Hálótervezés:

Bevezetés

Ezen fejezetben egy az eddigiektl eltér területet ismerhetünk meg, mely nem a lineráris programozásra, hanem gráf elméleti ismeretekre épít.
A fejezet elején összefoglalásra kerülnek a szükséges gráfelméleti ismeretek - így aki nem rendelkezik elismeretekkel ezen a területen az is megértheti a módszereket.
Példán keresztül kerül bemutatásra a - szemléletünknek jobban megfelel - tevékenység típusú háló, majd a számításainkban használt esemény típusú háló. Szintén példán a tevékenység láncolatok teljes idejének kiszámítása, az ezt alkotó tevékenységek láncolatának megkeresése és a tartalékid keresés módszere. Az úgynevezett kritikus út, kritikus id meghatározási módszert használjuk. Utalunk mindig az MS Project szoftver használatára, mely a valós problémák - nem csak ezen a területen, hanem az erforrások kezelése, pénzügyi tervezés területen is - modellezésére alkalmas. Mivel ez rendelkezésre áll, nem fektetünk hangsúlyt a valós feladatok hálójának megkonstruálása területre.
Tananyagunk új fejlesztései - az interaktív részben külön is megjelen, de itt is leírásra kerül - Maple eljárások, melyek felrajzolják - a Maple inputnak megfelelen megadott - irányított súlyozott gráfból a hálót, kiszámítják a kritikus idt és megkeresik és felrajzolják a kritikus utat.

Gráf elméleti ismeretek

Alapismeretek, definíciók:

Hálótervezési feladataink kezeléséhez szükséges néhány gráfelméleti alapismeret. Ezek fként definíciók, melyek elször egzakt matematikai megfogalmazásban kerülnek megadásra, majd köznapi megfogalmazásuk következik, végül a példával történ illusztrálás.
Mj: A példák természetesen a Maple rajzolási lehetségei alkalmazásával vannak megadva. A megadott utasítások Enter - el történ aktiválásával rajzolódik ki (ismét) a gráf. A megadások természetesen módosíthatóak és amennyiben jól történik a megadás módosítása a módosított gráf kerül akkor kirajzolásra. Azért, hogy módosítás után is elérhet legyen az eredeti megadás, a szöveg részben is megjelenítésre kerül az eredeti utasítás, "Példa Maple kódja:"felirattal.
Mj: Elsként (egyszer) a with(GraphTheory)utasításnak kell szerepelnie, hogy az ezen területhez tartozó utasítás készlet elérhet legyen.

Restart : *with*(*GraphTheory*) :

Gráf:=
$G_0 = (N,A)$ - egy véges ponthalmaz N - (csúcsok) és egy véges pontpár halmaz A - (élek) együttese.
ahol
$N=\{N_1, N_2, .., N_n\}$ ponthalmaz a csúcsok halmaza
$A=\{A_1, A_2, .., A_m\}$, ahol $A_k= (N_i , N_j) \in A$ pontpár halmaz az élek halmaza

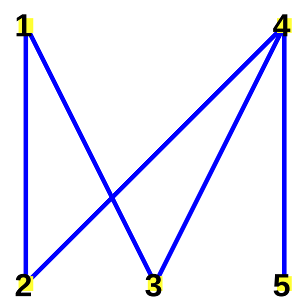
Megfogalmazás: A gráf csúcsokkal és élekkel rendelkez alakzat. (Az éleknek nincs sem irányítása sem súlya.)

(Nem irányított, nem súlyozott gráf példa.)
A példa Maple kódja és rajza:

```
> GNINS := Graph({{1, 2}, {1, 3}, {2, 4}, {3, 4}, {4, 5}});
```

GNINS := Graph 1: an undirected unweighted graph with 5 vertices and 5 edge(s) **(2.1.1)**

```
> DrawGraph(GNINS);
```



Irányított (de nem súlyozott) **gráf:=**
$G_0 = (N,A)$ - egy véges ponthalmaz N - (csúcsok) és egy véges pontpár halmaz A - (élek) olyan együttese,
ahol
$N=\{N_1, N_2, .., N_n\}$ ponthalmaz a csúcsok halmaza
$A=\{A_1, A_2, .., A_m\}$, ahol $A_k= (N_i , N_j) \in A$ rendezett pontpár halmaz (az élek halmaza) melyben N_i , az A_k él kezdpontja N_j pedig a végpontja.

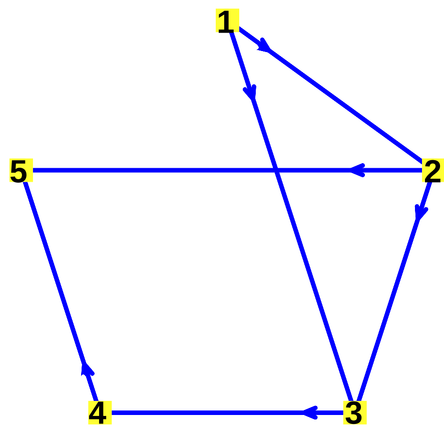
Megfogalmazás: Az irányított gráf olyan csúcsokkal és élekkel rendelkez alakzat, melyben az éleknek irányítása van. (Súly azonban nem szerepel az éleken.)

