

A gráfelmélet elemei

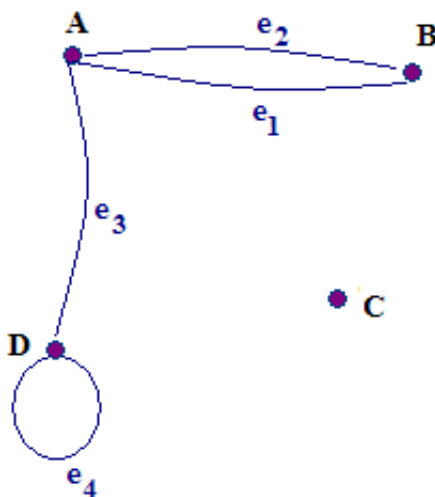
A gráfelmélet a tiszta és az alkalmazott matematikának különösen izgalmas fejezete, mert univerzális jelleg: alkalmazható az egzakt tudományokban, a társadalomtudományokban, a művészetben és a legutóbbi idők óta a bölcsészetben is. Ebben a fejezetben a gráfelmélet alapvető fogalomrendszerébe nyerünk betekintést. Néhány alkalmazással is megismerkedünk.

▼ Alapfogalmak

7.1.1 Definíció

A $G = (V, E)$ gráf a gráfpontok (csúcspontok, szögpontok) és élek olyan rendszere, amelyben minden él egy vagy két gráfcsúcsra illeszkedik.

Egy G gráf tehát két különböző típusú elembl, csúcsokból és élekből áll. Minden élnek két végpontja van, melyek a csúcsok halmazából kerülnek ki, azt mondjuk, hogy az él összeköti a két végpontot. Az élek halmaza tehát definiálható az összes csúcsokból képzett kételem halmazok részhalmazaként. Gyakran mégis úgy tekintenek az élek halmazára, mint egy halmaz amelyen értelmezve van egy illeszkedési reláció, amely egy élhez egy csúcspárt rendel, az él végpontjait. Ha az éleket elláthatjuk egy iránnyal (irányítással), akkor irányított gráfot kapunk. Tekintsük az alábbi gráfot:



A gráf csúcsai :

$$V = \{A, B, C, D\}$$

A gráf élei:

$$E = \{e_1, e_2, e_3, e_4\}$$

Ha egy él két csúcsra illeszkedik, akkor összeköti azokat. Az olyan élt, amely egy csúcsra illeszkedik, tehát kezd- és végpontja azonos, **hurokélnak** mondjuk.

A fenti gráfnak egy hurokéle van, ez az e_4 él. Ha két különböző nem hurokélnak végpontjai azonosak, akkor ezeket **többszörös éleknek** mondjuk. A fenti gráfnál többszörös élek az e_1 és az e_2 .

7.1.2. Definíció

Két **csúcs szomszédos**, ha legalább egy él összeköti ket.

A fenti gráfnál szomszédos csúcspár például az A és a B csúcs, de nem szomszédos az A és a D csúcs.

Azokat az éleket, amelyeknek van közös végpontja, **szomszédos éleknek** mondjuk.

A fenti gráfnál szomszédos élpár pl. az e_1 és az e_3 , de nem szomszédos az e_1 és az e_4 él

Az olyan csúcsot, amelyre egyetlen él sem illeszkedik, **izolált csúcsnak** nevezzük. A fenti gráfnak egy izolált csúcsa van, a C csúcs.

A csúcs fokszáma alatt a rá illeszked élek számát értjük. A fenti gráfnál az A csúcs harmadfokú, a B másodfokú, a C nulladfokú, a D harmadfokú (a hurokél kettvel emeli a fokszámot)

7.1.3. Definíció

Egy **gráf egyszer**, ha nincsenek benne többszörös és hurokélek, **multigráf**, ha vannak többszörös élei, és multigráf vagy pszeudográf, ha többszörös élek és hurokélek is vannak benne (nincs egységes elnevezés az irodalomban). A gráf (megkülönböztet jelz nélkül) többnyire egyszer gráfot jelöl.

7.1.4. Definíció

Teljes gráfnak nevezzük azt az egyszer gráfot, amelyben bármely két csúcsot él köt össze.

Az n csúcsú teljes gráfot teljes n -szögnek is szokás nevezni.

Azt a gráfot, amely csupa izolált csúcsból áll **üres gráfnak** nevezzük. Az üres gráfnál $E = \emptyset$.

```
[ > with(GraphTheory):with(RandomGraphs):  
[ > K4 := CompleteGraph(4);  
[ > Edges(K4);  
[ > DrawGraph(K4);  
[  
Teljes 4gráf
```

7.1.5. Definíció

Az n csúcsú G gráf $\bar{G}$ komplementerén azt a gráfot értjük, amelynek csúcsai a G csúcsai, élei az n csúcsú teljes gráfnak azok az élei, amelyek nem élei a G -nek.

7.1.6. Példa

Képezzünk véletlenszeren egy ötponτού gráfot, s adjuk meg a komplementerét !

Megoldás

```
[ > C := RandomGraph(5,5);  
[ > G := GraphComplement(C);  
[ > Edges(C);  
[ > Edges(G);  
[ > plots[display](array([DrawGraph(C), DrawGraph(G)]));  
[  
Gráf és komplementere
```

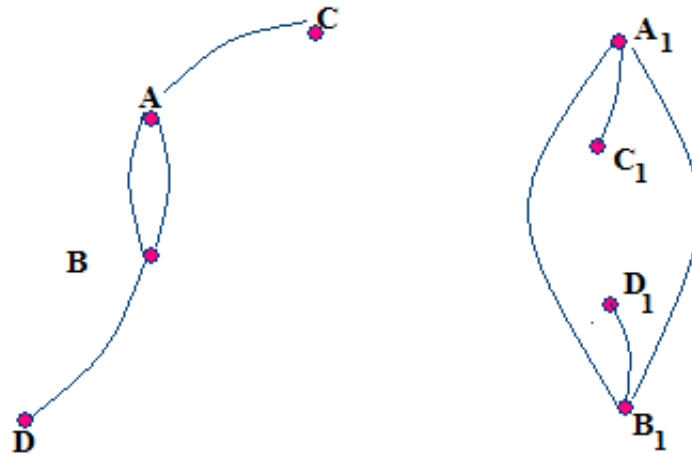
A **gráf címkézése** általában egyedi címkék (legtöbbször természetes számok) csúcsokhoz és élekhez rendelését jelenti. Ha egy gráf csúcsai, vagy élei címkézettek akkor címkézett gráfnak hívjuk, különben címkézetlennek. Különbséget tehetünk csúcs-címkézett és él-címkézett gráfok között, ha

hangsúlyozni szeretnénk, hogy csak a csúcs- vagy élhalmaz elemeit tekintjük megkülönböztethetnek.

