

Bevezetés a Számításelméletbe II. 6. előadás

Sali Attila

Budapest Műszaki és Gazdaságtudományi Egyetem

Számítástudományi Tsz.

I. B. 137/b

`sali@cs.bme.hu`

2002 március 19.

A kritikus út módszere (PERT-módszer)

A kritikus út módszere (PERT-módszer)

Az „emeletekre bontás” fontos alkalmazása az úgynevezett PERT-módszer.

A kritikus út módszere (PERT-módszer)

Az „emeletekre bontás” fontos alkalmazása az úgynevezett PERT-módszer. Az elnevezés az angol „Program Evaluation and Review Technique” rövidítéséből származik.

A kritikus út módszere (PERT-módszer)

Az „emeletekre bontás” fontos alkalmazása az úgynevezett PERT-módszer. Az elnevezés az angol „Program Evaluation and Review Technique” rövidítéséből származik.

Tegyük fel, hogy egy összetett feladatot több alvállalkozóval kell elvégeztetni. Az egyes részfeladatok nem végezhetőek el egymástól függetlenül: pl. egy házépítés során a kőművesmunkák nyilván megelőzik a festési munkákat.

A kritikus út módszere (PERT-módszer)

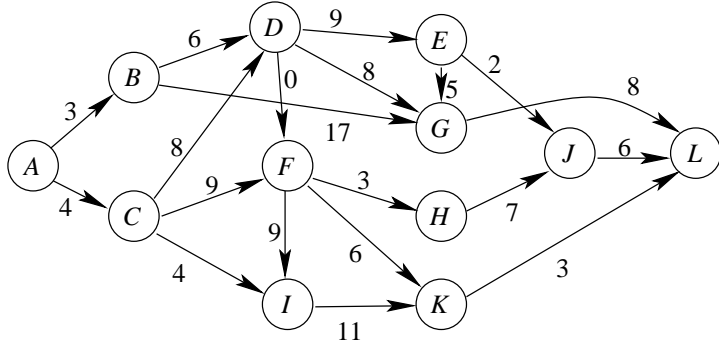
Az „emeletekre bontás” fontos alkalmazása az úgynevezett PERT-módszer. Az elnevezés az angol „Program Evaluation and Review Technique” rövidítéséből származik.

Tegyük fel, hogy egy összetett feladatot több alvállalkozóval kell elvégeztetni. Az egyes részfeladatok nem végezhetőek el egymástól függetlenül: pl. egy házépítés során a kőművesmunkák nyilván megelőzik a festési munkákat. A helyzetet egy G gráffal szemléltethetjük, melynek pontjai a részfeladatok, és egy l hosszúságú (x, y) irányított él azt fejezi ki, hogy az y részfeladat nem kezdhető el korábban, mint az x kezdése után l idővel.

A kritikus út módszere (PERT-módszer)

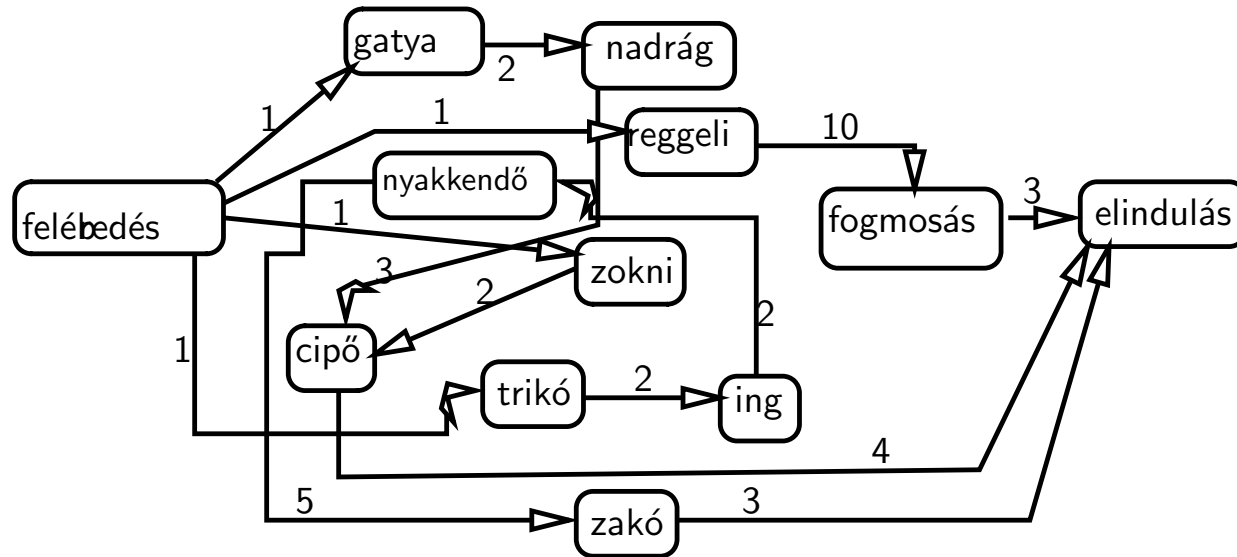
Az „emeletekre bontás” fontos alkalmazása az úgynevezett PERT-módszer. Az elnevezés az angol „Program Evaluation and Review Technique” rövidítéséből származik.

Tegyük fel, hogy egy összetett feladatot több alvállalkozóval kell elvégeztetni. Az egyes részfeladatok nem végezhetőek el egymástól függetlenül: pl. egy házépítés során a kőművesmunkák nyilván megelőzik a festési munkákat. A helyzetet egy G gráffal szemléltethetjük, melynek pontjai a részfeladatok, és egy l hosszúságú (x, y) irányított él azt fejezi ki, hogy az y részfeladat nem kezdhető el korábban, mint az x kezdése után l idővel. $l = 0$ is lehetséges: x és y ilyenkor kezdhető egyszerre, vagy y későbben.

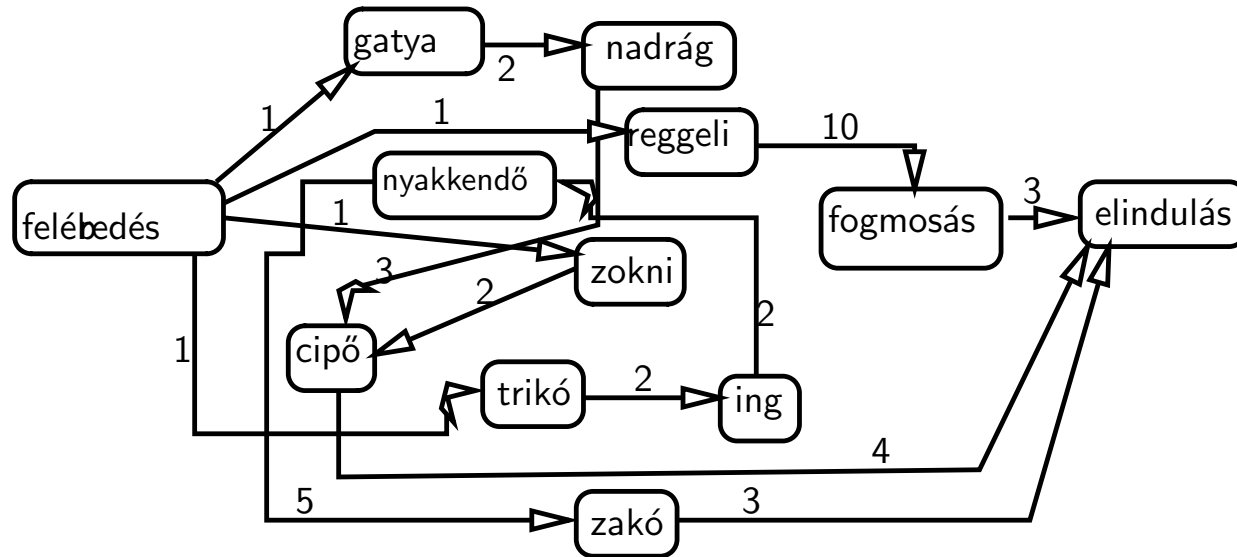


Mikor tud elindulni Kis Béla?

Mikor tud elindulni Kis Béla?

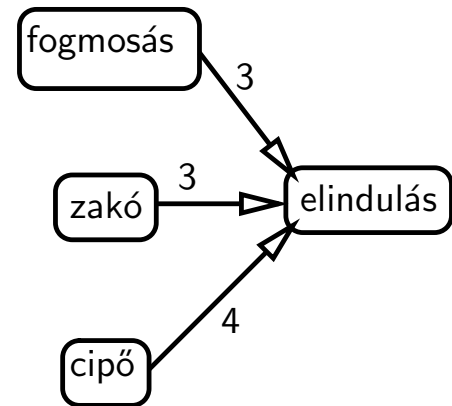
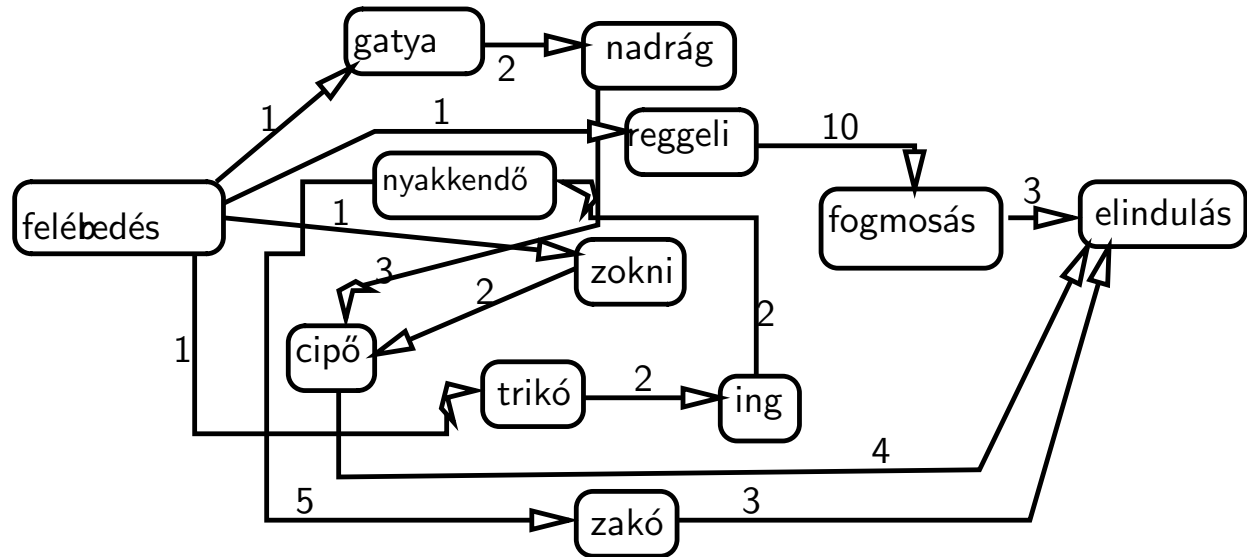


Mikor tud elindulni Kis Béla?

