Algoritmi pe grafuri - 3

Ștefan Trăușan-Matu

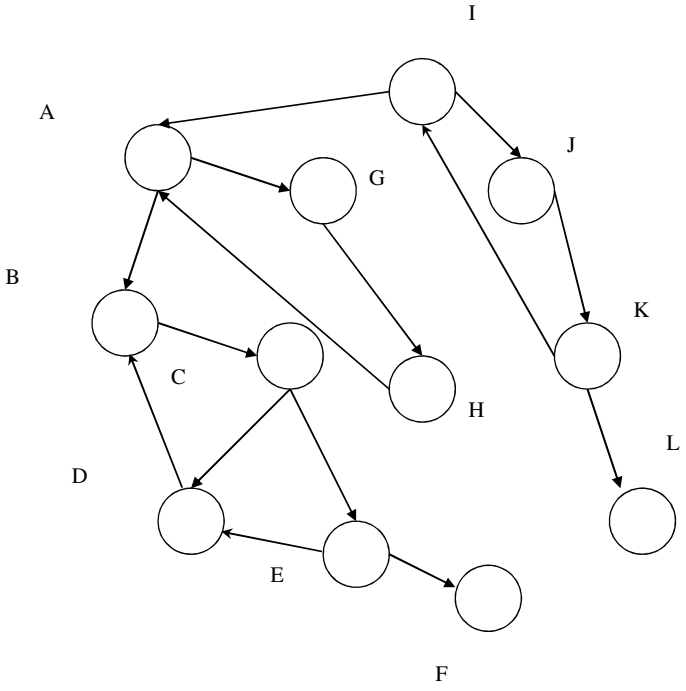
Componente ale unui graf

- G'=(N',A') este **subgraf** al grafului G=(N,A), ddacă N'$\subset$ N și A' $\subset$ A
- G'=(N',A') este **indus** de G=(N,A) ddacă $\forall$u,v $\in$ N', (u,v) $\in$ A =>(u,v) $\in$ A'
- O **componentă** cu o proprietate P a unui graf G este un subgraf maximal cu proprietatea P, indus de acel graf G

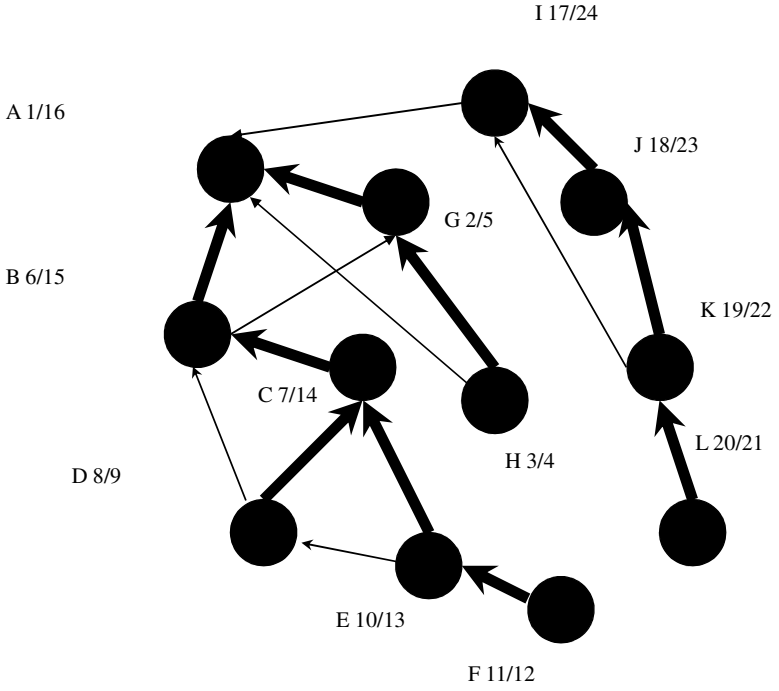
Componente

- **conexe** – grafuri neorientate
- **tare conexe** – grafuri orientate
- **biconectate**