7.1.7. Definíció

A $G_1 = (V_1, E_1)$ és $G_2 = (V_2, E_2)$ gráfok izomorfak, ha létezik egy $f: V_1 \leftrightarrow V_2$ éltartó bijekció, azaz olyan bijektív leképezés, amely szomszédos csúcsokat szomszédos csúcsokba, szomszédos éleket szomszédos élekbe visz át. Az izomorf gráfok között nem teszünk különbséget, azokat egyenlőknek tekintjük.

Az alábbi ábrán két izomorf gráfot látunk:



Egy **gráf végtelen**, ha csúcsainak vagy éleinek vagy akár mindkettőnek száma végtelen. Egy olyan gráfot, amelyben minden csúcs fokszáma véges lokálisan véges gráfnak hívjuk. Egy gráfról, ha nem állítjuk az ellenkezőjét, mindig feltehet, hogy véges.

A G gráf **részgráfja**, egy olyan gráf melynek csúcs- és élhalmazai részhalmazai G csúcs- és élhalmazainak. Azt mondjuk, hogy G tartalmazza H -t, ha G -nek van olyan részgráfja, mely megegyezik vagy izomorf H -val (az adott helyzettel függ).

Egy H részgráf **feszít részgráfja** G -nek, ha csúcshalmaza megegyezik G -ével. Ekkor H kifeszíti G -t.

Egy H részgráf **feszített részgráfja** G -nek, ha H bármely két csúcsára igaz, hogy szomszédosak H -ban akkor és csak akkor, ha szomszédosak G -ben. Másként fogalmazva H feszített részgráfja G -nek, ha úgy adódik, hogy vesszük G bizonyos szögpontjait és minden, köztük futó élt.

Gráfok kezelése Maple-vel

A networks csomag használata

```
> restart;
```

```
Töltsük be mindenekeltt a csomagot:
```

```
> with(networks);
```

```
A new(G) utasítással, s a csúcsok (addvertex) és élek (addedge) megadásával hozhatjuk létre a gráfot.
```

```
> new(G):
```

```

addvertex([A,B,C,D],G):
addedge([A,B],{A,B},{A,D},{D,D}],names=[e[1],e[2],e[3],e[4]
],G):

```

A draw utasítás felrajzolja a gráfot:

```

> draw(G);
> `A gráf csúcsai`=vertices(G);
> `A gráf élei`=edges(G);

```

A G gráf éleinek végpontjait az **ends(csúcsnév,G)** utasítással kaphatjuk meg:

```

> for i in edges(G) do
    i=ends(i,G)
od;

```

A G gráf csúcsainak fokszámát a **vdegree(csúcsnév, G)** utasítással kaphatjuk meg:

```

> for i in vertices(G) do
    i=vdegree(i,G)
od;

```

A gráfról részletes információt a **show(gráfnév)** utasítással kaphatunk:

```

> show(G);

```

7.2.1. Példa

Adjuk meg és ábrázoljuk a G gráfot a Maple-vel, ha csúcsai: $V = \{1, 2, 3, 4, 5, 6\}$ és élei: $E = \{\{1,2\}, \{1,5\}, \{2,3\}, \{2,5\}, \{3,4\}, \{4,5\}, \{4,6\}\}$

Megoldás

```

> new(G):
addvertex([1,2,3,4,5,6],G):
addedge([1,2],{1,5},{2,3},{2,5},{3,4},{4,5},{4,6}],G):
> `A gráf csúcsai`=vertices(G);
> `A gráf élei`=edges(G);

```

A G gráf éleinek végpontjait az **ends(csúcsnév,G)** utasítással kaphatjuk meg:

```

> for i in edges(G) do
    i=ends(i,G)
od;

```

A G gráf csúcsainak fokszámát a **vdegree(csúcsnév, G)** utasítással kaphatjuk meg:

```

> for i in vertices(G) do
    i=vdegree(i,G)
od;

```

Végül felrajzoljuk a G gráfot:

```

> draw(G);

```

G címkézett egyszer gráf.

▼ Gráf létrehozása

```
> restart;  
with(GraphTheory):
```

A gráf létrehozásának alapvető utasítása a [Graph](#) parancs. Irányított gráfok esetén a [Digraph](#) parancsot használjuk. Mindkét utasítás alkalmazása esetén a csúcsokat rendezett listával adhatjuk meg. A csúcsok neve lehet egész szám, szimbólum, vagy string. Stringek használata esetén tetszőleges szöveget adhatunk a csúcsok nevéül.

```
> G := Graph([a,b,c]);  
> Vertices(G); #A csúcsok  
> T := Digraph([1,2,3]);  
> Vertices(T);  
> H := Graph( ["0:0", "0:1", "1:0", "1:1"] );  
> Vertices(H);
```

Ha csupán a csúcsok számát adjuk meg, akkor a csúcsok címkéi automatikusan az 1,2,...,n értékek lesznek.

```
> G := Graph(4);  
> Vertices(G);
```

▼ Élek megadása

Az [AddEdge](#) utasítással egy vagy egyszerre több él is megadható.

Irányítás nélküli gráfok esetén az élek mindegyike az általa összekötött két csúcs halmazaként adható meg.

```
> G := Graph(4);  
> AddEdge(G, {1,2});  
AddEdge(G, {{2,3},{3,4}});  
> Edges(G); #A G gráf élei
```

Irányított gráfok éleinek megadására az [AddArc](#) utasítást használjuk. Ekkor az éleket az általuk összekötött két csúcs listájával adjuk meg. A lista első eleme az él kiindulási pontja, a második elem a végpontja.

```
> H := Digraph(4);  
> AddArc(H, {[1,2],[1,3],[2,3],[2,3],[3,1],[3,4]});  
> Edges(H);  
> Departures(H,3); # A 3-as csúcsból induló élek  
végpontjainak listája  
> Arrivals(H,3); # A 3-as csúcsba érkező élek  
kezdőpontjainak listája
```

