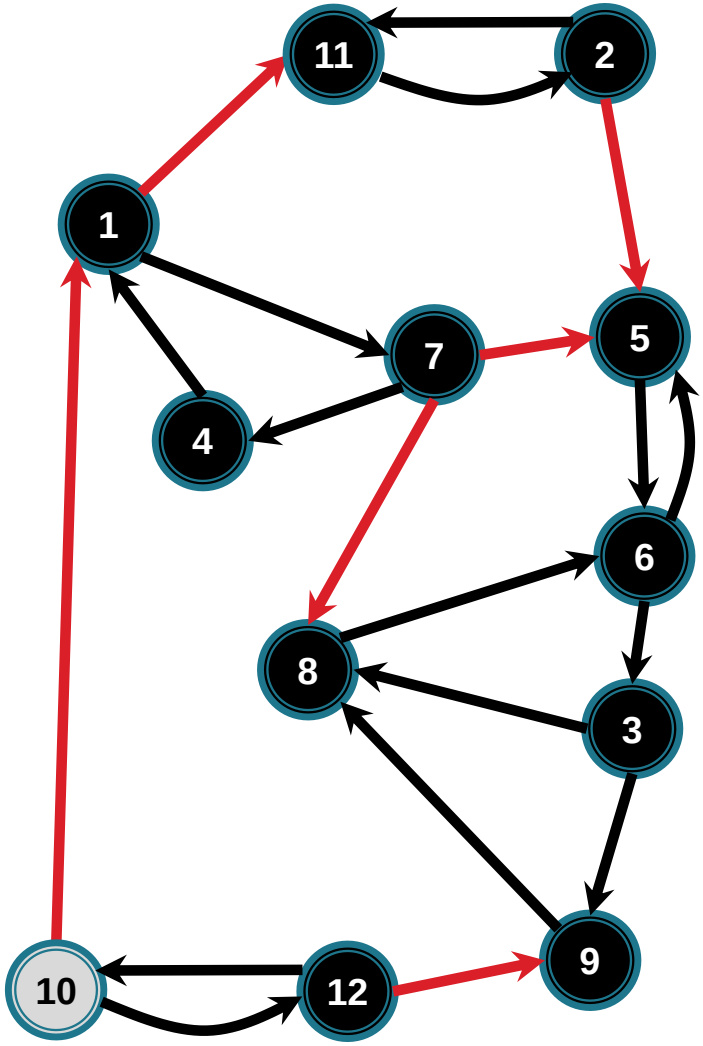


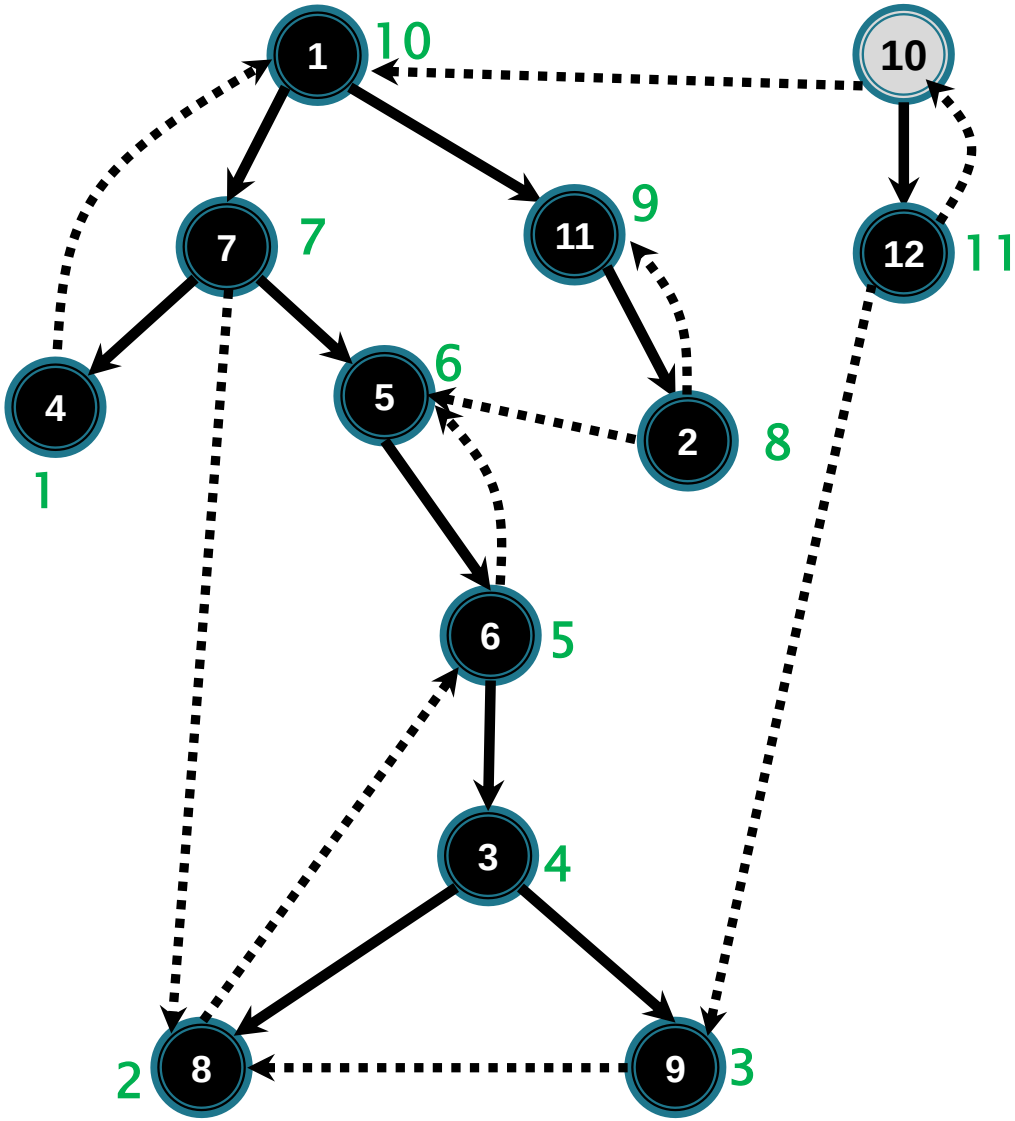
Ordinea descrescătoare finalizare: 1, 11, 2, 7, 5, 6, 3, 9, 8, 4

Algoritmul lui Kosaraju

Timp de finalizare

Pasul 1.