Példa Maple kódja és rajza:

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

GIINS := Graph 4: a directed unweighted graph with 5 vertices and 6 arc(s) **(2.1.2)**

```
> DrawGraph(GIINS);
```

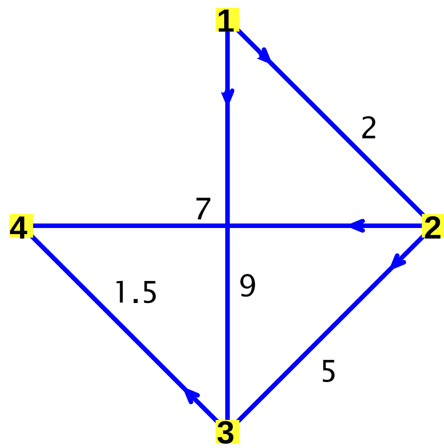


Irányított súlyozott gráf :=
$G_0 = (N,A)$ - egy véges ponthalmaz N - (csúcsok) és egy véges pontpár halmaz A - (élek) olyan együttese,
ahol
$N=\{N_1, N_2, .., N_n\}$ ponthalmaz a csúcsok halmaza
$A=\{A_1, A_2, .., A_m\}$, ahol $A_k= (N_i , N_j) \in A$ rendezett pontpár halmaz (az élek halmaza) melyben N_i , az A_k él kezdpontja N_j pedig a végpontja és minden A_k élhez egy súlyt rendelünk.

Megfogalmazás: Az irányított súlyozott gráf olyan csúcsokkal és élekkel rendelkez alakzat, melyben az éleknek irányítása és hozzátartozó súlya is van.

Példa Maple kódja és rajza:

```
> GIISS := Graph({[[1, 2], 2], [[1, 3], 9], [[2, 3], 5], [[3, 4], 3/2], [[2, 4], 7]});
        GIISS := Graph 6: a directed weighted graph with 4 vertices and 5 arc(s)        (2.1.3)
> DrawGraph(GIISS);
```



Irányítatlan gráf :=

G_0 = (N,A) - egy véges ponthalmaz N - (csúcsok) és egy véges pontpár halmaz A - (élek) olyan együttese,

ahol

N={N_1, N_2, .., N_n} ponthalmaz a csúcsok halmaza

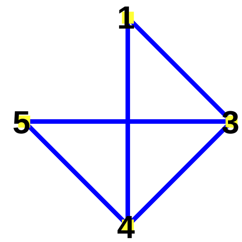
A={A_1, A_2, .., A_m}, ahol A_k= (N_i , N_j) ∈A pontpár halmaz (az élek halmaza) melyben (N_i , N_j) azonos (N_j , N_i) - vel

Megfogalmazás: Olyan gráf melyben a pontpárok nem rendezettek, vagyis nincs nyíl a gráf élein. Vagyis (N_i , N_j) azonos (N_j , N_i) - vel

Mj: Ilyen példa szerepelt az els gráf definícióban is.

Példa Maple kódja és rajza:

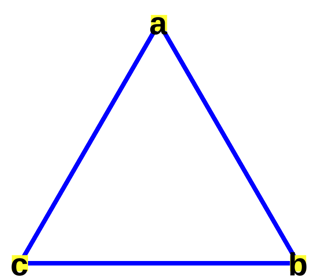
```
> GNINS := Graph({{1, 3}, {3, 4}, {1, 4}, {4, 5}, {3, 5}}) :
> DrawGraph(GNINS);
```



A Maple szimbolikus reprezentációs lehetsége miatt megadható karakterrel jellemzett csúcsokkal rendelkez gráf is az alábbiakban: (nem irányított nem súlyozott gráf példda)

> $GPNINS := Graph(\{\{a, b\}, \{b, c\}, \{c, a\}\})$:

> $DrawGraph(GPNINS)$;

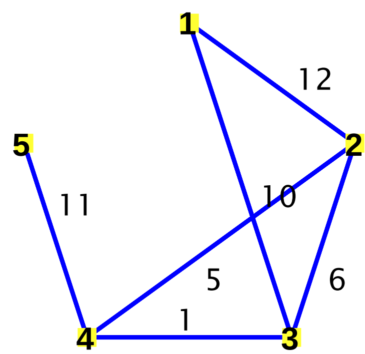


Irányítatlan súlyozott gráf :=
$G_0 = (N,A)$ - egy véges ponthalmaz N - (csúcsok) és egy véges pontpár halmaz A - (élek) olyan együttese,
ahol
$N=\{N_1, N_2, .., N_n\}$ ponthalmaz a csúcsok halmaza
$A=\{A_1, A_2, .., A_m\}$, ahol $A_k= (N_i , N_j) \in A$ pontpár halmaz (az élek nem irányított halmaza)
melyben (N_i , N_j) azonos (N_j , N_i) - vel

Megfogalmazás: Olyan gráf amelyben a pontpárok nem rendezettek, de minden élhez tartozik egy "súly" melyet az élen jelenítünk meg.

Példa Maple kódja és rajza:

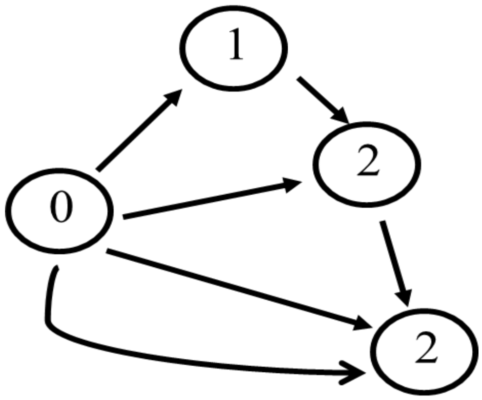
```
> GNISS := Graph(5, {[{1, 2}, 12], [{1, 3}, 10], [{2, 3}, 6], [{2, 4}, 5], [{3, 4}, 1],
      [{4, 5}, 11]});
    GNISS := Graph 7: an undirected weighted graph with 5 vertices and 6 edge(s)        (2.1.4)
> DrawGraph(GNISS);
```



A továbbiakban már nem adjuk meg a teljes matematikai definíciót, csak a gráf specialitásának leírását és példát.

