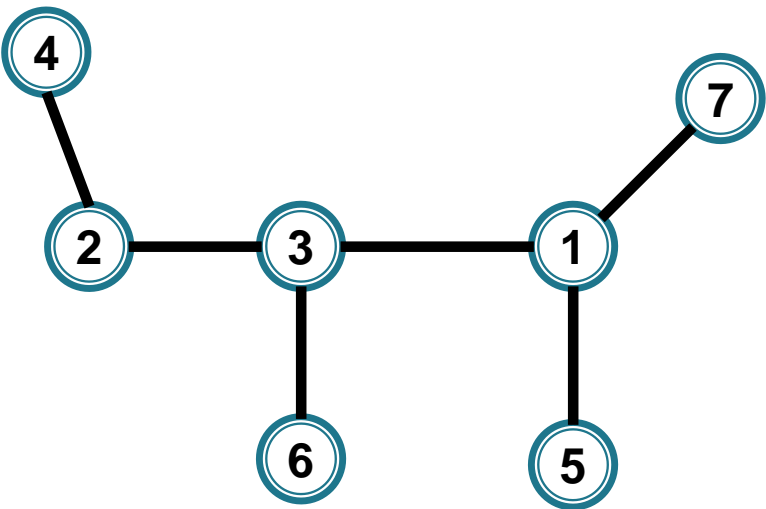
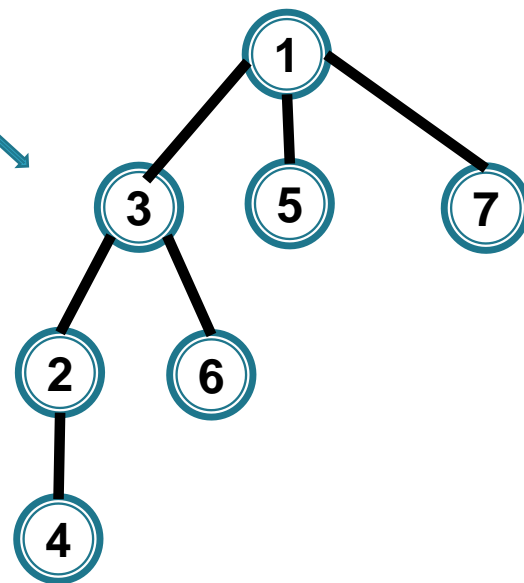
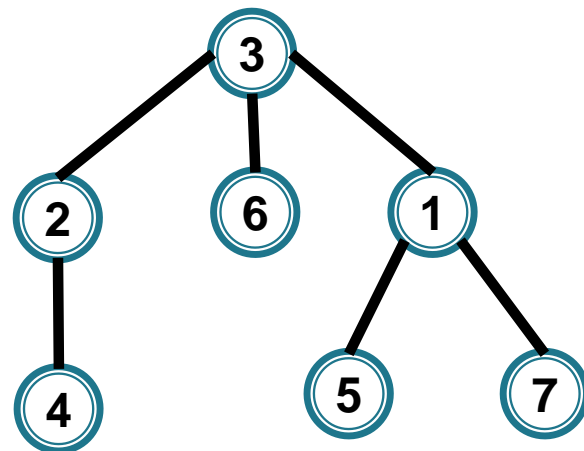
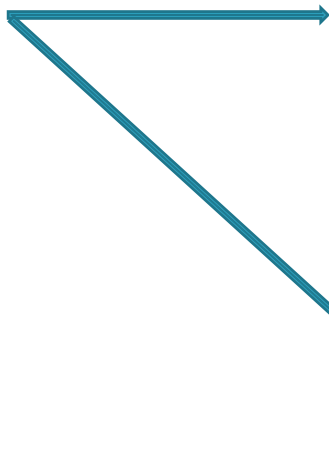


Arbori



Arbore = conex + aciclic



Arbore cu rădăcină

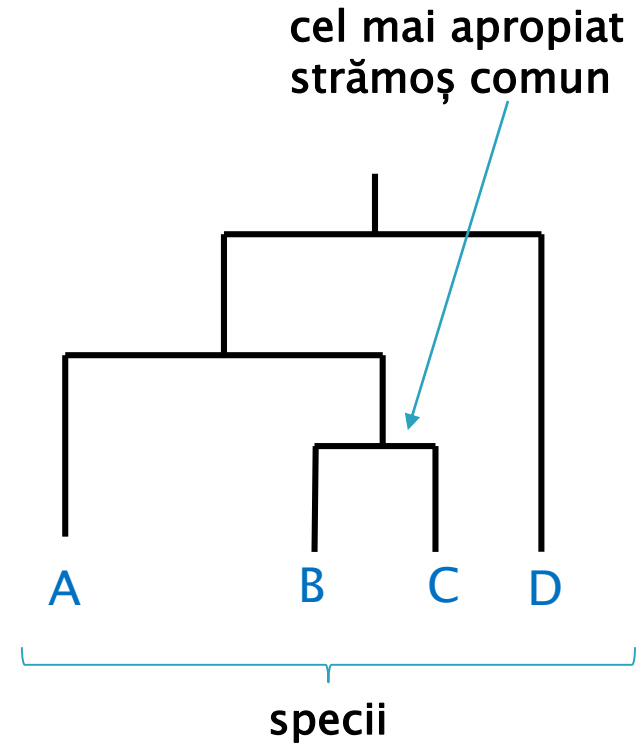
Arbori

- **Arbore** = graf neorientat conex și aciclic

Arbori

➤ **Arbore** = graf neorientat conex și aciclic

- **Arbori filogenetici** – ilustrează evoluții
- **Arbori de dependențe, de joc**
- **Probleme de rutare**
- **Arbori aleatorii**
- **Arbori economici (cu costul minim)**
- **Structuri de date...**