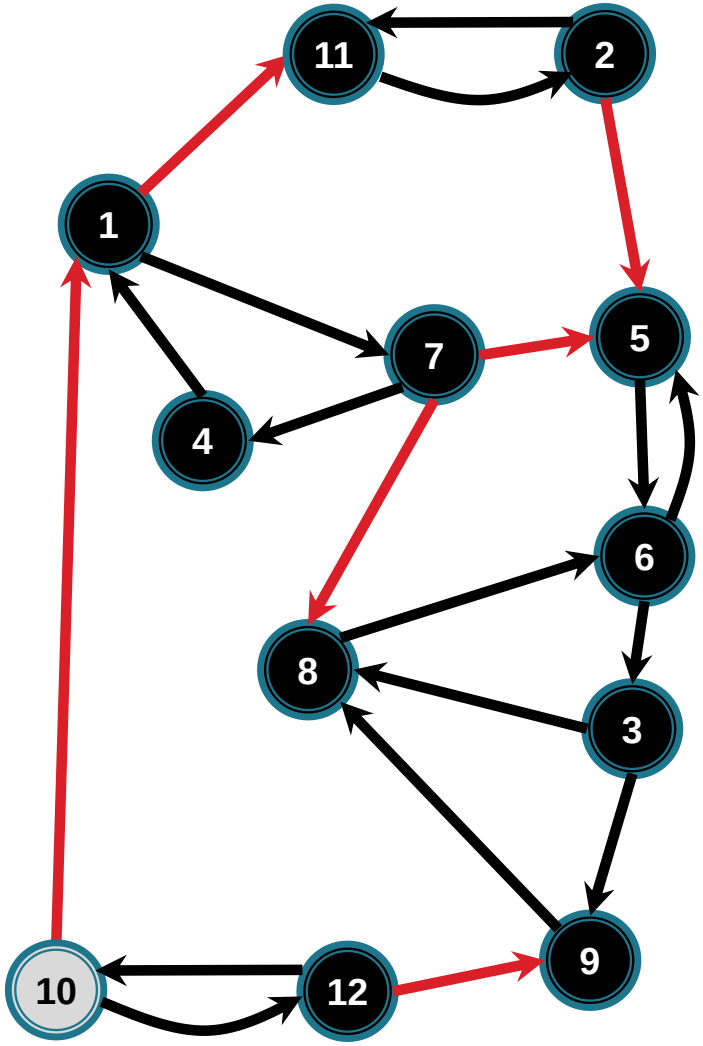


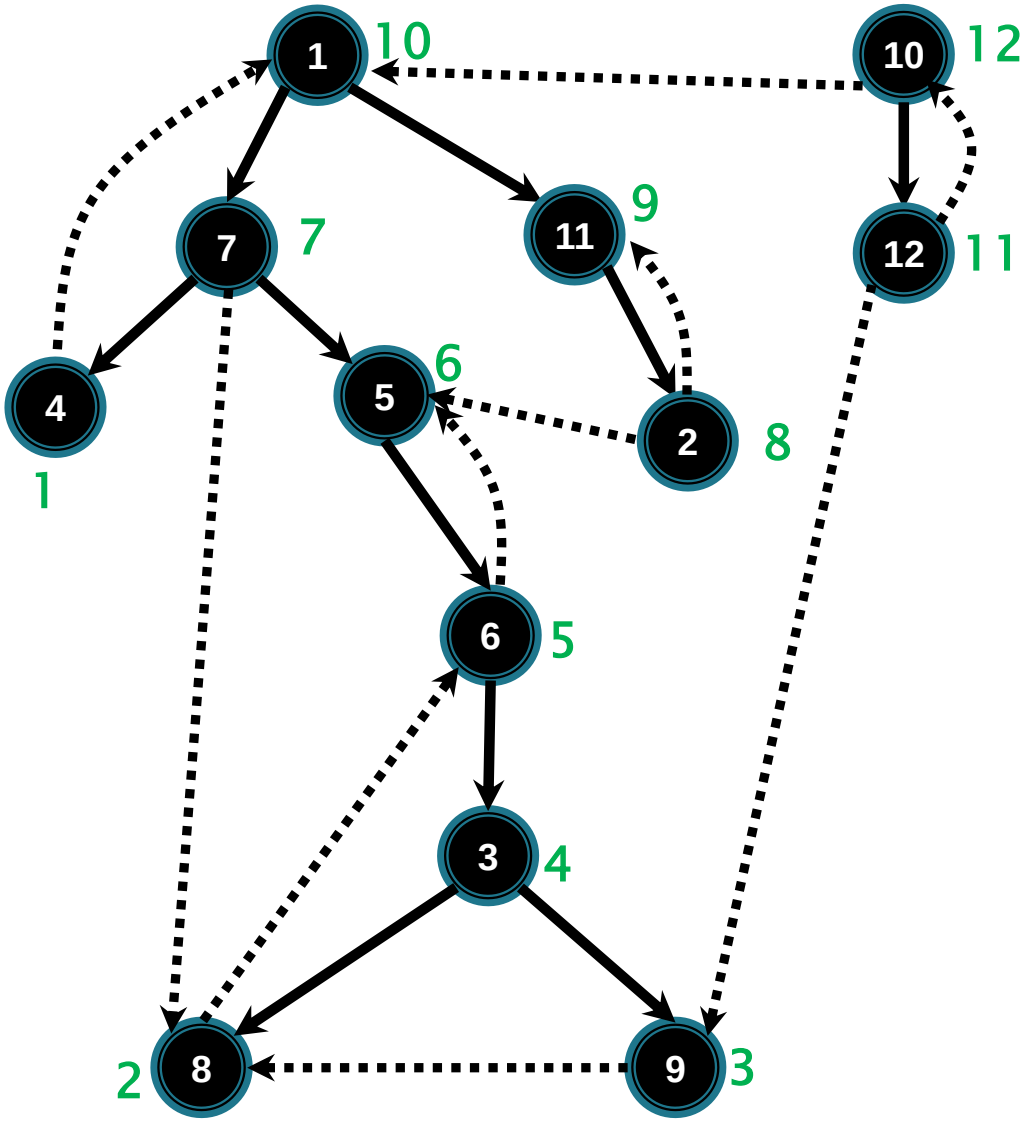
Ordinea descrescătoare finalizare: 12, 1, 11, 2, 7, 5, 6, 3, 9, 8, 4

Algoritmul lui Kosaraju

Timp de finalizare

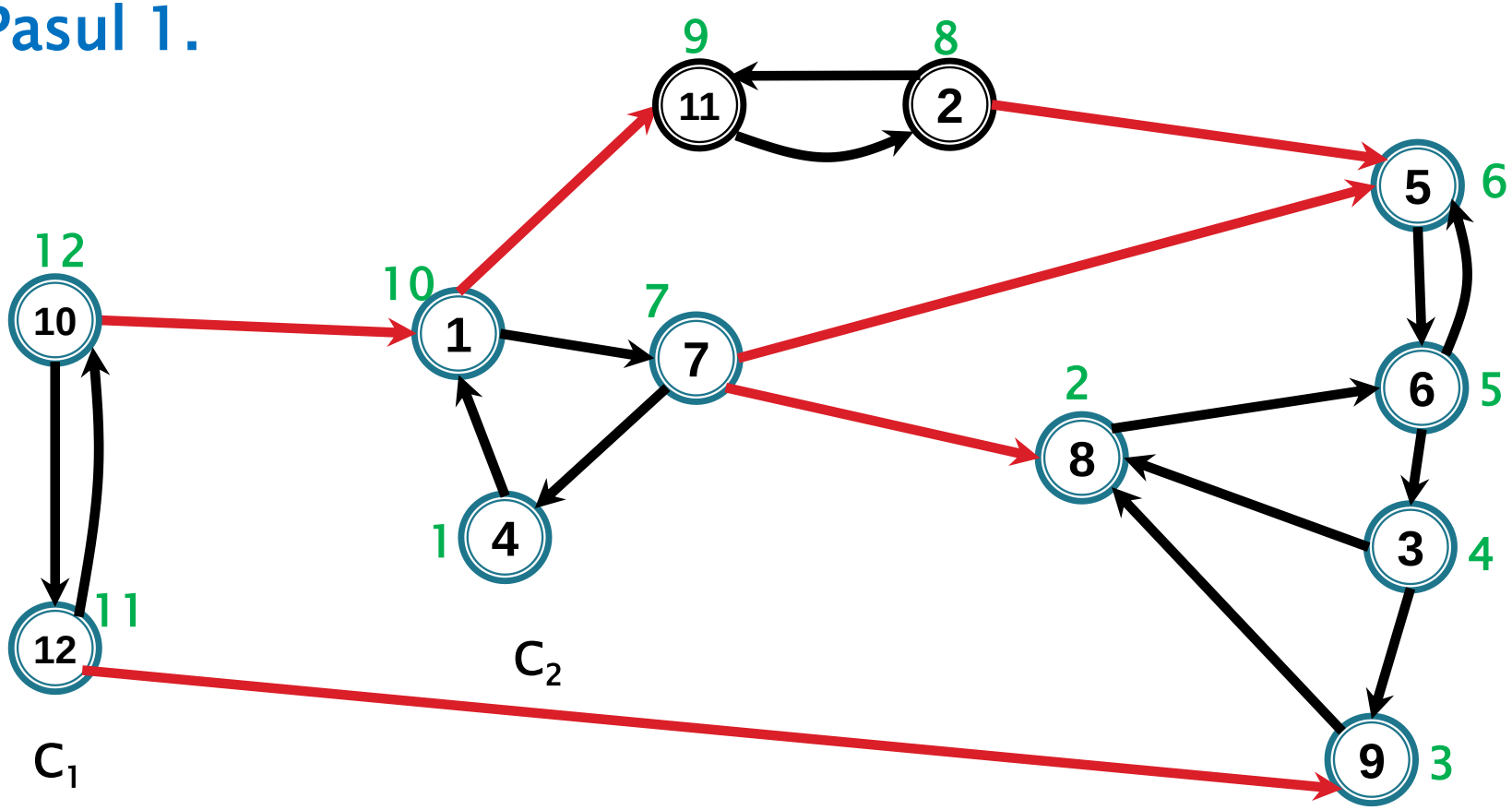
Pasul 1.





