



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI



Instrumente Structurale
2007-2013



Platformă de e-learning și curriculum e-content pentru învățământul superior tehnic

Proiectarea Algoritmilor

11. Componente tare conexe

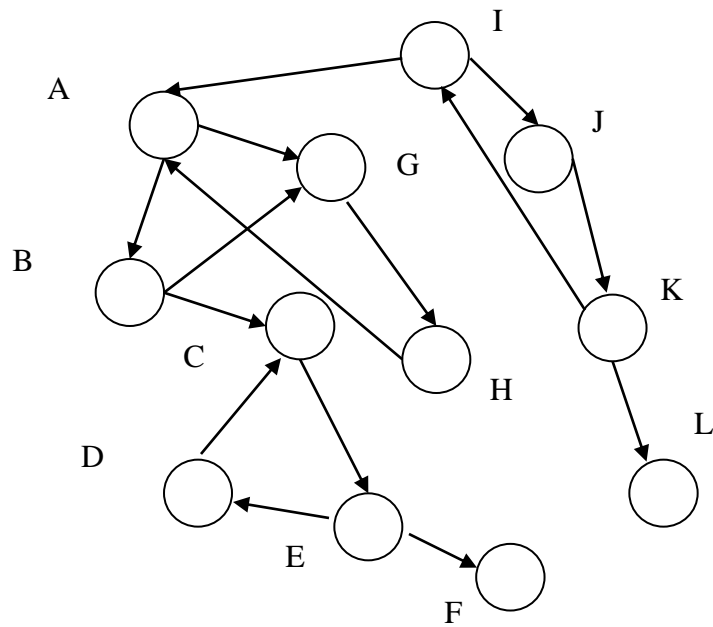
Bibliografie

- Giumale – Introducere in Analiza Algoritmilor cap. 5.2
- Cormen – Introducere în Algoritmi cap. 23.5
- http://en.wikipedia.org/wiki/Tarjan%27s_strongly_connected_components_algorithm

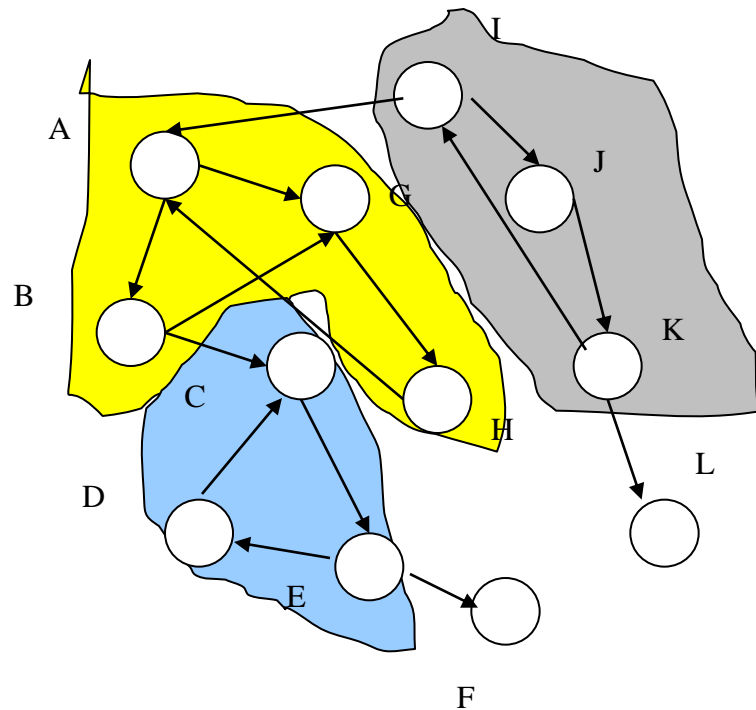
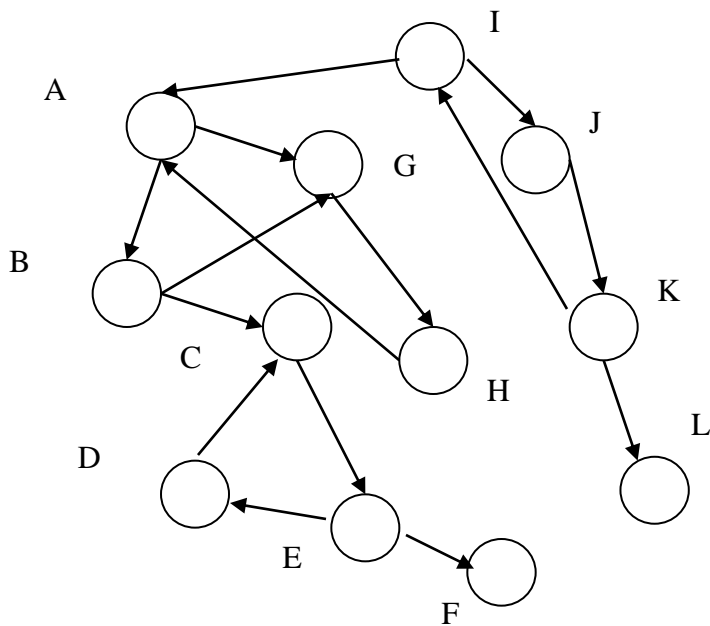
Componente Tare Conexa (CTC)