Arbori

Leme

1. Orice arbore T cu $n > 1$ are cel puțin două vârfuri terminale (de grad 1)

Arbori

Leme

1. Orice arbore T cu $n > 1$ are cel puțin două vârfuri terminale (de grad 1)



Fie P un lanț elementar maxim în T

Arbori

Leme

1. Orice arbore T cu $n > 1$ are cel puțin două vârfuri terminale (de grad 1)



Fie P un lanț elementar maxim în T

Extremitățile lui P sunt vârfuri terminale, altfel:

Arbori

Leme

1. Orice arbore T cu $n > 1$ are cel puțin două vârfuri terminale (de grad 1)



Fie P un lanț elementar maxim în T

Extremitățile lui P sunt vârfuri terminale, altfel:

– putem extinde lanțul cu o muchie
The diagram shows the path P from x to y (wavy line) being extended by a straight edge to a new vertex v . Vertices x , y , and v are all circled in blue.

sau

Arbori

Leme

1. Orice arbore T cu $n > 1$ are cel puțin două vârfuri terminale (de grad 1)



Fie P un lanț elementar maxim în T

Extremitățile lui P sunt vârfuri terminale, altfel:

– putem extinde lanțul cu o muchie



sau

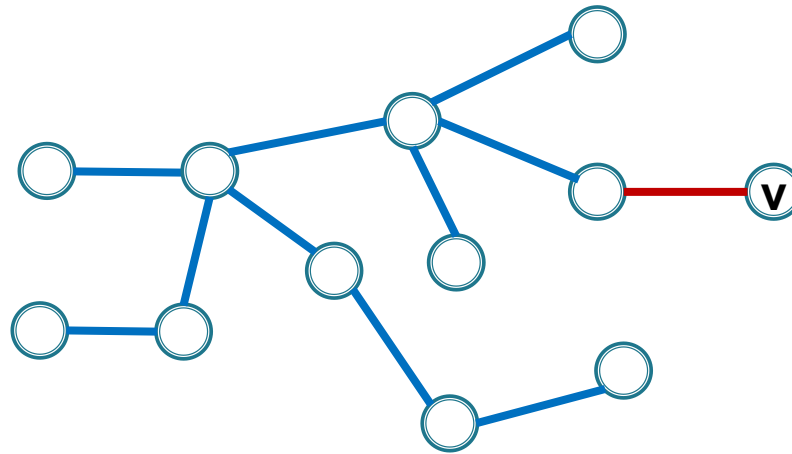
– se închide un ciclu în T



Arbori

Leme

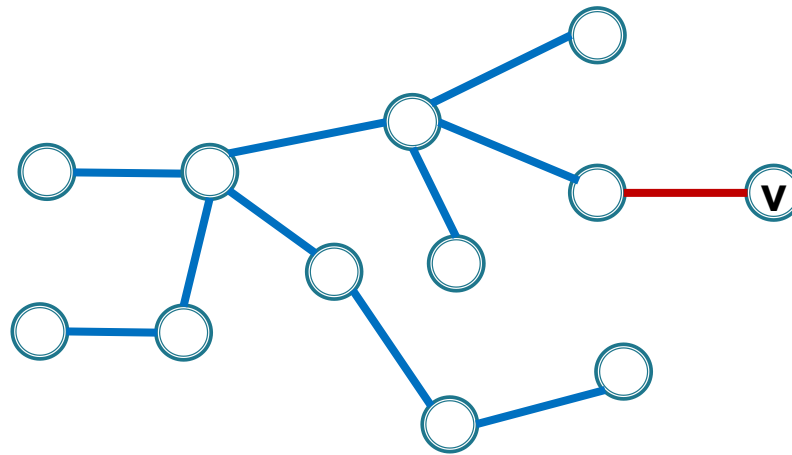
2. Fie T un arbore cu $n > 1$ vârfuri și v un vârf terminal în T .
Atunci $T - v$ este arbore.



Arbori

Leme

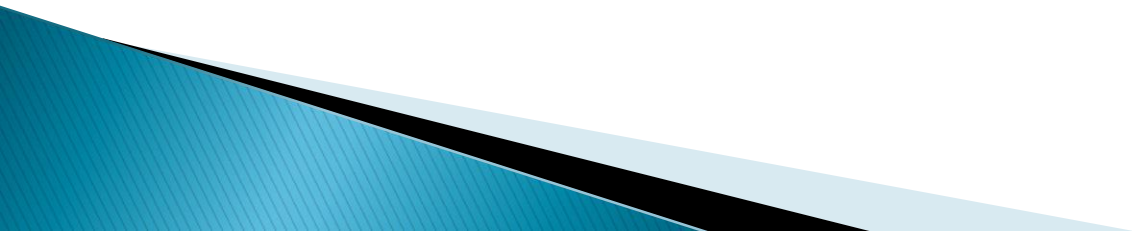
2. Fie T un arbore cu $n > 1$ vârfuri și v un vârf terminal în T .
Atunci $T - v$ este arbore.



Rezultă din definiția conexității + un vârf terminal nu poate fi vârf intern al unui lanț elementar