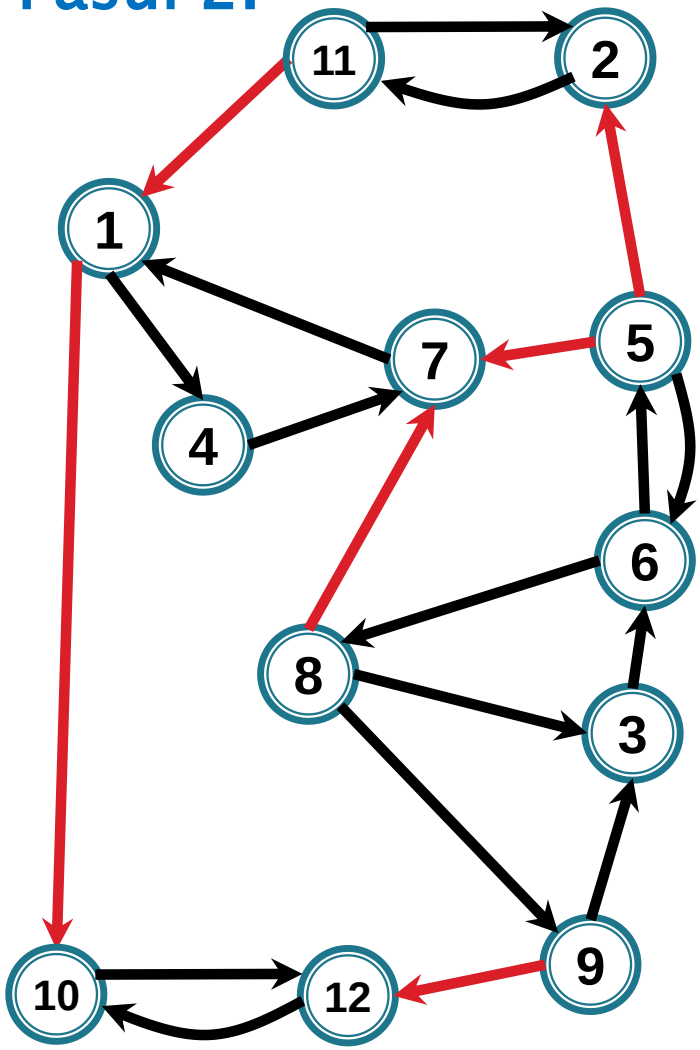
Ordinea descrescătoare finalizare: 10, 12, 1, 11, 2, 7, 5, 6, 3, 9, 8, 4

Pasul 1.



Algoritmul lui Kosaraju

Pasul 2.

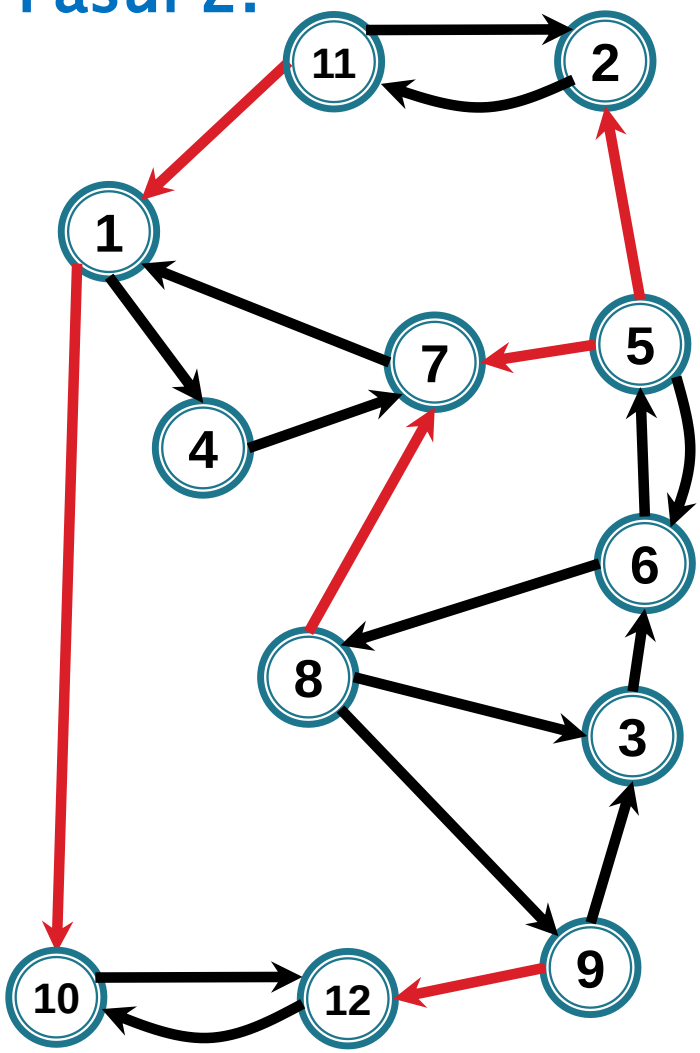


G^T

Ordinea descrescătoare finalizare: 10, 12, 1, 11, 2, 7, 5, 6, 3, 9, 8, 4

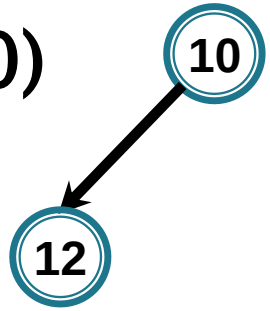
Algoritmul lui Kosaraju

Pasul 2.