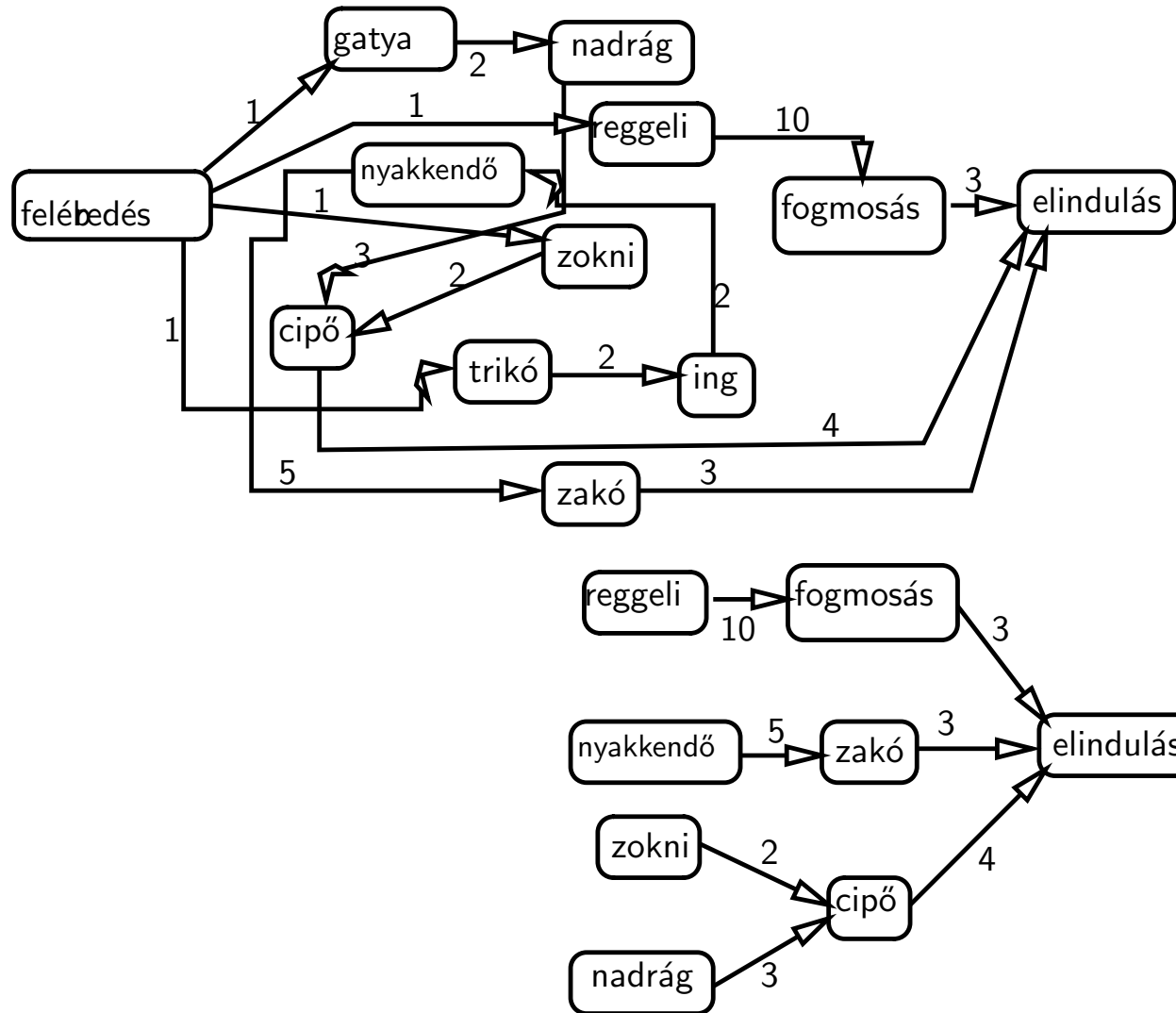


elindulás

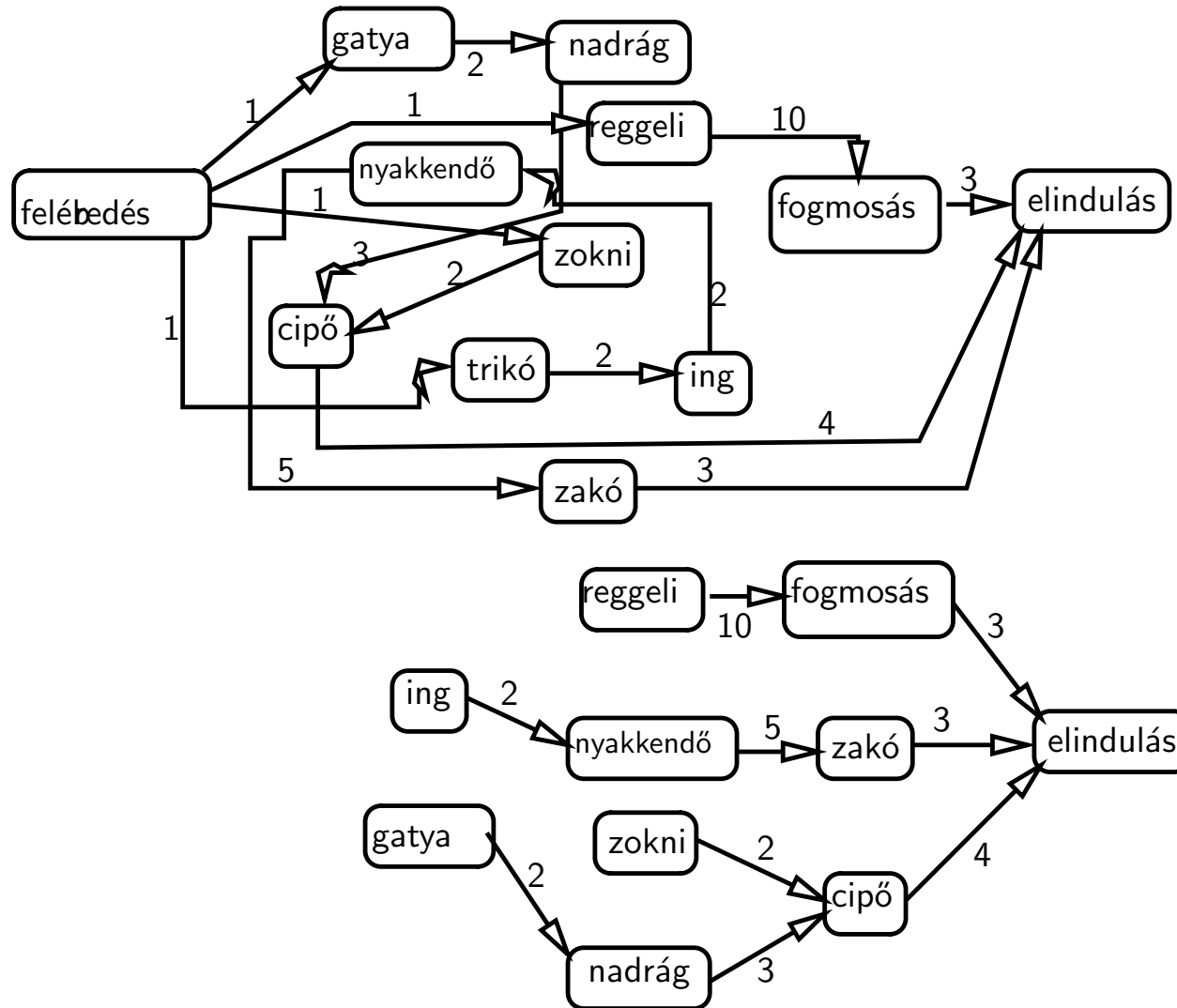
Mikor tud elindulni Kis Béla?



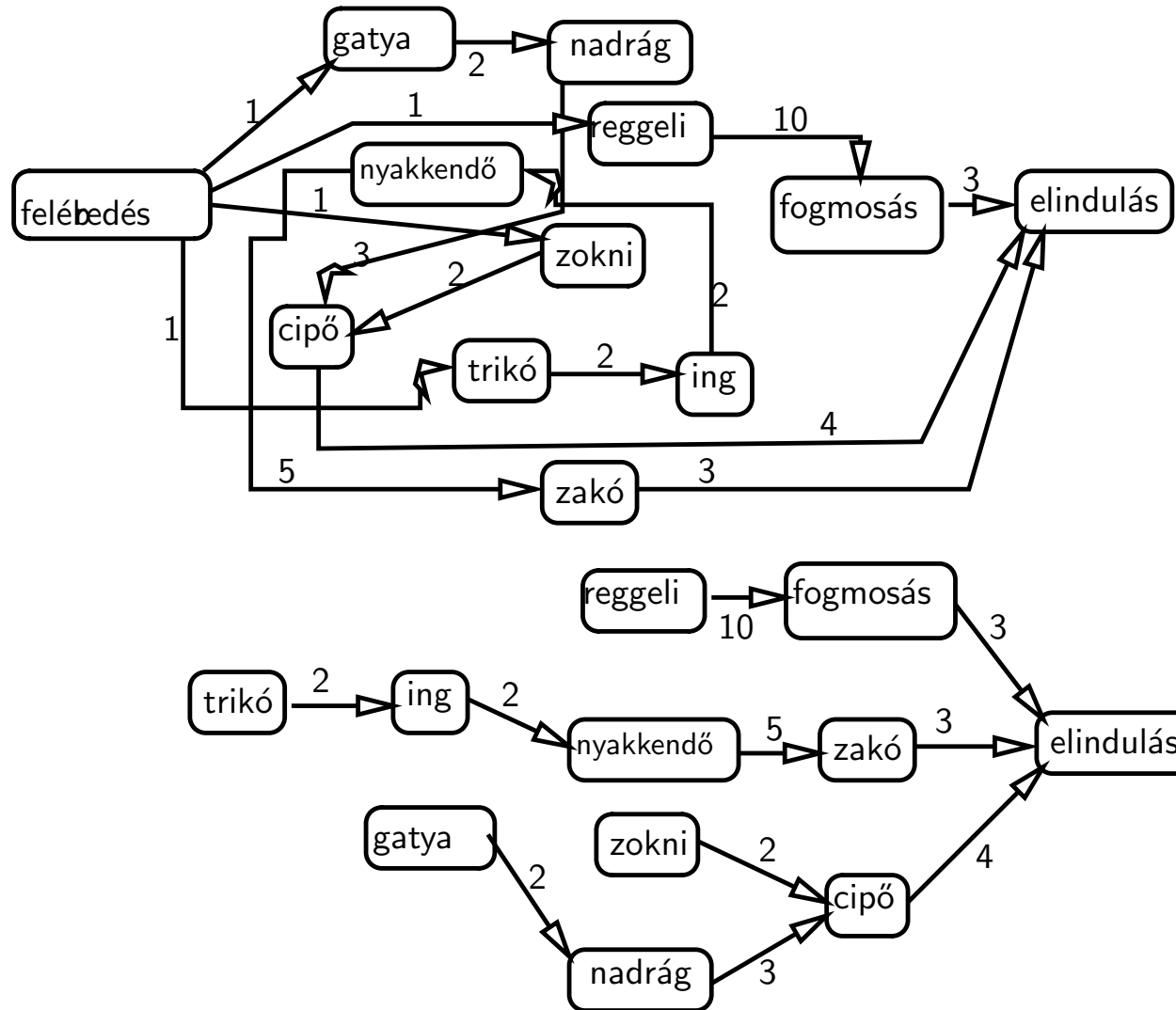
Mikor tud elindulni Kis Béla?



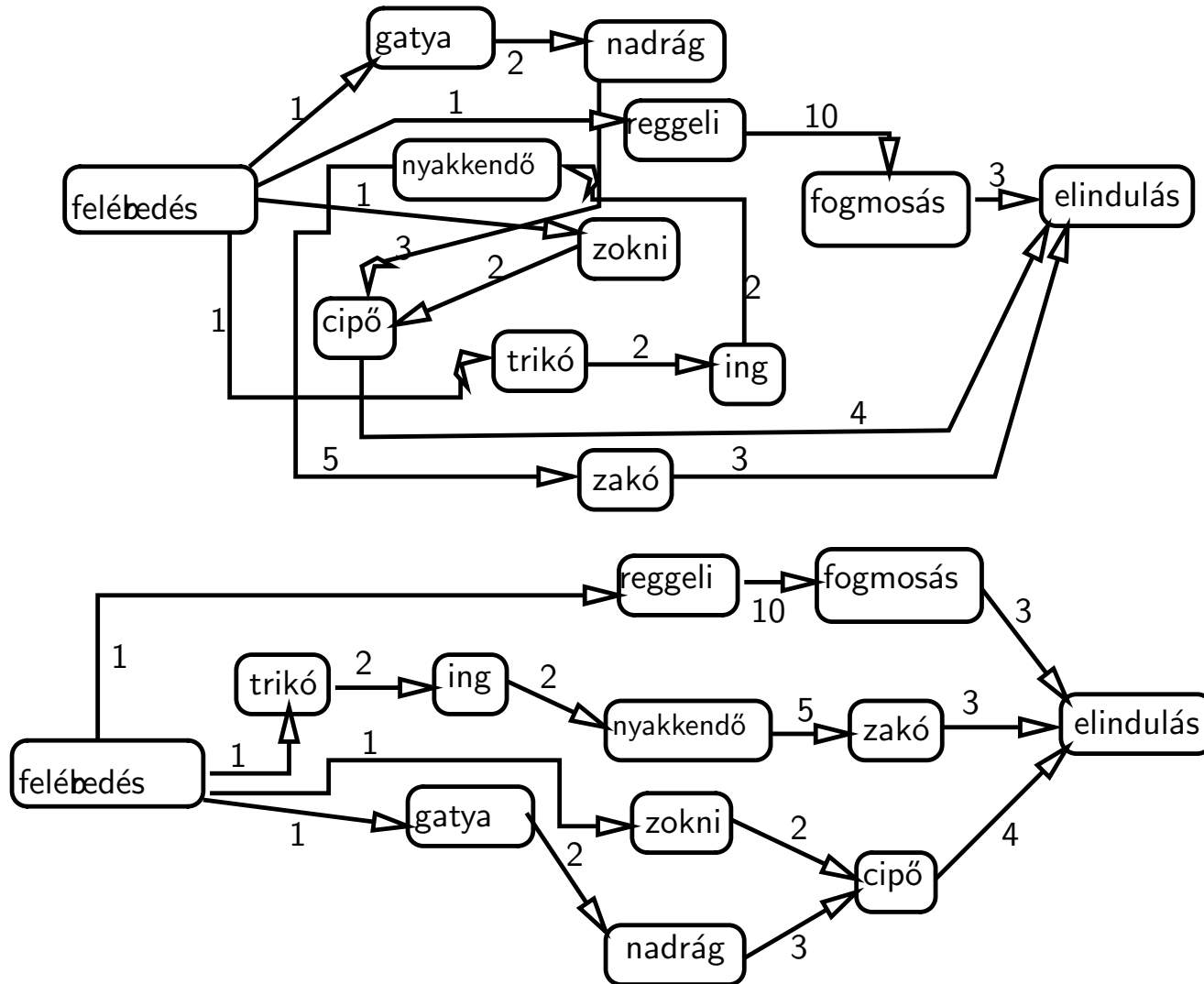
Mikor tud elindulni Kis Béla?



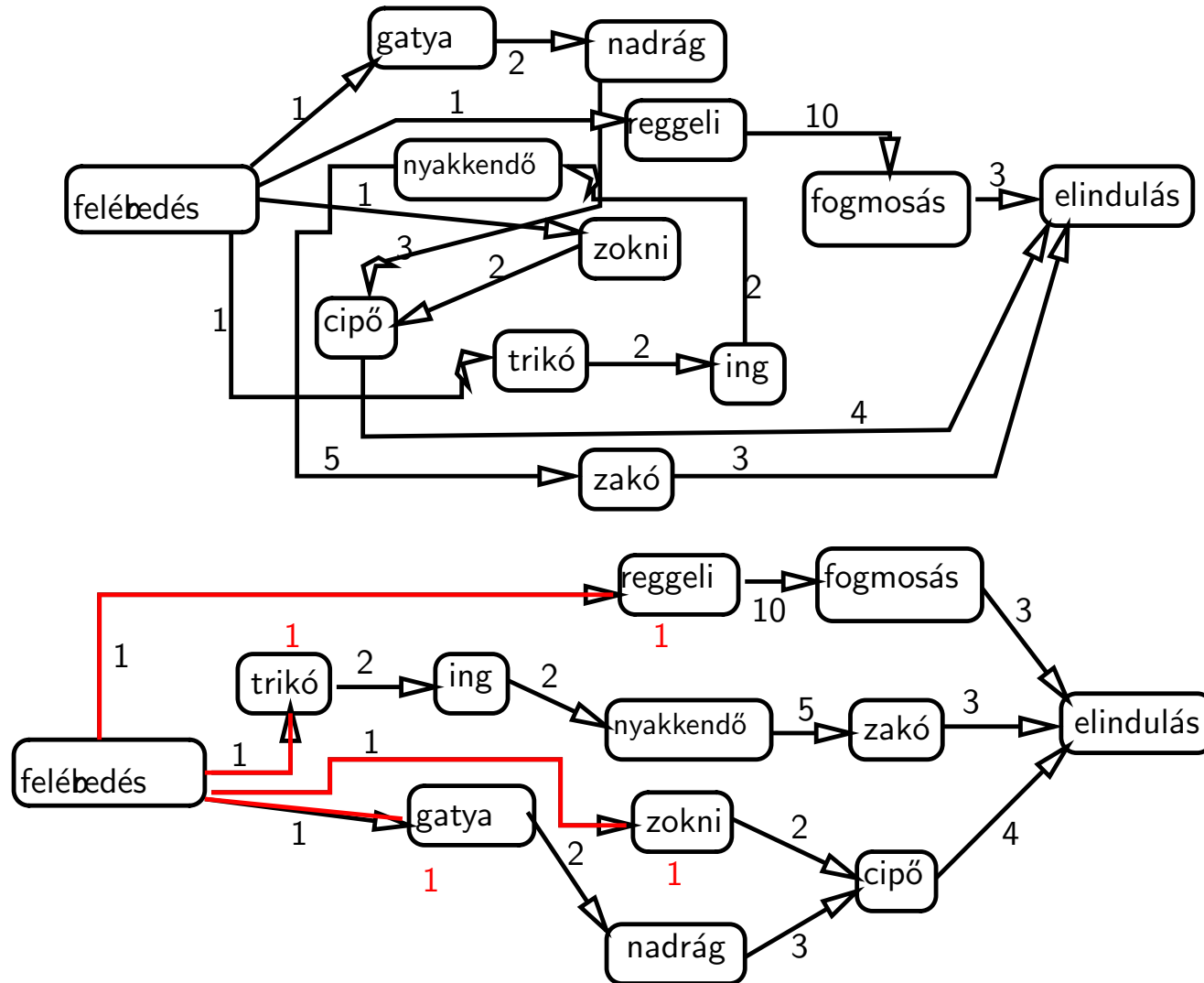
Mikor tud elindulni Kis Béla?