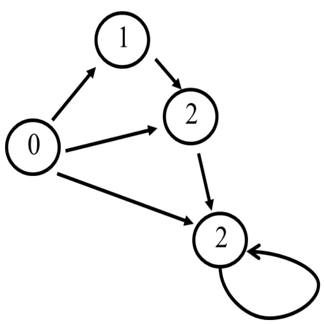
Többszörös él:=
Ha egy G_0 gráfnak $A_k$és A_r is eleme ($A_k \in A$ és $A_r \in A$),
ahol $A_k= (N_i , N_j)$ és ugyanekkor $A_r= (N_i , N_j)$
vagyis ha ugyanazon kezd és végpontú él többször is szerepel az élek halmazában.
Megjegyzés: ez lehet irányított vagy irányítatlan illetve súlyozott is.

Példa:
Példa többszörös éllel rendelkez irányított súlyozott gráfra.
(Az alábbi rajz nem a Maple-el készült, a rajz képként került ide.)



Hasonlóan lehetséges többszörös éllel rendelkez nem irányított súlyozott gráf illetve nem irányított, nem súlyozott gráf is.

Hurokél:=
Ha $A_k = (N_i, N_i) \in A$ akkor azt mondjuk, hogy $A_k \in A$ hurokél.
Példa:
Hurokél rajzolása is nehézkes Maple-el, ezért az erre vonatkozó példa is képként került ide.



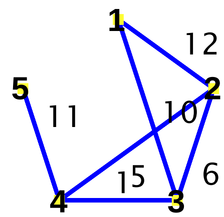
Egyszer gráf:=
Olyan gráf amely nem tartalmaz hurok élt és többszörös élt.
Ezen definíció egyaránt alkalmazható irányított és nem irányított esetekre.

Példa Maple kódja és rajza:

```
> GNISS := Graph(5, {[{1, 2}, 12], [{1, 3}, 10], [{2, 3}, 6], [{2, 4}, 5], [{3, 4}, 1],
      [{4, 5}, 11]});
```

GNISS := Graph 9: an undirected weighted graph with 5 vertices and 6 edge(s) **(2.1.5)**

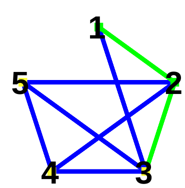
```
> DrawGraph(GNISS);
```



Szomszédos él :=
Két olyan nem irányított él melynek van közös csúcspontja. Az ábrán zölddel jelöve.

Példa Maple kódja és rajza:

```
> GNINS := Graph({{1, 2}, {2, 3}, {1, 3}, {3, 4}, {2, 4}, {2, 5}, {3, 5}, {4, 5}}) :
> neighbournods := {1, 2} :
  neighbouredges := {{1, 2}, {2, 3}} :
  HighlightEdges(GNINS, GNINS, blue) :
  HighlightVertex(GNINS, neighbournods, green) :
   HighlightEdges(GNINS, neighbouredges, green) :
   DrawGraph(GNINS);
```

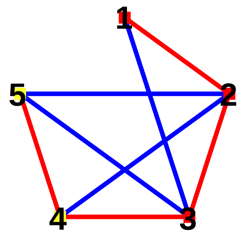


Irányítatlan út:=
Az élek olyan sorozata, egy irányítatlan gráfban, melyben bármely két szomszédos élnek van közös pontja. Az ábrán pirossal jelölve.

Példa Maple kódja és rajza:

```
> GNINS := Graph({{1, 2}, {2, 3}, {1, 3}, {3, 4}, {2, 4}, {2, 5}, {3, 5}, {4, 5}}) :
> edgesofrouteundirected := {1, 2, 3} :
  routeundirected := {{1, 2}, {2, 3}, {3, 4}, {4, 5}} :
  HighlightVertex(GNINS, edgesofrouteundirected, red) :
```

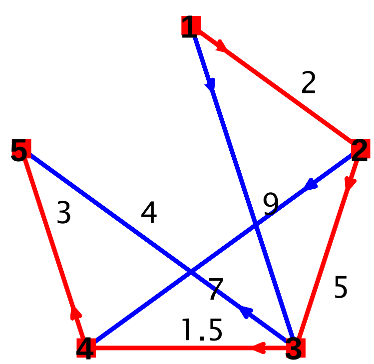
```
HighlightEdges(GNINS, routeundirected, red) :
DrawGraph(GNINS);
```



Irányított út:=
Élek olyan sorozata, egy irányított (esetünkben súlyozott) gráfban, melyben bármely él végpontja azonos a rákövetkez él kezdpontjával - amennyiben van rákövetkez él. (Az utolsó él esetében nincs rákövetkez él.)
Példánkban súlyozott irányított utat mutatunk be az alábbi utakkal:
[1,2] [2,3] [3,4] [4,5]
Mj: rajzunk a csomópontokat is kiemeli (piros színnel).

Példa Maple kódja és rajza:

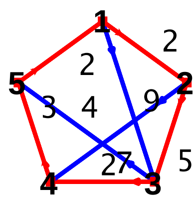
```
> GIISS := Graph({[[1, 2], 2], [[1, 3], 9], [[2, 3], 5], [[3, 4], 3/2], [[2, 4], 7], [[4, 5],
      3], [[3, 5], 4]}) :
> edgesofroutedirected := [1, 2, 3, 4, 5] :
  routedirected := {[1, 2], [2, 3], [3, 4], [4, 5]} :
  P := PathGraph(edgesofroutedirected) :
  HighlightVertex(GIISS, edgesofroutedirected, red) :
  HighlightEdges(GIISS, routedirected, red) :
  DrawGraph(GIISS);
```



Irányított kör:=
Olyan irányított út, melynek kezd és végpontja azonos.