G^T

DFS(10)

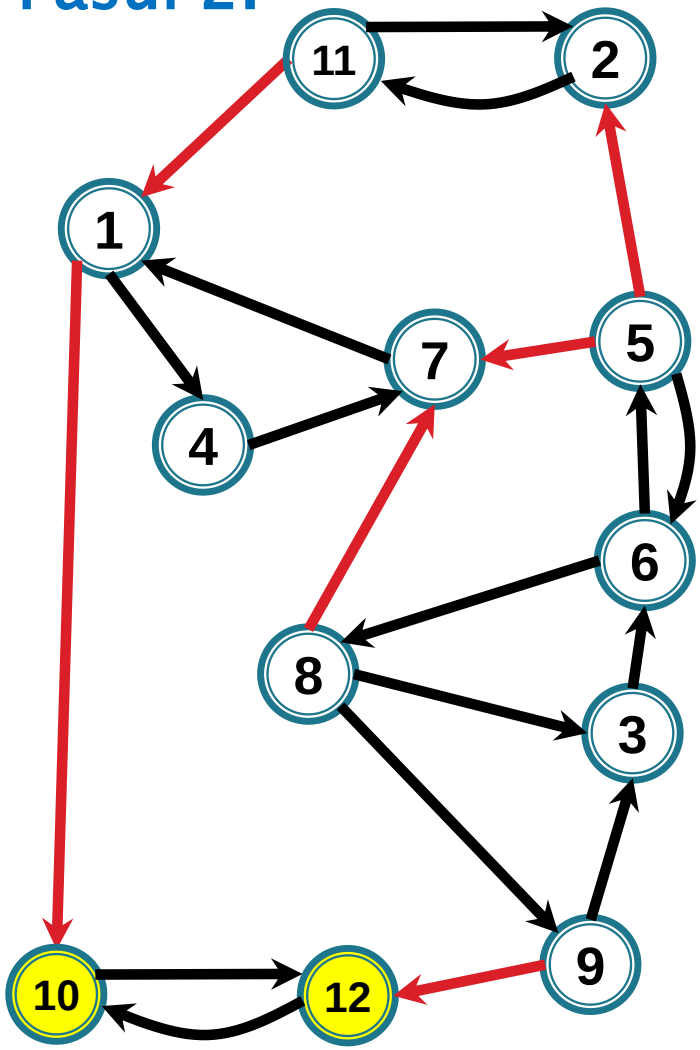


Componenta tare conexă: 10, 12

Ordinea descrescătoare finalizare: 10, 12, 1, 11, 2, 7, 5, 6, 3, 9, 8, 4

Algoritmul lui Kosaraju

Pasul 2.

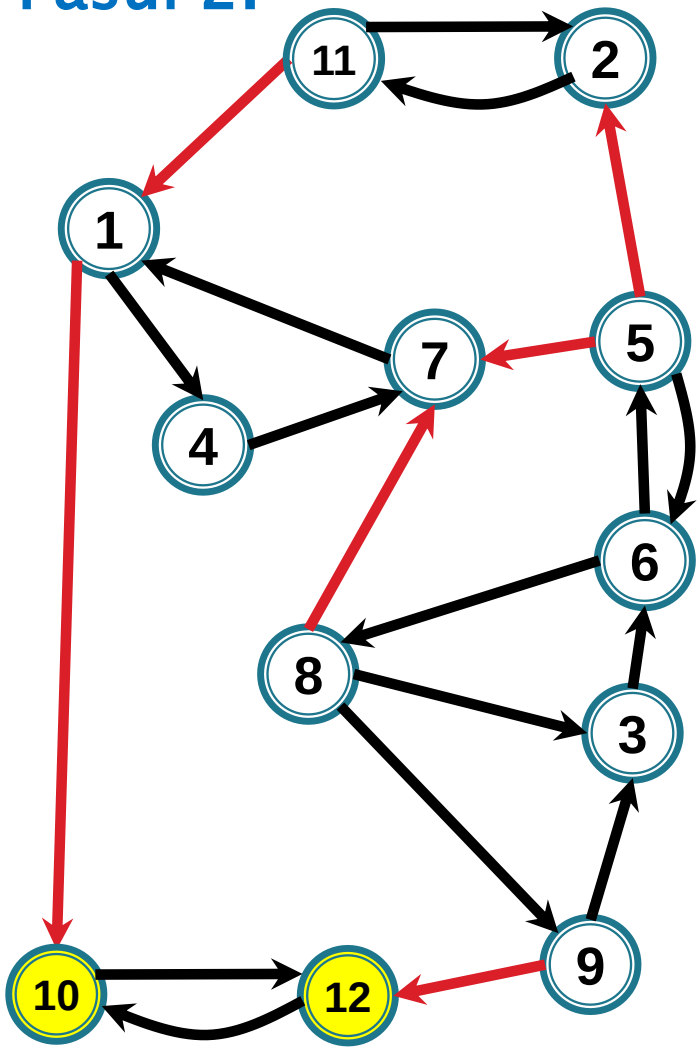


G^T

Ordinea descrescătoare finalizare: 10, 12, 1, 11, 2, 7, 5, 6, 3, 9, 8, 4

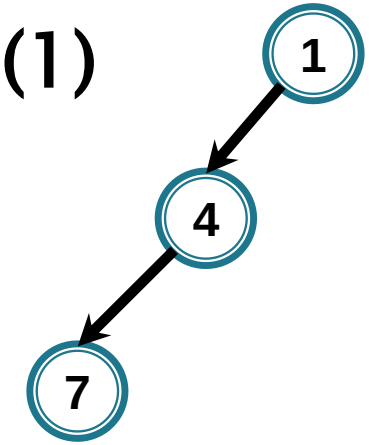
Algoritmul lui Kosaraju

Pasul 2.



G^T

DFS(1)

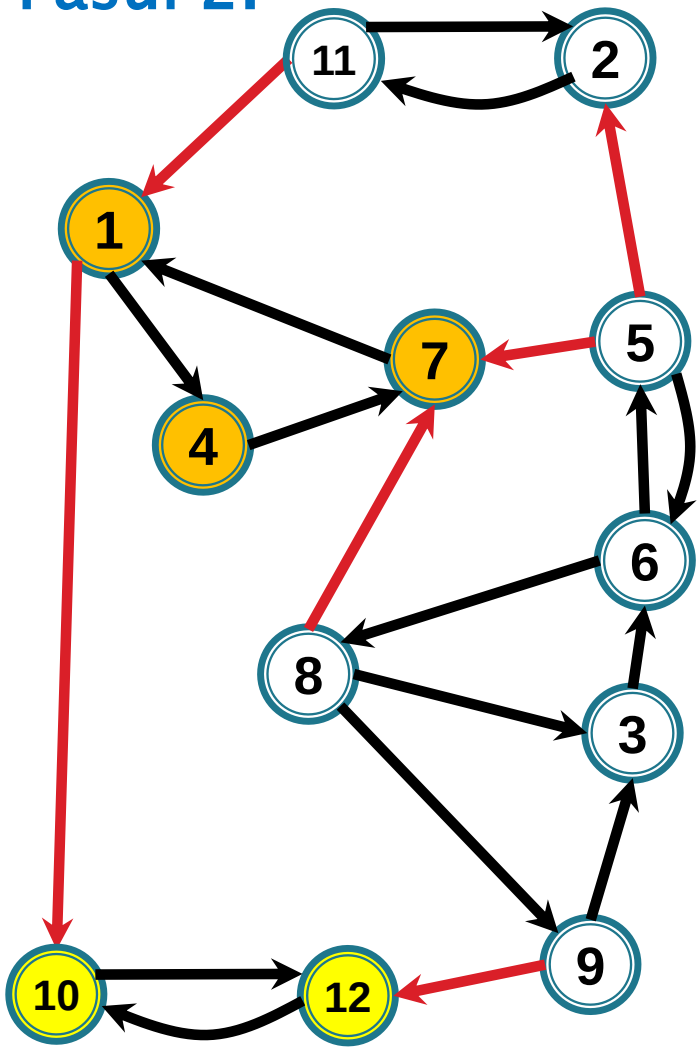


Componenta tare conexă: 1,4,7

Ordinea descrescătoare finalizare: 10, 12, 1, 11, 2, 7, 5, 6, 3, 9, 8, 4

Algoritmul lui Kosaraju

Pasul 2.