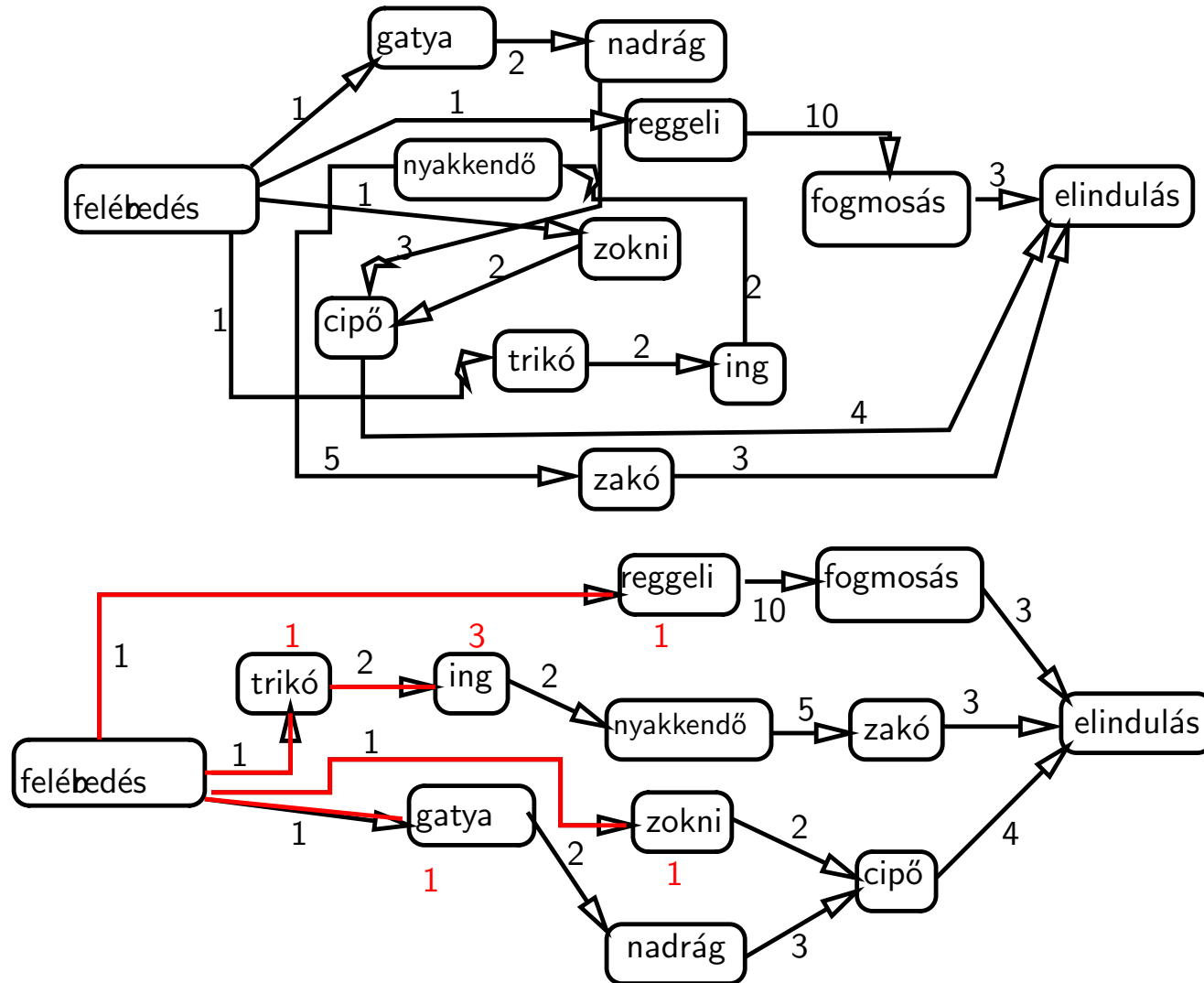
Mikor tud elindulni Kis Béla?



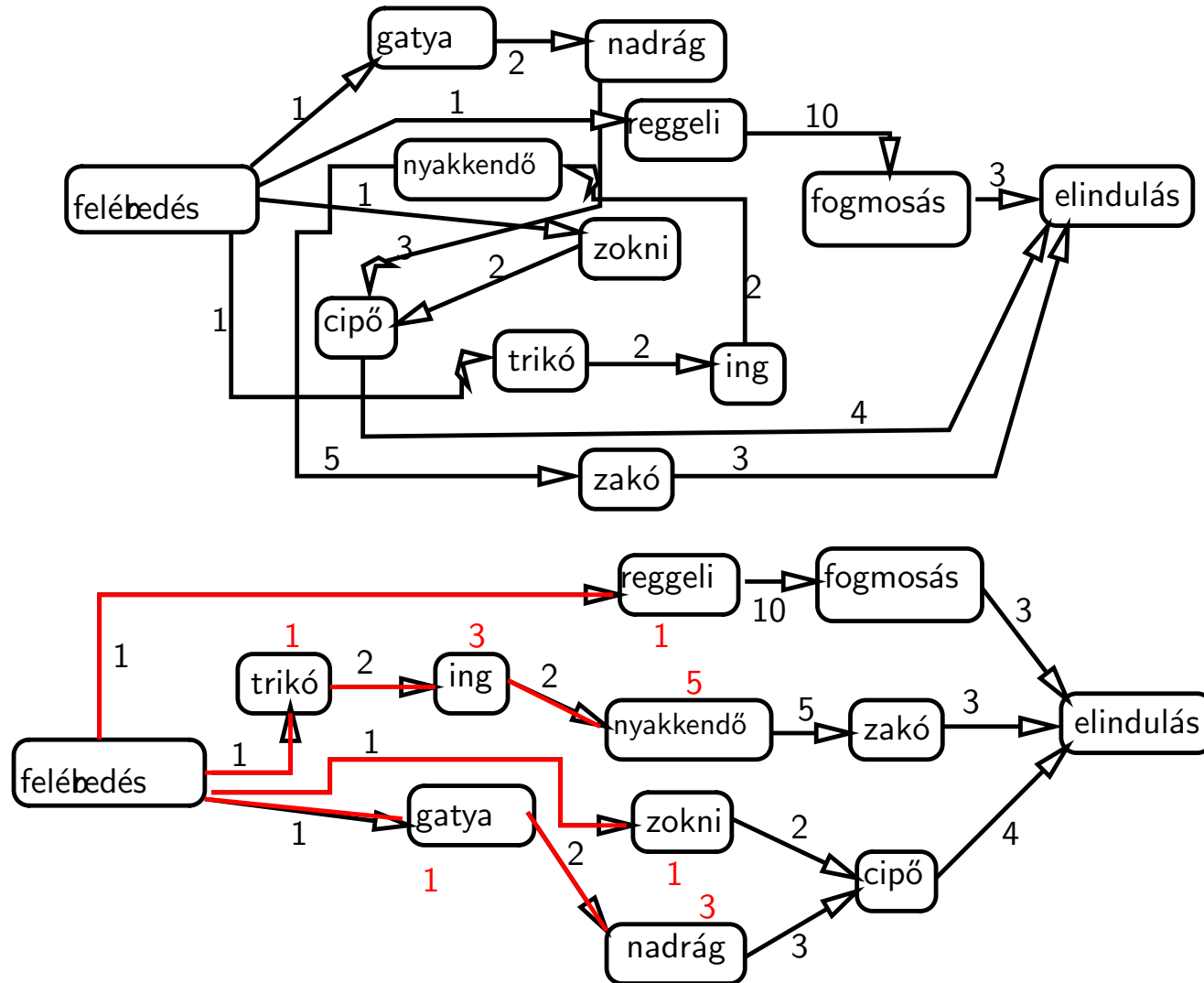
Mikor tud elindulni Kis Béla?



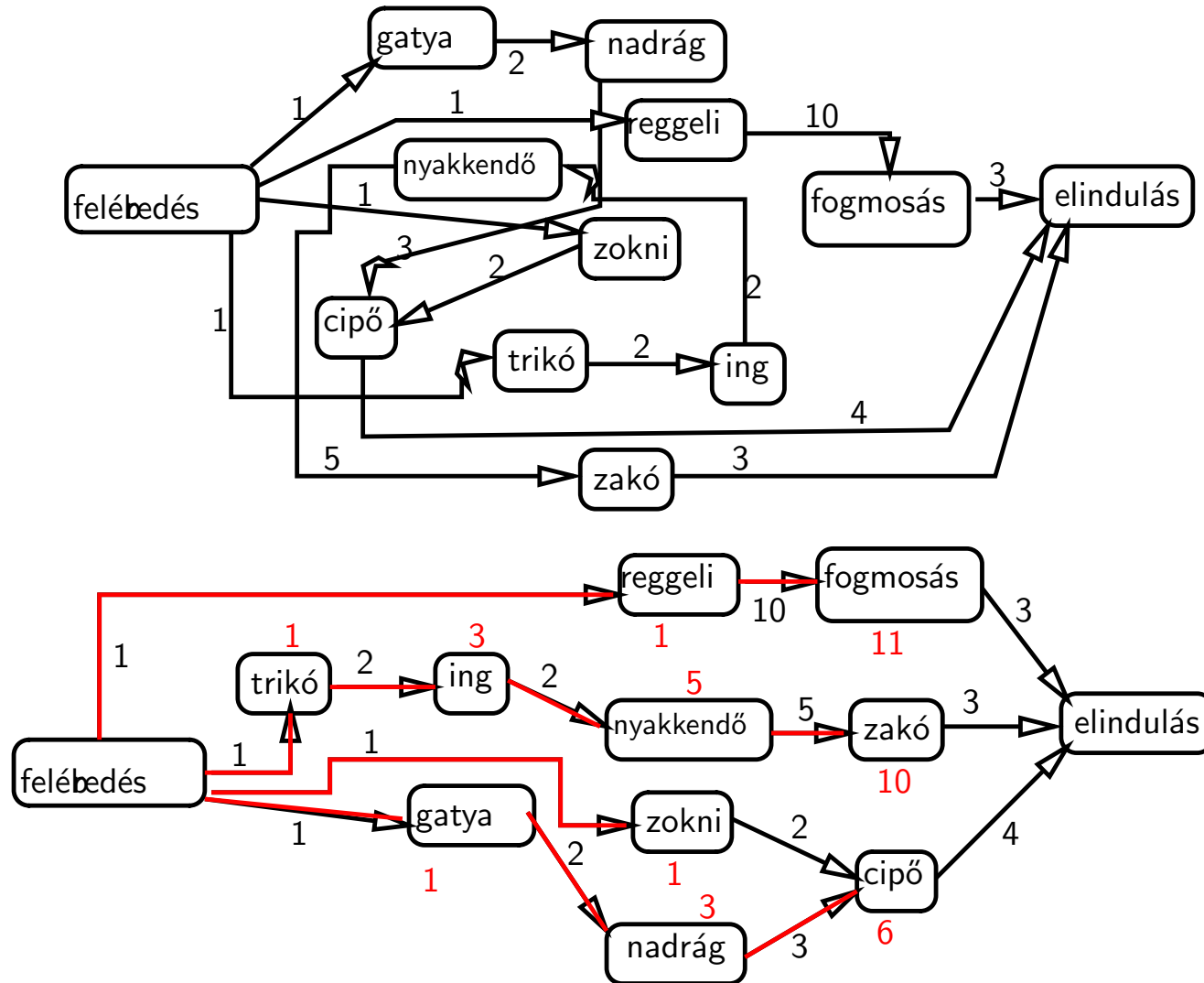
Mikor tud elindulni Kis Béla?