Példa Maple kódja és rajza:

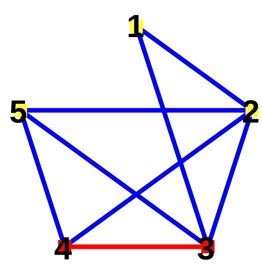
```
> GIISS := Graph({[[1, 2], 2], [[1, 3], 9], [[2, 3], 5], [[3, 4], 2], [[2, 4], 7], [[4, 5],
      3], [[3, 5], 4], [[5, 1], 2]}) :
> edgesofroutedirected := [1, 2, 3, 4, 5] :
  routedirected := {[1, 2], [2, 3], [3, 4], [4, 5], [5, 1]} :
  P := PathGraph(edgesofroutedirected) :
   HighlightVertex(GIISS, edgesofroutedirected, red) :
   HighlightEdges(GIISS, routedirected, red) :
   DrawGraph(GIISS);
```



Szomszédos csúcs:=
Két csúcs szomszédos, ha köztük van él. (Nem irányított és irányított esetre is vonatkozhat.)
Pl: a 3-as és a 4-es csúcs szomszédos mivel létezik {3,4} él.

Példa Maple kódja és rajza:

```
> GNINS := Graph({{1, 2}, {2, 3}, {1, 3}, {3, 4}, {2, 4}, {2, 5}, {3, 5}, {4, 5}}) :
> edgesofrouteundirected := {3, 4} :
  routeundirected := {{3, 4}} :
  HighlightEdges(GNINS, GNINS, blue) :
  HighlightVertex(GNINS, edgesofrouteundirected, red) :
   HighlightEdges(GNINS, routeundirected, red) :
   DrawGraph(GNINS);
```



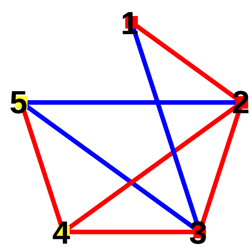
Valódi részgráf :=
Egy $G_1=(N_1,A_1)$ gráf (valódi) részgráfja egy $G_0=(N_0,A_0)$ gráfnak, ha $N_1 \subseteq N_0$, és $A_1 \subseteq A_0$
Jelölése: $Gp \subseteq G$

Megfogalmazás: Valódi részgráf az ha a részgráfnak minden csúcsa és minden éle eleme az t tartalmazó gráfnak.
(A piros rész valódi részgráfja a teljes gráfnak. Vagy a kék rész valódi részgráfja a teljes gráfnak.

)

Példa Maple kódja és rajza:

```
> GNINS := Graph({{1, 2}, {2, 3}, {1, 3}, {3, 4}, {2, 4}, {2, 5}, {3, 5}, {4, 5}}) :
> # subgraphnods :=
  ofrouteundirected := {1, 2, 3} :
  subgraphedges := {{1, 2}, {2, 3}} :
  edgesofrouteundirected := {1, 2, 3} :
  routeundirected := {{1, 2}, {2, 3}, {3, 4}, {2, 4}, {3, 4}, {4, 5}} :
  HighlightEdges(GNINS, GNINS, blue) :
  HighlightVertex(GNINS, edgesofrouteundirected, red) :
  HighlightEdges(GNINS, routeundirected, red) :
  DrawGraph(GNINS);
```




```
>
```

További gráf példák:

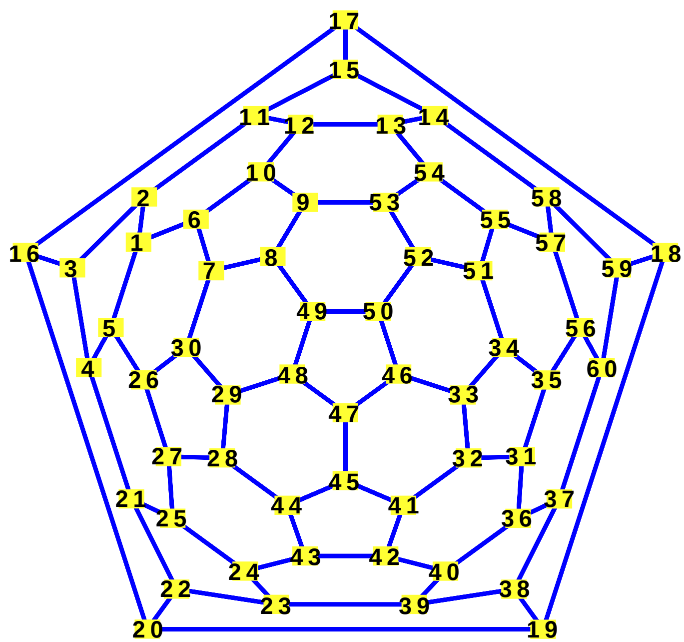
Méh sejt típusú gráf síkba és térbe rajzolva:

A példa Maple kódja és rajza:

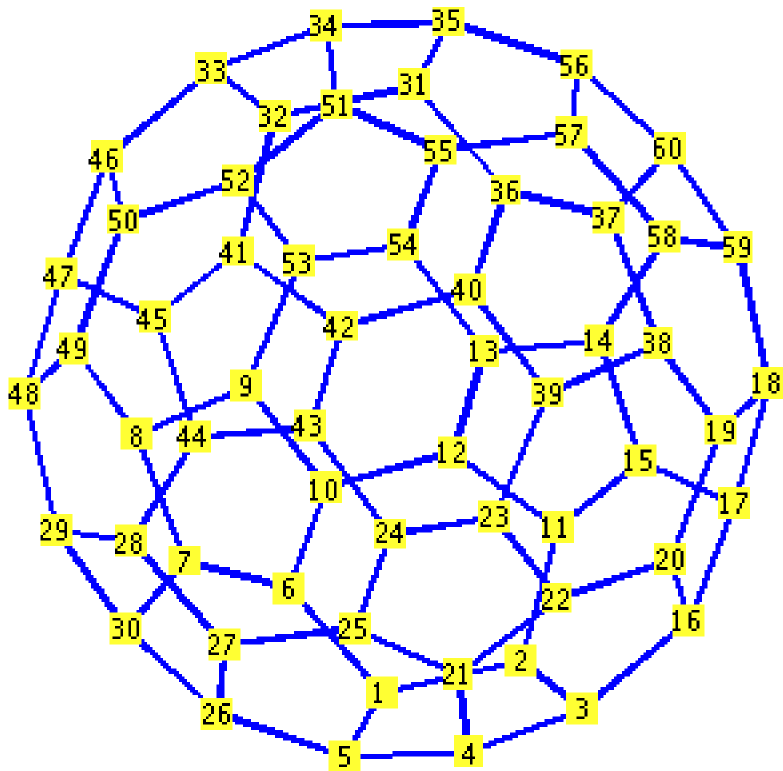
```
> S := SoccerBallGraph();
```

***S* := Graph 9: an undirected unweighted graph with 60 vertices and 90 edge(s)** **(2.1.6)**

```
> DrawGraph(S);
```

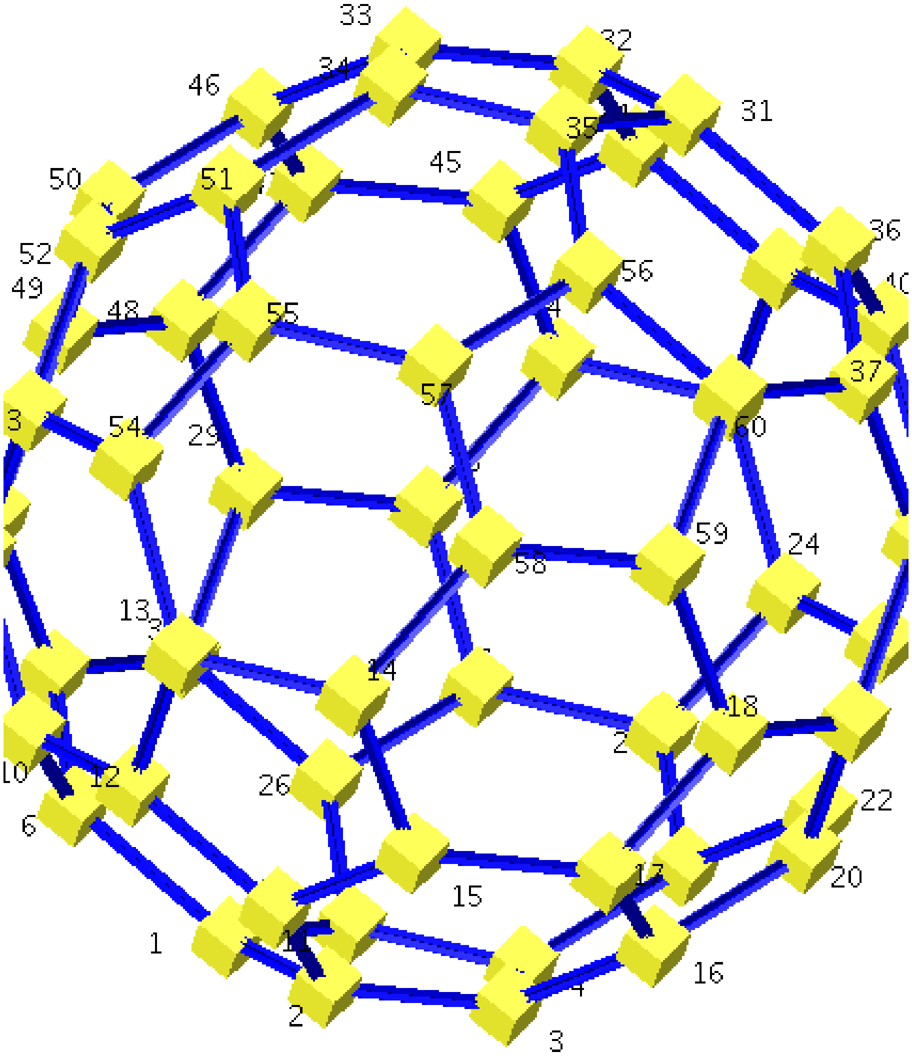



```
> DrawGraph(S,style=spring);
```



Térbeli gráf: (A nézpont egérrel változtatható.)

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



Gráfok megadása:

Szomszédossági mátrix:= (vagy adjecencia mátrix)
A szomszédossági mátrixban a sorokban a kiindulási csomópontok, az oszlopokban az érkezési csomópontok szerepelnek, a mátrixban pedig irányítatlan mátrix esetén 1-es ahol van összeköt él a csúcsok között, nulla ahol nincs. Szokásunk szerint csak kisebb sorszámú csomópontból nagyobb sorszámuba vezet él - melynek okát a késbbiekben látjuk majd csak.