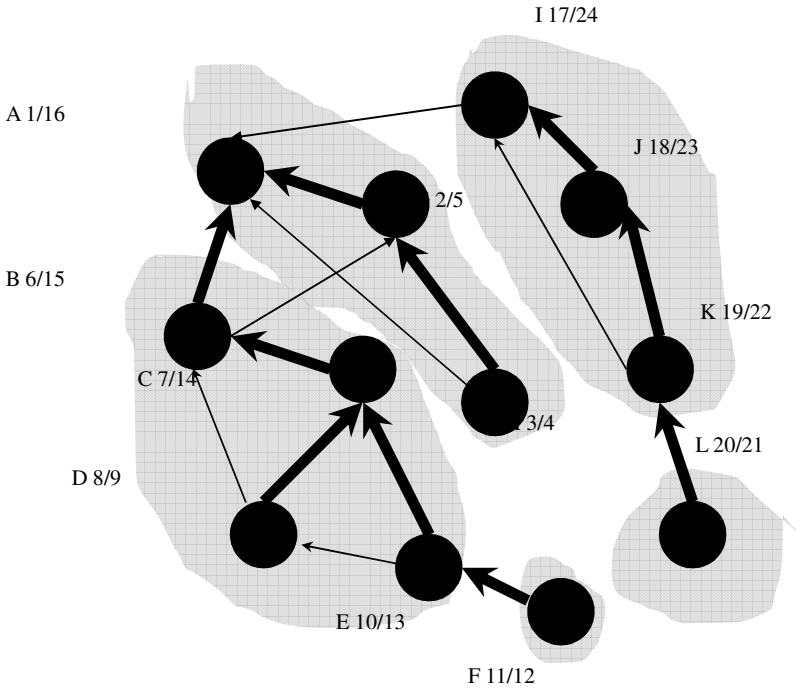
Componente tare conexe – graful inițial



Parcurgerea în adâncime a grafului inițial



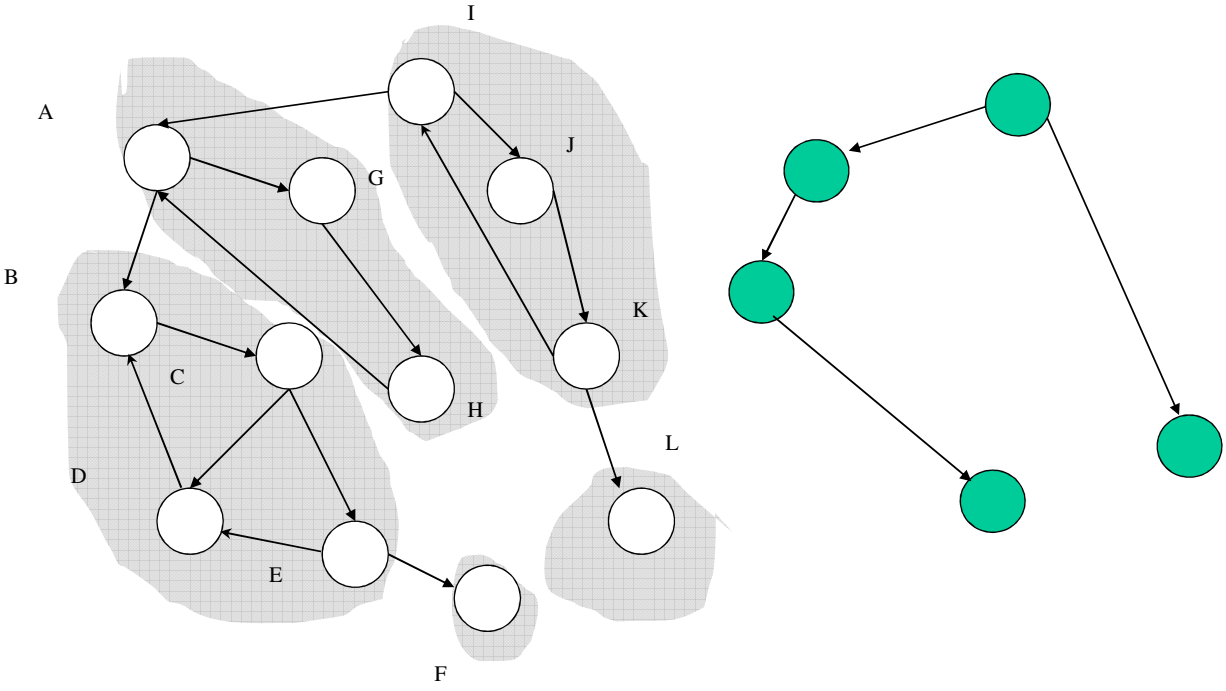
Proprietăți.....?