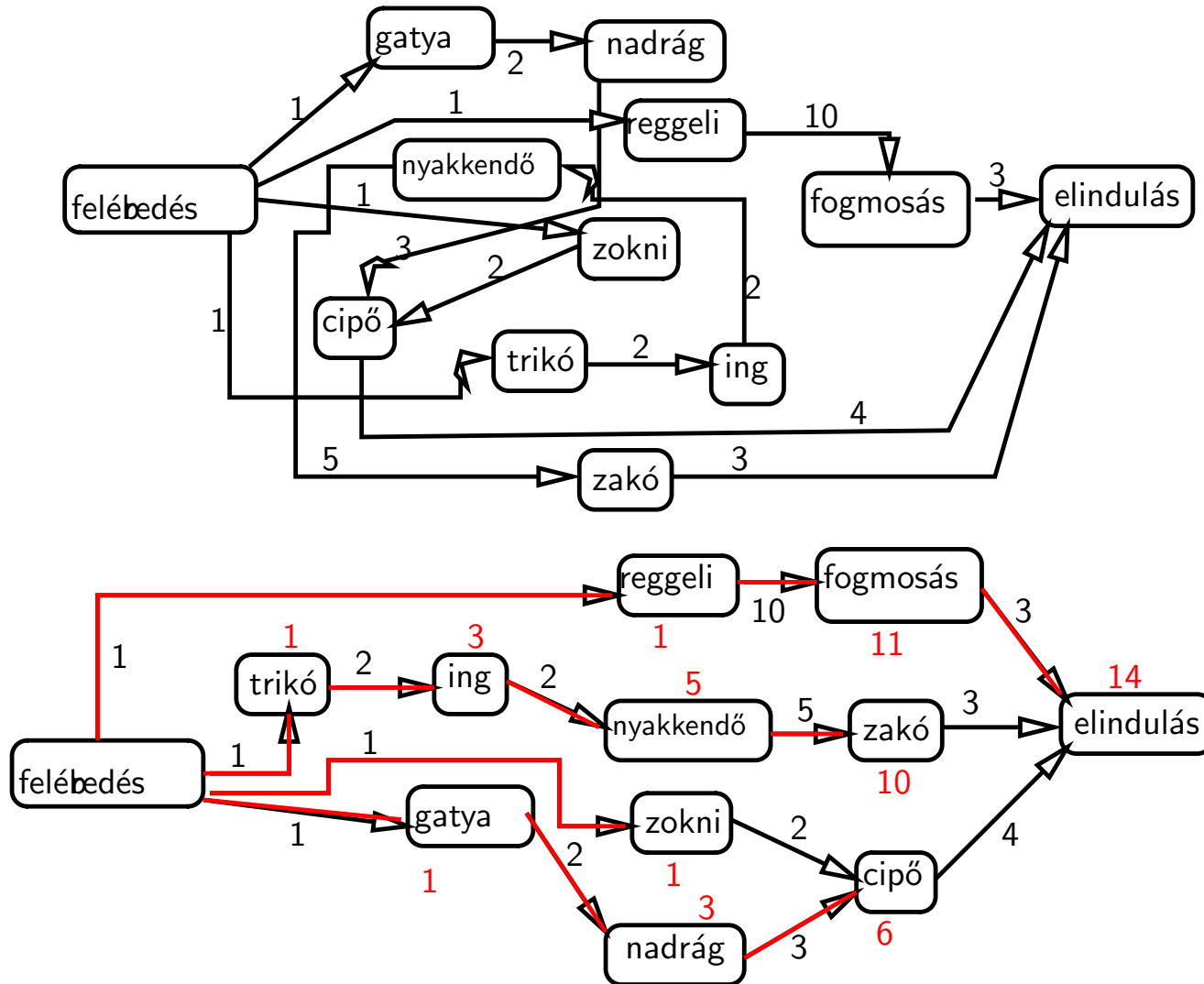
Mikor tud elindulni Kis Béla?



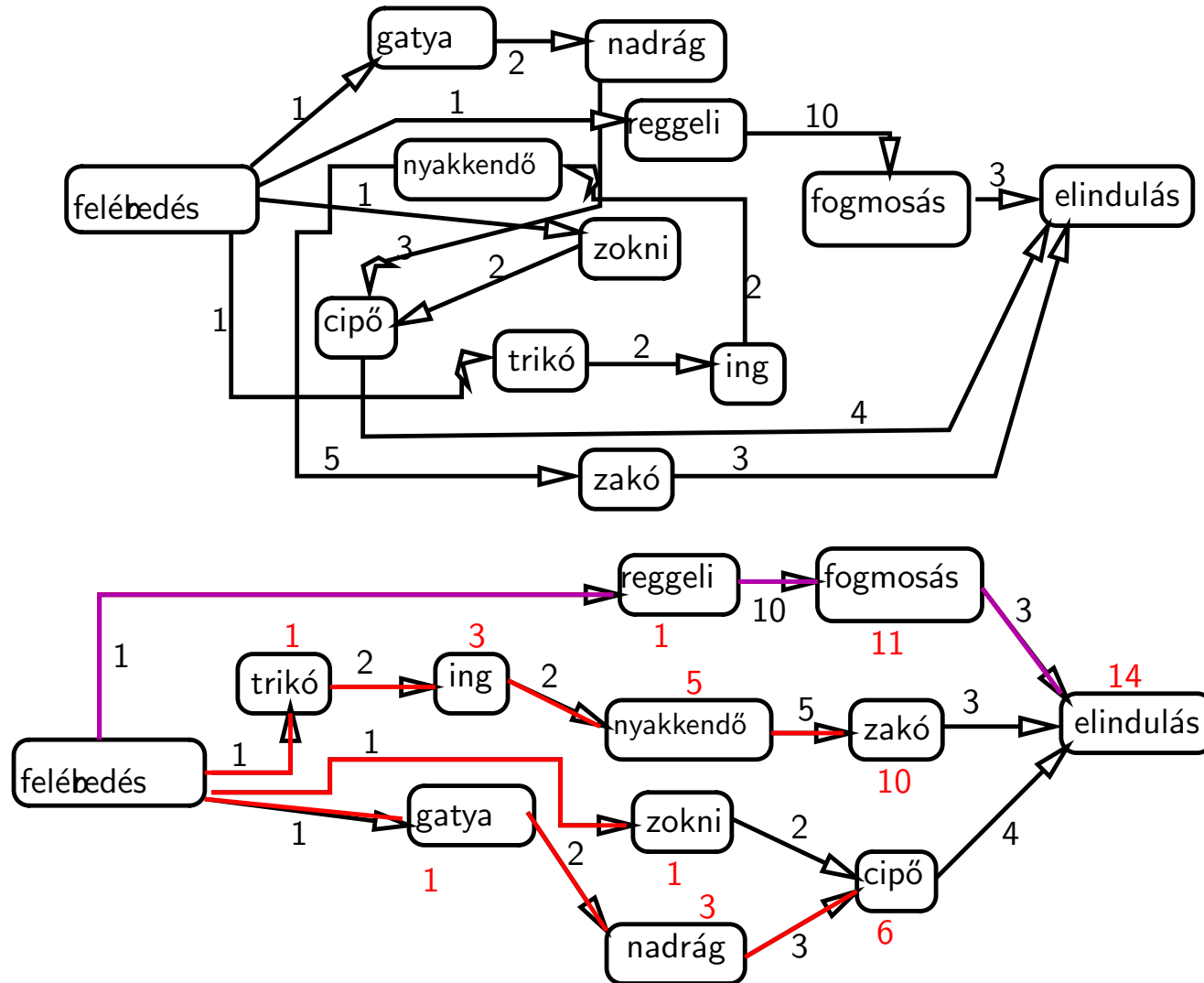
Mikor tud elindulni Kis Béla?



Mikor tud elindulni Kis Béla?



Mikor tud elindulni Kis Béla?



A feladat jellegénél fogva egy ilyen gráf nem tartalmazhat irányított kört.

A feladat jellegénél fogva egy ilyen gráf nem tartalmazhat irányított kört. Ha egy összehasonlítási gráfot irányítottan tekintünk (azaz $x \prec y \iff (x, y) \in E$), az ilyen.

A feladat jellegénél fogva egy ilyen gráf nem tartalmazhat irányított kört. Ha egy összehasonlítási gráfot irányítottan tekintünk (azaz $x \prec y \iff (x, y) \in E$), az ilyen. Megfordítva, ha egy irányított körmentes gráfba behúzzuk a *tranzitivitás*ból adódó éleket (képezzük a *tranzitív lezárt*át), akkor összehasonlítási gráfot kapunk.

A feladat jellegénél fogva egy ilyen gráf nem tartalmazhat irányított kört. Ha egy összehasonlítási gráfot irányítottan tekintünk (azaz $x \prec y \iff (x, y) \in E$), az ilyen. Megfordítva, ha egy irányított körmentes gráfba behúzzuk a *tranzitivitás*ból adódó éleket (képezzük a *tranzitív lezárt*át), akkor összehasonlítási gráfot kapunk.

A gráf emeletekre bontható:

A feladat jellegénél fogva egy ilyen gráf nem tartalmazhat irányított kört. Ha egy összehasonlítási gráfot irányítottan tekintünk (azaz $x \prec y \iff (x, y) \in E$), az ilyen. Megfordítva, ha egy irányított körmentes gráfba behúzzuk a *tranzitivtás*ból adódó éleket (képezzük a *tranzitív lezárt*át), akkor összehasonlítási gráfot kapunk.

A gráf emeletekre bontható: Először a nyelő(ke)t¹ helyezzük a jobbszélső halmazba,

¹ *Nyelő*: olyan csúcs melyből nem megy *ki* él

A feladat jellegénél fogva egy ilyen gráf nem tartalmazhat irányított kört. Ha egy összehasonlítási gráfot irányítottan tekintünk (azaz $x \prec y \iff (x, y) \in E$), az ilyen. Megfordítva, ha egy irányított körmentes gráfba behúzzuk a *tranzitivitás*ból adódó éleket (képezzük a *tranzitív lezárt*át), akkor összehasonlítási gráfot kapunk.

A gráf emeletekre bontható: Először a nyelő(ke)t¹ helyezzük a jobbszélső halmazba, ennek elhagyása után keletkező (és szintén irányított kört nem tartalmazó) gráf nyelőit a jobbról második halmazba és így tovább.

¹ *Nyelő*: olyan csúcs melyből nem megy *ki* él

A feladat jellegénél fogva egy ilyen gráf nem tartalmazhat irányított kört. Ha egy összehasonlítási gráfot irányítottan tekintünk (azaz $x \prec y \iff (x, y) \in E$), az ilyen. Megfordítva, ha egy irányított körmentes gráfba behúzzuk a **tranzitivitás**ból adódó éleket (képezzük a **tranzitív lezárt**át), akkor összehasonlítási gráfot kapunk.

A gráf emeletekre bontható: Először a nyelő(ke)t¹ helyezzük a jobbszélső halmazba, ennek elhagyása után keletkező (és szintén irányított kört nem tartalmazó) gráf nyelőit a jobbról második halmazba és így tovább.

Ezek után balról jobbra haladva, szintenként, meghatározhatjuk minden tevékenység elkezdésének lehetséges legkorábbi időpontját.

¹ **Nyelő**: olyan csúcs melyből nem megy **ki** él

A feladat jellegénél fogva egy ilyen gráf nem tartalmazhat irányított kört. Ha egy összehasonlítási gráfot irányítottan tekintünk (azaz $x \prec y \iff (x, y) \in E$), az ilyen. Megfordítva, ha egy irányított körmentes gráfba behúzzuk a **tranzitivitás**ból adódó éleket (képezzük a **tranzitív lezárt**át), akkor összehasonlítási gráfot kapunk.

A gráf emeletekre bontható: Először a nyelő(ke)t¹ helyezzük a jobbszélső halmazba, ennek elhagyása után keletkező (és szintén irányított kört nem tartalmazó) gráf nyelőit a jobbról második halmazba és így tovább.

Ezek után balról jobbra haladva, szintenként, meghatározhatjuk minden tevékenység elkezdésének lehetséges legkorábbi időpontját. A bal szélső tevékenység(ek) azonnal (0. időpontban) megkezdhető(ek),

¹ **Nyelő**: olyan csúcs melyből nem megy **ki** él

A feladat jellegénél fogva egy ilyen gráf nem tartalmazhat irányított kört. Ha egy összehasonlítási gráfot irányítottan tekintünk (azaz $x \prec y \iff (x, y) \in E$), az ilyen. Megfordítva, ha egy irányított körmentes gráfba behúzzuk a **tranzitivitás**ból adódó éleket (képezzük a **tranzitív lezárt**át), akkor összehasonlítási gráfot kapunk.

A gráf emeletekre bontható: Először a nyelő(ke)t¹ helyezzük a jobbszélső halmazba, ennek elhagyása után keletkező (és szintén irányított kört nem tartalmazó) gráf nyelőit a jobbról második halmazba és így tovább.

Ezek után balról jobbra haladva, szintenként, meghatározhatjuk minden tevékenység elkezdésének lehetséges legkorábbi időpontját. A bal szélső tevékenység(ek) azonnal (0. időpontban) megkezdhetők(ek), később egy y tevékenységhez tekintsük át az összes olyan x_1, x_2, \dots tevékenységet, melyre $(x_i, y) \in E(G)$, és ha ezek legkorábban a t_1, t_2, \dots időpontban kezdhetők el, akkor y elkezdésére legkorábban a

¹ **Nyelő**: olyan csúcs melyből nem megy **ki** él

A feladat jellegénél fogva egy ilyen gráf nem tartalmazhat irányított kört. Ha egy összehasonlítási gráfot irányítottan tekintünk (azaz $x \prec y \iff (x, y) \in E$), az ilyen. Megfordítva, ha egy irányított körmentes gráfba behúzzuk a **tranzitivitás**ból adódó éleket (képezzük a **tranzitív lezárt**át), akkor összehasonlítási gráfot kapunk.

A gráf emeletekre bontható: Először a nyelő(ke)t¹ helyezzük a jobbszélső halmazba, ennek elhagyása után keletkező (és szintén irányított kört nem tartalmazó) gráf nyelőit a jobbról második halmazba és így tovább.

Ezek után balról jobbra haladva, szintenként, meghatározhatjuk minden tevékenység elkezdésének lehetséges legkorábbi időpontját. A bal szélső tevékenység(ek) azonnal (0. időpontban) megkezdhetők(ek), később egy y tevékenységhez tekintsük át az összes olyan x_1, x_2, \dots tevékenységet, melyre $(x_i, y) \in E(G)$, és ha ezek legkorábban a t_1, t_2, \dots időpontban kezdhetők el, akkor y elkezdésére legkorábban a **$\max(t_1 + l(x_1, y), t_2 + l(x_2, y), \dots)$** időpontban kerülhet sor.

¹ **Nyelő**: olyan csúcs melyből nem megy **ki** él

Kritikus út

Kritikus út

Megjelöljük nyelő(k)ből visszafelé azokat az (x_i, y) éleket, melyeken a fenti maximumok felvételnek.

Kritikus út

Megjelöljük nyelő(k)ből visszafelé azokat az (x_i, y) éleket, melyeken a fenti maximumok felvételnek. A megjelölt élek a G gráf kritikus élei, az ezek által meghatározott részgráf mindig tartalmaz legalább egy irányított utat a forrásból a nyelőbe.

Kritikus út

Megjelöljük nyelő(k)ből visszafelé azokat az (x_i, y) éleket, melyeken a fenti maximumok felvételnek. A megjelölt élek a G gráf kritikus élei, az ezek által meghatározott részgráf mindig tartalmaz legalább egy irányított utat a forrásból a nyelőbe. Ezeket az utakat *kritikus út*nak nevezzük, nyilván ezek a leghosszabb utak a forrásból a nyelőbe.