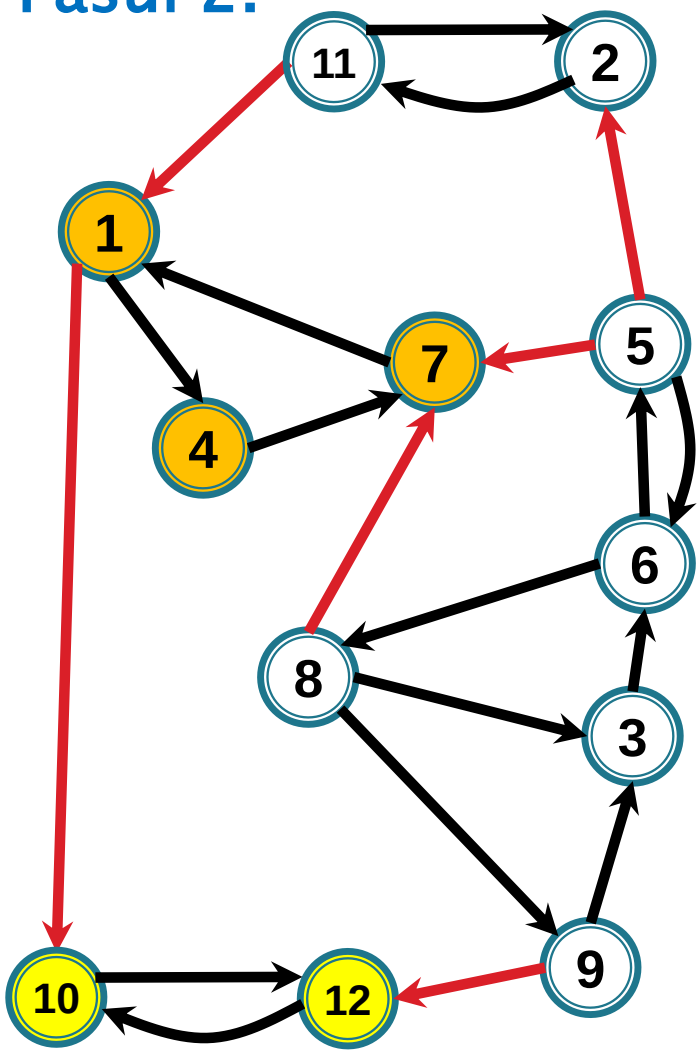


G^T

Ordinea descrescătoare finalizare: 10, 12, 1, 11, 2, 7, 5, 6, 3, 9, 8, 4

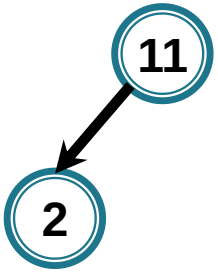
Algoritmul lui Kosaraju

Pasul 2.



G^T

DFS(11)

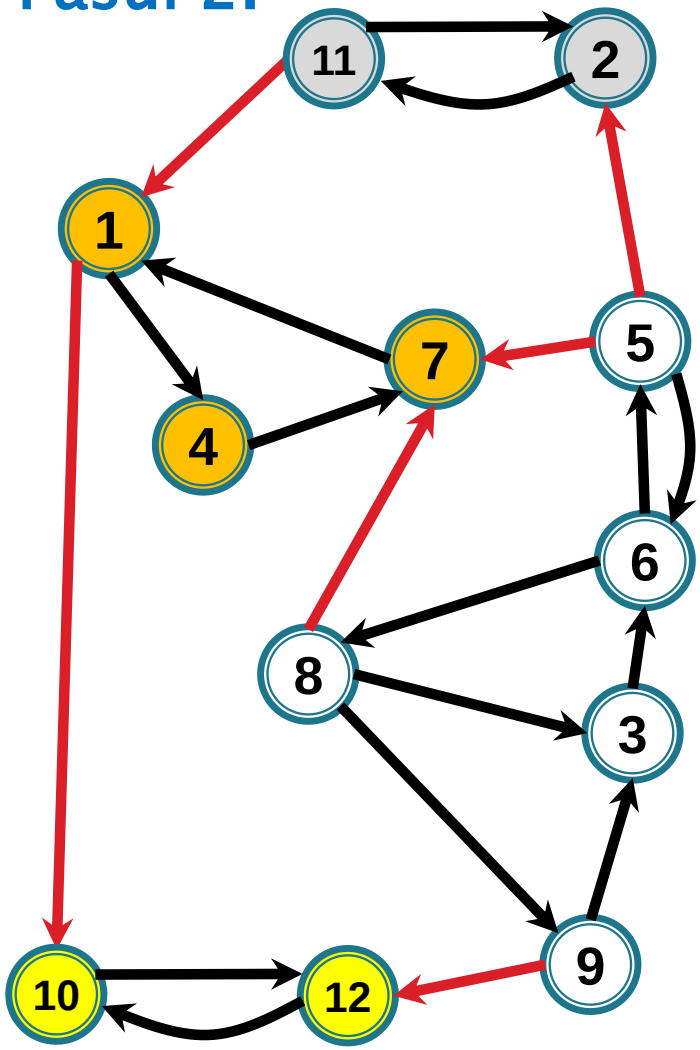


Componenta tare conexă: 11, 2

Ordinea descrescătoare finalizare: 10, 12, 1, 11, 2, 7, 5, 6, 3, 9, 8, 4

Algoritmul lui Kosaraju

Pasul 2.

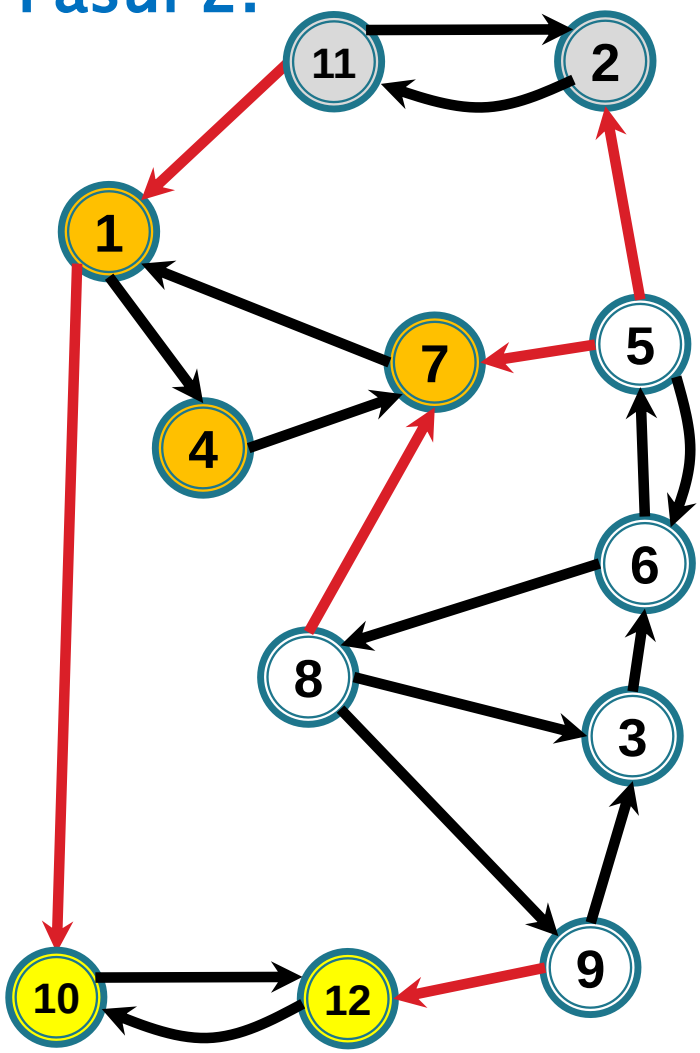


G^T

Ordinea descrescătoare finalizare: 10, 12, 1, 11, 2, 7, 5, 6, 3, 9, 8, 4

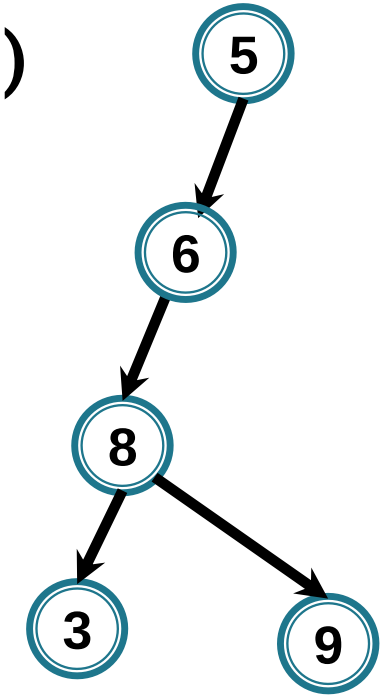
Algoritmul lui Kosaraju

Pasul 2.