▼ A gráf létrehozása élek halmazával

Az irányítás nélküli gráf létrehozható az élei halmazának megadásával is. Ilyenkor, ha külön nem adjuk meg a csúcsokat, akkor a csúcsok a megadott élek végpontjai lesznek. Ennek alapján az előző szekcióban szereplő G gráfot - amely egy egyszerű ösvény (út)-

megadhatjuk a következőképpen is:

```
> G := Graph({{1,2},{2,3},{3,4}});
```

```
> Vertices(G);
```

Az AdjacencyMatrix(G) utasítás a G gráf szomszédossági mátrixát adja meg. A szomszédossági mátrix sorai és oszlopai a gráf csúcsai szerint vannak indexelve. Az i-edik sor j-edik eleme 1, ha a gráf i-edik csúcsát él köti össze a j-edik csúccsal, egyébként az elem értéke 0. Irányítás nélküli gráf szomszédossági mátrixa szimmetrikus.

```
> AdjacencyMatrix(G);
```

Hasonlóan irányított gráf is megadható éleinek listájával.

```
> H := Digraph([1,2],[2,3],[3,4]);
```

Az irányított gráf szomszédossági mátrixa i-edik sorának j-edik eleme 1, ha az i-edik csúcsból él indul a j-edik csúcsba, egyébként az elem értéke 0.

```
> AdjacencyMatrix(H);
```

▼ Gráf létrehozása a szomszédossági mátrix-szal

Az irányítás nélküli és irányított gráf egyaránt definiálható a szomszédossági mátrixa segítségével is. Ha a csúcsok nevét külön nem adjuk meg egy listában, akkor azok címkéi az 1,2,...,n egészek lesznek, ahol n a szomszédossági mátrix dimenziója (rendje).

```
> A := Matrix([[0,1,0,0],[1,0,1,0],[0,1,0,1],[0,0,1,0]]);
```

```
> G := Graph(A);
```

A G gráf csúcsai:

```
> Vertices(G);
```

A G gráf élei:

```
> Edges(G);
```

Definiáljunk most irányított gráfot!

```
> B := Matrix([[0,1,0,0],[0,0,1,0],[0,0,0,1],[0,0,0,0]]);
```

Adjunk nevet a csúcsoknak. Legyenek ezek pl. a, b, c, d. Ekkor az irányítottgráf definíciója a következő:

```
> H := Digraph([a,b,c,d],B);
```

```
> Vertices(H);
```

```
> Edges(H);
```

▼ Szomszédok listájával

Ha a G gráfnak n csúcsa van, akkor a szomszédok listája is n elem, s a lista k-adik eleme a k-adik csúcs szomszédjainak listája.

```
> L := [[2],[1,3],[2,4],[3]];
```

```
> G := Graph(L);
```

```
> Edges(G);
```

```
> Neighbors(G);
```

Irányított gráf esetén a szomszédosság nem feltétlenül kölcsönös. A következő gráf irányított kör, irányított él vezet az 1-es csúcsból a 2-be, a 2-esből a 3-ba, a 3-ból a 4-be és a 4-ből az 1-be. Ez a lista megfelel a gráf kiindulási csúcsai (departures) listájának.

```

> L := [[2],[3],[4],[1]];
> H := Digraph(L);
> Vertices(H);
> Neighbors(H);
> Arrivals(H);
> Departures(H);

```

▼ Élsorozat használatával

Kényelmes megadási mód. Az 1,2,3,4 élsorozat az 1-ből 2-e majdinnen a 3-ba, s onnan tovább a 4-es csúcsba való haladást jelenti

```

> T := Trail(1,2,3,4,1);
> G := Graph(T);
> Edges(G);
> H := Digraph(Trail(a,b,c,d,c,b,a));
> Edges(H);

```

▼ Súlyozott élek

Súlyozott él gráf létrehozásakor a weighted opciót kell használnunk.

```

> G := Graph(5, weighted);

```

Az irányítás nélküli, w súlyú (u,v) él a $[\{u,v\},w]$ formátummal adható meg.

Írányított él esetén az $[[u,v],w]$ formátumot kell használnunk.

```

> AddEdge(G, [{1,2},2.3]);
> AddEdge(G, [{3,4},4.5]);
> AdjacencyMatrix(G);

```

A WeightMatrix utasítás megadja az élek súlyát

```

> W := WeightMatrix(G);

```

Az elz G gráf egy lépésben is létrehozható a következ utasítással

```

> G := Graph(5, { [{1,2},2.3], [{3,4},4.5] });

```

Az él súlya 0 is lehet.

```

> AddEdge(G, [{2,3},0.0]);
> Edges(G);
> AdjacencyMatrix(G);
> WeightMatrix(G);

```

▼ Gráfok ábrázolása

[Az alábbiakban néhány példával mutatjuk be a f opciókat. További példák találhatók a speciális gráfok szekciójában.

```

> G := Graph(9,{{1,2},{2,3},{3,1},{6,7},{7,8}});
> DrawGraph(G);

```

```
> H := Digraph(5, {[1,2], [2,3], [3,4], [2,5]});
```

```
> DrawGraph(H);
```

```
> DrawGraph(H, style=circle);
```

Ha elégedetlenek vagyunk a csúcsok elhelyezkedésével, akkor a SetVertexPositions utasítással változtathatunk helyzetükön.

```
> SetVertexPositions(H, [[0,0], [0.5,0], [1.0,0], [1.5,0], [0.5,1]]);
```

```
> DrawGraph(H);
```

```
> P := SpecialGraphs:-PetersenGraph();
```

A jól ismert Petersen graf nem rajzolható meg a síkban az élek metszése nélkül.

```
> DrawGraph(P);
```

A csúcsok elhelyezkedésére vonatkozó információkat a gráf definíciója tartalmazza.

```
> GetVertexPositions(G);
```

A "spring" módszer minden egyes végrehajtás során némiképp más, "szimmetrikus" ábrát hoz létre:

```
> DrawGraph(P, style=spring, redraw);
```

A spring módszer 3D-s ábra létrehozására is alkalmazható

```
> DrawGraph(P, style=spring, dimension=3);
```

A spring módszer nagyobb méretű gráfok ábrázolása esetén is hatékony (1000 csúcsig).

```
> S := SpecialGraphs:-SoccerBallGraph();
```

```
> DrawGraph(S, style=spring);
```

Nagy méretű súlyozott gráfokat az élek súlyai és a csúcsok címkéi túlzsúfolttá tehetnek.