Kritikus út

Megjelöljük nyelő(k)ből visszafelé azokat az (x_i, y) éleket, melyeken a fenti maximumok felvételnek. A megjelölt élek a G gráf kritikus élei, az ezek által meghatározott részgráf mindig tartalmaz legalább egy irányított utat a forrásból a nyelőbe. Ezeket az utakat *kritikus út*nak nevezzük, nyilván ezek a leghosszabb utak a forrásból a nyelőbe.

Az ilyen kritikus utakon lévő pontoknak megfelelő részfeladatok bármelyikének késedelmes elvégzése az egész összetett feladat befejezését késleltetné (innét a kritikus út elnevezés).

Kritikus út

Megjelöljük nyelő(k)ből visszafelé azokat az (x_i, y) éleket, melyeken a fenti maximumok felvételnek. A megjelölt élek a G gráf kritikus élei, az ezek által meghatározott részgráf mindig tartalmaz legalább egy irányított utat a forrásból a nyelőbe. Ezeket az utakat *kritikus út*nak nevezzük, nyilván ezek a leghosszabb utak a forrásból a nyelőbe.

Az ilyen kritikus utakon lévő pontoknak megfelelő részfeladatok bármelyikének késedelmes elvégzése az egész összetett feladat befejezését késleltetné (innét a kritikus út elnevezés). Ha viszont egy pont nincs kritikus úton, akkor a megfelelő feladat késedelmes elvégzése bizonyos határon belül még elfogadható.

Kritikus út

Megjelöljük nyelő(k)ből visszafelé azokat az (x_i, y) éleket, melyeken a fenti maximumok felvételnek. A megjelölt élek a G gráf kritikus élei, az ezek által meghatározott részgráf mindig tartalmaz legalább egy irányított utat a forrásból a nyelőbe. Ezeket az utakat *kritikus út*nak nevezzük, nyilván ezek a leghosszabb utak a forrásból a nyelőbe.

Az ilyen kritikus utakon lévő pontoknak megfelelő részfeladatok bármelyikének késedelmes elvégzése az egész összetett feladat befejezését késleltetné (innét a kritikus út elnevezés). Ha viszont egy pont nincs kritikus úton, akkor a megfelelő feladat késedelmes elvégzése bizonyos határon belül még elfogadható.

Bonyolultság

Bonyolultság

A részfeladatok „beprogramozásához” (vagyis a kezdési időpontok meghatározásához) szükséges lépések száma a G pontjainak fokszámösszegével (vagyis e -vel) arányos.

Bonyolultság

A részfeladatok „beprogramozásához” (vagyis a kezdési időpontok meghatározásához) szükséges lépések száma a G pontjainak fokszámösszegével (vagyis e -vel) arányos.

Ugyanis minden él hosszát pontosan egyszer vesszük figyelembe a maximumok számításakor.

Bonyolultság

A részfeladatok „beprogramozásához” (vagyis a kezdési időpontok meghatározásához) szükséges lépések száma a G pontjainak fokszámösszegével (vagyis e -vel) arányos.

Ugyanis minden él hosszát pontosan egyszer vesszük figyelembe a maximumok számításakor. A szintekre bontás is (alkalmas gráf tárolás esetén) fokszám összeggel arányos (minden élet figyelembe kell venni, amikor töröljük az egyik végpontját.)

Bonyolultság

A részfeladatok „beprogramozásához” (vagyis a kezdési időpontok meghatározásához) szükséges lépések száma a G pontjainak fokszámösszegével (vagyis e -vel) arányos.

Ugyanis minden él hosszát pontosan egyszer vesszük figyelembe a maximumok számításakor. A szintekre bontás is (alkalmas gráf tárolás esetén) fokszám összeggel arányos (minden élet figyelembe kell venni, amikor töröljük az egyik végpontját.)

A leghosszabb út meghatározása – ellentétben a legrövidebb útéval (lásd Algoritmelmélet) – általában nem végezhető el polinom időben. Ebben a speciális esetben azért tudtunk gyors algoritmust adni, mert G -ben nincsenek irányított körök.

Hogyan tároljunk gráfokat?

Hogyan tároljunk gráfokat?

Egy v pontú és e élű G gráf szomszédossági mátrixa v^2 , illeszkedési mátrixa ve helyet foglal el.

Hogyan tároljunk gráfokat?

Egy v pontú és e élű G gráf szomszédossági mátrixa v^2 , illeszkedési mátrixa ve helyet foglal el. Egyszerűs gráfokra $e \leq v(v-1)/2$ irányítatlan gráfok esetén és $e \leq v(v-1)$ irányított gráfok esetén.

Hogyan tároljunk gráfokat?

Egy v pontú és e élű G gráf szomszédossági mátrixa v^2 , illeszkedési mátrixa ve helyet foglal el. Egyszerűs gráfokra $e \leq v(v-1)/2$ irányítatlan gráfok esetén és $e \leq v(v-1)$ irányított gráfok esetén.

Az illeszkedési mátrix mindig feleslegesen sok helyet foglal el (hisz a ve szám között $(v-2)e$ darab zérus van);

Hogyan tároljunk gráfokat?

Egy v pontú és e élű G gráf szomszédossági mátrixa v^2 , illeszkedési mátrixa ve helyet foglal el. Egyszerűs gráfokra $e \leq v(v-1)/2$ irányítatlan gráfok esetén és $e \leq v(v-1)$ irányított gráfok esetén.

Az illeszkedési mátrix mindig feleslegesen sok helyet foglal el (hisz a ve szám között $(v-2)e$ darab zérus van); és ha egy gráf *ritka* (vagyis cv^2 -nél jóval kevesebb éle van), akkor a szomszédossági mátrixban is rengeteg a zérus.

Hogyan tároljunk gráfokat?

Egy v pontú és e élű G gráf szomszédossági mátrixa v^2 , illeszkedési mátrixa ve helyet foglal el. Egyszerűs gráfokra $e \leq v(v-1)/2$ irányítatlan gráfok esetén és $e \leq v(v-1)$ irányított gráfok esetén.

Az illeszkedési mátrix mindig feleslegesen sok helyet foglal el (hisz a ve szám között $(v-2)e$ darab zérus van); és ha egy gráf *ritka* (vagyis cv^2 -nél jóval kevesebb éle van), akkor a szomszédossági mátrixban is rengeteg a zérus. A feleslegesen nagy tárigény mellett a gráfelméleti algoritmusok lépésszámát (tehát időigényét) is növelné, ha a hasznos információkhoz csak számos felesleges zérus kiolvasásán keresztül jutnánk.

Hogyan tároljunk gráfokat?

Egy v pontú és e élű G gráf szomszédossági mátrixa v^2 , illeszkedési mátrixa ve helyet foglal el. Egyszerűs gráfokra $e \leq v(v-1)/2$ irányítatlan gráfok esetén és $e \leq v(v-1)$ irányított gráfok esetén.

Az illeszkedési mátrix mindig feleslegesen sok helyet foglal el (hisz a ve szám között $(v-2)e$ darab zérus van); és ha egy gráf *ritka* (vagyis cv^2 -nél jóval kevesebb éle van), akkor a szomszédossági mátrixban is rengeteg a zérus. A feleslegesen nagy tárigény mellett a gráfelméleti algoritmusok lépésszámát (tehát időigényét) is növelné, ha a hasznos információkhoz csak számos felesleges zérus kiolvasásán keresztül jutnánk.

Például egy csúcsról eldönteni hogy nyelő, szomszédossági mátrix esetén annak egy teljes oszlopát ($|v|$) adatot kell kiolvasni.

Szomszédossági tömbök és listák

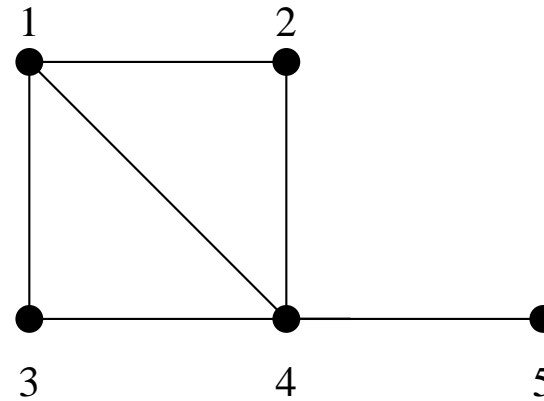
Szomszédossági tömbök és listák

Gyakran hasznos a gráfot úgy tárolni, hogy minden pontjához felsoroljuk a szomszédjait.

Szomszédossági tömbök és listák

Gyakran hasznos a gráfot úgy tárolni, hogy minden pontjához felsoroljuk a szomszédjait.

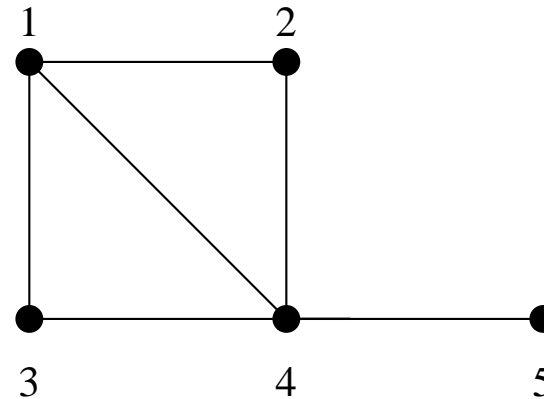
2 3 4
1 4
1 4
1 2 3 5
4



Szomszédossági tömbök és listák

Gyakran hasznos a gráfot úgy tárolni, hogy minden pontjához felsoroljuk a szomszédjait.

2 3 4
1 4
1 4
1 2 3 5
4

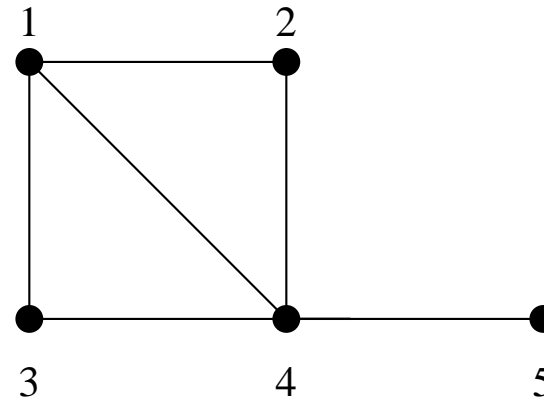


A szomszédossági listák általában különböző hosszúságúak,

Szomszédossági tömbök és listák

Gyakran hasznos a gráfot úgy tárolni, hogy minden pontjához felsoroljuk a szomszédjait.

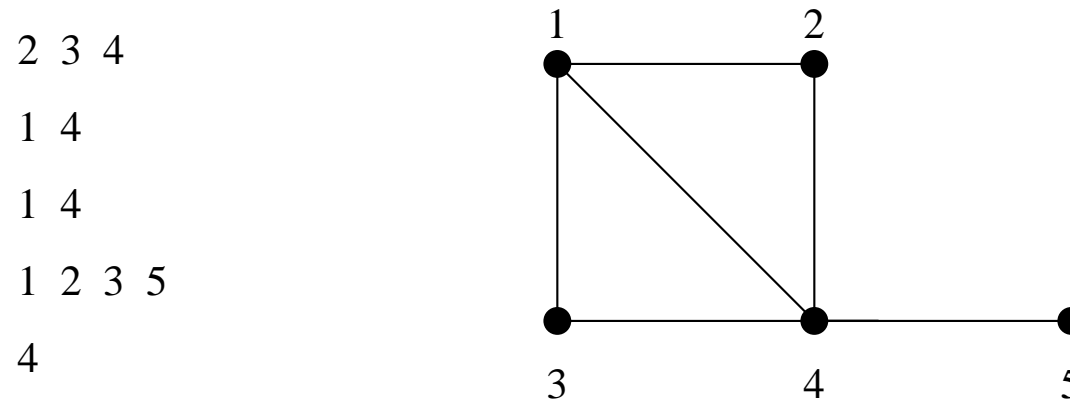
2 3 4
1 4
1 4
1 2 3 5
4



A szomszédossági listák általában különböző hosszúságúak, \implies érdemes őket egy nagy közös tömbben tárolni, és egy külön tömbben tárolni a „mutatókat” (pointer), hogy honnét kezdve kell olvasni egy adott pont szomszédait.

Szomszédossági tömbök és listák

Gyakran hasznos a gráfot úgy tárolni, hogy minden pontjához felsoroljuk a szomszédjait.

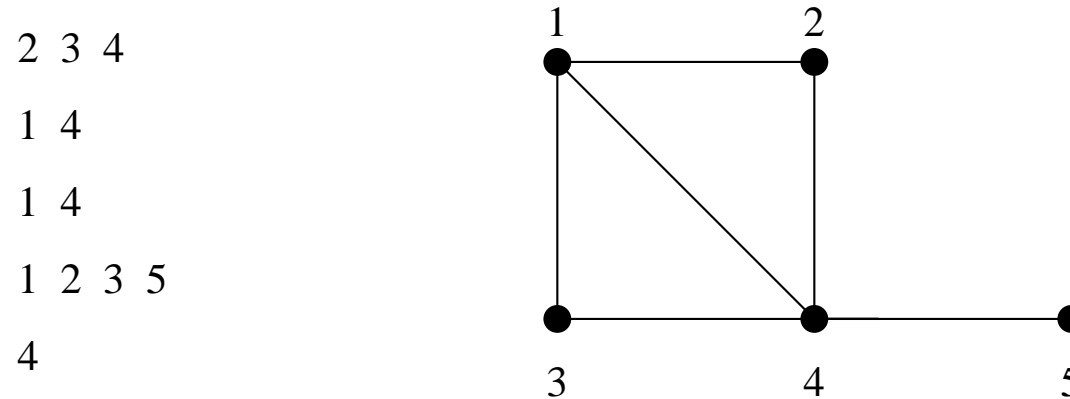


A szomszédossági listák általában különböző hosszúságúak, \implies érdemes őket egy nagy közös tömbben tárolni, és egy külön tömbben tárolni a „mutatókat” (pointer), hogy honnét kezdve kell olvasni egy adott pont szomszédait.

2 3 4 1 4 1 4 1 2 3 5 4 és 1 4 6 8 12

Szomszédossági tömbök és listák

Gyakran hasznos a gráfot úgy tárolni, hogy minden pontjához felsoroljuk a szomszédjait.



A szomszédossági listák általában különböző hosszúságúak, \implies érdemes őket egy nagy közös tömbben tárolni, és egy külön tömbben tárolni a „mutatókat” (pointer), hogy honnét kezdve kell olvasni egy adott pont szomszédait.

