Componente tare conexe

Componente tare conexe

Într-un graf **orientat**, avem 2 definiții de conexitate.

Un graf orientat este **slab conex** dacă există un drum de la oricare nod la oricare altul, **considerând muchiile grafului neorientate.**

Un graf orientat este **tare conex** dacă există un drum de la oricare nod la oricare altul.

Componente tare conexe

Într-un graf **orientat**, avem 2 definiții de conexitate.

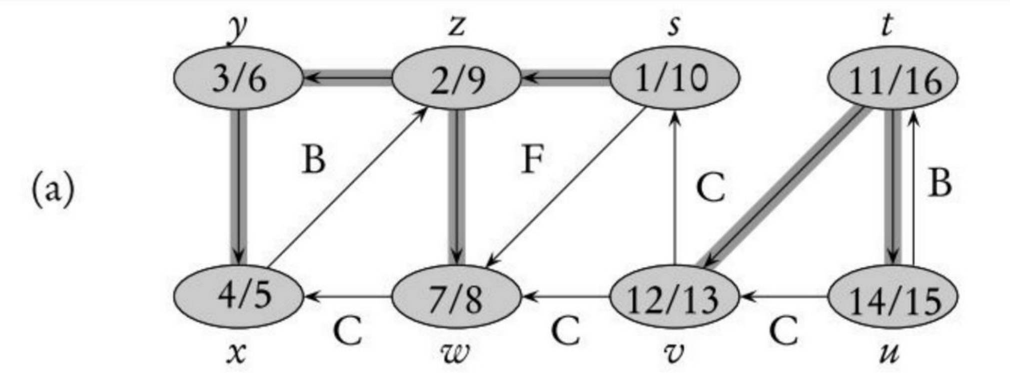
Un graf orientat este **slab conex** dacă există un drum de la oricare nod la oricare altul, **considerând muchiile grafului neorientate.**

Un graf orientat este **tare conex** dacă există un drum de la oricare nod la oricare altul.

Graful este slab conex.

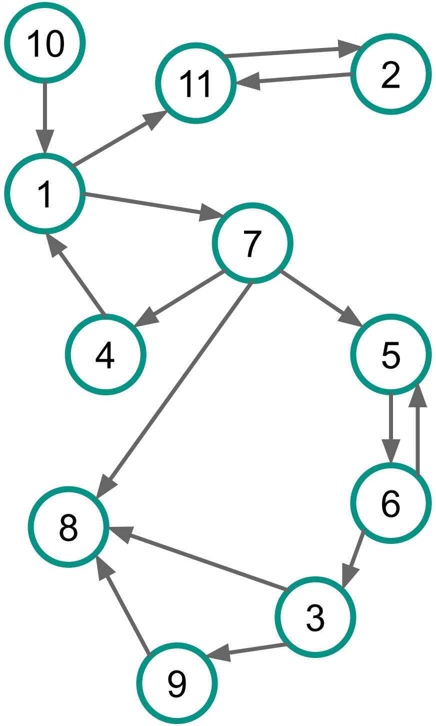
Graful **nu** este tare conex.