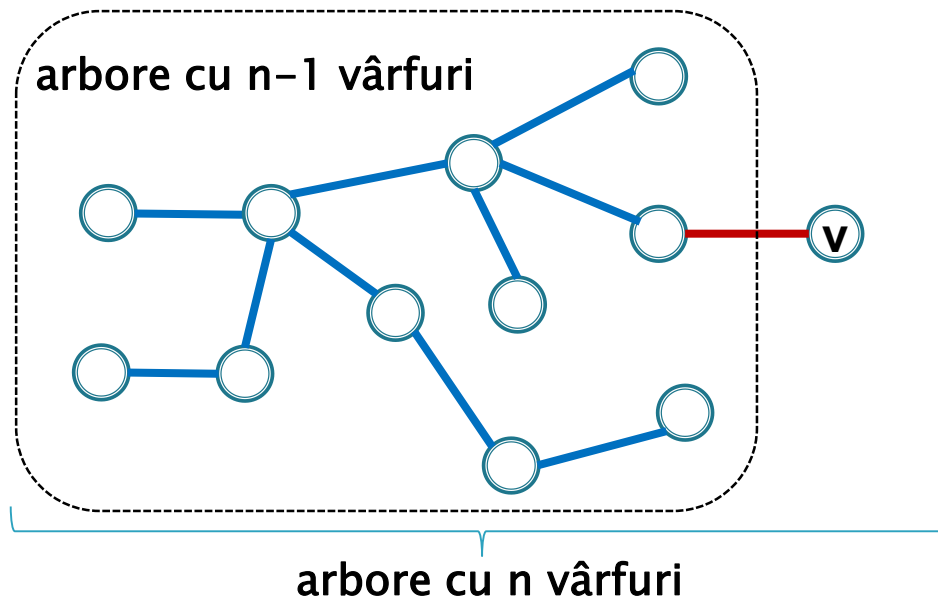
Arbori

3. Un arbore cu n vârfuri are $n-1$ muchii.



Arbori

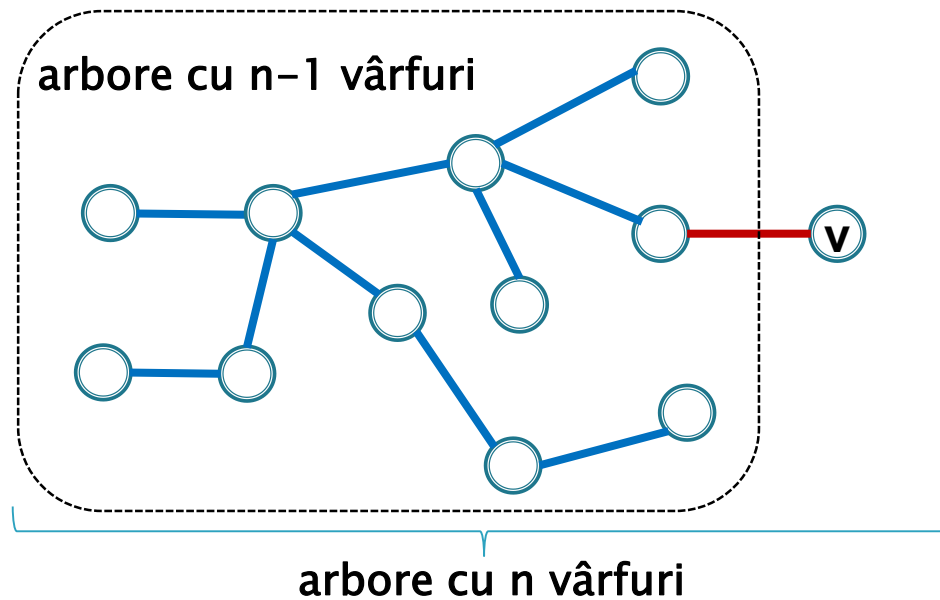
3. Un arbore cu n vârfuri are $n-1$ muchii.