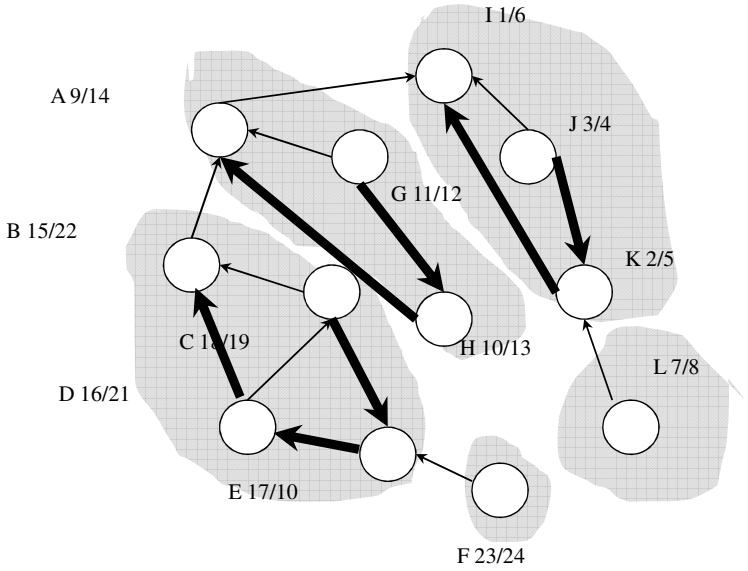
Lema: orice cale între 2 noduri ale unei CTC rămâne în acea CTC

Teorema 1: nodurile din aceeasi CTC sunt grupate in acelasi arbore DFS!

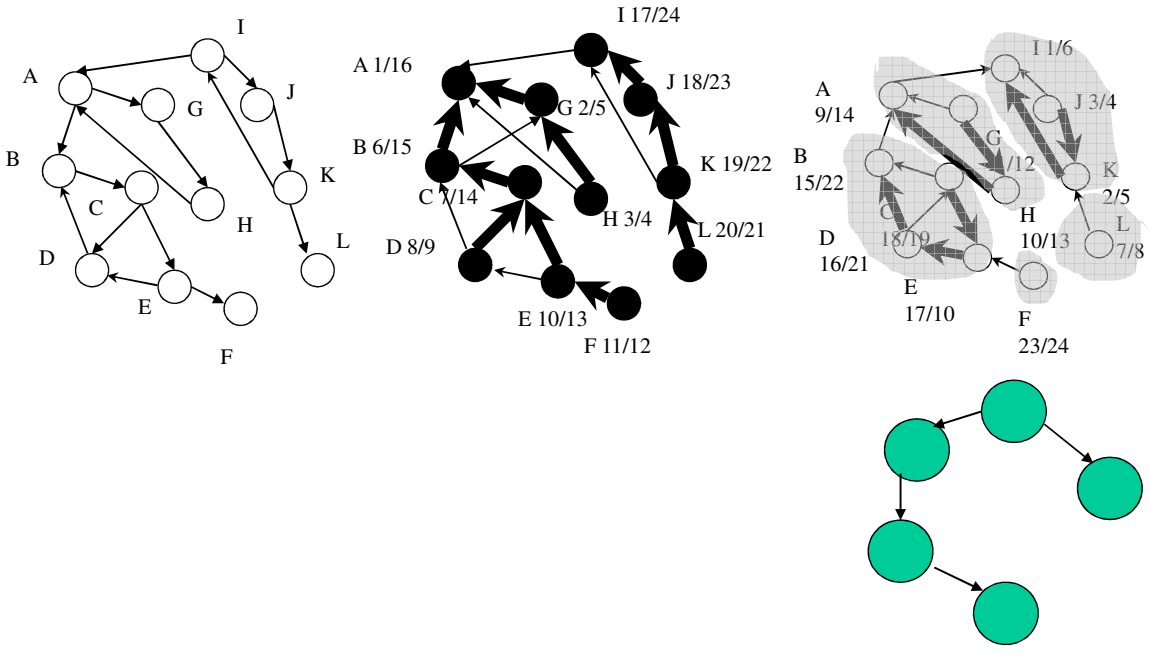
Componente tare conexe – graful inițial + graful componentelor