- **Definiție:** Fie $G = (V, E)$ un graf orientat. G este **tare-conex** $\Leftrightarrow \forall u, v \in V \exists$ o cale $u..v$ și o cale $v..u$ ($u \in R(v)$ && $v \in R(u)$).
- **Definiție:** $G = (V, E)$ graf orientat. $G' = (V', E')$, $V' \subseteq V$, $E' \subseteq E$. G' este o **CTC** a lui $G \Leftrightarrow G'$ e **tare-conex** ($\forall u, v \in V'$, $u \in R(v)$ && $v \in R(u)$) și G' este **maximal**.
- **Lema 5.6:** $G = (V, E)$ graf orientat, G' CTC, $\forall u, v \in V' \Rightarrow \forall u..v$ din G are noduri exclusiv în V'
 - **Dem:** $\forall z$ a.i. $u..z..v \Rightarrow z \in R(u)$ și $v \in R(z)$. Dar $u \in R(v) \Rightarrow z \in R(v) \Rightarrow v$ și z sunt în aceeași CTC.

Exemplu (I) – determinare CTC



Exemplu (II) – determinare CTC(2)

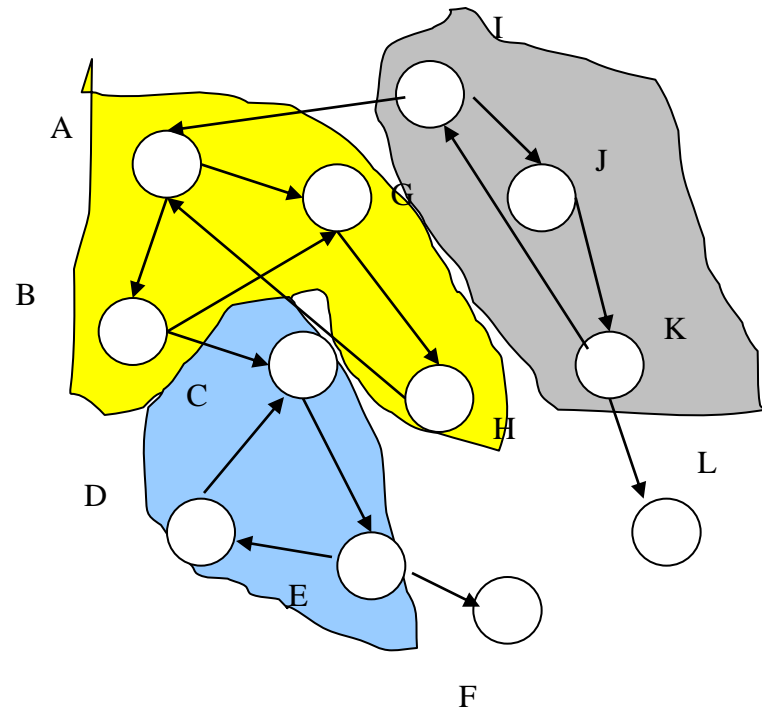
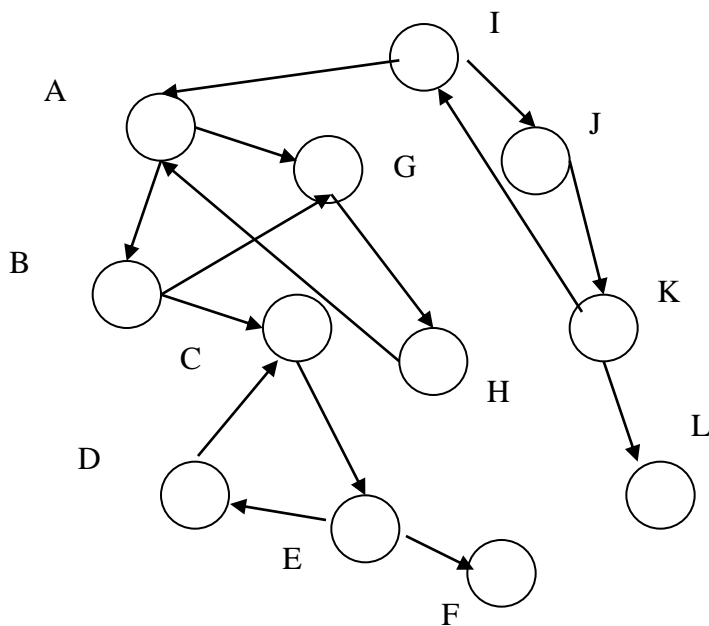