2 3 4 1 4 1 4 1 2 3 5 4 és 1 4 6 8 12

szomszédossági tömb

Az 1-1 ponthoz tartozó listák külön-külön lehetnek rendezettek, így hamarabb ellenőrizhetjük, hogy egy pont szomszédai között szerepel-e egy adott másik pont.

Az 1-1 ponthoz tartozó listák külön-külön lehetnek rendezettek, így hamarabb ellenőrizhetjük, hogy egy pont szomszédai között szerepel-e egy adott másik pont.

Az első tömb hossza a fokszámok összege, vagyis $2e$, a második tömbé pedig v . Így a teljes tárigény $2e + v$.

Az 1-1 ponthoz tartozó listák külön-külön lehetnek rendezettek, így hamarabb ellenőrizhetjük, hogy egy pont szomszédai között szerepel-e egy adott másik pont.

Az első tömb hossza a fokszámok összege, vagyis $2e$, a második tömbé pedig v . Így a teljes tárigény $2e + v$. Ez az elképzelhető minimális tárigénynek közel kétszerese (az $\{i, j\}$ élt i szomszédainál is, j szomszédainál is felsoroljuk). (megtérül!)

Az 1-1 ponthoz tartozó listák külön-külön lehetnek rendezettek, így hamarabb ellenőrizhetjük, hogy egy pont szomszédai között szerepel-e egy adott másik pont.

Az első tömb hossza a fokszámok összege, vagyis $2e$, a második tömbé pedig v . Így a teljes tárigény $2e + v$. Ez az elképzelhető minimális tárigénynek közel kétszerese (az $\{i, j\}$ élt i szomszédainál is, j szomszédainál is felsoroljuk). (megtérül!)

Irányított gráfok esetén minden i ponthoz felsoroljuk azokat a j pontokat, melyekbe (i, j) irányított él vezet i -ből;

Az 1-1 ponthoz tartozó listák külön-külön lehetnek rendezettek, így hamarabb ellenőrizhetjük, hogy egy pont szomszédai között szerepel-e egy adott másik pont.

Az első tömb hossza a fokszámok összege, vagyis $2e$, a második tömbé pedig v . Így a teljes tárigény $2e + v$. Ez az elképzelhető minimális tárigénynek közel kétszerese (az $\{i, j\}$ élt i szomszédainál is, j szomszédainál is felsoroljuk). (megtérül!)

Írányított gráfok esetén minden i ponthoz felsoroljuk azokat a j pontokat, melyekbe (i, j) irányított él vezet i -ből; vagy azokat a k pontokat, melyekből (k, i) irányított él vezet i -be.

Az 1-1 ponthoz tartozó listák külön-külön lehetnek rendezettek, így hamarabb ellenőrizhetjük, hogy egy pont szomszédai között szerepel-e egy adott másik pont.

Az első tömb hossza a fokszámok összege, vagyis $2e$, a második tömbé pedig v . Így a teljes tárigény $2e + v$. Ez az elképzelhető minimális tárigénynek közel kétszerese (az $\{i, j\}$ élt i szomszédainál is, j szomszédainál is felsoroljuk). (megtérül!)

Irányított gráfok esetén minden i ponthoz felsoroljuk azokat a j pontokat, melyekbe (i, j) irányított él vezet i -ből; vagy azokat a k pontokat, melyekből (k, i) irányított él vezet i -be. Sok esetben az a legjobb, ha mindkét listát megadjuk: a kétszeres tárigény számos algoritmusnál nagyságrenddel csökkenti a lépésszám-igényt.

Az 1-1 ponthoz tartozó listák külön-külön lehetnek rendezettek, így hamarabb ellenőrizhetjük, hogy egy pont szomszédai között szerepel-e egy adott másik pont.

Az első tömb hossza a fokszámok összege, vagyis $2e$, a második tömbé pedig v . Így a teljes tárigény $2e + v$. Ez az elképzelhető minimális tárigénynek közel kétszerese (az $\{i, j\}$ élt i szomszédainál is, j szomszédainál is felsoroljuk). (megtérül!)

Írányított gráfok esetén minden i ponthoz felsoroljuk azokat a j pontokat, melyekbe (i, j) irányított él vezet i -ből; vagy azokat a k pontokat, melyekből (k, i) irányított él vezet i -be. Sok esetben az a legjobb, ha mindkét listát megadjuk: a kétszeres tárigény számos algoritmusnál nagyságrenddel csökkenti a lépésszám-igényt.

*Rendezett szomszédossági tömb*ről beszélünk, ha az egyes pontok szomszédai már növekvő sorrendben elhelyezve kerülnek tárolásra (a példában is ez volt a helyzet). Világos, hogy a rendezések elvégzése további időt igényel, de ez később megtérülhet.

Láncolt szomszédossági listák

Láncolt szomszédossági listák

Ha egy olyan algoritmus (probléma) adódik, amikor gyakran kell a gráfból egy élt (vagy akár pontot) elhagyni, akkor a szomszédsági tömb nem megfelelő:

Láncolt szomszédossági listák

Ha egy olyan algoritmus (probléma) adódik, amikor gyakran kell a gráfból egy élt (vagy akár pontot) elhagyni, akkor a szomszédsági tömb nem megfelelő: a beszúrandó vagy elhagyandó elem után következők eggyel eltolása akár n darab további lépést is igényelhet.

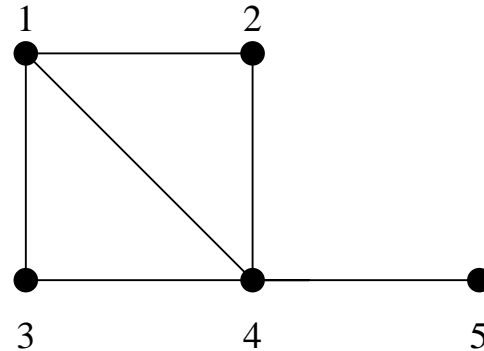
Láncolt szomszédossági listák

Ha egy olyan algoritmus (probléma) adódik, amikor gyakran kell a gráfból egy élt (vagy akár pontot) elhagyni, akkor a szomszédsági tömb nem megfelelő: a beszúrandó vagy elhagyandó elem után következők eggyel eltolása akár n darab további lépést is igényelhet. \implies Olyan listát adunk, aminek első tömbje a szomszédossági lista elemeit tetszőleges sorrendben tartalmazhatja.

Láncolt szomszédossági listák

Ha egy olyan algoritmus (probléma) adódik, amikor gyakran kell a gráfból egy élt (vagy akár pontot) elhagyni, akkor a szomszédsági tömb nem megfelelő: a beszúrandó vagy elhagyandó elem után következők eggyel eltolása akár n darab további lépést is igényelhet. \implies Olyan listát adunk, aminek első tömbje a szomszédossági lista elemeit tetszőleges sorrendben tartalmazhatja.

2 3 4
1 4
1 4
1 2 3 5
4



Láncolt szomszédossági listák

Ha egy olyan algoritmus (probléma) adódik, amikor gyakran kell a gráfból egy élt (vagy akár pontot) elhagyni, akkor a szomszédsági tömb nem megfelelő: a beszúrandó vagy elhagyandó elem után következők eggyel eltolása akár n darab további lépést is igényelhet. \implies Olyan listát adunk, aminek első tömbje a szomszédossági lista elemeit tetszőleges sorrendben tartalmazhatja.

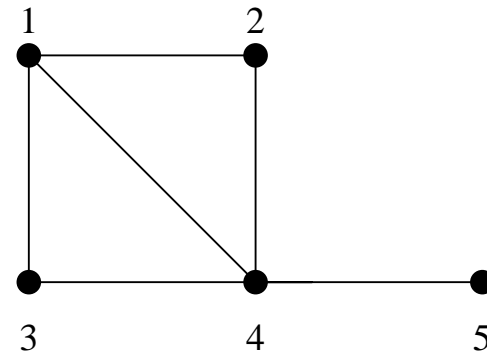
2 3 4

1 4

1 4

1 2 3 5

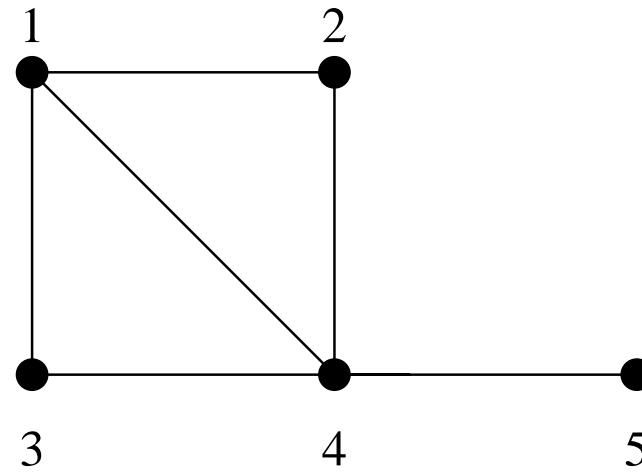
4



Most két darab, egyenként $2e$ hosszú és változatlanul egy darab v hosszú tömb kell:

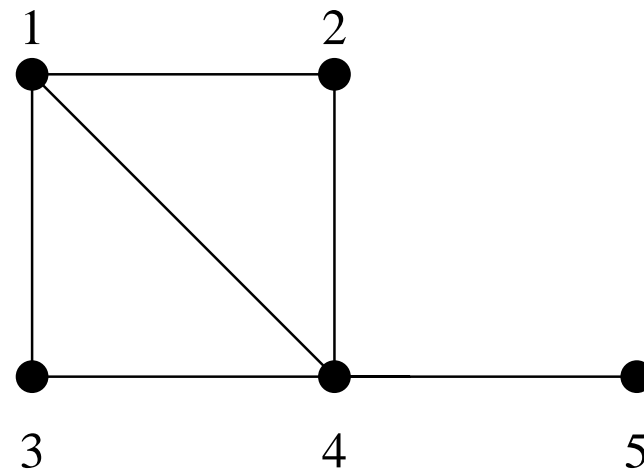
2	1	1	3	2	1	4	4	5	3	4	4		1	2	6	3	11
4	12	5	7	10	8	*	*	*	9	*	*						

2 3 4
 1 4
 1 4
 1 2 3 5
 4



2 1 1 3 2 1 4 4 5 3 4 4 1 2 6 3 11
 4 12 5 7 10 8 * * * 9 * *

2 3 4
 1 4
 1 4
 1 2 3 5
 4



2	1	1	3	2	1	4	4	5	3	4	4		1	2	6	3	11
4	12	5	7	10	8	*	*	*	9	*	*						

Most is a v hosszú tömb i -ik eleme mutatja meg, hogy hol kezdjük el az i -ik pont szomszédainak kiolvasását az első $2e$ hosszú tömbből. Azt azonban az alatta lévő szám (tehát a második $2e$ hosszú tömb megfelelő eleme) mutatja meg, hogy hol folytassuk az olvasást, illetve egy speciális $*$ szimbólum jelzi, hogy vége van a listának.

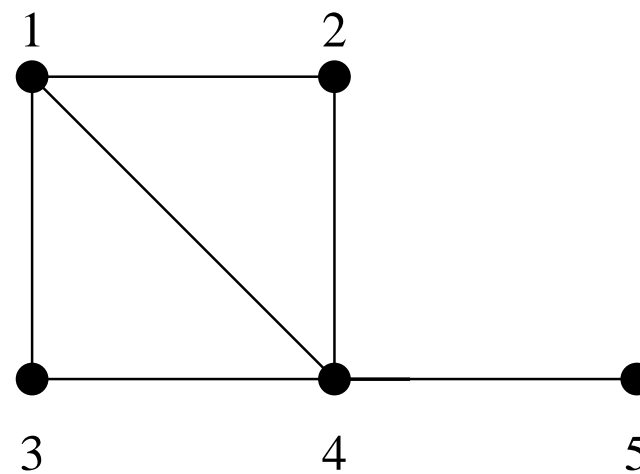
2 3 4

1 4

1 4

1 2 3 5

4



$\{1,4\}$ él elhagyása után:

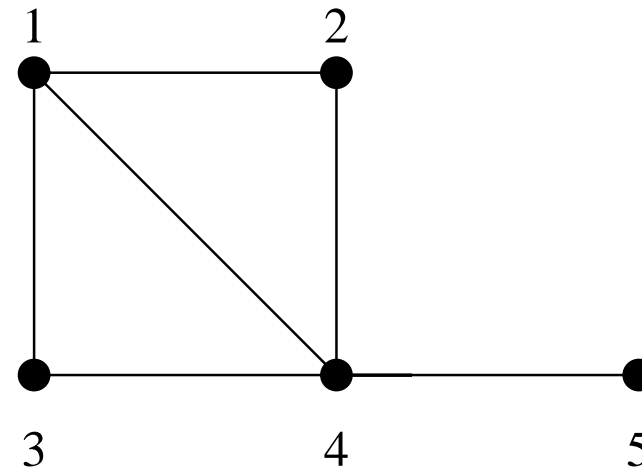
2 3 4

1 4

1 4

1 2 3 5

4



$\{1,4\}$ él elhagyása után:

2	1	1	3	2	1	4	4	5	3	4	4		1	2	6	5	11
4	12	*	*	10	8	*	*	*	9	*	*						

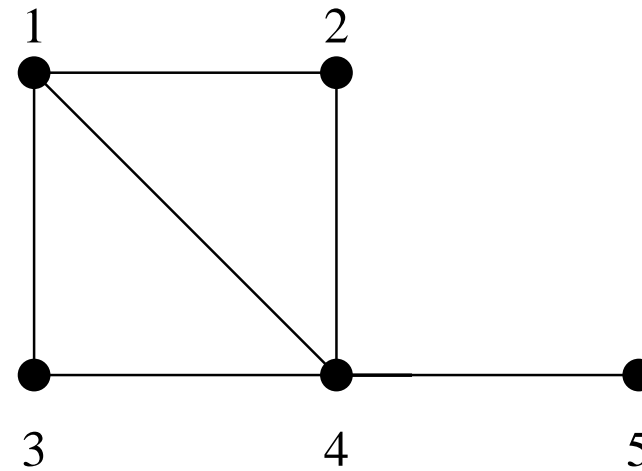
2 3 4

1 4

1 4

1 2 3 5

4

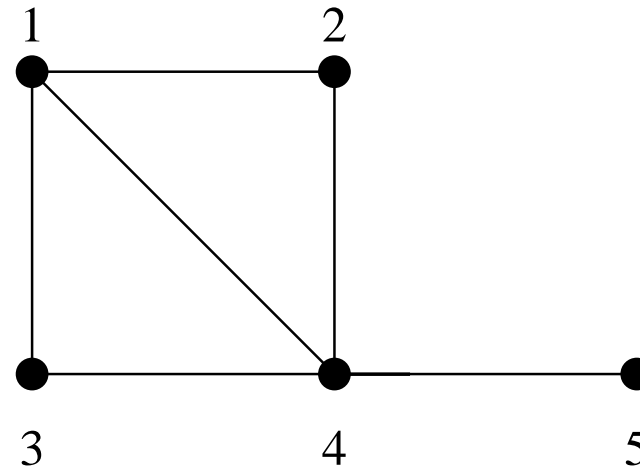


$\{1,4\}$ él elhagyása után:

2	1	1	3	2	1	4	4	5	3	4	4		1	2	6	5	11
4	12	*	*	10	8	*	*	*	9	*	*						

$\{2,5\}$ él bevétele után:

2 3 4
 1 4
 1 4
 1 2 3 5
 4



{1,4} él elhagyása után:

2	1	1	3	2	1	4	4	5	3	4	4		1	2	6	5	11
4	12	*	*	10	8	*	*	*	9	*	*						

{2,5} él bevétele után:

2	1	1	3	2	1	4	4	5	3	4	4	5	2		1	2	6	3	11
4	12	5	7	10	8	*	*	*	9	14	13	*	*						

1. táblázat. Tárigény és a különféle gráfelméleti műveletek időigénye, ha a gráfot szomszédossági mátrixszal (**A**), szomszédossági tömbbel (**B**), rendezett szomszédossági tömbbel (**C**) vagy láncolt szomszédossági listával (**D**) adjuk meg.