Ilyenkor el kell tekintenünk az utóbbi információk használatától.

```
> G := Graph({[a,b],2}, [{b,c},3], [{c,d},4], [{d,b},5]);
```

```
> DrawGraph(G);
```

```
> DrawGraph(G, showweights=false);
```

```
> DrawGraph(G, showlabels=false);
```

▼ Gráf reprezentációja mátrix-szal

Az [AdjacencyMatrix](#) utasítás eredményeként a gráfhoz rendelt négyzetes mátrixot, a szomszédossági mátrixot kapjuk. A mátrix i -edik sorának j -edik eleme 1, ha az i -edik csúcsot él köti össze a j -edik csúccsal, 0 egyébként. Tekintsük a korábban definiált G gráfot!

```
> `Szomszédossági mátrix`=AdjacencyMatrix(G);
```

```
> DrawGraph(G, style=circle);
```

```
> Edges(G);
```

▼ A gráf bejárása

▼ Fogalmak

Egy **séta** (élsorozat) csúcsok és élek váltakozó sorozata, mely csúccsal kezdik és csúcsban végzik, és minden csúcs szomszédos az t megelőző és t követő éllel, illetve minden él két végpontja az t megelőző és az t követő csúcs. Egy **séta** zárt, ha első és utolsó csúcsai megegyeznek, különben nyitott.

A **séta** (élsorozat) **L hossza** az őt alkotó élek száma. Minden nyitott sétára $L = n - 1$, ahol n a meglátogatott csúcsok száma (minden csúcsot annyiszor számolunk, ahányszor meglátogattuk), minden zárt sétára $L = n$ (a kezd- és végcsúcs ugyan kétszer szerepel, de csak egyszer számoljuk)

Az élsorozatot **vonalként** mondjuk, ha egyetlen él sem fordulhat el benne egynél többször. Az **út** olyan vonal, amelyben egyetlen csúcs sem fordulhat el egynél többször. Ha az út kezd- és végpontja egybeesik, **körrel** beszélünk. Az utat egyszer nyílt sétának, a kört egyszer zárt sétának is mondják.

Egy séta, ha minden élet pontosan egyszer használ **Euler-út** vagy **Euler-kör**, aszerint, hogy nyílt vagy zárt. A definíciójából eredően egy ilyen séta egy csúcsot akár többször is érinthet.

Két út pontdiszjunkt (vagy pontidegen, de néha hívják függetlennek is), ha nincs közös csúcsuk, a kezd- és végpontjukat leszámítva. Két út éldiszjunkt (élidegen), ha nincs közös élük.

Az n hosszúságú kört C_n -nel jelöljük. C_1 egy hurokél, C_2 kétszög, azaz egy többszörösél-pár, C_3 egy háromszög és így tovább.

7.3.1. Definíció

A G gráfot **összefüggnek** mondjuk, ha bármely két csúcsához létezik a G -ben út, amely összeköti ket.

A **fa** egy összefüggő, körmentes, egyszeres gráf.

A fa első fokú csúcsait **levélnek** hívjuk. Egy nem levél csúcs a fában **belső csúcs**. Néha van a fának egy megkülönböztetett csúcsa, a **gyökér**. A **gyökeres fa** olyan fa, amelyben van gyökér. Az irányított gyökeres fák éleit általában a gyökértől elfelé mutató irányítással látjuk el.

A T fa részfája, egy összefüggő részgráfja T -nek.

Erdnek hívunk egy körmentes egyszeres gráfot, amely nem feltétlenül összefügg. (Az erdők pontdiszjunkt fák uniói, innen az elnevezés.)

Az F erd egy részedje F egy részgráfja.

A **feszít fa** egy olyan feszít részgráf, amely fa. Minden gráfnak van feszít erdje, de feszít fája, csak az összefüggő gráfoknak.

Az n -ed fokú fa (n -áris fa) egy gyökeres fa, amelyben minden csúcsnak legfeljebb n gyereke van. A teljes n -ed fokú fában minden csúcs gyerekeinek száma pontosan n vagy 0 (az előbbiek a belső csúcsok, az utóbbiak a levelek). Az első fokú fa egy út. A **másodfokú fát** hívják **bináris fának** is (a bináris fa fogalma általában ezen felül azt is magában foglalja, hogy különbség van bal- és jobboldali gyerekek között) nevezik.

▼ A gráf éleinek bejárása



Leonhard Euler
(1707—1783)
portréja, 1753
Emanuel Handmann
(1718—1781)
festménye

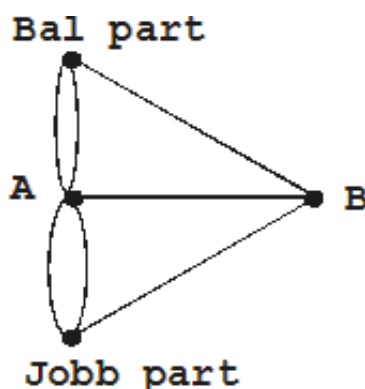
[Pach János: A megtestesült analízis — Leonhard Euler](#)
[Leonhard Euler élete és munkássága](#)

Euler gráfok



1. ábra

Leonard Euler (1707-1783) nevéhez kapcsolódik az els gráfelméleti munka, mely 1736-ban jelent meg a Szentpétervári Tudományos Akadémia közleményeiben. Az értekezését Euler az ún. Königsbergi hidak problémájával kezdte. A Pregel folyó A, B szigeteit hidak kötötték össze egymással és a partokkal is. Az A szigetet két párhuzamos híd kötötte össze a jobb parttal, egy híd a B szigettel, s ugyancsak két párhuzamos híd vezetett az A-ról a bal partra is. B-t egy-egy híd kötötte össze a bal és a jobb parttal is és B-ről vezetett egy híd A-ra is, melyet az elbb már említettünk. A kérdés az volt, be lehet e járni a hidakat valamely fix C pontból oly módon, hogy minden hídon átmegyünk pontosan egyszer. Euler lényegében teljes általánosságban megoldotta a feladatot.



2. ábra

7.3.2. Definíció

Egy adott G gráf összes élét tartalmazó vonalat a G gráf Euler-vonalának nevezzük. A gráfot Euler-gráfnak mondjuk, ha nincs izolált csúcsa, és van zárt Euler-vonala.

7.3.3. Tétel (szükséges feltétel)

Ha a G gráf Euler-gráf, akkor összefügg, és minden csúcsa páros foksámú.

Bizonyítás

A feltétel szerint G -nek van zárt Euler-vonala. Ekkor, ha valamely P csúcson k -szor aladunk át, akkor P foka $2 \cdot k$ (hiszen k -szor bemegyünk P -be, de mindannyiszor ki is jövünk belé). Másrészt G -nek nincs izolált csúcsa, tehát minden csúcsa illeszkedik valamely élre, s így az Euler-vonalon fekszik, ezért G összefügg.