Componente Tare Conexa (CTC)

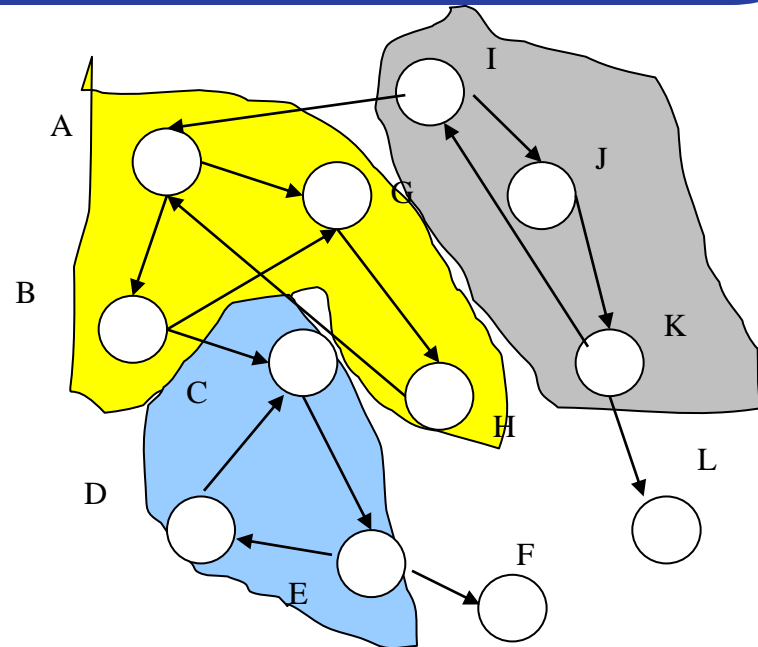
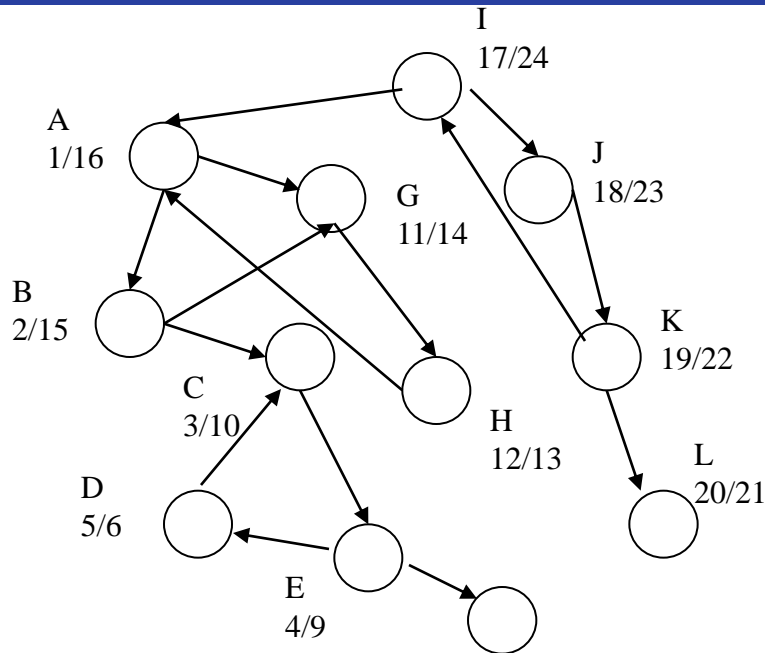
- **Teorema 5.10:** $G = (V, E)$ orientat, $G' = (V', E')$ o CTC a lui G . Toate nodurile $v \in V'$ sunt grupate in același $Arb(u)$ construit de $DFS(G)$, unde u este primul nod descoperit al componentei.
 - **Dem:** $\forall v \in V', v \neq u, \exists u..v$ drum cu noduri albe la momentul descoperirii $d(u)$; toate nodurile drumului sunt in V' (conf. [Lema 5.6](#))
 \Rightarrow (din [Teorema drumurilor albe](#)) v este descendent al lui u in $Arb(u)$