Inducție după n

Arbori

3. Un arbore cu n vârfuri are $n-1$ muchii.

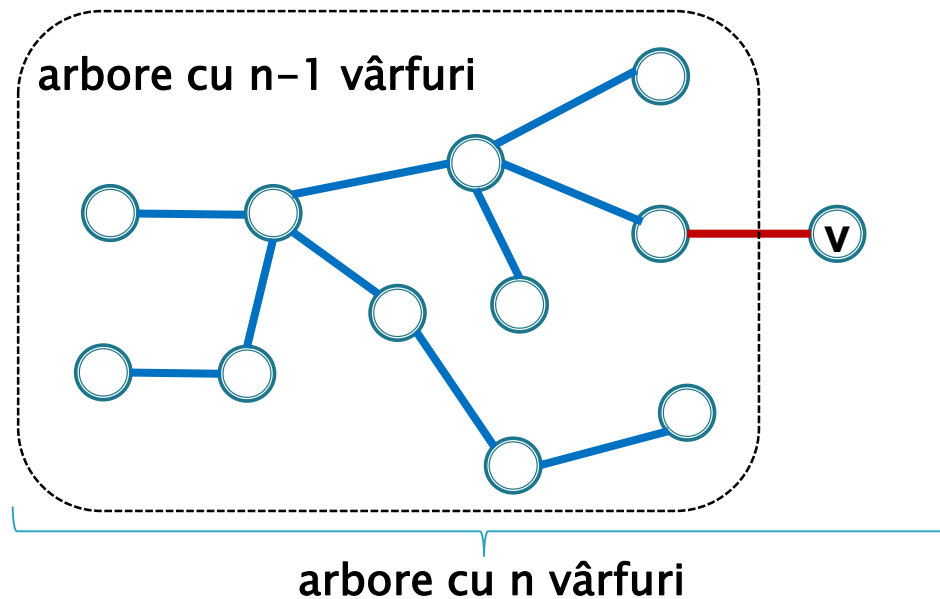


Inducție după n

Verificare - $n=1$

Arbori

3. Un arbore cu n vârfuri are $n-1$ muchii.

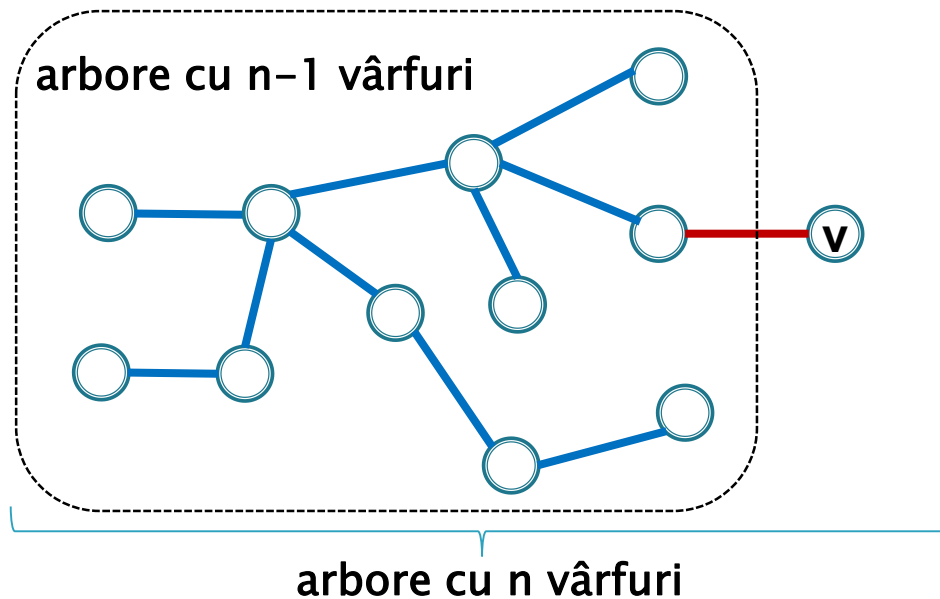


Inducție după n

- Fie T este un arbore cu n vârfuri. Fie v vârf **terminal** în T (\exists , lema 1)

Arbori

3. Un arbore cu n vârfuri are $n-1$ muchii.



Inducție după n

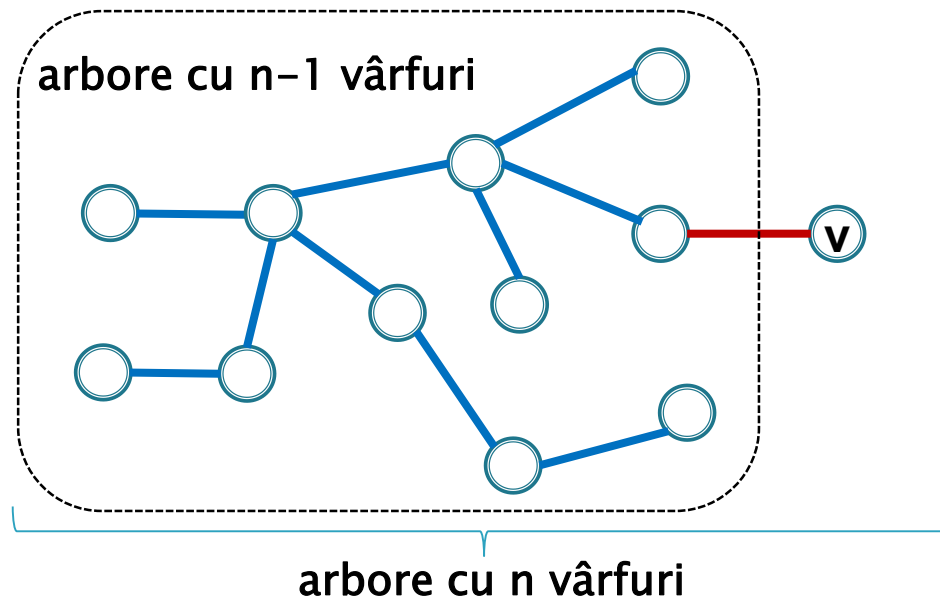
- Fie T este un arbore cu n vârfuri. Fie v vârf terminal în T (\exists , Lema 1)

$\Rightarrow T - v$ este arbore cu $n-1$ vârfuri (Lema 2)

- Aplicăm ipoteza de inducție pentru $T-v$

Arbori

3. Un arbore cu n vârfuri are $n-1$ muchii.



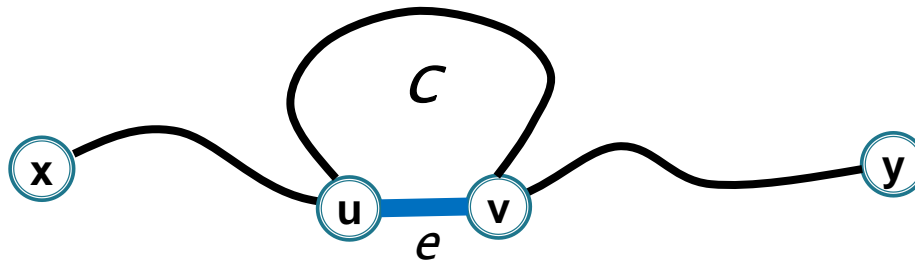
$$\Rightarrow |E(T-v)| = |V(T-v)| - 1 = (n-1) - 1 = n-2$$

$$\Rightarrow |E(T)| = |E(T-v)| + 1 = n-1$$

Arbori

Leme

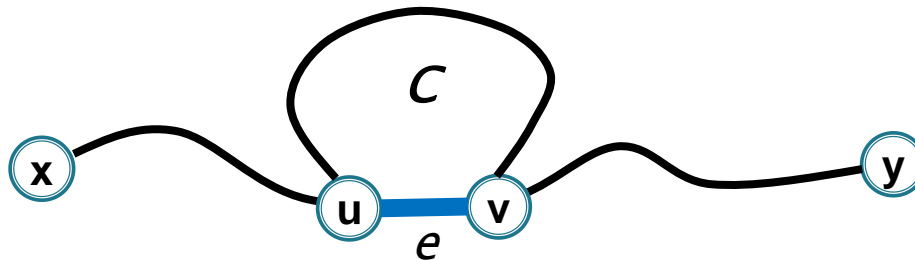
4. Fie G un graf neorientat conex și C un ciclu în G .
Fie $e \in E(C)$ o muchie din ciclul C .
Atunci $G - e$ este tot un graf conex.