7.3.4. Következmény

A königsbergi hidak problémája megoldhatatlan. Igaz a tétel megfordítása is, ezt bizonyítás nélkül közöljük:

7.3.5. Tétel (elégéses feltétel)

Ha a G gráf összefügg és minden csúcsa páros fokú, akkor Euler-féle.

7.3.6. Tétel (Euler tétele)

A G gráf akkor és csak akkor Euler-féle, ha összefügg, és minden csúcsa páros foksámú.

▼ **Ejáráás annak eldöntésére, hogy a gráf Euler-féle-e, vagy sem**

Töltsük be mindenekeltt a GraphTheory csomagot!

```
> with(GraphTheory):
```

```
== Tekintsük elször a négy csúcsú teljes gráf-ot. Rajzoljuk föl a gráfot!
```

```
> DrawGraph(CompleteGraph(4));
```

```
== Az IsEulerian eljárás eldönti, hogy a gráf Euler-féle-e, vagy sem:
```

```
> IsEulerian(CompleteGraph(4));
```

```
== A gráf nem Euler-féle, hiszen nem csúcsai páratlan foksámúak. Tekintsük most a teljes 5-gráfot:
```

```
> DrawGraph(CompleteGraph(5));
```

```
> IsEulerian(CompleteGraph(5), 'T');
```

Ennek minden csúcsa páros foksámú, így a tétel értelmében Euler-féle. Egy zárt Euler-vonala:

```
> T;
```

A gráf csúcsainak bejárása

7.3.7. Definíció

Az olyan kört, amely a G gráf minden csúcsán áthalad a G gráf **Hamilton körének** nevezzük. Nincsen minden gráfnak Hamilton köre. Nem ismeretes a Hamilton kör létezésének szükséges és elégséges feltétele.

Egy elégséges feltételt közlünk:

7.3.8. Tétel

Ha egy n pontú egyszerű gráf minden pontjának foka legalább $\frac{n}{2}$ ($n \geq 3$), akkor van a gráfnak Hamilton-köre.

```
> with(GraphTheory):
```

```
> with(SpecialGraphs):
```

Az IsHamiltonian utasítással eldönthetjük a gráfról, hogy van-e Hamilton köre. Az alábbi beállítás alapján üzenetet kapunk arról, hogy milyen eljárást alkalmaz az IsHamiltonian utasítás:

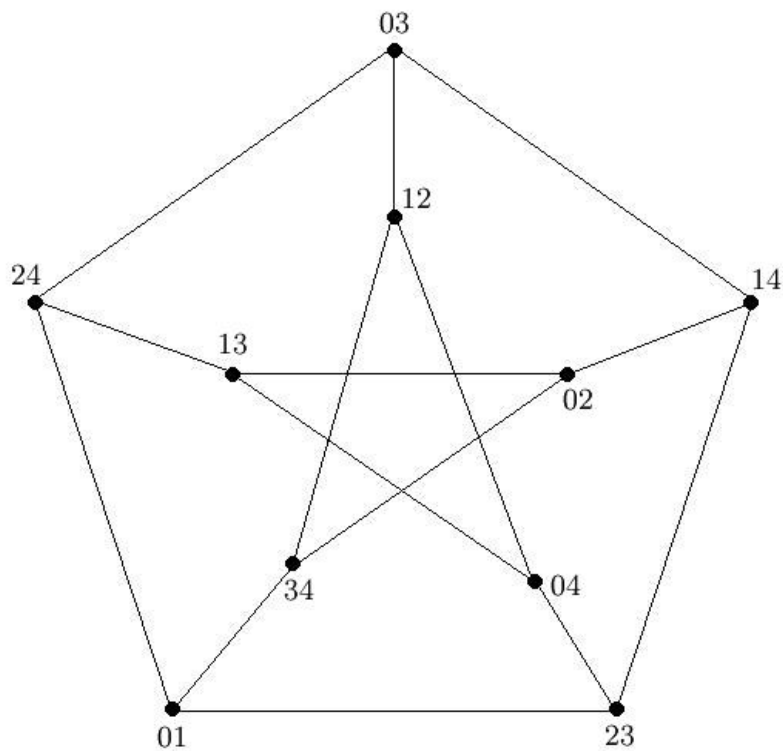
7.3.9. Példa

Tekintsük a nevezetes [Petersen-gráf](#) -ot! Legyen egy 5 elem halmaz, $P = \{0,1,2,3,4\}$ Ebből a

halmazból kiválasztjuk az összes kételem részhalmazt. Ezek száma: $\binom{5}{2} = 10$ Ezeket a kételem halmazokat megfeleltetjük a gráf csúcsainak: $V = \{\{0,1\}, \{0,2\}, \{0,3\}, \{0,4\}, \{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}, \{2,4\}, \{3,4\}\}$

Akkor van él a gráfban két csúcs között, ha a csúcsoknak megfelelő halmazok diszjunktak, azaz

$$E = \{\{u,v\} \in \binom{V}{2} : u \cap v = \emptyset\}$$



```
> infolevel[IsHamiltonian] := 2;
```

```
Adjuk meg a gráfot!
```

```
> P := PetersenGraph();
```

```
Rajzoljuk föl a gráfot!
```

```
> DrawGraph(P);
```

```
Vizsgáljuk meg az IsHamiltonian utasítással, hogy van-e a gráfnak Hamilton-köre!
```

```
> IsHamiltonian(P);
```

```
A gráfnak nincs Hamilton-köre. Vegyünk hozzá a gráfhoz egy élet, kössük össze az 1-es és a 3-as csúcsot!
```

```
> AddEdge(P, {1,3});
```

```
> IsHamiltonian(P, 'C');
```

```
Ennek a gráfnak már van Hamilton köre.
```

```
> HighlightTrail(P, C, red);
```

```
> `A Hamilton kör`=C;
```

```
Jelenítsük meg a Hamilton-kört!
```

```
> DrawGraph(P);
```

```
Tekintsük az általánosított Petersen-gráfot
```

```
> M:=GeneralizedPetersenGraph(6,2);
```

```
> DrawGraph(M);
```

```
> IsHamiltonian(M, 'C');
```

```
Ennek a gráfnak is van Hamilton-köre
```

```
> C;
```

```
> HighlightTrail(M, C, red);
```

```
> DrawGraph(M);
```

```
>
```