Exemplu (III) – DFS

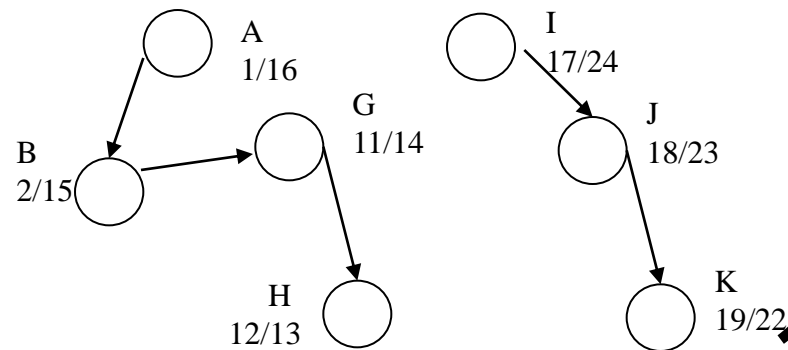
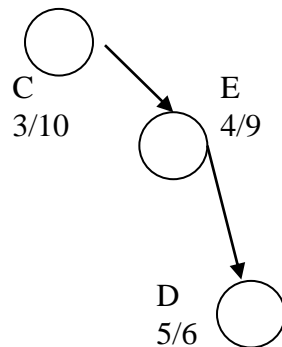
- Aplicare DFS pornind din primul nod al fiecărei CTC



Exemplu (IV) - DFS(2)



Conform Teoremei
nodurile din aceeași
CTC sunt grupate în
același arbore DFS!



Componente Tare Conexa (CTC)

- **Definiție:** $G = (V, E)$ orientat, $u \in V$. $\Phi(u)$ = **strămoș** DFS al lui u determinat în cursul $\text{DFS}(G)$ dacă:
 - $\Phi(u) \in R(u)$
 - $f(\Phi(u)) = \max\{f(v) \mid v \in R(u)\}$
- **Ce e $\Phi(u)$?** $\Phi(u)$ este primul nod din CTC descoperit de $\text{DFS}(G)$
- **Teorema 5.11:** $\Phi(u)$ satisface următoarele proprietăți:
 1. $f(u) \leq f(\Phi(u))$ **când e egalitate?** u este primul nod din CTC
 2. $\forall v \in R(u), f(\Phi(v)) \leq f(\Phi(u))$ **ce înseamnă ca e egalitate?**
 u și v sunt în aceeași CTC
 3. $\Phi(\Phi(u)) = \Phi(u)$ **Dem:** $\Phi(\Phi(u)) \in R(\Phi(u)) \stackrel{2}{\Rightarrow} f(\Phi(\Phi(u))) \leq f(\Phi(u))$ și
 $\stackrel{1}{\Rightarrow} f(\Phi(\Phi(u))) \geq f(\Phi(u)) \rightarrow f(\Phi(\Phi(u))) = f(\Phi(u)) \rightarrow \Phi(\Phi(u)) = \Phi(u)$

Exemplu (V) – stramosi

A, C, I sunt strămoși DFS ai nodurilor din componenta conexă din care fac parte.

Proprietățile strămoșilor:

- $f(u) \leq f(\Phi(u))$

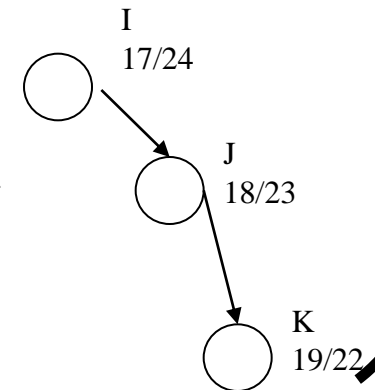
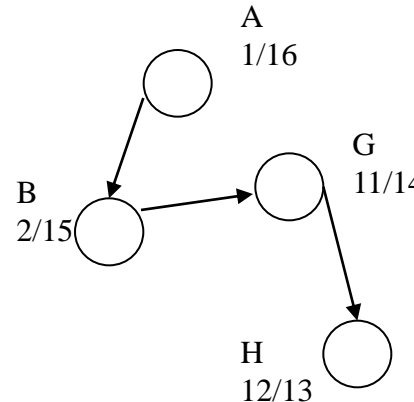
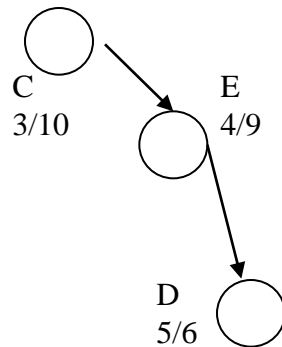
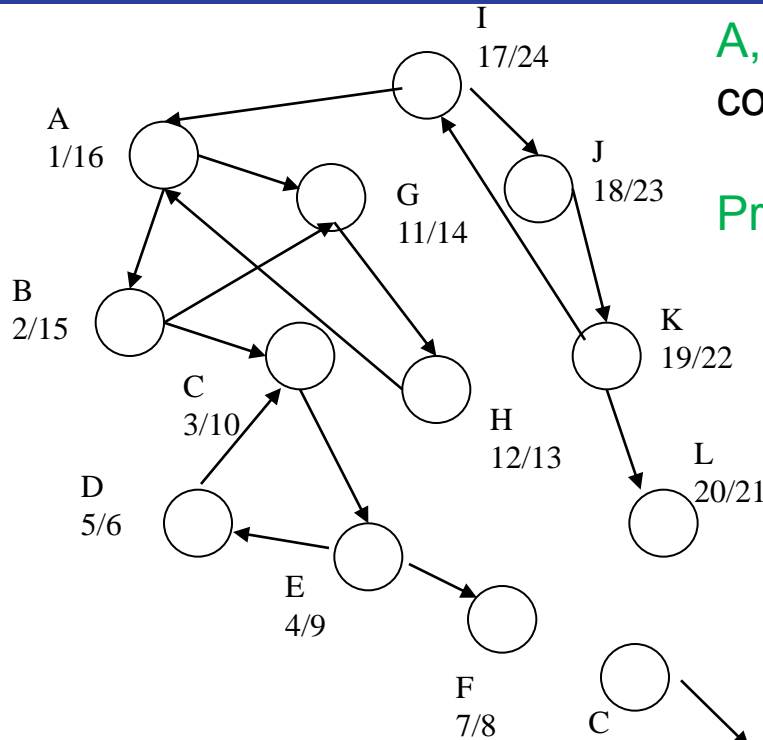
Ex: $f(B) \leq f(\Phi(B)) = f(A)$

