# CONEXITATE

# 3. CONEXITATE

- Grafuri conexe
- Conexitate
- Determinarea componentelor conexe ale unui graf
- Teoreme de caracterizare
- Puncte de articulatie
- k-conexitate; 2-conexitate

**Algoritmi** (descriere+pseudocod+ex.)

det. componenta **tare conexa** in graf

- DepthFirstSearch (DFS) cu marcaje de timp

det. componente tare conexe (1-4)

- DepthFirstSearch cu marcaje de timp si leader

det. componente **conexe** ale unui graf

- BFS si DFS pt. det. componente conexe + modicari

## Definitii:

- graf tare conex
- componenta tare conexa in graf

(DFS marcaje de timp, marcaje timp, lider)

- graf conex
- componenta conexa a unui graf
- componente conexe (clase de exchivalenta)
- nr. componentelor conexe
- nod punct de articulatie
- nod punct de articulatie <=> eliminarea sa nr. componentelor conexe creste
- (muchie) istm <=>graful partial nu e conex
- (muchie) istm <=>eliminarea sa nr. componentelor conexe creste
- k-conex relativ la noduri
- conexitatea relativ la noduri a grafului
- k-conex operatii: stergere, adaugare, diviz.
- Subdiviziune graf

**Teoreme de caracterizare**

- gradele nodurilor (conditie) => graf conex (+demonstratie)
- numarul de muchii (conditie)=> graf conex (+demonstratie)
- T. Bondy (grade (conditie) =>graf conex)
- graf eulerian fara noduri izolate <=> (gradul int. nod==gr.ext)
- graf eulerian fara noduri izolate <=> gr.noduri pare
- x punct articulatie <=> exista 2 noduri (y,z) a.i. orice lant (y,z) contine nodul x
- T. Bondy (graf k-conex)
- T. Menger (k-conex <=> k lanturi mutual disjuncte)
- T. Menger (2-conex <=>ciclu elementar)
- T. 2-conexitate (construit triunghi (K3) diviz. + adaug. muchii)

## Proprietati. Teoreme

-Graf tare conex -> circuit

-R Relatia binara (multimea nodurilor grafului neorientat) ⇔ exista lant

-R este o relatie de echivalenta (RST)

-graf conex ⇔ are o singura componenta conexa

# Grafuri conexe

## Problemă

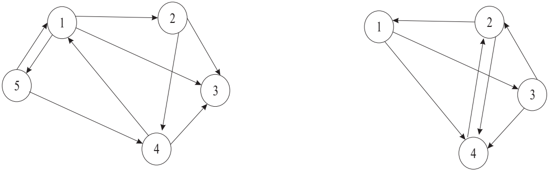
Într-o organizație sunt persoane care se cunosc sau nu (eventual reciproc). Conducerea organizației are o informație care trebuie să ajungă la toți membrii organizației. O persoană care are informația o poate transmite tuturor cunoscuților săi. Care este numărul minim de persoane care trebuie să dețină informația astfel încât aceasta să ajungă la toată lumea?

- se poate modela ca problemă de teoria grafurilor: $G = (X, U)$ cu $X$ mulțimea nodurilor, câte un nod pentru fiecare persoană, și $(x, y) \in U$ dacă și numai dacă $x$ și $y$ se cunosc.

# Conexitate

## Definiție

Un graf orientat $G = (X, U)$ se numește **graf tare conex** dacă și numai dacă între oricare două noduri există drum.




a) b)

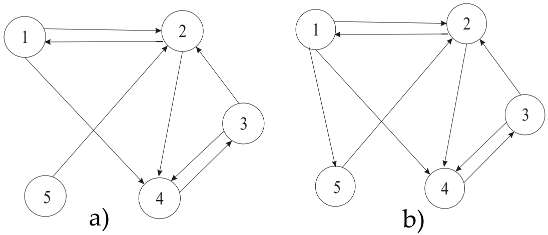
a) nu tare conex, b) tare conex

## Teoremă

Într-un graf $G = (X, U)$ tare conex oricare două noduri se găsesc pe un circuit.