Parcurgerea în adâncime a grafului transpus în ordinea ....



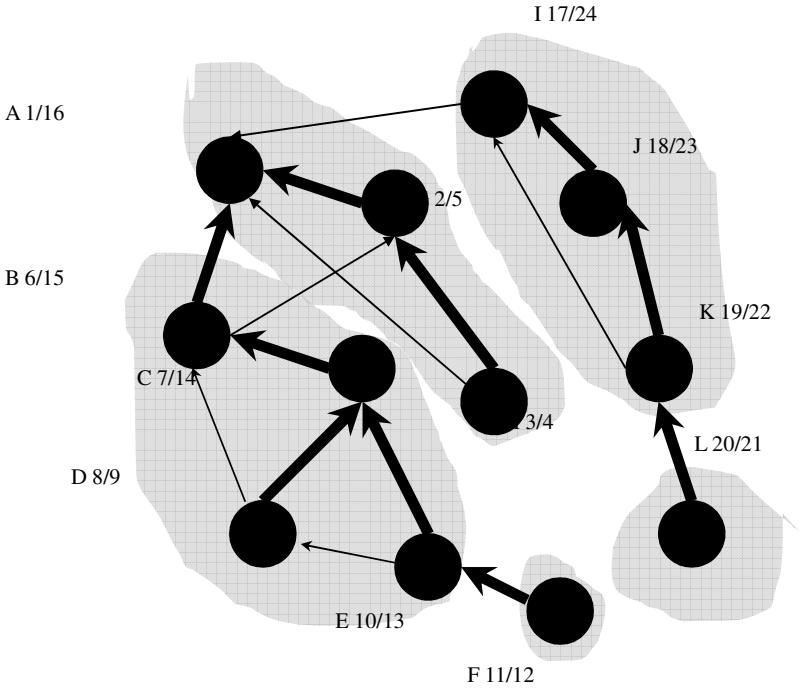
Graful componentelor (GCTC)



Strămoșul

- G=(N,A), u∈N, Φ(u)= stramoșul lui u este accesibil din u și este terminat ultimul într-o parcurgere în adâncime a lui G
  - Φ(u)∈R(u)
  - Φ(u).finis=max{v.finis|v∈R(u)}
- Teoremă Φ(u) satisface urmatoarele proprietati
  - u.finis≤ Φ(u).finis
  - ∀v∈R(u) Φ(v).finis≤Φ(u).finis
  - Φ(Φ(u))= Φ(u)
- Φ(u) este primul nod din CTC descoperit de DFS(G)

A, C, I, F, L sunt strămoși ai nodurilor din componenta conexa din care fac parte



Componente Tare Conexe (CTC)