	A	B	C	D
Tárigény	v^2	$2e + v$	$2e + v$	$4e + v$
Két pont szomszédosságának eldöntése	1	d	$\log d$	d
Pont szomszédainak megjelölése	v	d	d	d
Minden él megjelölése	v^2	e	e	e
Új él hozzávétele	1	e	e	1
Régi él elvétele	1	e	e	d
Régi pont elvétele	v	e	e	$\min(e, d^2)$

jelölések: v pontszám, e élszám, d maximális fokszám

Mantel tétele

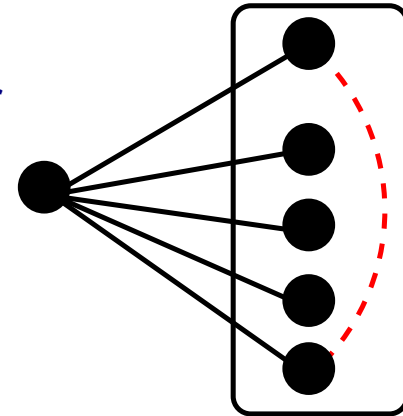
Legfeljebb hány éle lehet egy n -csúcsú egyszerű gráfnak, ha nem tartalmaz háromszöget?

Mantel tétele

Legfeljebb hány éle lehet egy n -csúcsú egyszerű gráfnak, ha nem tartalmaz háromszöget?

1. Tétel. *Ha egy n -csúcsú egyszerű gráf nem tartalmaz háromszöget, akkor éleinek száma legfeljebb $\lceil \frac{n}{2} \rceil \cdot \lfloor \frac{n}{2} \rfloor$.*

BIZONYÍTÁS Ha G nem tartalmaz háromszöget, akkor $\alpha(G) \geq \Delta(G)$.

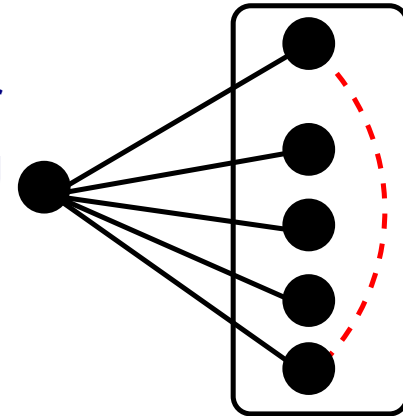


Mantel tétele

Legfeljebb hány éle lehet egy n -csúcsú egyszerű gráfnak, ha nem tartalmaz háromszöget?

1. Tétel. *Ha egy n -csúcsú egyszerű gráf nem tartalmaz háromszöget, akkor éleinek száma legfeljebb $\lceil \frac{n}{2} \rceil \cdot \lfloor \frac{n}{2} \rfloor$.*

BIZONYÍTÁS Ha G nem tartalmaz háromszöget, akkor $\alpha(G) \geq \Delta(G)$. Ugyanakkor $|E(G)| \leq \tau(G) \cdot \Delta(G)$ mindig igaz.

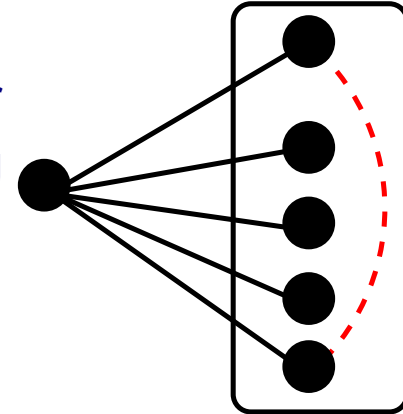


Mantel tétele

Legfeljebb hány éle lehet egy n -csúcsú egyszerű gráfnak, ha nem tartalmaz háromszöget?

1. Tétel. *Ha egy n -csúcsú egyszerű gráf nem tartalmaz háromszöget, akkor éleinek száma legfeljebb $\lceil \frac{n}{2} \rceil \cdot \lfloor \frac{n}{2} \rfloor$.*

BIZONYÍTÁS Ha G nem tartalmaz háromszöget, akkor $\alpha(G) \geq \Delta(G)$. Ugyanakkor $|E(G)| \leq \tau(G) \cdot \Delta(G)$ mindig igaz. Gallai tétele szerint $\alpha(G) + \tau(G) = |V(G)| = n$.

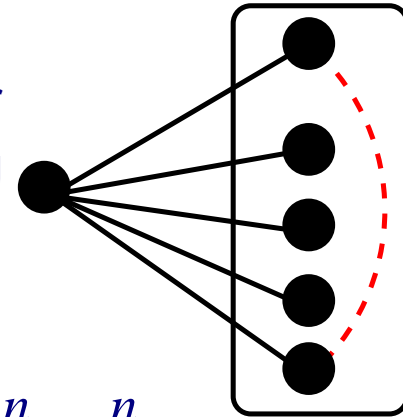


Mantel tétele

Legfeljebb hány éle lehet egy n -csúcsú egyszerű gráfnak, ha nem tartalmaz háromszöget?

1. Tétel. *Ha egy n -csúcsú egyszerű gráf nem tartalmaz háromszöget, akkor éleinek száma legfeljebb $\lceil \frac{n}{2} \rceil \cdot \lfloor \frac{n}{2} \rfloor$.*

BIZONYÍTÁS Ha G nem tartalmaz háromszöget, akkor $\alpha(G) \geq \Delta(G)$. Ugyanakkor $|E(G)| \leq \tau(G) \cdot \Delta(G)$ mindig igaz. Gallai tétele szerint $\alpha(G) + \tau(G) = |V(G)| = n$.
Összerakva:



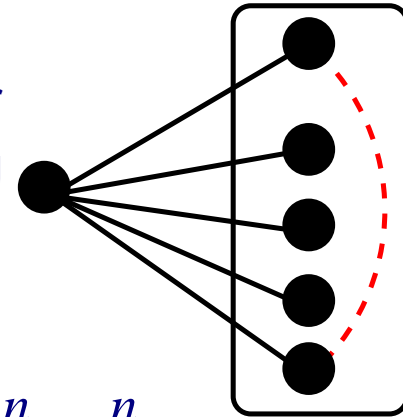
$$|E(G)| \leq \tau(G) \cdot \Delta(G) \leq \tau(G) \cdot \alpha(G) = (n - \alpha(G)) \cdot \alpha(G) \leq \lceil \frac{n}{2} \rceil \cdot \lfloor \frac{n}{2} \rfloor.$$

Mantel tétele

Legfeljebb hány éle lehet egy n -csúcsú egyszerű gráfnak, ha nem tartalmaz háromszöget?

1. Tétel. *Ha egy n -csúcsú egyszerű gráf nem tartalmaz háromszöget, akkor éleinek száma legfeljebb $\lceil \frac{n}{2} \rceil \cdot \lfloor \frac{n}{2} \rfloor$.*

BIZONYÍTÁS Ha G nem tartalmaz háromszöget, akkor $\alpha(G) \geq \Delta(G)$. Ugyanakkor $|E(G)| \leq \tau(G) \cdot \Delta(G)$ mindig igaz. Gallai tétele szerint $\alpha(G) + \tau(G) = |V(G)| = n$.
Összerakva:



$$|E(G)| \leq \tau(G) \cdot \Delta(G) \leq \tau(G) \cdot \alpha(G) = (n - \alpha(G)) \cdot \alpha(G) \leq \lceil \frac{n}{2} \rceil \cdot \lfloor \frac{n}{2} \rfloor.$$

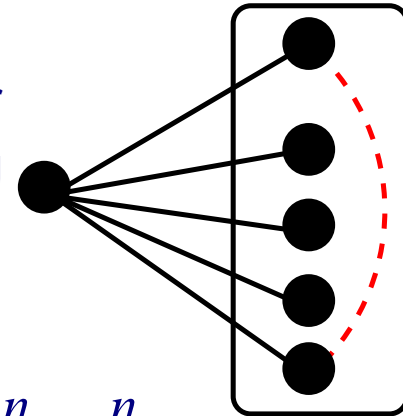
A $K_{\lceil \frac{n}{2} \rceil, \lfloor \frac{n}{2} \rfloor}$ teljes páros gráf mutatja, hogy a tétel állítása éles.

Mantel tétele

Legfeljebb hány éle lehet egy n -csúcsú egyszerű gráfnak, ha nem tartalmaz háromszöget?

1. Tétel. *Ha egy n -csúcsú egyszerű gráf nem tartalmaz háromszöget, akkor éleinek száma legfeljebb $\lceil \frac{n}{2} \rceil \cdot \lfloor \frac{n}{2} \rfloor$.*

BIZONYÍTÁS Ha G nem tartalmaz háromszöget, akkor $\alpha(G) \geq \Delta(G)$. Ugyanakkor $|E(G)| \leq \tau(G) \cdot \Delta(G)$ mindig igaz. Gallai tétele szerint $\alpha(G) + \tau(G) = |V(G)| = n$.
Összerakva:

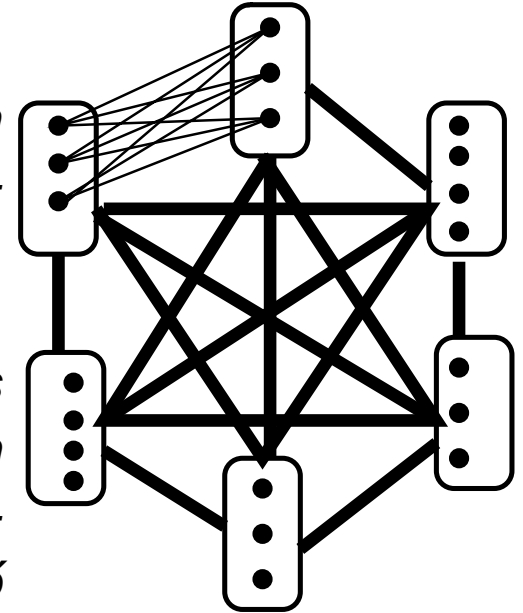


$$|E(G)| \leq \tau(G) \cdot \Delta(G) \leq \tau(G) \cdot \alpha(G) = (n - \alpha(G)) \cdot \alpha(G) \leq \lceil \frac{n}{2} \rceil \cdot \lfloor \frac{n}{2} \rfloor.$$

A $K_{\lceil \frac{n}{2} \rceil, \lfloor \frac{n}{2} \rfloor}$ teljes páros gráf mutatja, hogy a tétel állítása éles. Azt, hogy ez az egyetlen ilyen gráf, általánosabban bizonyítjuk.

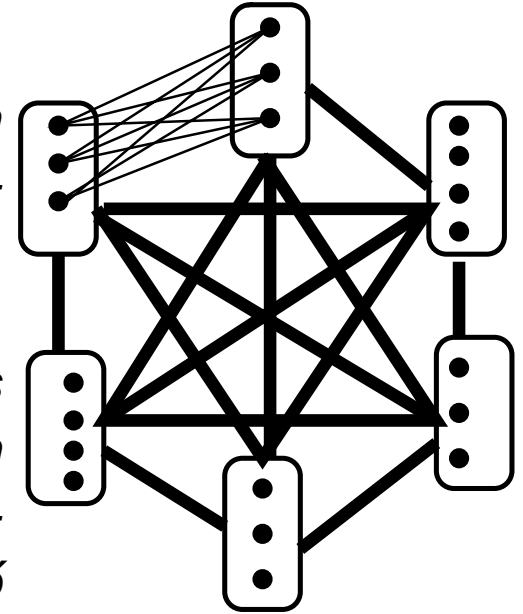
Turán gráfok

2. Definíció. Definiáljuk a $T_{n,m}$ ($n \geq m$) gráfot a következőképpen. Osszuk el maradékosan n -et m -mel, azaz legyen $n = qm + r$, ahol $0 \leq r < m$. A gráf n pontját osszuk m osztályra, r osztály álljon $q + 1$ pontból, a többi $m - r$ pedig q pontból. A gráfban két pont akkor és csak akkor legyen összekötve, ha különböző osztályban vannak. **m -osztályú gráfnak** nevezünk egy gráfot, ha a pontjai m osztályba oszthatók úgy, hogy az egy osztályban levő pontok között nem fut él. $T_{n,m}$ -et másképpen m -osztályú teljes gráfnak nevezzük.



Turán gráfok

2. Definíció. Definiáljuk a $T_{n,m}$ ($n \geq m$) gráfot a következőképpen. Osszuk el maradékosan n -et m -mel, azaz legyen $n = qm + r$, ahol $0 \leq r < m$. A gráf n pontját osszuk m osztályra, r osztály álljon $q + 1$ pontból, a többi $m - r$ pedig q pontból. A gráfban két pont akkor és csak akkor legyen összekötve, ha különböző osztályban vannak. **m -osztályú gráfnak** nevezünk egy gráfot, ha a pontjai m osztályba oszthatók úgy, hogy az egy osztályban levő pontok között nem fut él. $T_{n,m}$ -et másképpen m -osztályú teljes gráfnak nevezzük.