# Conexitate

- se numeste **componenta** tare **conexa** in graful $G=(X, U)$ un subgraf tare conex maximal $C=G(Y)$ al grafului $G$, adica $Y \subset X$ si intre oricare doua noduri ale lui $Y$ exista drum si ın plus, orice subgraf $G(Y_1)$ cu $Y \subset Y_1$ nu este tare conex




- graful a) nu este tare conex și are două componente conexe și anume $G(\{5\})$ respectiv $G(\{1, 2, 3, 4\})$.

# Conexitate

- pentru determinarea componentelor tare conexe ale unui graf orientat $G$ se folosește algoritmul $DFS$ cu timpi de marcare
- acest algoritm se aplică atât grafului inițial $G$ cât și grafului $G^{rev}$ obținut prin inversarea sensurilor arcelor grafului inițial
- fiecare vârf $x$ are două marcaje de timp:

- $ti[x]$ momentul când $x$ este atins prima oară (gri, moment inițial)
- $tf[x]$ momentul când căutarea termină de examinat lista de succesorilor lui $x$ ( se colorează $x$ negru, moment final)
- marcajele de timp sunt valori întregi între 0 și $2 \cdot |X| - 1$ (eveniment de descoperire și de terminare pentru fiecare $x \in X$)
- $\forall\ x \in X,\ ti[x] < tf[x]$
- $x$ este alb înainte de momentul $ti[x]$, gri între momentul $ti[x]$ și momentul $tf[x]$ și negru după aceea

vf. alb: initial: inainte de timp initial ti
vf. gri intre timp initial ti si timp final tf
vf negru: la final

DepthFirstSearch cu marcaje de timp

Pentru $x \in X$ execută
  $color[x] = alb$;   vf. alb initializare: vf. nevizitat
Sfârșit pentru;
timp=0;
Pentru $x \in X$ execută
  Dacă ($color[x] = alb$) atunci   daca este nevizitat (alb) se aplica recursiv DFS
    $DFS(x)$;
  Sfârșit dacă;
Sfârșit pentru;
Sfârșit algoritm;

## Conexitate

DFS recursiv

$DFS(x)$

$color[x] = gri;$ — vf. gri intre timp initial ti si timp final tf

$ti[x] = timp;$ — timpul initial al vf.x = timpul curent

$timp = timp + 1;$ — timpul se incrementeaza

Pentru fiecare $y \in X$ cu $(x, y) \in U$ execută — fiecare y vecin a lui x

Dacă $color[y] = alb$ atunci

$DFS(y);$ — daca y este nevizitat (alb) se aplica recursiv DFS

Sfârșit dacă;

Sfârșit pentru;

$culoare[x] = negru;$ — vf. negru daca s-a inchieiat cu toti vecnii lui x

$tf[x] = timp;$ — timp final vf x= timp curent

$timp = timp + 1;$ — momentul de timp se incrementeaza

Sfârșit DFS;

G rev= reverse orientare arce
(arc orientat x->y devine orientat y->x)

Determinarea componentelor tare conexe

1. Construiește graful $G^{rev}$.
2. Execută DFS în graful $G^{rev}$ pentru a calcula timpii finali $tf[x], x \in X$.
3. Execută DFS pentru graful $G$, considerând nodurile în ordinea descrescătoare a timpilor $tf$ calculați la pasul 2 pentru a atașa fiecărui nod $x \in X$ un nod leader.
4. Componentele conexe sunt formate din acele noduri care au un leader comun.