Arbori

Leme

4. Fie G un graf neorientat conex și C un ciclu în G .
Fie $e \in E(C)$ o muchie din ciclul C .
Atunci $G - e$ este tot un graf conex.

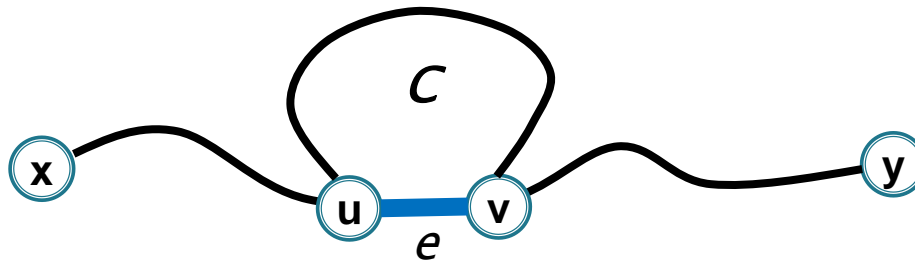


Rezultă din definiția conexității + observația:

Arbori

Leme

4. Fie G un graf neorientat conex și C un ciclu în G .
Fie $e \in E(C)$ o muchie din ciclul C .
Atunci $G-e$ este tot un graf conex.



Rezultă din definiția conexității + observația:

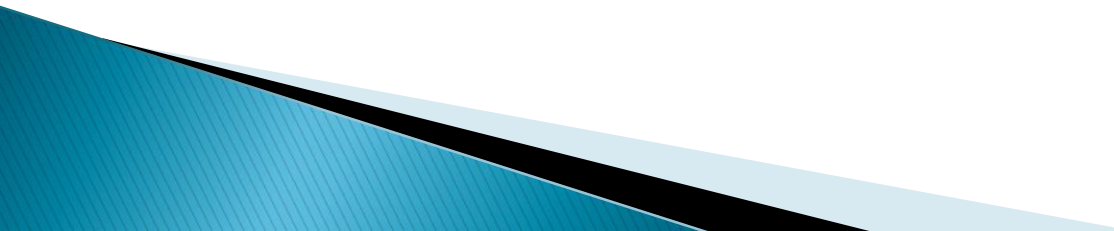
- dintr-un x - y lanț în G care conține muchia e se poate obține un x - y lanț în $G-e$ înlocuind muchia e cu lanțul $C-e$.

Arbori

Definiții echivalente

Fie T un graf neorientat cu $n > 1$ vârfuri.

Următoarele afirmații sunt echivalente.

1. T este arbore (conex și aciclic)
 - 2.
 - 3.
 - 4.
 - 5.
 - 6.
- 

Arbori

Definiții echivalente

Fie T un graf neorientat cu $n > 1$ vârfuri.

Următoarele afirmații sunt echivalente.

1. T este arbore (conex și aciclic)
2. T este conex muchie-minimal
- 3.
- 4.
- 5.
- 6.

prin eliminarea unei muchii din T se obține un graf care nu mai este conex

Arbori

Definiții echivalente

Fie T un graf neorientat cu $n > 1$ vârfuri.

Următoarele afirmații sunt echivalente.

1. T este arbore (conex și aciclic)
2. T este conex muchie-minimal
3. T este aciclic muchie-maximal
- 4.
- 5.
- 6.

prin eliminarea unei muchii din T se obține un graf care nu mai este conex

Arbori

Definiții echivalente

Fie T un graf neorientat cu $n > 1$ vârfuri.

Următoarele afirmații sunt echivalente.

1. T este arbore (conex și aciclic)
2. T este conex muchie-minimal
3. T este aciclic muchie-maximal
4. T este conex și are $n-1$ muchii
5. T este aciclic și are $n-1$ muchii
- 6.

Arbori

Definiții echivalente

Fie T un graf neorientat cu $n > 1$ vârfuri.

Următoarele afirmații sunt echivalente.

1. T este arbore (conex și aciclic)
2. T este conex muchie-minimal
3. T este aciclic muchie-maximal
4. T este conex și are $n-1$ muchii
5. T este aciclic și are $n-1$ muchii
6. Între oricare două vârfuri din T există un unic lanț elementar.

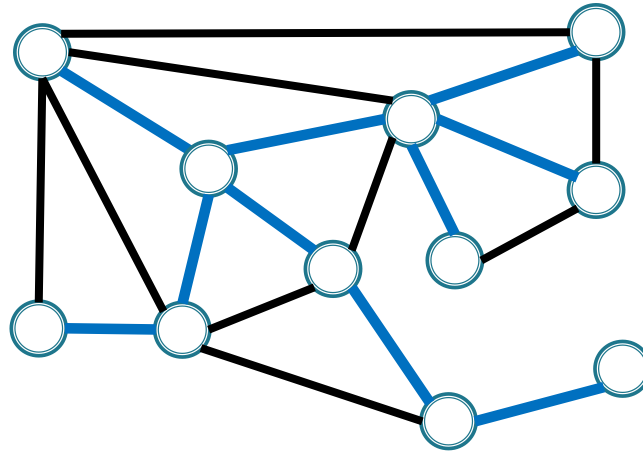
Arbori

Demonstrații echivalențe – Temă (seminar)