Teorema 2

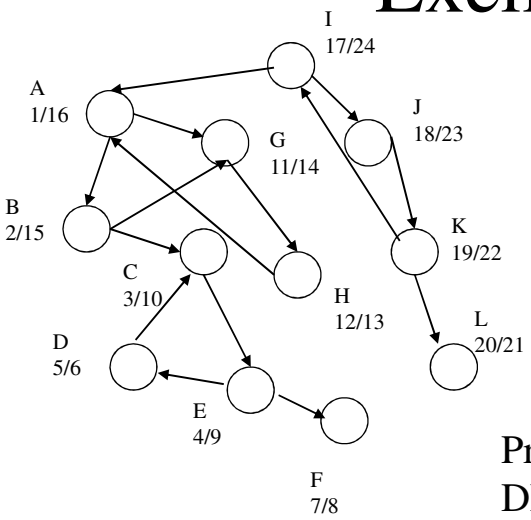
G=(N,A), $\forall u \in N$, u este descendent al lui $\Phi(u)$ in Arb($\Phi(u)$)

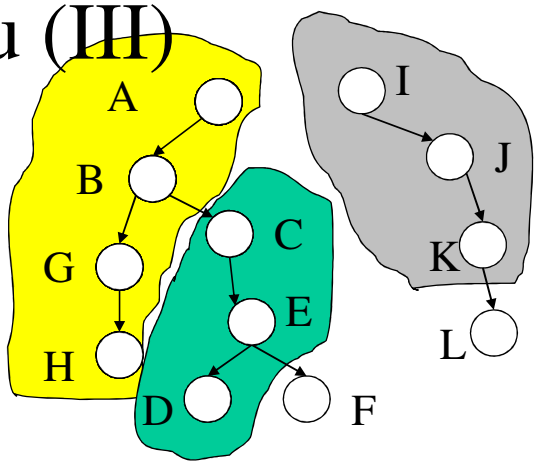
Corolar u și $\Phi(u)$ sunt în aceeași CTC

Teorema 3

G=(N,A), u,v$\in$ N; u si v apartin aceleiași CTC <=> $\Phi(u)=\Phi(v)$

Exemplu (III)



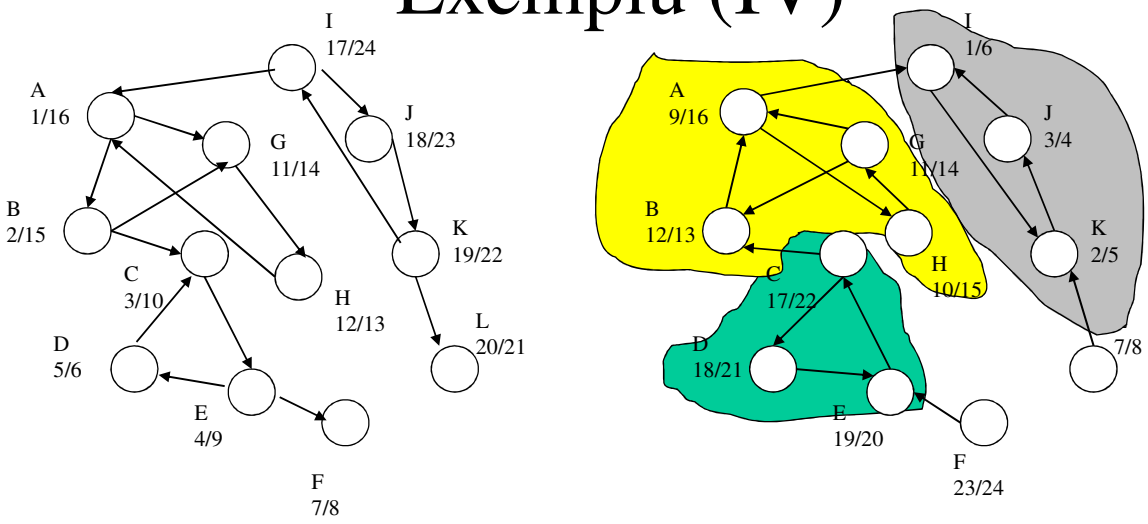


Primul nod dintr-o CTC descoperit la DFS va avea copii in arborele generat de DFS toate elementele componentei conexe!