- $\forall v \in R(u), f(\Phi(v)) \leq f(\Phi(u))$

Ex: $E \in R(B), f(C) = f(\Phi(E)) \leq f(\Phi(B)) = f(A)$

- $\Phi(\Phi(u)) = \Phi(u)$

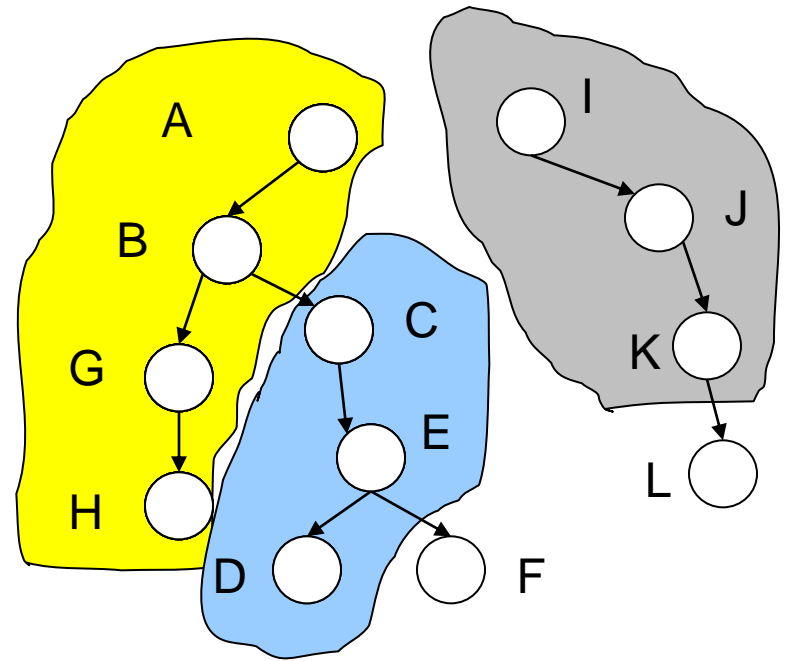
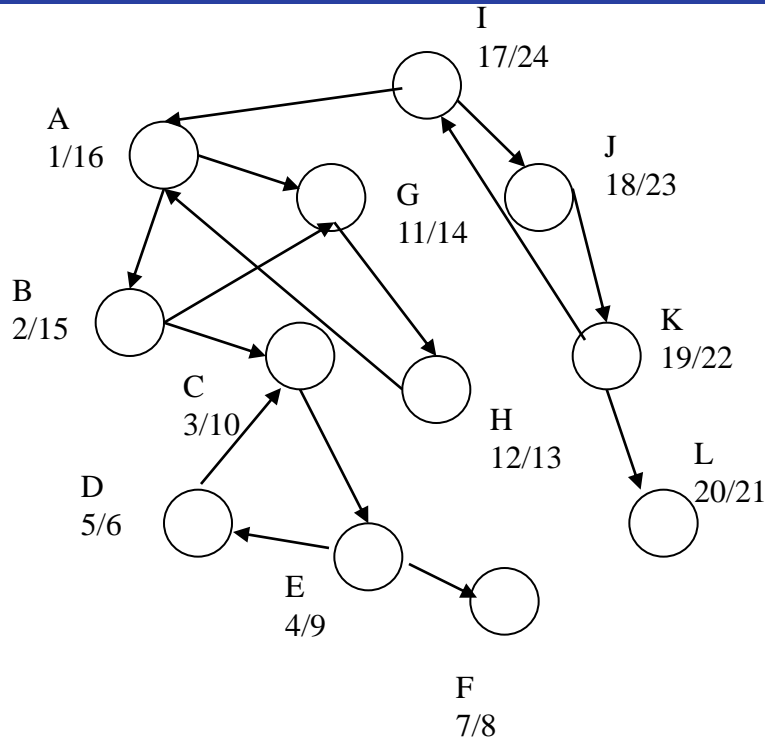
Ex: $\Phi(\Phi(E)) = \Phi(C) = C = \Phi(E)$