Arbori parțiali ai unui graf

Arbori parțiali

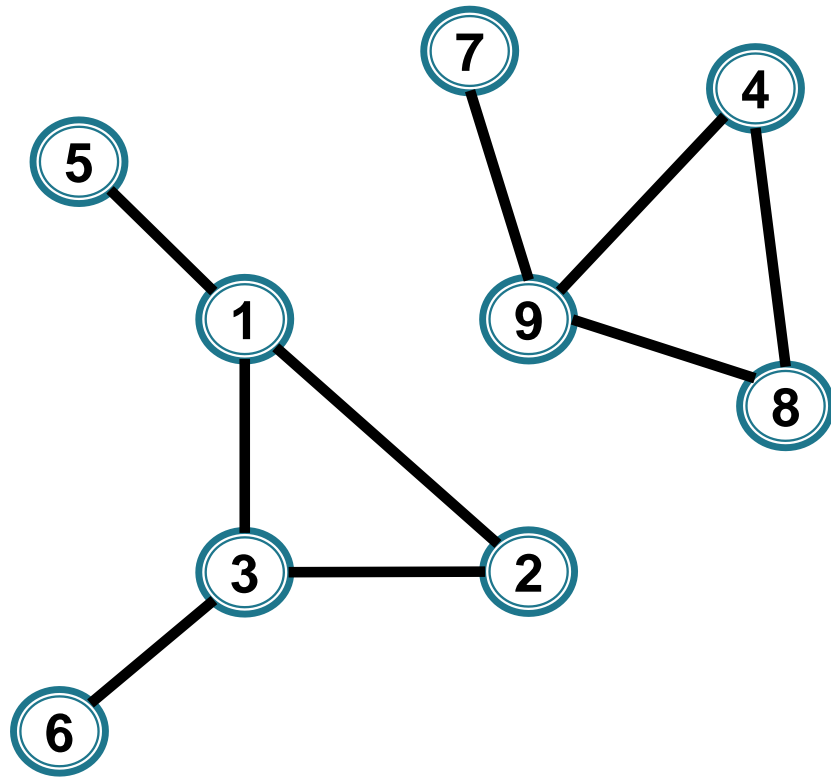


- “Scheletul” grafului
- Transmiterea de mesaje în rețea astfel încât mesajul să ajungă o singură dată în fiecare vârf
- Conectare fără redundanță + cu cost minim

Aplicații

- ▶ **Determinarea unui arbore parțial al unui graf conex**
- ▶ **Transmiterea unui mesaj în rețea:** Între participanții la un curs s-au legat relații de prietenie și comunică și în afara cursului. Profesorul vrea să transmită un mesaj participanților și știe ce relații de prietenie s-au stabilit între ei. El vrea să contacteze cât mai puțini participanți, urmând ca aceștia să transmită mesajul între ei. Ajutați-l pe profesor să decidă cui trebuie să transmită inițial mesajul și să atașeze la mesaj o listă în care să arate fiecărui participant către ce prieteni trebuie să trimită mai departe mesajul, astfel încât mesajul să ajungă la fiecare participant la curs o singură dată.