▼ Irányított gráfok

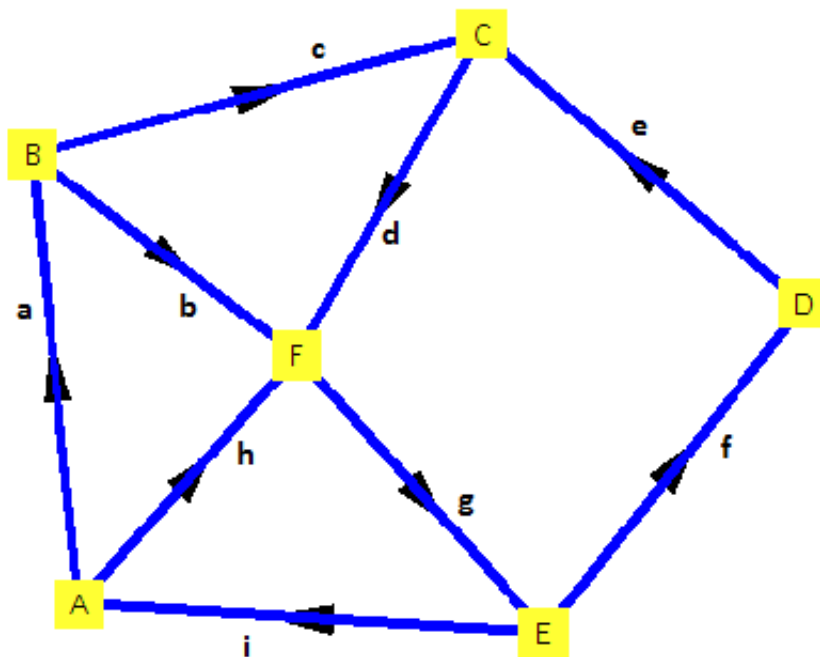
Ha egy gráf éleihez meghatározott irányt rendelünk, akkor irányított gráfhoz jutunk. Egy élt akkor nevezünk irányítottnak, ha megmondjuk, hogy a két végpontja közül melyiket nevezzük kezdpontnak és melyiket végpontnak. Az élek irányítását általában nyílhegyekkel jelölik. Legyen a H gráf irányított. A

$$P_0, k_1, P_1, k_2, P_2, \dots, k_s, P_s$$

élsorozatot folytatólagosan irányított élsorozatnak nevezünk, ha minden i -re ($i = 1, 2, \dots, s$) a k_i él kezdpontja P_{i-1} , végpontja P_i .

A folytatólagosan irányított élsorozatot elgondolhatjuk, mint egy autókázást egyirányú utcákon. A továbbiakban akörülményes "folytatólagosan" irányított" kifejezés helyett egyszerűen az "irányított" szót használjuk.

Az irányított gráfoknál is bevezethetjük a vonal és az út fogalmát, természetesen figyelembe kell vennünk az élek irányítását. Az alábbi ábra irányított grájában ***CdFgEfDeC*** irányított kör, ***AhFgEfDeCdF*** irányított vonal.

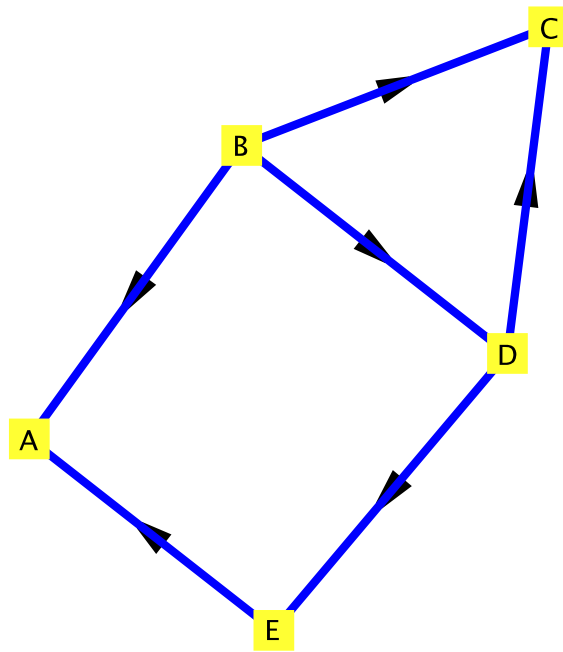


Az irányított gráfot összefüggnek mondjuk, ha az élek irányításnak megszüntetésével kapott irányítás nélküli gráf összefügg. Ez a fogalom azonban nem jellemzi megfelelően az irányított gráfban a csúcsok elérhetőségét. Ha egy településen csupa egyirányú utca van, akkor hiába összefügg az úthálózat, nem biztos, hogy bárhonnan bárhova el tudunk jutni a közlekedési szabályok betartásával. Ezért az irányított gráfoknál egy új fogalomra, az ers összefüggés fogalmára van szükség.

7.4.1. Definíció

Azt mondjuk, hogy az irányított G gráf ersen összefügg, ha bármely csúcsából bármely csúcsba el lehet jutni irányított úton.

Ha G ersen összefügg, akkor összefügg, de megfordítva ez nem mindig igaz. Az alábbi ábrán látható gráf pl. összefügg, de nem ersen összefügg, mert az A csúcsból a B csúcsba nem vezet irányított út.



Az irányított gráfok esetében is beszélhetünk (irányított) Euler-vonalról és (irányított) Hamilton körrel.

Irányított gráfokban egy csúcs fokszáma a belé kiinduló és a benne végződő élek számának a különbsége. Legyen $f(P)$ a P -ben kezdődő, $n(P)$

akkor P -t *forrásmentes* csúcsnak, ha

$$f(P) \leq n(P),$$

$$n(P) \leq f(P),$$

akkor P -t *nyelmentes* csúcsnak nevezzük. Ha egy gráf minden csúcsa forrásmentes, akkor a G forrásmentes gráf, ha minden csúcsa nyelmentes, akkor nyelmentes gráf.

7.4.2. Tétel

Minden forrásmentes gráf egyidejleg nyelmentes is. Ugyanígy, minden nyelmentes gráf forrásmentes.