- nu există drumul s→v



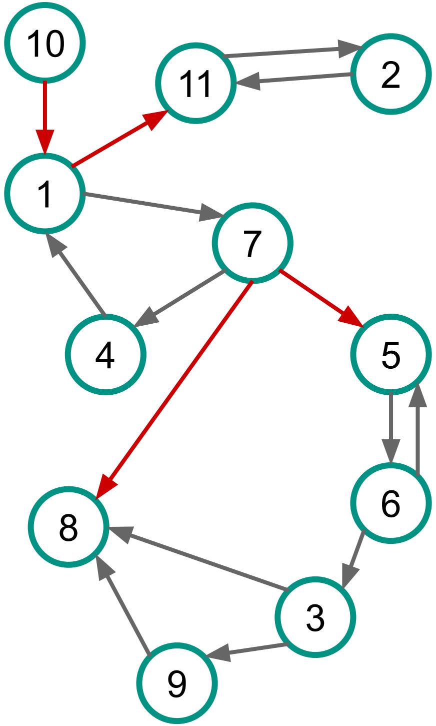
Componente tare conexe

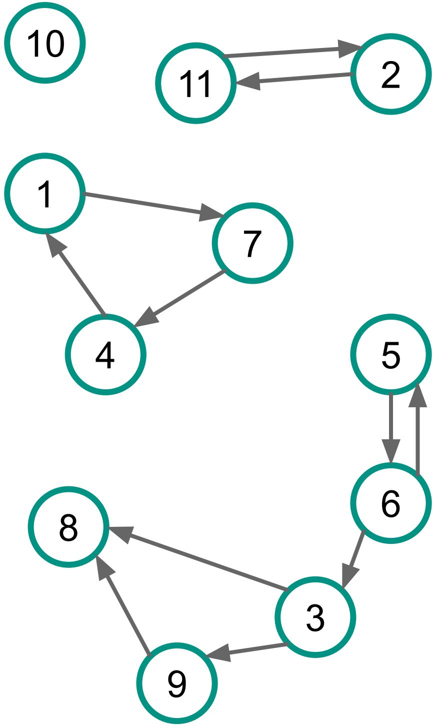
Ce element al grafului ne poate da informații despre componentele tare conexe?



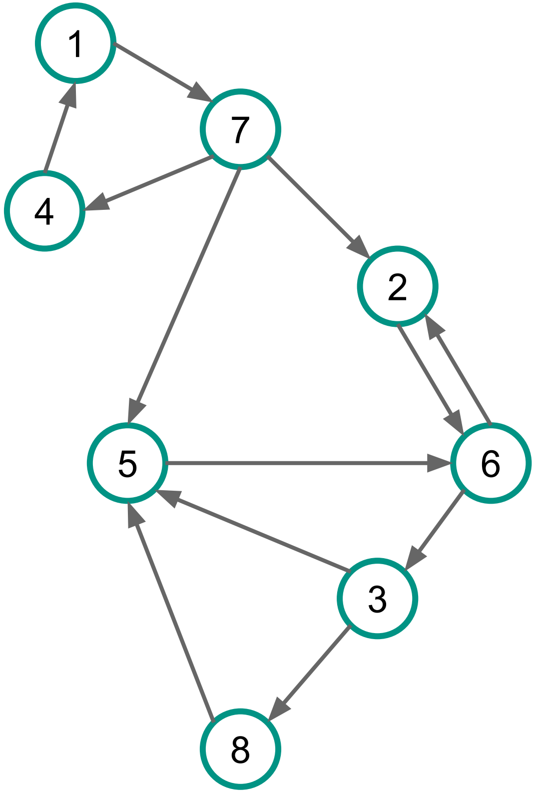
Componente tare conexe

Ce element al grafului ne poate da informații despre componentele tare conexe?