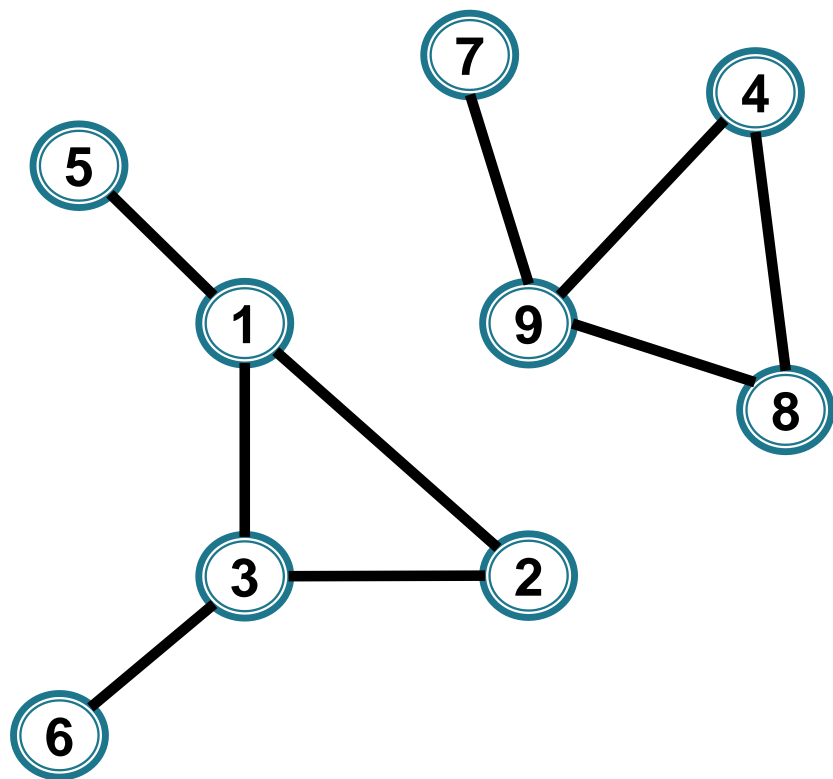
relații de prietenie/comunicare



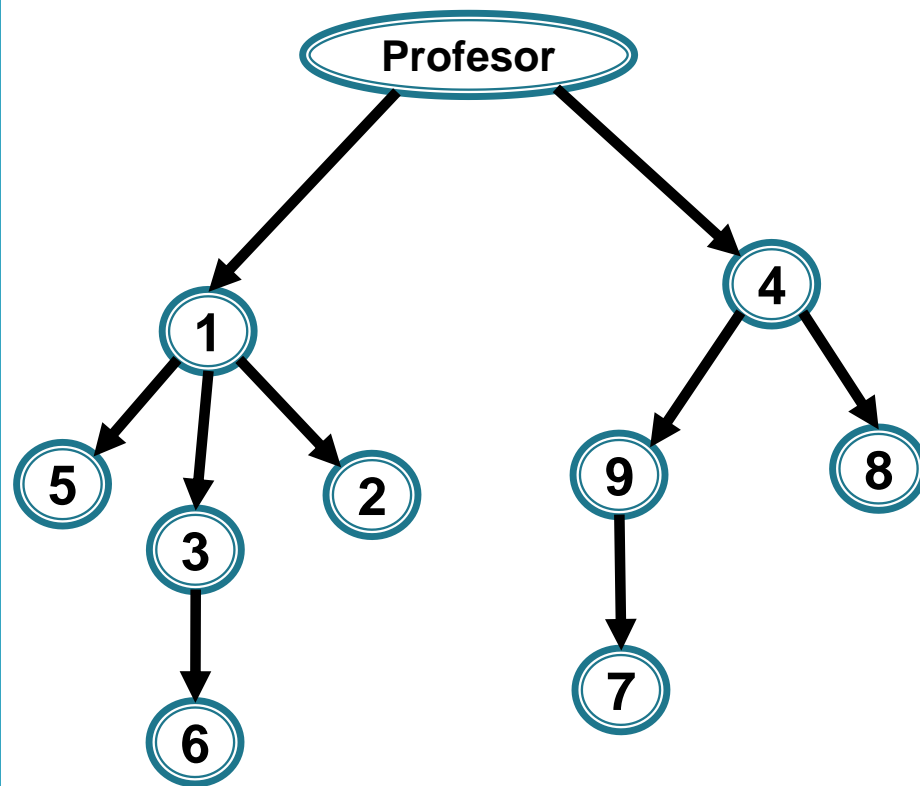
traseu de transmitere a unui mesaj



relații de prietenie/comunicare



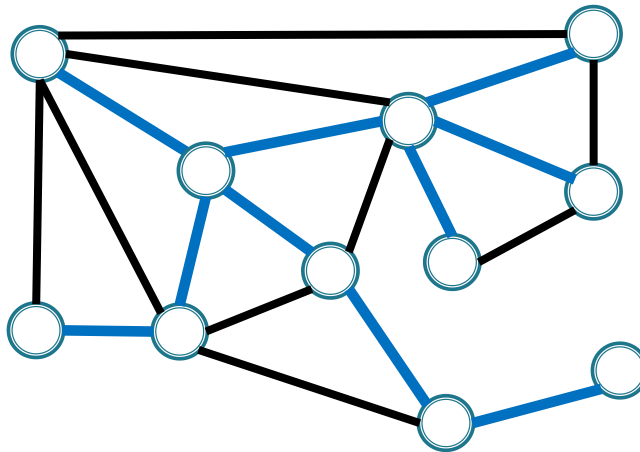
traseu de transmitere a unui mesaj

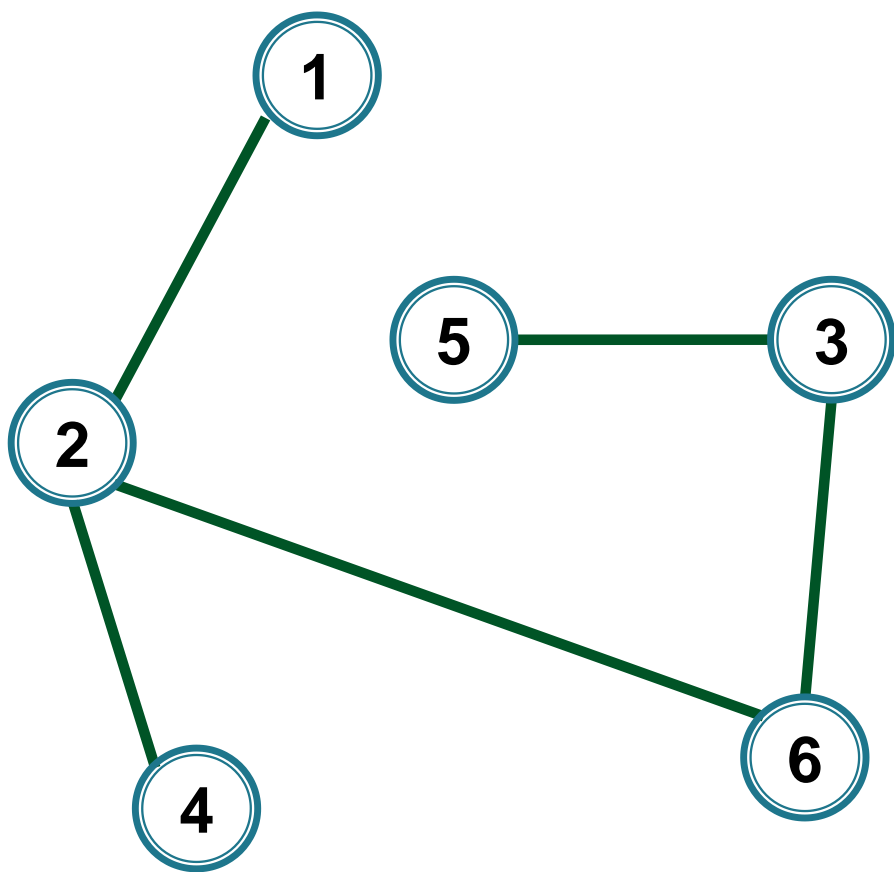
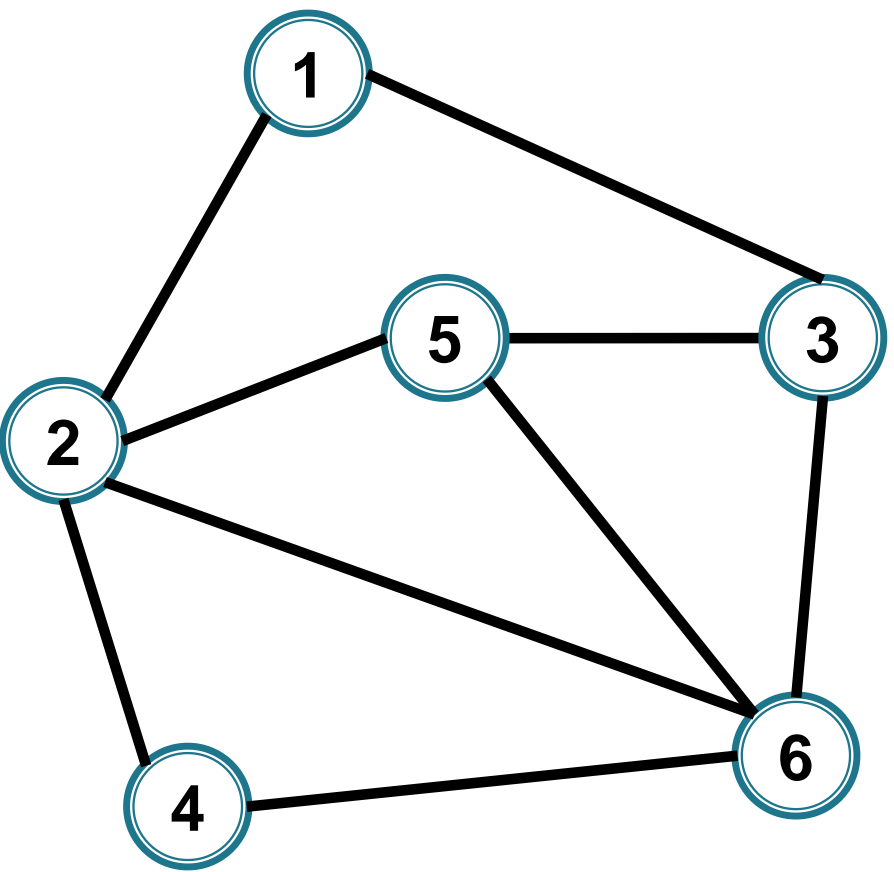