3. Tétel. Ha egy n pontú G gráf nem tartalmaz K_{m+1} -et, akkor

$$e(G) \leq e(T_{n,m}).$$

Ha pedig $e(G) = e(T_{n,m})$, akkor $G \cong T_{n,m}$.

Turán tétel bizonyítása

Turán tétel bizonyítása

1. Az m -osztályú gráfok közül $T_{n,m}$ -nek van a legtöbb éle.

Turán tétel bizonyítása

1. Az m -osztályú gráfok közül $T_{n,m}$ -nek van a legtöbb éle. Tegyük fel, hogy az a G gráf, amelyiknek a legtöbb éle van, nem a $T_{n,m}$ gráf. Ebben a gráfban kell, hogy legyen két olyan osztály, hogy az egyikben x pont van, a másikban legalább $x + 2$.

Turán tétel bizonyítása

1. Az m -osztályú gráfok közül $T_{n,m}$ -nek van a legtöbb éle. Tegyük fel, hogy az a G gráf, amelyiknek a legtöbb éle van, nem a $T_{n,m}$ gráf. Ebben a gráfban kell, hogy legyen két olyan osztály, hogy az egyikben x pont van, a másikban legalább $x + 2$. Ha a nagyobból a kisebbbe átteszünk egy pontot, akkor legfeljebb x él szűnik meg, viszont legalább $x + 1$ új élet húzunk be. Vagyis növeltük az élszámot, ez pedig ellentmond a feltevésünknek.

Turán tétel bizonyítása

1. Az m -osztályú gráfok közül $T_{n,m}$ -nek van a legtöbb éle. Tegyük fel, hogy az a G gráf, amelyiknek a legtöbb éle van, nem a $T_{n,m}$ gráf. Ebben a gráfban kell, hogy legyen két olyan osztály, hogy az egyikben x pont van, a másikban legalább $x + 2$. Ha a nagyobból a kisebbbe átteszünk egy pontot, akkor legfeljebb x él szűnik meg, viszont legalább $x + 1$ új élet húzunk be. Vagyis növeltük az élszámot, ez pedig ellentmond a feltevésünknek.
2. Ha G egy K_{m+1} -et nem tartalmazó n -pontú gráf, akkor ugyanazon a ponthalmazon konstruálható egy olyan m osztályú teljes H gráf, melyben minden pont fokszáma legalább akkora mint G -ben, vagyis minden $v \in V(G) = V(H)$ -ra $d_G(v) \leq d_H(v)$.

Turán tétel bizonyítása

1. Az m -osztályú gráfok közül $T_{n,m}$ -nek van a legtöbb éle. Tegyük fel, hogy az a G gráf, amelyiknek a legtöbb éle van, nem a $T_{n,m}$ gráf. Ebben a gráfban kell, hogy legyen két olyan osztály, hogy az egyikben x pont van, a másikban legalább $x + 2$. Ha a nagyobból a kisebbbe átteszünk egy pontot, akkor legfeljebb x él szűnik meg, viszont legalább $x + 1$ új élet húzunk be. Vagyis növeltük az élszámot, ez pedig ellentmond a feltevésünknek.
2. Ha G egy K_{m+1} -et nem tartalmazó n -pontú gráf, akkor ugyanazon a ponthalmazon konstruálható egy olyan m osztályú teljes H gráf, melyben minden pont fokszáma legalább akkora mint G -ben, vagyis minden $v \in V(G) = V(H)$ -ra $d_G(v) \leq d_H(v)$.
3. m -re való teljes indukcióval bizonyítunk. $m = 1$ -re az állítás triviális.

Turán tétel bizonyítása

1. Az m -osztályú gráfok közül $T_{n,m}$ -nek van a legtöbb éle. Tegyük fel, hogy az a G gráf, amelyiknek a legtöbb éle van, nem a $T_{n,m}$ gráf. Ebben a gráfban kell, hogy legyen két olyan osztály, hogy az egyikben x pont van, a másikban legalább $x + 2$. Ha a nagyobból a kisebbbe átteszünk egy pontot, akkor legfeljebb x él szűnik meg, viszont legalább $x + 1$ új élet húzunk be. Vagyis növeltük az élszámot, ez pedig ellentmond a feltevésünknek.
2. Ha G egy K_{m+1} -et nem tartalmazó n -pontú gráf, akkor ugyanazon a ponthalmazon konstruálható egy olyan m osztályú teljes H gráf, melyben minden pont fokszáma legalább akkora mint G -ben, vagyis minden $v \in V(G) = V(H)$ -ra $d_G(v) \leq d_H(v)$.
3. m -re való teljes indukcióval bizonyítunk. $m = 1$ -re az állítás triviális.
4. Legyen x olyan pont, hogy $d_G(x) = \Delta_G$. Legyen $V_1 = \{u \mid \{u, x\} \in E(G)\}$, vagyis x szomszédainak halmaza, V_2 pedig a többi pont, vagyis $V_2 = V(G) - V_1$ Így persze $x \in V_2$. G_1 legyen G -nek a V_1 által feszített részgráfja.

Turán tétel bizonyítása

1. Az m -osztályú gráfok közül $T_{n,m}$ -nek van a legtöbb éle. Tegyük fel, hogy az a G gráf, amelyiknek a legtöbb éle van, nem a $T_{n,m}$ gráf. Ebben a gráfban kell, hogy legyen két olyan osztály, hogy az egyikben x pont van, a másikban legalább $x + 2$. Ha a nagyobból a kisebbbe átteszünk egy pontot, akkor legfeljebb x él szűnik meg, viszont legalább $x + 1$ új élet húzunk be. Vagyis növeltük az élszámot, ez pedig ellentmond a feltevésünknek.
2. Ha G egy K_{m+1} -et nem tartalmazó n -pontú gráf, akkor ugyanazon a ponthalmazon konstruálható egy olyan m osztályú teljes H gráf, melyben minden pont fokszáma legalább akkora mint G -ben, vagyis minden $v \in V(G) = V(H)$ -ra $d_G(v) \leq d_H(v)$.
3. m -re való teljes indukcióval bizonyítunk. $m = 1$ -re az állítás triviális.
4. Legyen x olyan pont, hogy $d_G(x) = \Delta_G$. Legyen $V_1 = \{u \mid \{u, x\} \in E(G)\}$, vagyis x szomszédainak halmaza, V_2 pedig a többi pont, vagyis $V_2 = V(G) - V_1$. Így persze $x \in V_2$. G_1 legyen G -nek a V_1 által feszített részgráfja. Nyilván G_1 -ben nincs K_m , hiszen

ez x -szel együtt G -ben K_{m+1} -et alkotna. Így alkalmazhatjuk az indukciós feltevést G_1 -re. Tehát van olyan teljes $m - 1$ -osztályú H_1 gráf, hogy minden $v \in V(G_1)$ -re $d_{G_1}(v) \leq d_{H_1}(v)$.

ez x -szel együtt G -ben K_{m+1} -et alkotna. Így alkalmazhatjuk az indukciós feltevést G_1 -re. Tehát van olyan teljes $m - 1$ -osztályú H_1 gráf, hogy minden $v \in V(G_1)$ -re $d_{G_1}(v) \leq d_{H_1}(v)$.

5. H gráf a következő: Vegyük a V_1 ponthalmazon a H_1 gráfot, majd V_1 minden pontját kössük össze V_2 minden pontjával, viszont hagyjunk el minden két V_2 -beli pontot összekötő élet. Nyilvánvaló, hogy ez a H gráf m osztályú.

ez x -szel együtt G -ben K_{m+1} -et alkotna. Így alkalmazhatjuk az indukciós feltevést G_1 -re. Tehát van olyan teljes $m - 1$ -osztályú H_1 gráf, hogy minden $v \in V(G_1)$ -re $d_{G_1}(v) \leq d_{H_1}(v)$.

5. H gráf a következő: Vegyük a V_1 ponthalmazon a H_1 gráfot, majd V_1 minden pontját kössük össze V_2 minden pontjával, viszont hagyjunk el minden két V_2 -beli pontot összekötő élet. Nyilvánvaló, hogy ez a H gráf m osztályú. Ha $v \in V_2$, akkor $d_H(v) = |V_1| = \Delta_G$, a definícióink szerint viszont $d_G(v) \leq \Delta_G$. Ha $v \in V_1$, akkor $d_H(v) = d_{H_1}(v) + |V_2| \geq d_{G_1}(v) + |V_2| \geq d_G(v)$.

ez x -szel együtt G -ben K_{m+1} -et alkotna. Így alkalmazhatjuk az indukciós feltevést G_1 -re. Tehát van olyan teljes $m - 1$ -osztályú H_1 gráf, hogy minden $v \in V(G_1)$ -re $d_{G_1}(v) \leq d_{H_1}(v)$.

5. H gráf a következő: Vegyük a V_1 ponthalmazon a H_1 gráfot, majd V_1 minden pontját kössük össze V_2 minden pontjával, viszont hagyjunk el minden két V_2 -beli pontot összekötő élet. Nyilvánvaló, hogy ez a H gráf m osztályú. Ha $v \in V_2$, akkor $d_H(v) = |V_1| = \Delta_G$, a definícióink szerint viszont $d_G(v) \leq \Delta_G$. Ha $v \in V_1$, akkor $d_H(v) = d_{H_1}(v) + |V_2| \geq d_{G_1}(v) + |V_2| \geq d_G(v)$.
6. Így ha egy G gráfban nincs K_{m+1} , de nem izomorf $T_{n,m}$ -mel, akkor konstruáltunk egy nála nagyobb élszámú m -osztályú teljes gráfot (ugyanis ekkor valamelyik egyenlőtlenség biztosan nem egyenlőség), ennek az élszáma pedig nem nagyobb $T_{n,m}$ élszámánál. Egyben beláttuk az állítás második részét is.

Két érdekes tétel

Két érdekes tétel

4. Tétel (Erdős–Stone). *Ha*

$$e(G) \geq e(T_{n,m}) + \varepsilon n^2,$$

akkor G -ben nemcsak hogy van legalább egy K_{m+1} , hanem létezik olyan $c(\varepsilon, m)$ konstans is, hogy G -ben van olyan teljes $m + 1$ -osztályú részgráf, amelyben az osztályok pontszáma legalább $c \log n$.

Két érdekes tétel

4. Tétel (Erdős–Stone). Ha

$$e(G) \geq e(T_{n,m}) + \varepsilon n^2,$$

akkor G -ben nemcsak hogy van legalább egy K_{m+1} , hanem létezik olyan $c(\varepsilon, m)$ konstans is, hogy G -ben van olyan teljes $m+1$ -osztályú részgráf, amelyben az osztályok pontszáma legalább $c \log n$.

Azaz, ha csak kicsit nagyobb az élsűrűség, mint a Turán gráfé, akkor már *rengeteg* K_{m+1} van a gráfban.

5. Tétel (Erdős–Simonovits). *Ha G_1, G_2, \dots, G_k adott gráfok, akkor létezik olyan $ex(n; G_1, G_2, \dots, G_k)$ függvény, amelyre teljesül, hogy minden olyan G gráfnak, amelyre $v(G) = n$ és $e(G) \geq ex(n; G_1, G_2, \dots, G_k)$, van valamelyik G_i gráffal izomorf részgráfja. Az ex függvényre teljesül, hogy*

$$\lim_{n \rightarrow \infty} \frac{ex(n; G_1, G_2, \dots, G_k)}{\binom{n}{2}} = 1 - \frac{1}{\min_{i=1, \dots, k} \chi(G_i) - 1}. \quad (1)$$

5. Tétel (Erdős–Simonovits). *Ha G_1, G_2, \dots, G_k adott gráfok, akkor létezik olyan $ex(n; G_1, G_2, \dots, G_k)$ függvény, amelyre teljesül, hogy minden olyan G gráfnak, amelyre $v(G) = n$ és $e(G) \geq ex(n; G_1, G_2, \dots, G_k)$, van valamelyik G_i gráffal izomorf részgráfja. Az ex függvényre teljesül, hogy*

$$\lim_{n \rightarrow \infty} \frac{ex(n; G_1, G_2, \dots, G_k)}{\binom{n}{2}} = 1 - \frac{1}{\min_{i=1, \dots, k} \chi(G_i) - 1}. \quad (1)$$

Azaz a maximális G_i -t nem tartalmazó gráf élsűrűségének nagyságrendje G_i kromatikus számától függ, ha az legalább 3.

5. Tétel (Erdős–Simonovits). *Ha G_1, G_2, \dots, G_k adott gráfok, akkor létezik olyan $ex(n; G_1, G_2, \dots, G_k)$ függvény, amelyre teljesül, hogy minden olyan G gráfnak, amelyre $v(G) = n$ és $e(G) \geq ex(n; G_1, G_2, \dots, G_k)$, van valamelyik G_i gráffal izomorf részgráfja. Az ex függvényre teljesül, hogy*

$$\lim_{n \rightarrow \infty} \frac{ex(n; G_1, G_2, \dots, G_k)}{\binom{n}{2}} = 1 - \frac{1}{\min_{i=1, \dots, k} \chi(G_i) - 1}. \quad (1)$$

Azaz a maximális G_i -t nem tartalmazó gráf élsűrűségének nagyságrendje G_i kromatikus számától függ, ha az legalább 3.

Ha valamelyik kizárandó gráf páros, akkor a fenti tétel nem határozza meg az élsűrűség nagyságrendjét.

$T_{n,m}$ élszáma

$$e(T_{n,m}) = \binom{n}{2} - r \binom{q+1}{2} - (m-r) \binom{q}{2} \approx \binom{n}{2} \left(1 - \frac{1}{m}\right). \quad (2)$$

$T_{n,m}$ élszáma

$$e(T_{n,m}) = \binom{n}{2} - r \binom{q+1}{2} - (m-r) \binom{q}{2} \approx \binom{n}{2} \left(1 - \frac{1}{m}\right). \quad (2)$$

$k = 1$ és $G = K_{m+1}$ esetén $\chi(K_{m+1}) = m + 1$, azaz ebben az esetben az Erdős-Simonovits következik a Turánból.