# Conexitate

ex. (5,2) orientat reverse (2,5)

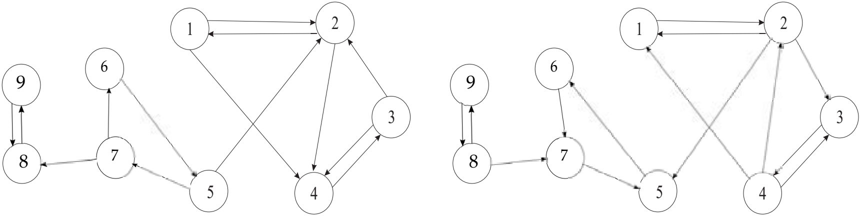



Figure: Grafurile $G$ și $G^{rev}$

1. construire graf reverse orientat
2. DFS G rev, calcul timpi finali ai fiecarui vf.
3. Descrescator noduri in ordinea timpilor finali
- DFS pe G
- atasare fiecare nod, un leader
4. Componente conexe=noduri cu leader comun

DepthFirstSearch cu marcaje de timp și leader

Pentru $x \in X$ execută initializari

$color[x] = alb;$

$leader[x] = 0;$

Sfârșit pentru;

timp=0;

Pentru $x \in X$ execută

Dacă $(color[x] = alb)$ atunci

$nod\_leader = x;$

$DFS(x);$

Sfârșit dacă;

Sfârșit pentru;

fiecare nod
daca este nevizitat (alb)
devine leader
se aplica DFS(x)

# Conexitate

DFS recursiv

$DFS(x)$

$color[x] = gri;$

$ti[x] = timp;$

$timp = timp + 1;$

Pentru fiecare $y \in X$ cu $(x, y) \in U$ execută

Dacă $color[y] = alb$ atunci

$leader[y] = nod\_leader;$

$DFS(y);$

Sfârșit dacă;

Sfârșit pentru;

$culoare[x] = negru;$

$tf[x] = timp;$

$timp = timp + 1;$

gri=intre ti si tf
vf x timp initial=timp curent
timp incrementat

y fiecare vecin a lui x
daca y nevizitat (alb)
leader(y)=nod leader
aplica DFS pe acest nod y

vf. negru daca s-a inchieiat cu toti vecnii lui x
timp final vf x= timp curent

momentul de timp se incrementeaza

- aplicând *DFS* grafului $G^{rev}$ se obțin timpii inițiali respectiv finali pentru fiecare nod

- cercetarea nodurilor în $G^{rev}$ începe cu nodul 1

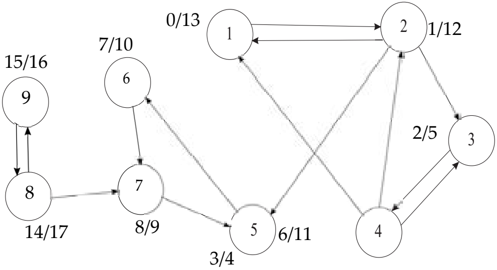
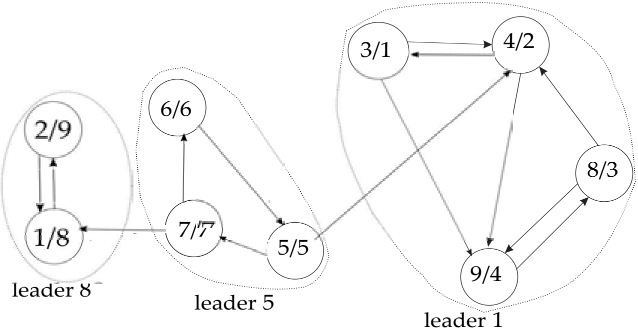



Figure: $G^{rev}$ cu timpi inițiali și finali

# Conexitate

- se aplică *DFS* pentru *G* cercetând nodurile în ordinea descrescătoare a timpilor finali: ordinea $8, 9, 1, 2, 5, 7, 6, 3, 4$