G^T

DFS(5)

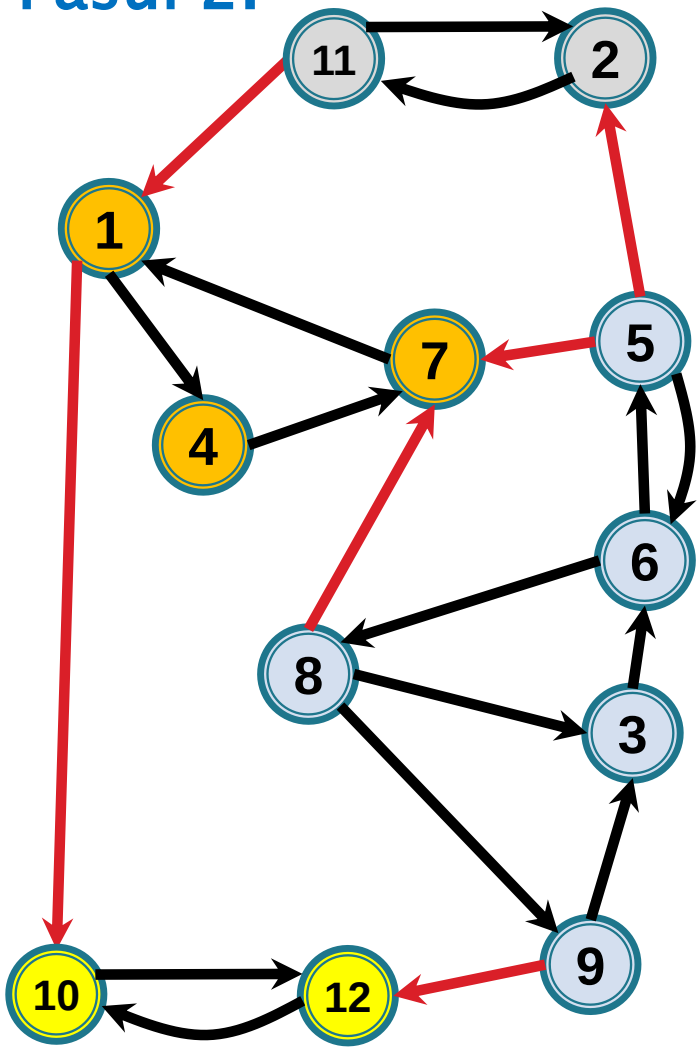


Componenta tare conexă: 5, 6, 8, 3, 9

Ordinea descrescătoare finalizare: 10, 12, 1, 11, 2, 7, 5, 6, 3, 9, 8, 4

Algoritmul lui Kosaraju

Pasul 2.



G^T

Componenta tare conexă: 5, 6, 8, 3, 9

Ordinea descrescătoare finalizare: 10, 12, 1, 11, 2, 7, 5, 6, 3, 9, 8, 4

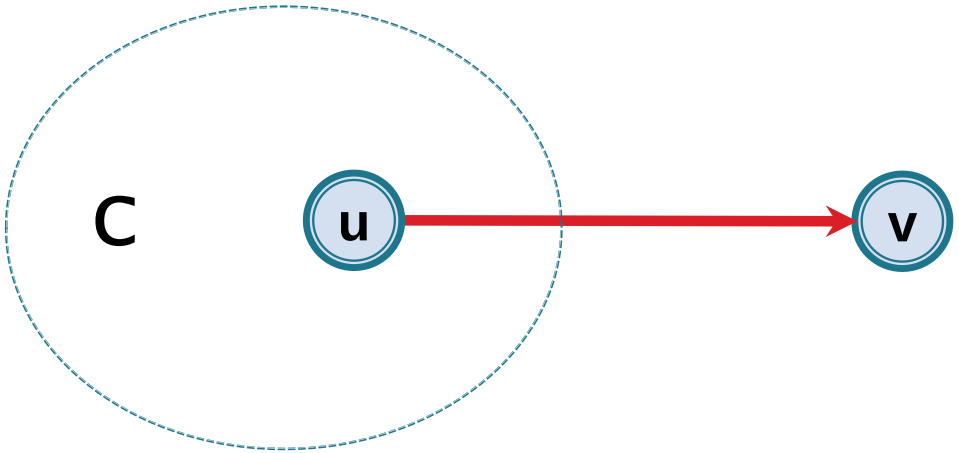
Corectitudine

Corectitudine – Algoritmul lui Kosaraju

Lemă

Fie C o componentă tare conexă si $v \notin C$, astfel încât există un drum de la C la v. Atunci

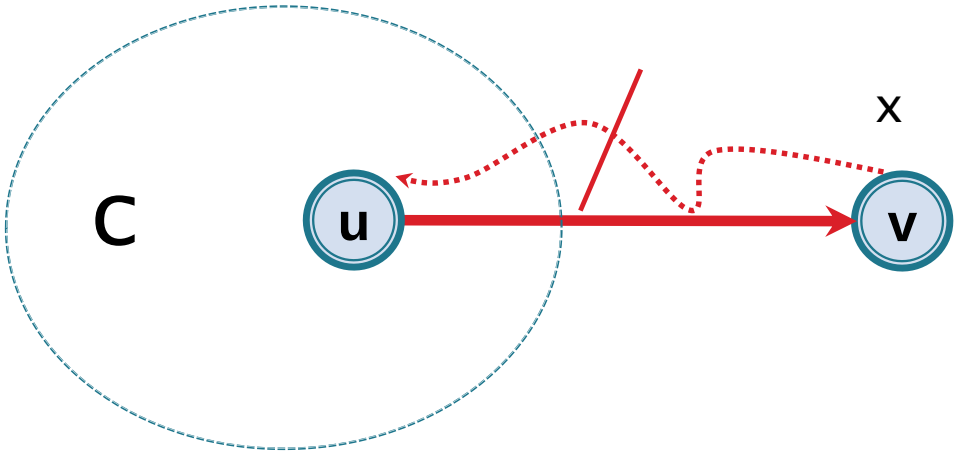
$$\max\{ fin[u] \mid u \in C \} > fin[v]$$



Corectitudine – Algoritmul lui Kosaraju

Demonstrație: Fie x primul vârf din $C \cup \{v\}$ vizitat la pasul 1

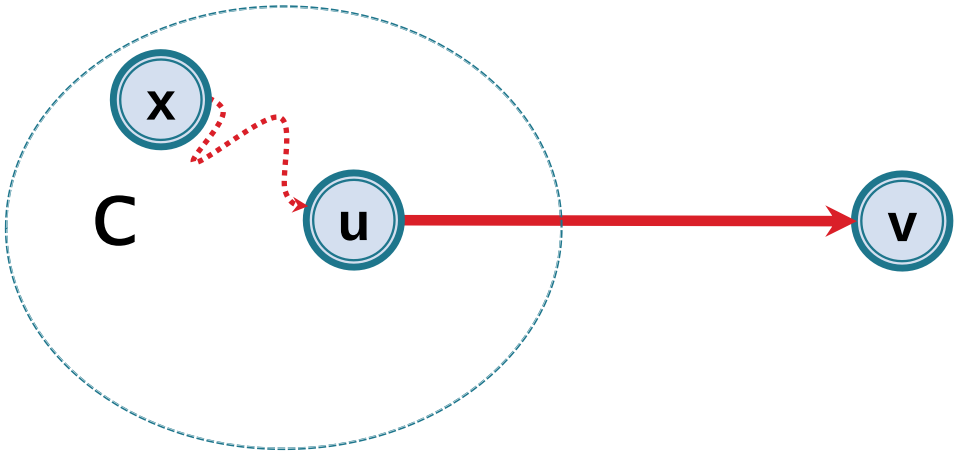
Cazul 1: $x = v$.