Componente Tare Conexa (CTC)

- Din **Definiție**: $\Phi(u) \in R(u)$ && $f(\Phi(u)) = \max\{f(v) \mid v \in R(u)\}$.
- **Teorema 5.12.** $G = (V, E)$ orientat, $\forall u \in V$, u este descendent al lui $\Phi(u)$ în $Arb(\Phi(u))$ construit de DFS.
 - **Dem:** prin considerarea tuturor culorilor posibile ale lui $\Phi(u)$ la momentul $d(u)$.
- **Teorema 5.13.** $G = (V, E)$ orientat, $\forall u, v \in V$; u și v aparțin aceleiași CTC $\Leftrightarrow \Phi(u) = \Phi(v)$.
 - **Dem folosind proprietățile strămoșilor :**
 - $\forall u, v \in$ aceleiași CTC $\Rightarrow \Phi(u) = \Phi(v)$: $v \in R(u)$, $f(\Phi(v)) \leq f(\Phi(u))$ și $u \in R(v)$, $f(\Phi(u)) \leq f(\Phi(v)) \Rightarrow f(\Phi(u)) = f(\Phi(v)) \rightarrow \Phi(u) = \Phi(v)$
 - $\Phi(u) = \Phi(v) \Rightarrow \Phi(u) \in R(u) \Rightarrow u$ și $\Phi(u) \in$ aceleiași CTC și $\Phi(u) \in R(v) \Rightarrow v$ și $\Phi(u) \in$ aceleiași CTC $\Rightarrow u$ și $v \in$ aceleiași CTC

Exemplu (VI)

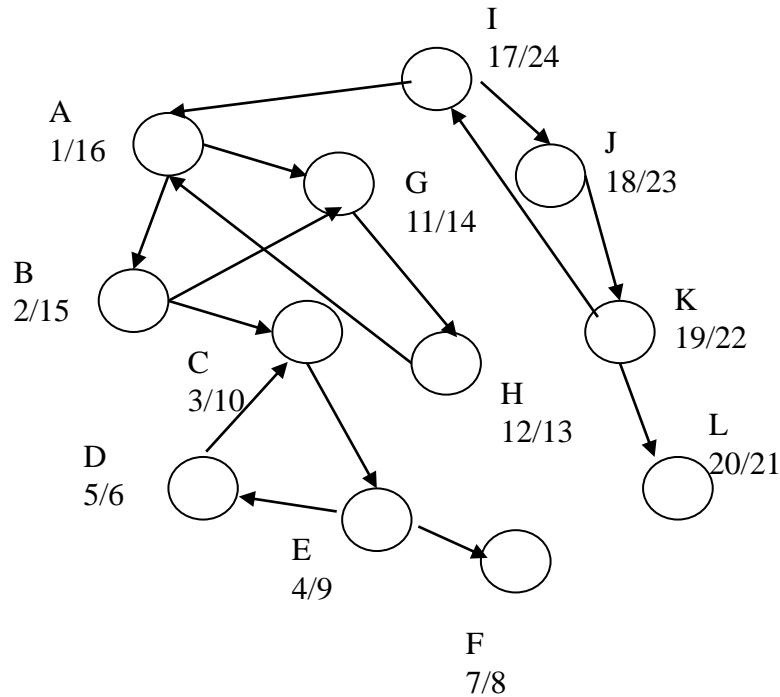


Primul nod dintr-o CTC descoperit la DFS va avea copii in arborele generat de DFS toate elementele componentei conexe!