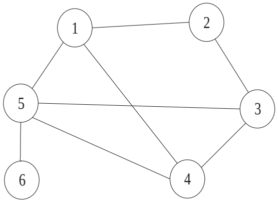


# Conexitate

**Definiție**

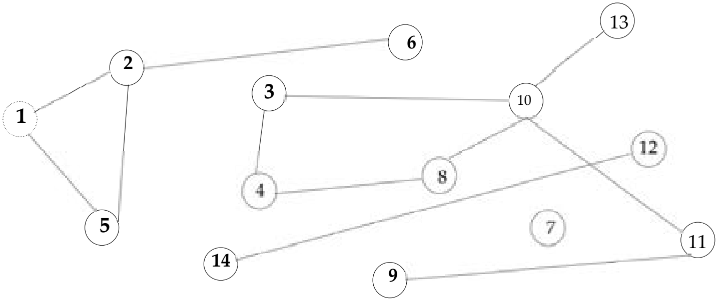
Un graf neorientat $G = (X, U)$ se numește **graf conex** dacă și numai dacă între oricare două noduri există lanț.

$G = (X, U)$




este graf conex.

- se numește componentă conexă a unui graf o mulțime maximală de noduri $Y \subset X$ cu proprietatea că subgraful $G(Y)$ al lui $G$ este graf conex




$C_1 = \{1, 2, 5, 6\},\ C_2 = \{3, 4, 8, 9, 10, 11, 13\},\ C_3 = \{12, 14\},\ C_4 = \{7\}$
componente conexe

# Conexitate

- pe mulțimea nodurilor grafului neorientat $G = (X, U)$ se definește relația binară $R \subset X \times X$ astfel: $xRy$ dacă și numai dacă există lanț de la $x$ la $y$

- relația $R$ este o relație de echivalență

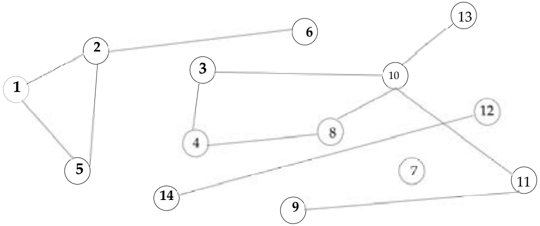
- $R$ este reflexivă, adică $xRx, \forall x \in X$.

- dacă $\exists\ \mu = \{x = x_{i_1}, x_{i_2}, ..., x_{i_k} = y\}$ de la $x \in X$ la $y \in X$, atunci $\mu' = \{y = x_{i_k}, ..., x_{i_2}, x_{i_1} = x\}$ lanț de la $y$ la $x \Rightarrow R$ simetrică

- $x, y, z \in X$ și $\mu_{xy} = \{x = x_{i_1}, x_{i_2}, ..., x_{i_k} = y\}$ lanț de la $x$ la $y$, respectiv $\mu_{yz} = \{y = y_{j_1}, y_{j_2}, ..., y_{j_l} = z\}$ lanț de la $y$ la $z \Rightarrow$ $\mu_{xz} = \{x = x_{i_1}, x_{i_2}, ..., x_{i_k} = y = y_{j_1}, y_{j_2}, ..., y_{j_l} = z\} = \mu_{xy} \cdot \mu_{yz}$ lanț de la $x$ la $z \Rightarrow R$ este tranzitivă

- clasele de echivalență determinate de relația de echivalență $R$ pe mulțimea nodurilor grafului neorientat $G = (X, U)$ se numesc **componente conexe** ale grafului $G$

- numărul claselor de echivalență determinate de relația $R$ pe mulțimea $X$ a nodurilor unui graf este **numărul componentelor conexe** ale grafului $G$ și se notează cu $p$




$C_1 = \{1, 2, 5, 6\} = [1]$, $C_2 = \{3, 4, 8, 9, 10, 11, 13\} = [3]$,
$C_3 = \{12, 14\} = [12]$, $C_4 = \{7\} = [7]$, $p = 4$