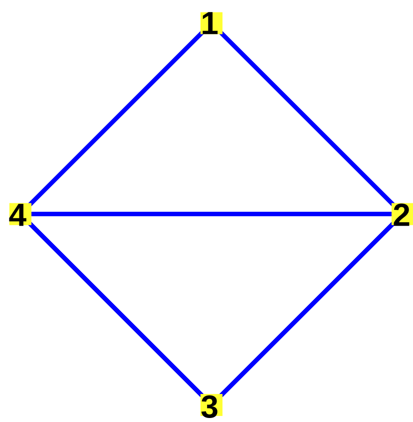
Példa: irányítatlan gráf szomszédossági mátrixára

	1	2	3	4
1	0	1	0	1
2	0	0	1	1
3	0	0	0	1
4	0	0	0	0

A fenti gráf rajza Maple-el:

A példa Maple kódja és rajza:

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



További megadási lehetség:

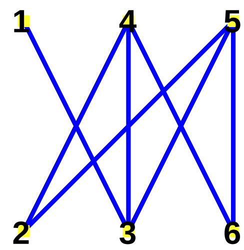
Adjecencia lista:=
A gráf éleinek megadása lista formájában.
Ez történhet több féle képpen, mi az alábbi reprezentációkat preferáljuk:

Példa: adjecancia lista irányítatlan gráfra:

i	1	2	2	3	3	4	5
j	3	4	5	4	5	6	6

A példa Maple kódja és rajza:

```
> Gadjaclist := Graph({{1, 3}, {2, 4}, {2, 5}, {3, 4}, {3, 5}, {4, 6}, {5, 6}}) :
> DrawGraph(Gadjaclist);
```



Adjecancia lista irányított gráfra:=
A gráf éleinek és a súlyoknak a megadása lista formájában.

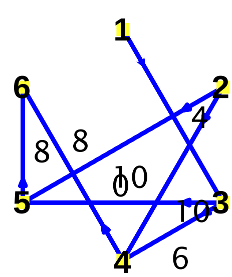
i	1	2	2	3	3	4	5
j	3	4	5	4	5	6	6
$t_{i,j}$	4	10	10	6	0	8	8

A példa Maple kódja és rajza:

```
> Gadjacir := Graph({[[1, 3], 4], [[2, 4], 10], [[2, 5], 10], [[3, 4], 6], [[3, 5], 0], [[4,
     6], 8], [[5, 6], 8]});
```

Gadjacir := Graph 10: a directed weighted graph with 6 vertices and 7 arc(s) **(2.2.1)**

```
> DrawGraph(Gadjacir);
```



Incidencia mátrix:=
Az Incidencia mátrixnak annyi oszlopa van ahány éle a mátrixnak és annyi sora ahány csomópontja. (Ritkán alkalmazott reprezentáció, ezért nem mutatunk be példát rá.)

Léteznek még további megadási lehetségek de számunkra ennyi már elégséges.

Tevékenységláncolatok tervezése elméleti bevezetés:

Egymásra épül részmunkafolyamatok idbeli lefolyásának optimális ütemezése. Divatos szóval projektek elemzése, a résztevékenységek idbeli összehangolása, egymás után vagy egymással párhuzamosan végezhet tevékenységek ütemezése.
Az elemzés kiterjedhet az
- Idtervezésre (mi a tevékenységláncolat befejezésének várható ideje, mely tevékenységek késhetnek végbefejezés megzavarása nélkül)
- Költségtervezésre (hogyan alakul a teljes tevékenység láncolat költsége a résztevékenységek költségeinek eredjeként.)
- Erforrás allokálásra (mely feladatot ki (vagy mely szervezet) végzi, ki milyen feladatot végez, ki milyen javadalmazásban részesül)

A tevékenységláncolat reprezentálásának eszköze a súlyozott körmentes, irányított gráf, amelynek egy kezd és egy végpontja van. Ezt egyszeren "háló"-nak szoktuk nevezni. A módszereket pedig "hálótervezési módszereknek". Ezeket csoportosíthatjuk (például) az alábbiak szerint.
A tevékenységek idtervezése lehet determinisztikus vagy sztochasztikus.
Determinisztikus - elre meghatározott, nem véletlenszer, vagyis a tevékenység id elre megadott, fix érték.
Szochasztikus - véletlen, valószínségi változóval leírhatóak a tevékenység idk.

A háló típusa lehet tevékenység vagy esemény típusú háló.
Tevékenység típusú háló esetén a gráf csomópontjai jelölik a tevékenységeket.
Esemény típusú háló esetén a csomópontok eseményeket jelentenek, és az élek reprezentálják a tevékenységeket.

Hálótervezési ismeretek példákkal:

Példa tevékenység típusú hálóra:

Tekintsük az alábbi (Garzon lakás takarítás) tevékenységláncolatot:

Sorszám :	*Tevékenység megnevezése :*	*Tevékenység idő :*
1.	*Szoba porszívózás*	20 *perc*
2.	*Szobában letörölgetni*	15 *perc*
3.	*Fürdőszobában a csempe és a szerelvények lesúrolása*	25 *perc*
4.	*Fürdőszoba felmosása*	5 *perc*

5.	*Konyhában bútorok lesúrolása*	25 *perc*
6.	*Konyha felmosása*	5 *perc*
7.	*Előszoba felsöprése*	5 *perc*
8.	*Előszoba felmosása*	5 *perc*