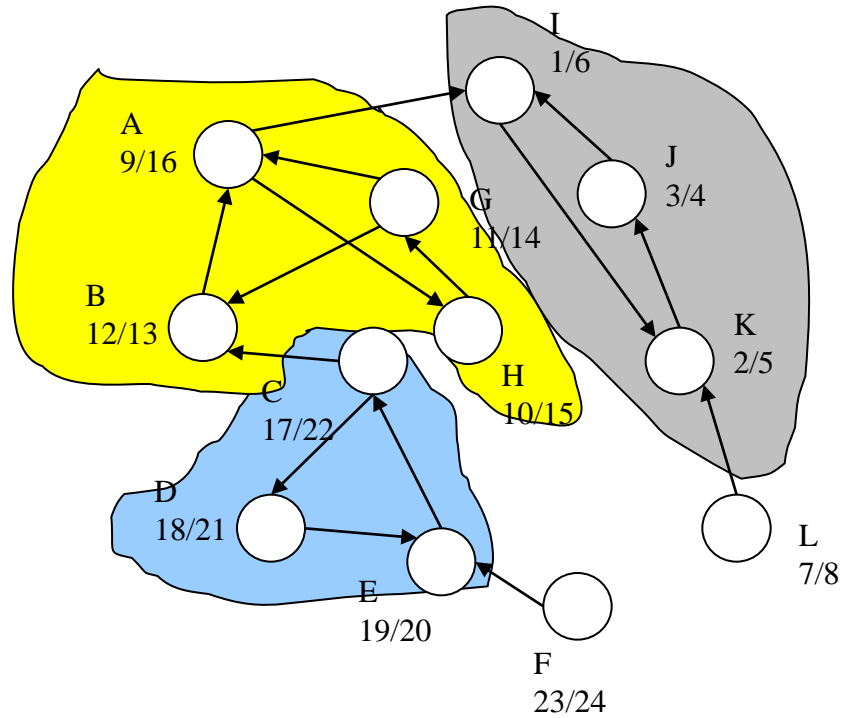
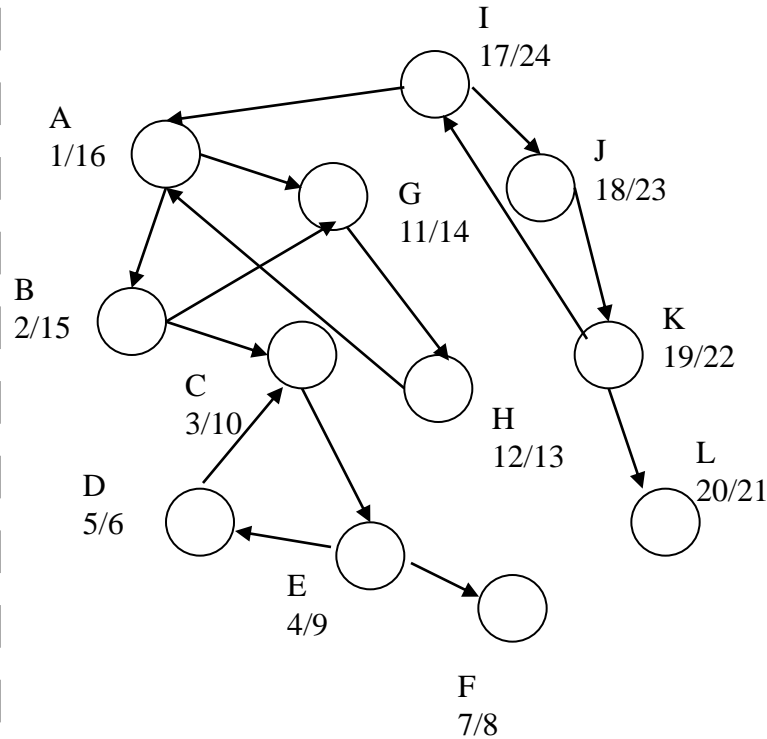
Componente Tare Conexa (CTC)

- **Probleme:**
 - trebuie să eliminăm nodurile care nu sunt în componenta conexă.
 - vrem ca fiecare arbore construit să conțină o CTC.
- **Idee** – eliminăm nodurile ce nu aparțin CTC!
Cum?
 - Dacă aparțin $\text{Arb}(u)$ și nu CTC $\Rightarrow \exists u..v$ și $\nexists v..u$.
 - \rightarrow DFS pe graful transpus, în ordinea inversă a timpilor de finalizare obținuți din DFS pe graful normal!

Exemplu (VII) – DFS (G^T)

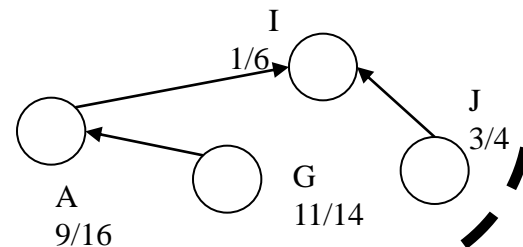
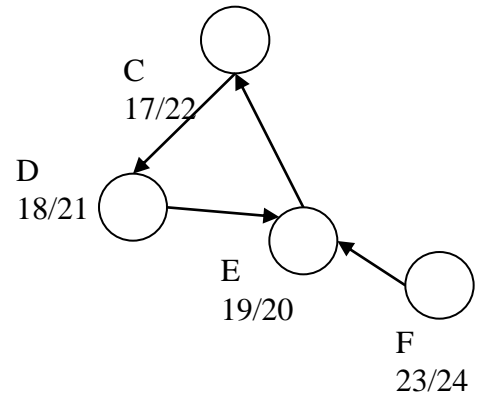
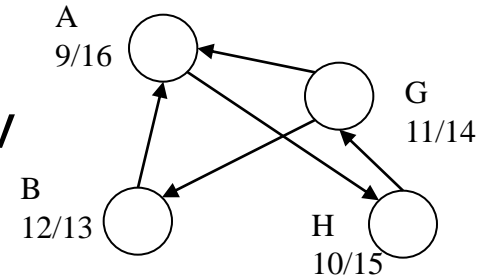