- graful neorientat $G = (X, U)$ se numește graf conex dacă și numai dacă are o singură componentă conexă, adică $p = 1$

# Determinarea componentelor conexe ale unui graf

- se pune problema determinării componentelor conexe ale unui graf

- se pot folosi algoritmi de tipul celor de parcurgere, *BFS* și *DFS* pentru determinarea componentelor conexe, cu mici modificări:

  - se începe prelucrarea cu un nod $s$ al grafului

  - se marchează cu un același număr toate nodurile accesibile din $s$ folosind *BFS* sau *DFS* (acestea vor forma componenta conexă ce conține nodul $s$)

  - se continuă procedeul până nu mai sunt noduri nemarcate în graf

# Determinarea componentelor conexe ale unui graf cu BFS

Pentru $x \in X$ execută $compc[x] = 0$;
$p = 0$; // numărul curent de componente conexe
Câttimp ( $\exists\ s \in X$ cu $compc[s] = 0$) execută
    $p = p + 1$;            // o nouă componentă conexă
// se aplică *BFS* cu nod de pornire $s$ și se marchează nodurile din componenta $p$
    Adaugă $s$ într-o coadă $Q$ inițial vidă;
    $compc[s] = p$;    // nodul $s$ se marchează vizitat în componenta conexă $p$
    Cât timp ($Q \neq \emptyset$ ) execută    // mai există noduri nemarcate
        $x = cap(Q)$;
        Pentru toți $y \in X$ cu $(x, y) \in U$ execută
            Dacă ($compc[y] = 0$) atunci
                    $compc[y] = compc[x]$;    // $y$ în aceeași componentă cu $x$
                    Adaugă $y$ în coada $Q$;    // $y$ ultimul nod marcat
            Sfârșit dacă;
        Sfârșit pentru;
        Se scoate $x$ din coada $Q$;
    Sfârșit cât timp;
Sfârșit cât timp;

# Determinarea componentelor conexe ale unui graf

Pentru $x \in X$ execută $compc[x] = 0$;
$p = 0$; // numărul curent de componente conexe
Câttimp (există $x \in X$ cu $compc[x] = 0$) execută //există încă noduri nemarcate (nevizitate)
    $p = p + 1$; //se construiește o nouă componentă conexă $p$
    $compc[x] = p$; // se adaugă nodul la componenta respectivă
    Câttimp (există $y \in X$ nemarcat adiacent cu $z \in X$ marcat) execută
        $compc[y] = compc[z]$;
    // vecinii nemarcați ai nodurilor marcate deja se pun în aceeași componentă conexă
Sfârșit câttimp;

# Determinarea componentelor conexe ale unui graf

- cercetarea muchiilor grafului
- inițial p=0, nodurile grafului sunt nemarcate
- la citirea unei muchii $u \in U$ ne putem afla în una din următoarele situații:

- ambele capete ale sale sunt nemarcate - apare o componentă conexă nouă ce va conține cele două noduri, $p = p + 1$, cele două noduri se marchează cu $p$
- un capăt este marcat și celălalt nemarcat - nodul nemarcat se marchează cu aceeași valoare cu cel marcat
- cele două extremități ale muchiei sunt marcate cu valori diferite - cele două componente conexe se vor reuni și p=p-1; este posibil să fie nevoie de o renumerotare a componentelor conexe

- marcarea nodurilor izolate

# Teoreme de caracterizare

**Teoremă**

Fie $G = (X, U)$ un graf cu $n = |X|$ noduri. Dacă gradele nodurilor îndeplinesc condiția $g(x) \geq n/2$, $\forall x \in X$, atunci graful $G$ este conex.