Bizonyítás

Tegyük föl, hogy G forrásmentes, vagyis legyen $f(P) \leq n(P)$ a G minden P pontjára. Számoljuk össze a G gráf éleit úgy, hogy minden csúcsnál az onnan kiinduló éleket vesszük figyelembe. Az élek száma így

$$f(P_1) + f(P_2) + \dots + f(P_n).$$

Ugyanezt a számot kapjuk akkor is, ha minden csúcsnál az ott végződő éleket vesszük számításba:

$$n(P_1) + n(P_2) + \dots + n(P_n).$$

Mivel feltevésünk szerint minden i -re ($i = 1, 2, \dots, n$)

$$f(P_i) \leq n(P_i),$$

mindenütt az egyenlőség érvényesül, tehát

$$f(P_i) = n(P_i)$$

minden i -re, tehát G nyelmentes. Analóg módon igazolható a tétel második állítása is. Ezután könnyen átfogalmazhatjuk az Euler-tételt irányított gráfokra.

7.4.3. Tétel

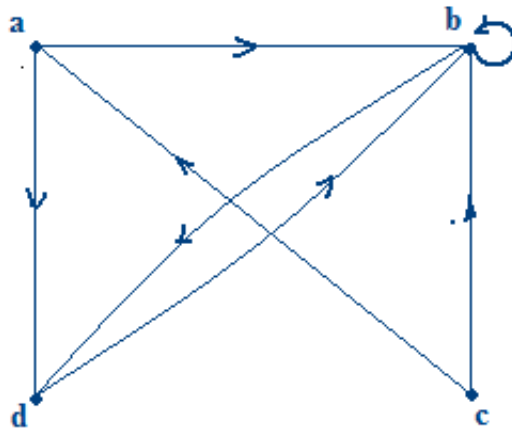
A G gráfban akkor és csak akkor létezik zárt irányított Euler-vonal, ha G összefügg és forrásmentes.

▼ Relációk reprezentációja irányított gráfokkal

Az A halmazon értelmezett $R \subseteq A \times A$ homogén reláció - a mátrix-reprezentáció mellett - olyan irányított gráffal is reprezentálható, amelynek csúcsai az A halmaz elemei, élei pedig pontosan azok az (a, b) élek, amelyekre $(a, b) \in R$. Ezzel a megállapodással egy-egyértelmű megfeleltetést hoztunk létre az A halmazon értelmezett relációk, és az olyan irányított gráfok között, amelyeknek csúcsai az A halmaz elemei.

Ez azt jelenti, hogy minden, a relációkra érvényes állításnak megfeleltethet a gráfokra vonatkozó kijelentés, és megfordítva.

Az alább ábrán látható irányított gráf az a, b, c és d csúcsokból és az (a, b) , (a, d) , (b, b) , (b, d) , (c, a) , (c, b) és (d, b) élekből áll. Ennek megfelelően az R reláció $R = \{(a, b), (a, d), (b, b), (b, d), (c, a), (c, b), (d, b)\}$



▼ Eljárás a reláció mátrixához tartozó gráf meghatározására

```
> with(networks):
> graf:=proc(S)
> local elek,sulyok,csucsok,i,j:
> elek:=NULL:sulyok:=NULL:csucsok:=NULL:
> for i to linalg[rowdim](S) do
>   for j to linalg[coldim](S) do
>     if S[i,j]<>0 then
>       elek:=elek,[i,j]:
>       sulyok:=sulyok,S[i,j]:
```

```

> fi:
> od:
> csucsok:=csucsok,i:
> od:
> new(G):
  addvertex(csucsok,G):
  addedge([elek], weights=[sulyok],G):
> print(`A gráf élei, a relációt alkotó párok`=ends(G)):
> draw(G);
> end:
>

```

7.4.5. Példa

Legyen adott az R reláció M_R mátrixa:

$$M_R = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Adjuk meg a relációt reprezentáló gráfot!

Megoldás

```

> M:=Matrix(4,4,[1,0,1,1,1,1,0,0,0,1,0,1,1,0,1,0]);
> G:='G':
> graf(M);

```

A gráf képe nem nyújt teljes képet a gráfról, mert a gráf irányított, s az ábra nem mutatja az irányítást, ezért az eljárás az éleket is kiírja.

A GraphTheory csomagot is használhatjuk. A gráfot közvetlenül is létrehozhatjuk a szomszédossági mátrix segítségével. Ezen mátrix i -edik sorának j -edik eleme 1, ha az i -edik csúcsot él köti össze a j -edik csúccsal, 0 egyébként. Itt viszont nem jeleníthet meg a hurokélek, ezért itt is megadjuk a gráf éleit.

```

> with(GraphTheory):M1:=Graph(M);DrawGraph(M1);Edges(M1);
>

```

Fagráfok

Ebben a paragrafusban csak röviden foglalkozunk a fagráfokkal, de a 8. fejezetben visszatérünk rájuk. Idézzük föl először a fagráf fogalmát!

7.5.1. Definíció

Az összefügg, körmentes gráfot *fagráf*nak nevezzük

Az elnevezést az indokolja, hogy a fagráfok alakja sokszor valóban a csupasz ágú téli fákra hasonlít. A definícióból rögtön következik, hogy egy G fában bármely két csúcsot pontosan egy út köt össze. Az összefüggőség miatt ui. van ilyen összekötő út, a körmentességbl pedig levezethet, hogy egynél több nem lehet.

A fa els fokú csúcsait **levél**nek hívjuk. Egy nem levél csúcs a fában **bels csúcs**. Néha van a fának egy megkülönböztetett csúcsa, a **gyökér**. A **gyökeres fa** olyan fa, melyben van gyökér. Az ilyen fákat **ültetett fának** is nevezzük. Az irányított gyökeres fák éleit általában a gyökértl elfelé mutató irányítással látjuk el.

7.5.2. Tétel

Az n -pontú összefügg gráfnak legalább $n - 1$ éle van.

7.5.3. Tétel

Összefügg gráf bármely körének egy tetszleges élét törölve ismét összefügg gráfot kapunk.

Bizonyítás

Feltehetjük, hogy a törölt él nem hurokél, hiszen hurokél törlése nem szünteti meg az összefüggőséget. Töröljük az összefügg G gráf egy K körének $\{a,b\}$ élet. Ha G -ben az $\{a,b\}$ élen megyünk végig, a -ból b -be - vagy b -ből a -ba - jutunk; ezt pedig a törlés után is megtehtjük K megmaradt élein haladva. Tehát az $\{a,b\}$ él törlése után is eljuthatunk G bármely pontjából bármely másikba, nem törölt éleken haladva. Ha közben valamely kör éleit is bejárnók, azokat kihagyva, út élein haladhatunk.