Componente Tare Conexe (CTC)

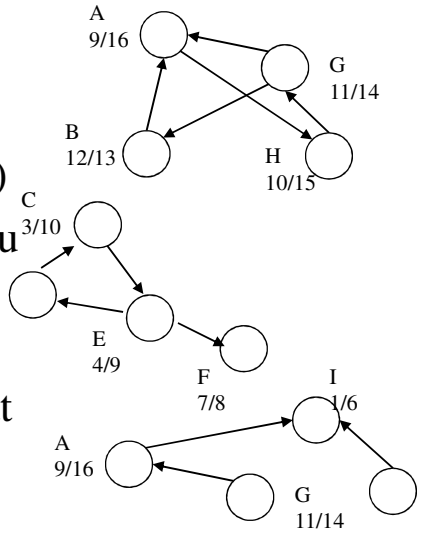
- Problema: trebuie sa eliminam nodurile care nu sunt in componenta conexa
- Vrem ca fiecare arbore construit sa contina o CTC
- =>idee – eliminam nodurile ce nu apartin CTC
  - Daca apartin Arb(u) si nu CTC => ∃ u..v si nu v..u
  - =>DFS pe graful transpus

Exemplu (IV)

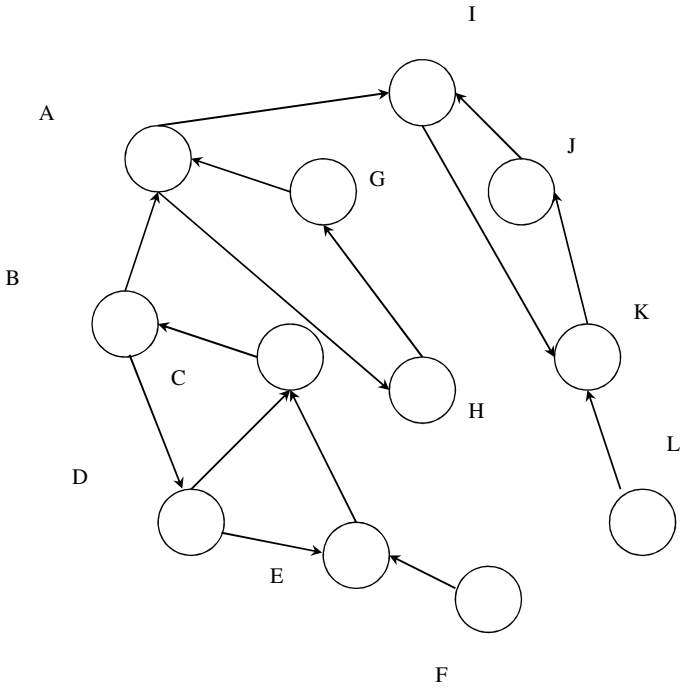

I
17/24
A
1/16
J
18/23
G
11/14
B
2/15
K
19/22
C
3/10
H
12/13
L
20/21
D
5/6
E
4/9
F
7/8
I
1/6
A
9/16
J
3/4
G
11/14
B
12/13
K
2/5
H
10/15
C
17/22
D
18/21
7/8
E
19/20
F
23/24

Componente Tare Conexe (CTC)

- Cazuri in DFS(G^T)
  - v este in CTC descoperita din u=> v poate fi descoperit din u si in DFS(G^T)
  - v$\notin$ CTC dar v$\in$ Arb(u) in DFS(G)=> nu va fi atins in DFS(G^T) din u
  - v$\notin$ CTC dar $\exists$ v..u in G =>v.finis>u.finis=> v va fi deja colorat in negru cand se exploreaza u

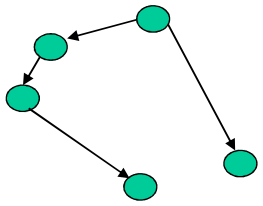


Graful transpus



Observatii

- Inlocuind componentele tare conexe cu noduri obtinem un graf aciclic
- Prima parcurgere DFS este o sortare topologica (de ce?)