Demonstrație:

- presupunem că $G$ nu conex și fie $G_1 = (X_1, U_1)$, $G_2 = (X_2, U_2)$ două componente conexe ale sale
- fie $x \in X_1$, $y \in X_2 \Rightarrow$ între cele două noduri nu există lanț
- $y$ nu are sigur ca noduri adiacente pe $x$ și nici pe vreunul dintre cele $g(x)$ noduri adiacente cu $x$
- atunci

$$g(y) \leq n - 1 - 1 - g(x) \leq n - 2 - n/2 = n/2 - 2 < n/2,$$

contradicție cu faptul că $g(y) \geq n/2$

- graful este deci conex.

# Teoreme de caracterizare

**Teoremă**

Fie $G = (X, U)$ un graf cu $n = |X|$ noduri și $m = |U|$ muchii. Dacă numărul de muchii $m > \frac{(n-1)(n-2)}{2}$ atunci graful $G$ este conex.

Demonstrație:

- se calculează numărul maxim de muchii într-un graf neconex cu $n$ noduri

1. dacă într-un graf cu nr. fixat noduri numărul componentelor conexe crește atunci numărul maxim de muchii scade $\Rightarrow$ numărul maxim de muchii se obține pentru 2 componente conexe

2. dacă un graf are 2 componente conexe, numărul maxim de muchii se obține când una dintre componente este un nod izolat

# Teoreme de caracterizare

**Teoremă Bondy**

Fie un graf $G$ cu $n = |X| \geq 2$ noduri și fie $d_1 \leq ... \leq d_2 \leq d_n$ gradele celor $n$ noduri. Dacă $d_j \leq j - 1$ implică $d_n \geq n - j$ pentru orice $1 \leq j \leq [n/2]$ atunci graful $G$ este conex.

## Teoreme de caracterizare

**Teoremă**

Fie $G = (X, U)$ graf orientat cu $|X| = n > 1$ noduri. Atunci $G$ este graf eulerian fără noduri izolate dacă și numai dacă gradul interior al fiecărui nod este egal cu cel exterior, $g^{-}(x) = g^{+}(x) > 0, \ \forall\, x \in X$.

**Teoremă**

Fie $G = (X, U)$ graf neorientat cu $|X| = n > 1$ noduri. Atunci $G$ este graf eulerian fără noduri izolate dacă și numai dacă gradele nodurilor sunt pare, $g(x) = par > 0, \ \forall\, x \in X$.

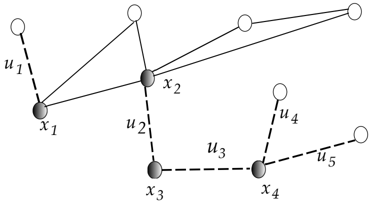
# Puncte de articulație

- un nod $x$ al unui graf conex $G = (X, U)$ se numește **punct de articulație** al grafului $G$ dacă și numai dacă subgraful $G(X \setminus \{x\})$ nu este conex

- un nod al unui graf este punct de articulație dacă și numai dacă prin eliminarea sa numărul componentelor conexe ale grafului crește

- muchie $u$ a unui graf conex $G = (X, U)$ se numește **istm** al grafului $G$ dacă și numai dacă graful parțial $G' = (X, U \setminus \{u\})$ nu este conex

- muchie $u$ a unui graf $G$ se numește istm al grafului dacă și numai dacă prin eliminarea sa din graf numărul componentelor conexe crește




- puncte de articulație $x_1, x_2, x_3, x_4$
- muchii istm $u_1, u_2, u_3, u_4, u_5$

**Teoremă**

Un nod $x$ al grafului conex $G = (X, U)$ este punct de articulație dacă și numai dacă există două noduri $y,\ z \in X \setminus \{x\}$ astfel încât orice lanț între $y$ și $z$ conține nodul $x$.

# $k$-conexitate

**Definiție**

Un graf $G = (X, U)$ se numește **$k$-conex relativ la noduri** dacă și numai dacă are cel puțin $k + 1$ noduri și prin eliminarea oricăror $k - 1$ noduri rămâne conex.