$x = v$ va fi finalizat inainte de a fi descoperit vreun vârf din C

Corectitudine – Algoritmul lui Kosaraju

Cazul 2: $x \in C$



v este accesibil din x și este încă nevizitat =>

va fi descoperit și finalizat în DFS(x,G), deci înaintea lui x:

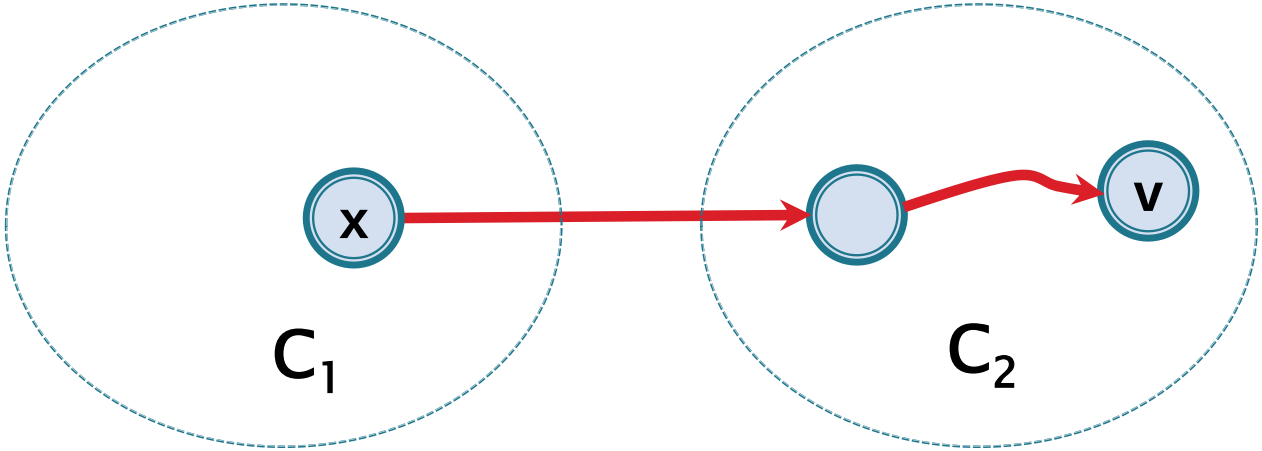
$$fin[v] < fin[x]$$

Corectitudine – Algoritmul lui Kosaraju

Consecința 1

Fie C_1, C_2 componente tare conexe astfel încât există un drum de la C_1 la C_2. Atunci

$$\max\{ fin[u] \mid u \in C_1 \} > \max\{fin[v] \mid v \in C_2 \}$$



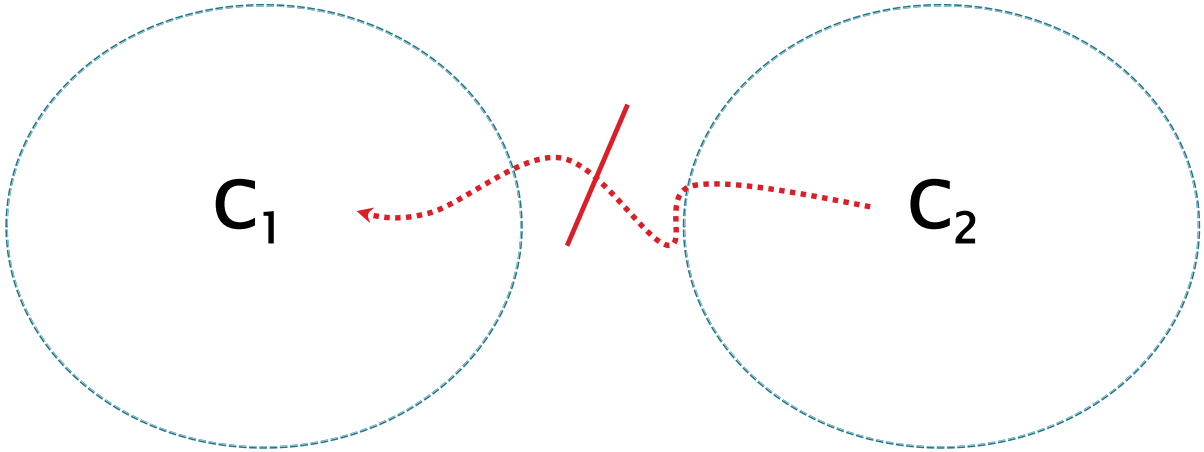
$$fin[x] > fin[v]$$

Corectitudine – Algoritmul lui Kosaraju

Consecința 2

Fie C_1, C_2 componente tare conexe astfel încât

$$\max\{ fin[u] \mid u \in C_1 \} > \max\{ fin[v] \mid v \in C_2 \}$$



Atunci

- nu există drum de la C_2 la C_1 în G
- nu există drum de la C_1 la C_2 în G^T

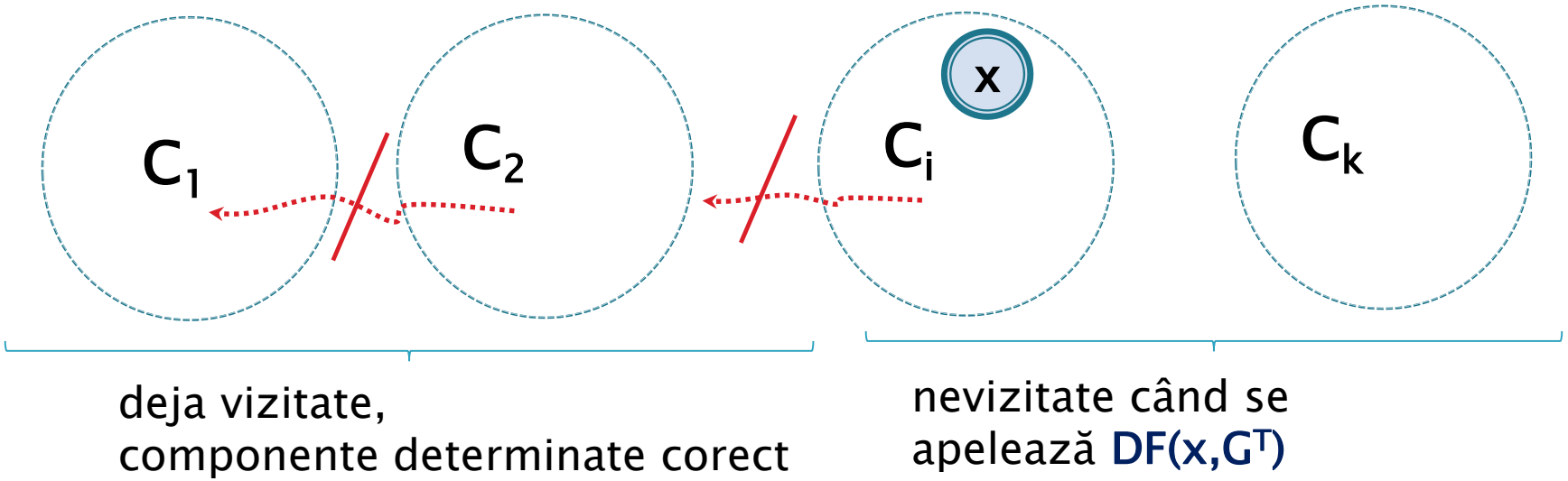
Corectitudine – Algoritmul lui Kosaraju

Teoremă – Corectitudinea Algoritmului Kosaraju

Fie $C_1, C_2, \ldots, C_k$ toate componente tare conexe ale lui G aî

$$\max\{ fin[u] \mid u \in C_1 \} > \max\{ fin[u] \mid u \in C_2 \} > \ldots > \max\{ fin[u] \mid u \in C_k \}.$$