Exemplu (VIII) – DFS (G^T) (2)



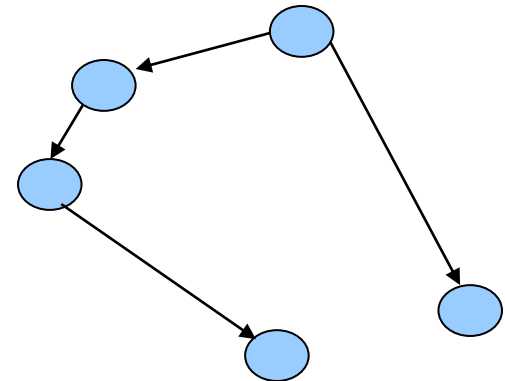
Componente Tare Conexa (CTC)

- Cazuri in $\text{DFS}(G^T)$:
 - v este in CTC descoperita din $u \rightarrow v$ poate fi descoperit din u si in $\text{DFS}(G^T)$.
 - $v \notin \text{CTC}$ dar $v \in \text{Arb}(u)$ in $\text{DFS}(G) \rightarrow$ nu va fi atins in $\text{DFS}(G^T)$ din u .
 - $v \notin \text{CTC}$ dar $\exists v..u$ in $G \rightarrow f(v) > f(u) \rightarrow v$ va fi deja colorat in negru când se explorează u .



Observatii

- Înlocuind componentele tare conexe cu noduri obținem un graf aciclic. **De ce?**
- Pentru că altfel am avea o singură CTC!
- Prima parcurgere DFS este o sortare topologică. **De ce?**
- Pentru că sortează nodurile in ordinea inversă a timpilor!



Pseudocod algoritm CTC

- Algoritmul lui Kosaraju:
- CTC(G)
 - DFS(G)
 - $G^T = \text{transpune}(G)$
 - DFS(G^T) (in bucla principală se tratează nodurile în ordinea descrescătoare a timpilor de finalizare de la primul DFS)
- Componentele conexe sunt reprezentate de pădurea de arbori generați de DFS(G^T).

Complexitate?

Algoritmul lui Kosaraju

Complexitate

$O(n+m)$

n = numar noduri

m = numar muchii

Corectitudine algoritm CTC (1)

- **Teoremă:** Algoritmul CTC calculează corect componentele tare conexe ale unui graf $G = (V, E)$.
- **Dem. prin inducție după nr. de arbori de adâncime găsiți de PAD a G^T că vârfurile din fiecare arbore formează o CTC:**
 - Fiecare pas demonstrează ca arborele format în acel pas e o CTC, presupunând că toți arborii produși deja sunt CTC.
 - P_1 : trivial pt. că \nexists arbori anteriori.
 - $P_n \rightarrow P_{n+1}$: Fie arborele T obținut în pasul curent având rădăcina r . Notăm $C_r = \{v \in V \mid \Phi(v) = r\}$.

Corectitudine algoritm CTC (2)

- Demonstrăm că $u \in T \Leftrightarrow u \in C_r$:
- $\Rightarrow u \in C_r \rightarrow \exists r..u \rightarrow$ toate nodurile din C_r ajung in același arbore de PAD. Dar $r \in C_r$ și r e rădăcina lui $T \rightarrow \forall u \in C_r \Rightarrow u \in T$
- \Leftarrow demonstrăm că $\forall w$ a.i. $f(\Phi(w)) > f(r)$ sau $f(\Phi(w)) < f(r)$, $w \notin T$
 - Dacă $f(\Phi(w)) > f(r) \rightarrow$ la $d(r)$, w e deja pus in CTC cu rădăcina $\Phi(w)$ pt. că nodurile sunt considerate in ordinea inversa a timpilor de finalizare $\rightarrow w \notin T$
 - Dacă $f(\Phi(w)) < f(r) \rightarrow w \notin T$ pt. că altfel ($w \in T$) $\rightarrow \exists r..w$ in $G^T \rightarrow \exists w..r$ in $G \rightarrow r \in R(w)$, $\rightarrow f(\Phi(w)) \geq f(\Phi(r)) = f(r)$ (F)
 - $\rightarrow T$ conține doar nodurile pt. care $\Phi(w) = r \rightarrow T = C_r$