**Definiție**

Valoarea maximă a lui $k$ pentru care graful $G = (X, U)$ este $k$-conex relativ la noduri se numește **conexitatea relativ la noduri** a grafului $G$.

# $k$-conexitate

**Definiție**

Un graf $G = (X, U)$ se numește ***k*-conex relativ la muchii** dacă și numai dacă are cel puțin $k + 1$ muchii și prin eliminarea oricăror $k - 1$ muchii rămâne conex.

**Definiție**

Valoarea maximă a lui $k$ pentru care graful $G = (X, U)$ este $k$-conex relativ la muchii se numește **conexitatea relativ la muchii** a grafului $G$.

# $k$-conexitate

**Teoremă Bondy**

Fie un graf $G$ cu $n = |X| \geq 2$ noduri, $d_1 \leq ... \leq d_2 \leq d_n$ gradele celor $n$ noduri și $1 \leq j \leq n-1$. Dacă $d_j \leq j + k - 2$ implică $d_{n-k+1} \geq n - j$ pentru orice $1 \leq j \leq [(n-k+1)/2]$ atunci graful $G$ este $k$-conex.

- dacă graful este o reprezentare a unei rețele pubice de transport, de telefonie,... conexitatea sa definește numărul de componente/legături care se pot defecta fără a afecta funcționarea rețelei

# $k$-conexitate

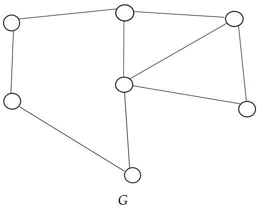
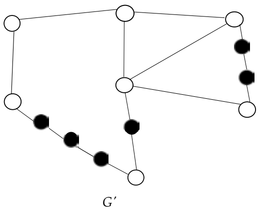
Fie $G = (X, U)$ cu $X = \{x_1, ..., x_n\}$ și $U = \{u_1, u_2, ..., u_m\}$

- se definesc asupra grafului $G$ următoarle operații:

- ștergere a unui nod: se elimină un nod $x \in X$ din graf; se obține subgraful $G(X \setminus \{x\})$, $G - x$;
- ștergere a unei muchii: se elimină o muchie $u \in U$ din graf; se obține graful parțial cu muchiile $U \setminus \{u\}$, $G(U \setminus \{u\})$, $G - u$;
- adăugare a unei muchii: se adaugă o muchie între noduri neadiacente în $G$, $u' \notin U$; graful obținut este $G' = (X, U \cup \{u'\})$, $G + u'$;
- divizare a unei muchii: se adaugă un nod $z \notin X$ pe o muchie $u = (x, y) \in U$; se obține $G' = (X \cup \{z\}, U \setminus \{(x, y)\} \cup \{(x, z), (z, y)\})$, $G\%u$.

# $k$-conexitate

- un graf $G'$ se numește subdiviziune a grafului $G$ dacă se obține din $G$ prin aplicarea de operații de divizare

# $k$-conexitate

**Teoremă (Menger)**

Un graf $G = (X, U)$ este $k$-conex dacă și numai dacă oricare ar fi $x, \ y \in X$ două noduri ale sale, există $k$ lanțuri mutual disjuncte între nodurile $x$ și $y$ (nodurile $x$ și $y$ sunt singurele noduri comune).

# 2-conexitate

**Definiție**

Un graf $G = (X, U)$ se numește **2-conex** dacă și numai dacă are cel puțin 3 noduri și prin eliminarea oricărui nod rămâne conex.

**Teoremă (Menger)**

Un graf $G = (X, U)$ este 2-conex dacă și numai dacă oricare două noduri $x,\ y \in X$ se află pe un ciclu elementar.

# 2-conexitate

## Teoremă

Un graf $G$ este 2-conex dacă și numai dacă poate fi construit dintr-un triunghi (graful $K_3$) prin operații de divizare și adăugare de muchii.