Garzon lakás takarítás

Ezen tevékenység láncolat nem feltétlenül csak egymás után ""sorosan" végezhet el. Hanem lehetnek benne párhuzamosságok, st akár több is (ha több ember dolgozik egyszerre). Ennek struktúrája a most megismert gráf reprezentációval jeleníthet meg hatékonyan.
A gráf megalkotásához, meg kell adnunk mely tevékenység melyiket kövesse - ebbl az már adódik, hogy melyek mehetnek párhuzamosan.
Kezd csomópont (0.) megadása is szükséges, mely a tevékenységek megkezdését jelenti illetve végcsomópontot is, mely a takarítás befejezését jelenti. Így tudunk egy átlátható struktúrát konstruálni.
Ahhoz, hogy a tevékenységek egymásutániságát fel tudjuk rajzolni meg kell adnunk minden tevékenységre, hogy mely tevékenység után következhet. Ezt hívjuk elfeltételnek.
(Angolul predecessor)
Ezek megadása személy, projekt, szervezési technika függ. Mi tegyük fel, hogy a különböz helyiségbeli tevékenységek függetlenek, mindegyik a takarítás megkezdése után azonnal kezdhet. Így a szoba porszívózás után jöhet a letörölgetés (a porszívó porát is letöröljük), ettl függetlenül kezdve a fürdszobai súrolás után jöhet a fürdszoba felmosása. A konyha is független, és ott is csak a bútorok lesúrolása után érdemes felmosni. Mivel minden helyiség az elszobából nyílik azt felseperni és felmosni csak ha az összes helyiséggel végeztünk akkor érdemes. Természetesen ezen párhuzamosságok csak akkor valósíthatók meg, ha elegend megvalósító, munkaer áll rendelkezésre. A megvalósítókat azonban ezen technikánkkal nem kezeljük. Ezt az MSProject programmal készített project reprezentációban könny kezelni, hagyjuk ezért ezt a területet addigra.
Jelen modellezési ismeretek fként a fogalmak megértését az SM projekt használatának elkészítését segítik, nem törekszik teljességre.
Mj: Ha ezen elosztással valakinek párhozamosan végzend feladatai lennének - és azt MS project-el történt modellezésünk majdan jelzi, akkor majd módosítjuk tervünket. Ha lehetséges a megvalósító személyét, Vagy a megvalósító személyét, vagy az elfeltételeket módosítjuk majd.

Az elfeltételekkel kiegészített táblázat:

Sorszám :	*Tevékenység megnevezése* :	*Tevékenység rövidítése* :	*Tevékenység idő* :	*Az előfeltételi tevékenység száma* :
0.	*Kezdés* (*Start*)	*Start*	0	-
1.	*Szoba porszívózás*	*Szobaporsziv*	10 *perc*	0
2.	*Szobában letörölgetni*	*Szobaletorol*	15 *perc*	1

3.	*Szobában feltörölni*	*Szobafeltor\ ol*	5 *perc*	2
4.	*Fürdőszobában szerelvények lesúrolása*	*Furdosurol*	25 *perc*	0
5.	*Fürdőszoba felmosása*	*Furdofelmo\ s*	5 *perc*	4
6.	*Konyhában bútorok lesúrolása*	*Konyhasur\ ol*	25 *perc*	0
7.	*Konyha felmosása*	*Konyhafelm\ os*	5 *perc*	6
8.	*Előszoba felsöprése*	*Eloszobaso\ por*	5 *perc*	3; 5; 7
9.	*Előszoba felmosása*	*Eloszobafel\ mos*	5 *perc*	8
10.	*Kész a takarítás*	*Finish*	0	9.

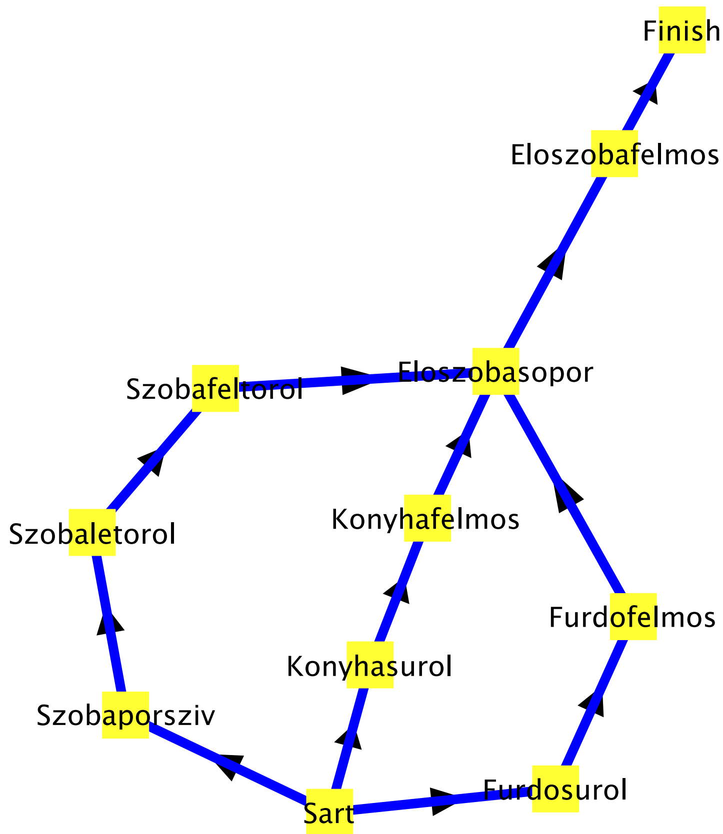
Garzon lakás takarítás elfeltételek magadásával

Ennek - könnyen megkonstruálható - gráfja az alábbi:

> *Gtakaritas* := *Graph*({[*Sart*, *Szobaporsziv*], [*Sart*, *Furdosurol*], [*Sart*, *Konyhasurol*], [*Szobaporsziv*, *Szobaletorol*], [*Szobaletorol*, *Szobafeltorol*], [*Furdosurol*, *Furdofelmos*], [*Konyhasurol*, *Konyhafelmos*], [*Szobafeltorol*, *Eloszobasopor*], [*Furdofelmos*, *Eloszobasopor*], [*Konyhafelmos*, *Eloszobasopor*], [*Eloszobasopor*, *Eloszobafelmos*], [*Eloszobafelmos*, *Finish*], });

Gtakaritas := Graph 23: a directed unweighted graph with 11 vertices and 12 arc(s) **(4.1)**

> *DrawGraph*(*Gtakaritas*, *style* = *spring*);