Arbori parțiali

Proprietate

Orice graf neorientat conex conține un arbore parțial
(un graf parțial care este arbore).





Arbori parțiali

Proprietate

Orice graf neorientat conex conține un arbore parțial

Demonstrație – două tipuri de algoritmi de construcție a unui arbore parțial al unui graf conex $G=(V,E)$:

Prin adăugare de muchii (bottom – up)	Prin eliminare de muchii (cut –down)

Arbori parțiali

Proprietate

Orice graf neorientat conex conține un arbore parțial

Demonstrație – două tipuri de algoritmi de construcție a unui arbore parțial al unui graf conex $G=(V,E)$:

Prin adăugare de muchii (bottom – up)	Prin eliminare de muchii (cut –down)
$T \leftarrow (V, \emptyset)$	

Arbori parțiali

Proprietate

Orice graf neorientat conex conține un arbore parțial

Demonstrație – două tipuri de algoritmi de construcție a unui arbore parțial al unui graf conex $G=(V,E)$:

Prin adăugare de muchii (bottom – up)	Prin eliminare de muchii (cut –down)
$T \leftarrow (V, \emptyset)$ cat timp T nu este conex executa <ul style="list-style-type: none">alege $e \in E(G) - E(T)$ care unește două componente conexe din T (nu formează cicluri cu muchiile din T)$E(T) \leftarrow E(T) \cup \{e\}$ returneaza T	

Arbori parțiali

Proprietate

Orice graf neorientat conex conține un arbore parțial

Demonstrație – două tipuri de algoritmi de construcție a unui arbore parțial al unui graf conex $G=(V,E)$:

Prin adăugare de muchii (bottom – up)	Prin eliminare de muchii (cut –down)
$T \leftarrow (V, \emptyset)$ cat timp T nu este conex executa <ul style="list-style-type: none">alege $e \in E(G) - E(T)$ care unește două componente conexe din T (nu formează cicluri cu muchiile din T)$E(T) \leftarrow E(T) \cup \{e\}$ returneaza T	
În final T este conex și aciclic, deci arbore	

Arbori parțiali

Proprietate

Orice graf neorientat conex conține un arbore parțial

Demonstrație – două tipuri de algoritmi de construcție a unui arbore parțial al unui graf conex $G=(V,E)$:

Prin adăugare de muchii (bottom – up)	Prin eliminare de muchii (cut –down)
$T \leftarrow (V, \emptyset)$ cat timp T nu este conex executa <ul style="list-style-type: none">alege $e \in E(G) - E(T)$ care unește două componente conexe din T (nu formează cicluri cu muchiile din T)$E(T) \leftarrow E(T) \cup \{e\}$ returneaza T	$T \leftarrow (V, E)$ cat timp T conține cicluri executa <ul style="list-style-type: none">alege $e \in E(T)$ o muchie dintr-un ciclu$E(T) \leftarrow E(T) - \{e\}$ returneaza T
În final T este conex și aciclic, deci arbore	

Arbori parțiali

Proprietate

Orice graf neorientat conex conține un arbore parțial

Demonstrație – două tipuri de algoritmi de construcție a unui arbore parțial al unui graf conex $G=(V,E)$:

Prin adăugare de muchii (bottom – up)	Prin eliminare de muchii (cut –down)
$T \leftarrow (V, \emptyset)$ cat timp T nu este conex executa <ul style="list-style-type: none">alege $e \in E(G) - E(T)$ care unește două componente conexe din T (nu formează cicluri cu muchiile din T)$E(T) \leftarrow E(T) \cup \{e\}$ returneaza T	$T \leftarrow (V, E)$ cat timp T conține cicluri executa <ul style="list-style-type: none">alege $e \in E(T)$ o muchie dintr-un ciclu$E(T) \leftarrow E(T) - \{e\}$ returneaza T
În final T este conex și aciclic, deci arbore	În final T este aciclic și conex (s-au eliminat doar muchii din ciclu), deci arbore

Arbori parțiali

Algoritmi de determinare a unui arbore parțial al unui graf conex



Algoritm de determinare a unui arbore parțial?

Complexitate algoritm?

Arbori parțiali

Algoritmi de determinare a unui arbore parțial al unui graf conex

Complexitate algoritm?



arborele asociat unei parcurgeri este arbore parțial \Rightarrow
determinăm un arbore parțial printr-o parcursere