Algoritmul lui Kosaraju

- CTC(G)

  Parc-ad(G)

  G^T=transpune(G)

  Parc-ad(G^T) (in bucla principala se trateaza nodurile in ordinea descrescatoare a timpilor de finiș de la primul Parc-ad)
- Componentele conexe sunt reprezentate de padurea de arbori generati de Parc-ad(G^T)

Algoritmul lui Tarjan (I)

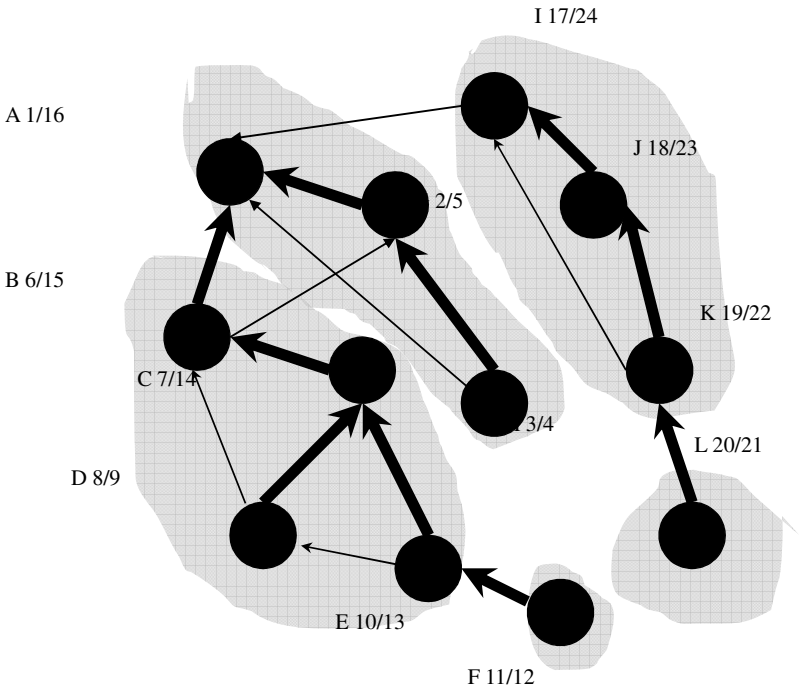
- Bazat tot pe DFS
- Foloseste o singura parcurgere in adancime
- Determina din parcurgere care sunt "radacinile" CTC

Algoritmul lui Tarjan (II)

pentru fiecare n din N **repetă**
    n.debut← -1 // echivalent cu n.culoare←alb
    S←∅; // S stiva
    timp← 0
pentru fiecare n din N **repetă**
    dacă n.debut=-1 **atunci** Parc-ad-rec_Tarjan(n)

Algoritmul lui Tarjan (III)

```
Parc-ad-rec_Tarjan(n)
    n.debut←timp
    n.minim←timp
    timp← timp+1
    Push(S,n)
    Pentru fiecare v succesor al lui n repetă
      dacă v.debut=-1
        atunci
          Parc-ad-rec_Tarjan(v)
          n.minim←min(n.minim,v.minim)
        altfel dacă v∈ S
              atunci n.minim←min(n.minim,v.debut)
      dacă n.minim =n.debut
        tipărește "CTC"
        repetă
            u=pop(S)
            tipărește u
        până când u=n
```


I 17/24
A 1/16
J 18/23
2/5
B 6/15
K 19/22
C 7/14
3/4
L 20/21
D 8/9
E 10/13
F 11/12