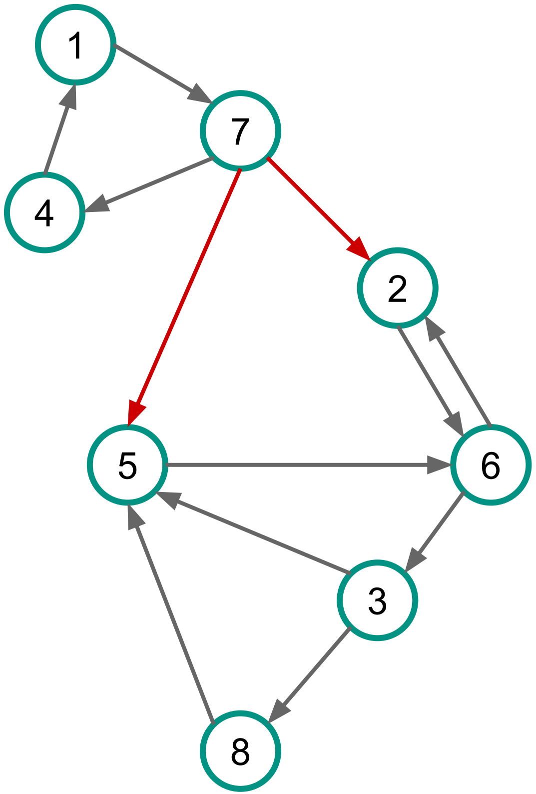


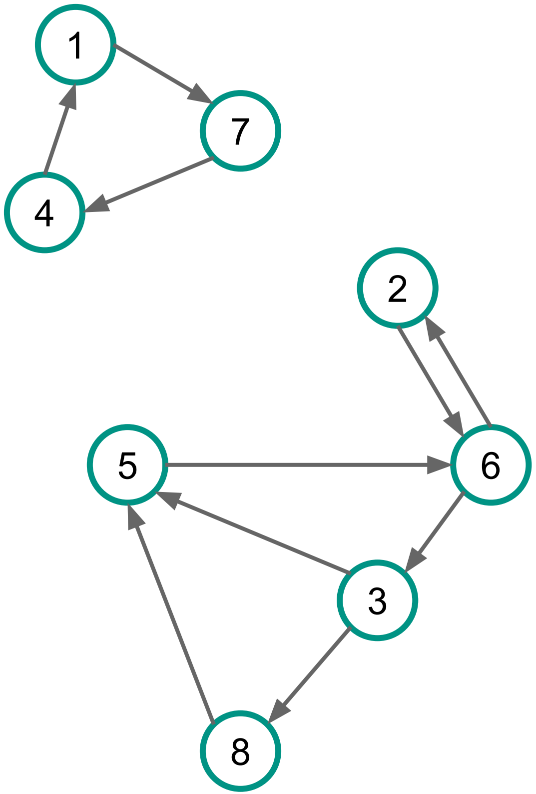


Componente tare conexe



Componente tare conexe

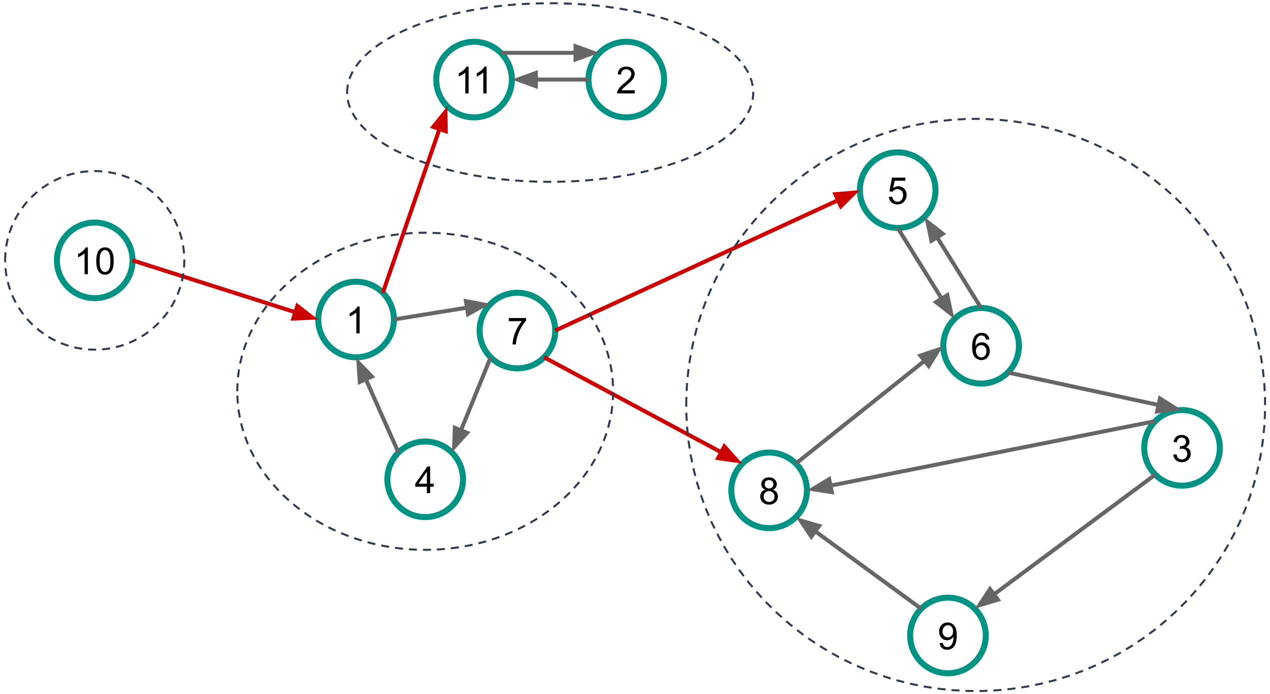




Componente tare conexe

Observație

Componentele tare conexe ale lui G = componentele tare conexe ale lui G^T

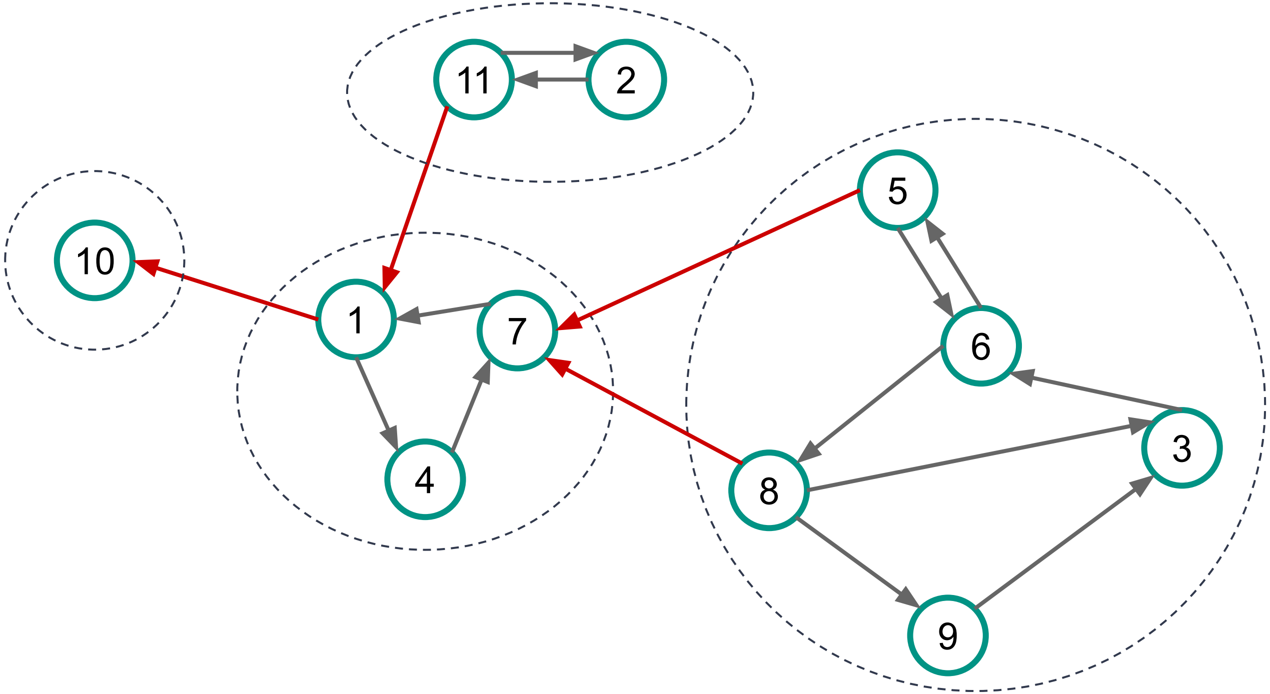


G

Componente tare conexe

Observație

Componentele tare conexe ale lui G = componentele tare conexe ale lui G^T



G^T

Algoritm

Componente tare conexe – Algoritm

Următorul algoritm de timp liniar (adică **θ(V+E)**) determină componentele tare conexe ale unui graf orientat G = (V, E), folosind două căutări în adâncime, una în G și una în G^T.

Componente-Tare-Conexe(G)

1. apelează CA(G) pentru a calcula timpii de terminare f[u] pentru fiecare vârf u
2. calculează G^T
3. apelează CA(G^T), dar, în bucla principală a lui CA, consideră vârfurile în ordinea **descrescătoare** a timpilor f[u] (calculați la linia 1)
4. afișează vârfurile fiecărui arbore din pădurea de adâncime din pasul 3 ca o componentă tare conexă separată

Componente tare conexe - Algoritm Kosaraju

Următorul algoritm de timp liniar (adică **θ(V+E)**) determină componentele tare conexe ale unui graf orientat G = (V, E), folosind două căutări în adâncime, una în G și una în G^T.

Componente-Tare-Conexe(G)

1. apelează CA(G) pentru a calcula timpii de terminare f[u] pentru fiecare vârf u
2. calculează G^T
3. apelează CA(G^T), dar, în bucla principală a lui CA, consideră vârfurile în ordinea **descrescătoare** a timpilor f[u] (calculați la linia 1)
4. afișează vârfurile fiecărui arbore din pădurea de adâncime din pasul 3 ca o componentă tare conexă separată

90-100p pe infoarena :)

Componente tare conexe - Schiță demonstrație

Lemă: Dacă două vârfuri se află în aceeași componentă tare conexă, atunci niciun drum între ele nu părăsește, vreodată, această componentă tare conexă.

Demonstrație

Fie u și v două noduri din componenta tare conexă.

Presupunem că există w în afara componentei și există drum u→v prin w.

Atunci avem drum de la u la w, dar avem și drumul w→v→u, deci și drum de la w la u.

Deci, w este în componenta tare conexă.