A tétel alapján arra következtethetünk, hogy az n pontú és $n - 1$ él összefügg gráfok körmentesek, ugyanis, ha volna egy n pontú és $n - 1$ él, kört tartalmazó összefügg gráf, akkor annak egy körbe tartozó élet törölve n -pontú és $n - 2$ él összefügg gráfot kapnánk, ellentétben azzal, hogy egy n -pontú összefügg gráfnak legalább $n - 1$ éle van. Érvényes tehát a következő állítás, amely egyúttal a minimális számú él tartalmazó összefügg gráfok jellemző tulajdonságát szolgáltatja:

7.5.4. Tétel

Az n -pontú és $n-1$ él összefügg gráfok fák.

7.5.5. Tétel

Ha egy n -pontú gráfnak legalább n éle van, akkor van a gráfban kör.

Ha ezek után tekintetbe vesszük, hogy minden n -pontú összefügg gráfnak legalább $n - 1$ éle van, akkor kiondhatjuk, hogy minden n -pontú összefügg körmentes gráfnak pontosan $n - 1$ éle van. Érvényes tehát a következő tétel:

7.5.6. Tétel

Az n -pontú fagráf éleinek száma $n-1$.

A két tétel összefoglalva:

7.5.7. Tétel

Az n pontú összefügg gráf akkor és csak akkor fagráf, ha $n-1$ éle van.

A következőkben bemutatunk néhány, a fagráfok kezelésére vonatkozó Maple-utasítást.

```
[> restart;  
[> with(networks):  
[> with(GraphTheory):  
[Létrehozunk a T gráfot. A gráf irányítás nélküli, éleinek végpontjaival definiáljuk.  
[> T := Graph({{1,2},{2,3},{2,4},{3,5},{3,6},{3,7},{4,8},{4,9}});  
[Felrajzoljuk a gráfot:
```

```
> DrawGraph(T);
```

A TreeHeight(T,r) utasítás megadja a gyökérnek tekintett r csúctól való maximális távolságot, tehát a legtávolabbi csúcs(ok)ba vezető út éleinek számát.

```
> TreeHeight(T, 4);
```

Tehát, ha a 4-es csúcsot tekintjük a gyökérnek, akkor a legtávolabbi csúcs(ok)nak távolsága (ezek itt most az 5,6,7 címkeű csúcsok) tle 3.

```
> TreeHeight(T, 9);
```

A 4-es számú csúcsot gyökérként tekintve 3 a legnagyobb távolság, a 9-es esetén pedig 4. Jelenítsünk meg egy 4 hosszúságú élsorozatot!

```
> H := InducedSubgraph(T, [9,4,2,3,5]);
```

```
> HighlightSubgraph(T,H,red,green);
```

```
> DrawGraph(T);
```

Az IsTree utasítással eldönthetjük, hogy a tekintett gráf fa-e vagy sem.

```
> `Fagráf-e a T ?`=IsTree(T);
```

A RandomGraphs csomagot betöltve véletlenszerűen generált gráfokat hozhatunk létre

```
> with(RandomGraphs);
```

```
> T := RandomTree(10,degree<5);
```

```
> DrawGraph(T);
```

Az éleknek véletlenszerűen (is) adhatunk súlyt:

```
> T := RandomTree(8, weights=1..9);
```

```
> IsTree(T);
```

A Weightmatrix utasítás megadja a súlyozott éleket. Az i-edik sor j-edik eleme az i-edik csúcsot a j-edik csúccsal összekötő él súlyát adja meg.

```
> WeightMatrix(T);
```

```
> DrawGraph(T);
```

Különleges szerepek azok a fák amelyeknek minden elágazása (legfeljebb) kétirányú. Az ilyen fákat **bináris fák**nak nevezzük. Az alábbi ábrán egy bináris fa látható. A fa gyökere az a csúcs.

```
> G2:= Graph([a,b,c,d,e,f,g,h,i,j,k]):  
  AddEdge(G2,{a,b},{a,c},{b,d},{b,e},{c,f},{c,g},{d,h},{d,i},{f,  
  j},{f,k});
```

```
> DrawGraph(G2,style=tree,root=a);
```

```
>
```

Ellenrz kérdések

1. Határozza meg a gráf fogalmát!
2. Mikor nevezünk egy gráfban két élt, két csúcsot szomszédosnak?
3. Mikor beszélünk hurokélről és mikor többszörös élekről?
4. Mit értünk izolált csúcson?
5. Mikor beszélünk véges, illetve végtelen gráfról?
6. Határozza meg a részgráf fogalmát!
7. Mit értünk séta (élsorozat), vonal, út, kör alatt?
8. Igaz-e, hogy minden út vonal?

9. Mit értünk gráfelméleti szempontból hatszögön?
10. Mit értünk a gráfban egy csúcs fokán?
11. Határozza meg az egyszer gráf fogalmát!
12. Mit értünk teljes gráf, illetve üres gráf alatt?
13. Hány éle van egy teljes n -gráfnak?
14. Mit értünk egy gráf komplementerén?
15. Mekkora egy k -adfokú P csúcs fokszáma az n csúcsú G gráf komplementerében?
16. Határozza meg a gráfok izomorfizmusát!
17. Mikor tekintünk két gráfot egyenlőnek?
18. Mi az Euler-vonal?
19. Mikor nevezünk egy gráfot Euler-gráfnak?
20. Mi a szükséges és elégséges feltétele annak, hogy a gráf Euler-féle legyen?
21. Mit értünk egy gráf Hamilton-körén?
22. Határozza meg az irányított gráf fogalmát!
23. Mit értünk folytatólagosan irányított élsorozaton?
24. Milyen fogalmakat használunk az irányított gráfok összefüggésének jellemzésére?
25. Milyen kapcsolatban van az összefüggőség és az erősen összefüggőség?
26. Rajzoljon fel olyan irányított gráfot, amelyik összefügg, de nem erősen összefügg!
27. Határozza meg a csúcsok forrásmentességének és nyelmentességének fogalmát!
28. Mikor nevezünk egy irányított gráfot forrásmentesnek, illetve nyelmentesnek?
29. Milyen kapcsolatban van a nyelmentesség és a forrásmentesség, mint gráf tulajdonság?
30. Hogyan fogalmazhatjuk meg az Euler-tételt irányított gráfokra?
31. Milyen kapcsolat van a relációk és a gráfok között?
32. Hogyan adhatunk meg gráfokat mátrixok segítségével?
33. Határozza meg a fagráf fogalmát!
34. Milyen összefüggés van a fákban a csúcsok száma és az élek száma között?
35. Mikor beszélünk ültetett fáról?
36. Milyen gráfot nevezünk bináris fának?