Fie x cel de al i-lea vârf pentru care se apelează $DF(x,G^T)$ la Pasul 2.



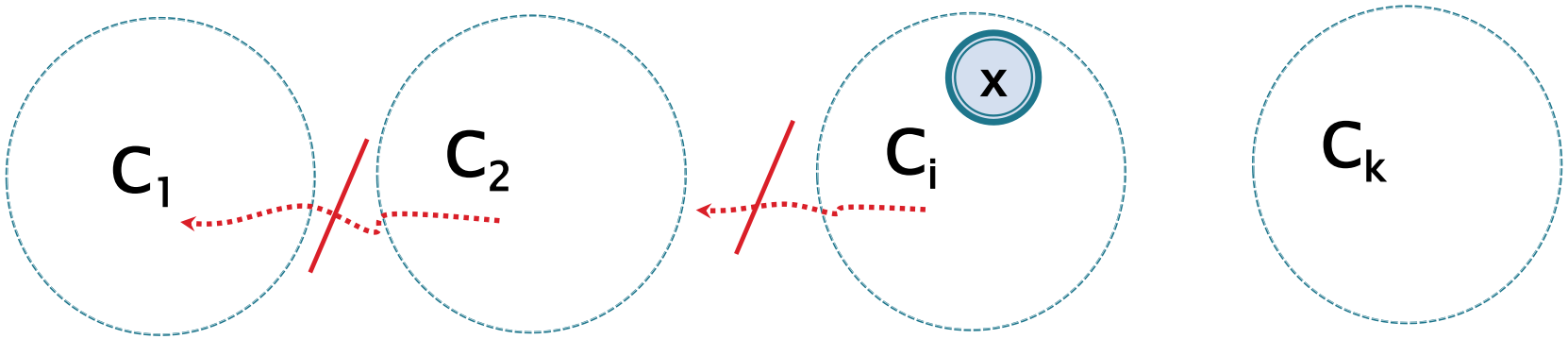
Corectitudine – Algoritmul lui Kosaraju

Teoremă – Corectitudinea Algoritmului Kosaraju

Fie $C_1, C_2, \ldots, C_k$ toate componente tare conexe ale lui G aî

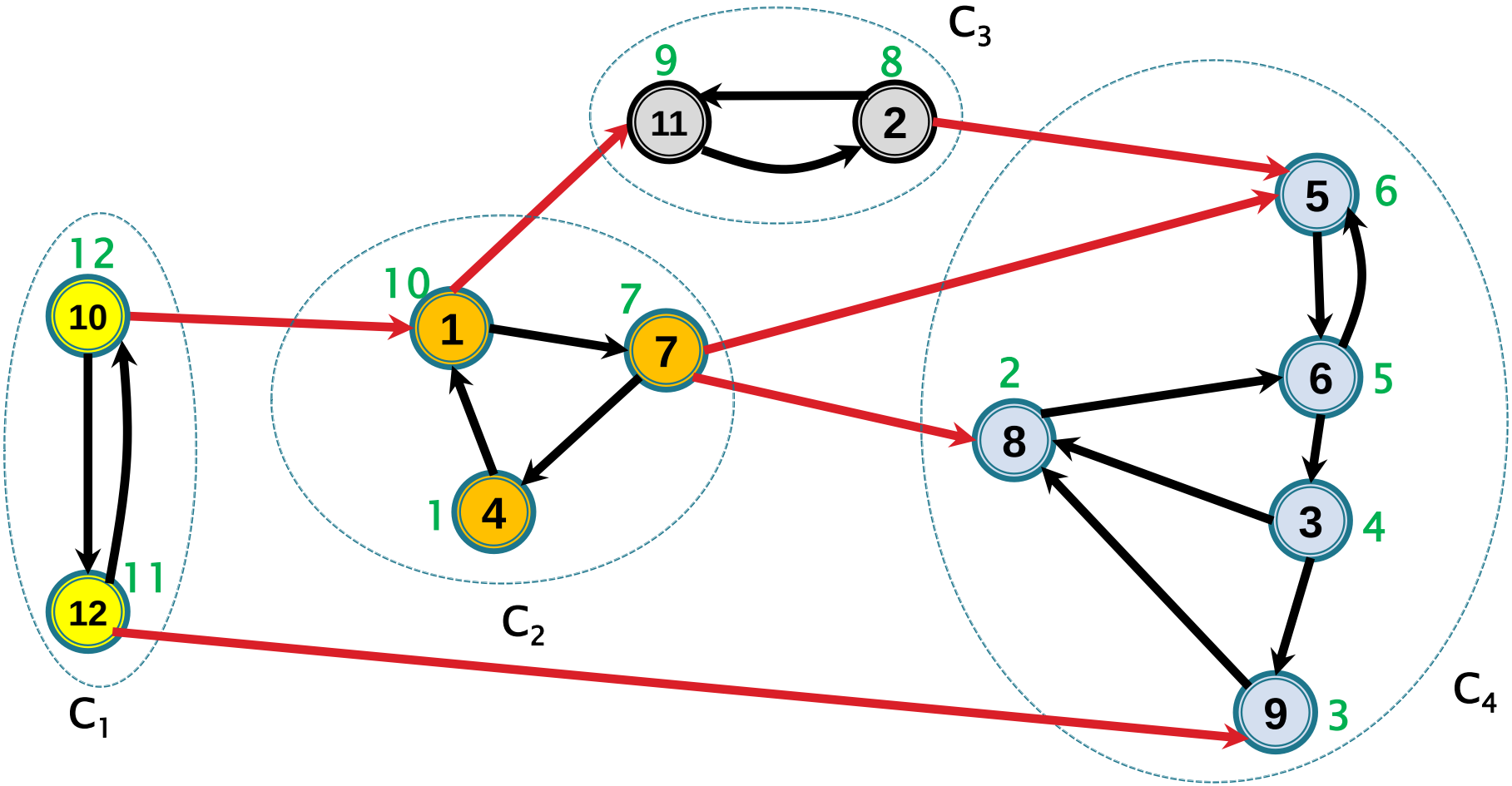
$$\max\{ fin[u] \mid u \in C_1 \} > \max\{ fin[u] \mid u \in C_2 \} > \ldots > \max\{ fin[u] \mid u \in C_k \}.$$

Fie x cel de al i-lea vârf pentru care se apelează $DF(x,G^T)$ la Pasul 2.



Atunci, la momentul acestui apel toate vârfurile din $C_1,\ldots, C_{i-1}$ au fost vizitate, iar cele din $C_i,\ldots, C_k$ nu

În plus, în apelul $DF(x,G^T)$ se vor vizita toate vârfurile componentei C_i și doar acestea



G