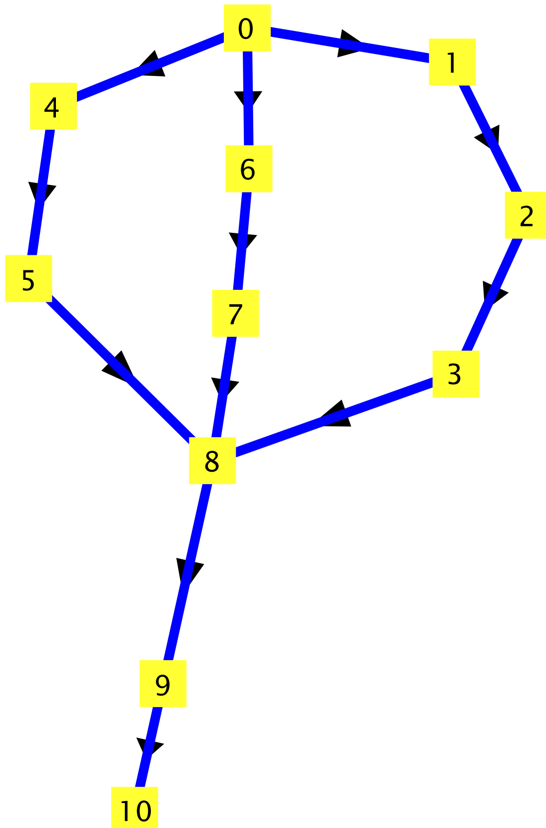
>

Vagy csak számokkal megjelenítve - mert ekkor "szebb" lefelé haladó gráfot rajzol a Maple beépített algoritmusa:

> *Gtakaritasnum* := *Graph*({[0, 1], [0, 4], [0, 6], [1, 2], [2, 3], [4, 5], [6, 7], [3, 8], [5, 8], [7, 8], [8, 9], [9, 10]});

Gtakaritasnum := Graph 25: a directed unweighted graph with 11 vertices and 12 arc(s) **(4.2)**

> *DrawGraph*(*Gtakaritasnum*, *style* = *spring*);



A fentit nevezzük tevékenység típusú gráfnak, mivel a csomópontok tevékenységeket jelentenek. Azonban mivel a tevékenységeknek ideje van a gráfoknál pedig súlyok az éleken jeleníthetek meg, további reprezentációnkban a tevékenységeket nem a csomópontokban, hanem az éleken fogjuk megjeleníteni. Ezt nevezzük esemény típusú gráfnak, ahol a csomópontok az oda befutó éleken végzett tevékenységek befejezését (és az innen kiindulók kezdését) jelentik.
Eljárásaink a továbbiakban ilyen - esemény - típusú gráfokra vonatkoznak majd.

Tevékenység típusú gráf:=
Tevékenység láncolat olyan gráf reprezentálása, ahol a csomópont tevékenységet az irányított élek pedig egymásutániságot jelentenek.

Esemény típusú gráf:=
Tevékenység láncolat olyan gráf reprezentálása, ahol az élek jelentik a tevékenységeket a csomópontok pedig események, melyek az adott csomópontba befutó éleken végzett tevékenységek befejezésének eseményét jelentik. (És természetesen az innen kiinduló éleken végzett tevékenységek kezdését is.)

Példa esemény típusú hálóra:
Konstruáljuk meg fenti példa esemény típusú hálóját. Ekkor nincs szükség Start tevékenységre, mert az a 0 esemény, csomópont jelenti a kezdést, hasonlóan az utolsó esemény a befejezés.

(Megkülönböztetésül nem teszünk pontot a csomópontok száma után.)
Most nincs szükség a tevékenységek rövidítésére, mivel az éleken a tevékenység idt fogjuk megjeleníteni.
Nincs szükség az elfeltételek megjelenítésére sem, mivel azt az élek egymásutánisága tartalmazza.

A tevékenységek élekké történt transzformálásakor kapott táblázat:

Indulási csomópont száma :	*Érkezési csomópont száma* :	*Tevékenység megnevezése* :	*Tevéken\ység idő* :
0	1	*Szoba porszívózás*	10 *perc*
1	2	*Szobában letörölgetni*	15 *perc*
2	3	*Szobában feltörölni*	5 *perc*
0	4	*Fürdőszobában szerelvények lesúrolása*	25 *perc*
4	5(3)	*Fürdőszoba felmosása*	5 *perc*
0	6	*Konyhában bútorok lesúrolása*	25 *perc*
6	7(3)	*Konyha felmosása*	5 *perc*
7(3)	8	*Előszoba felsöprése*	5 *perc*
8	9	*Előszoba felmosása*	5 *perc*
9	10	*Kész a takarítás*	0

Garzon lakás takarítás elfeltételek magadásával

Átalakításunk közben természetes módon adnánk meg 5-ös és 7-es csomópontokat, eseményeket mikor is a "Fürdszoba felmosása" illetve a "Konyha felmosása" tevékenységek befejezdnek. Tovább haladva azonban látjuk, hogy ezeknek célszer azonos csomópontot megadni. Ezért szerepel mellettük zárójelben - az elször megjelen - 3-as csomópont száma.
Reprezentációnkat elkészíthetjük tehát egyrészt úgy, hogy jelenítünk meg 5-ös és 7-es csomópontot, de - az ezen eset kezelésére alkalmas technikával - nullás tevékenység idej, logikai kapcsolatot leíró tevékenységgel azonos idpontban bekövetkezvé tesszük a legutolsó (7-es) tevékenységet.
Vagy már a háló rajzolása eltt beépítjük ezen felismerésünket és csak 3-as csomópontként jelenítjük meg mindhárom tevékenység befejezdésének eseményét.
Így tudjuk az "Elszoba felsöprés" tevékenységet megelzként megadni mindhárom tevékenységet.
Lássuk mindkét fajta reprezentációt:

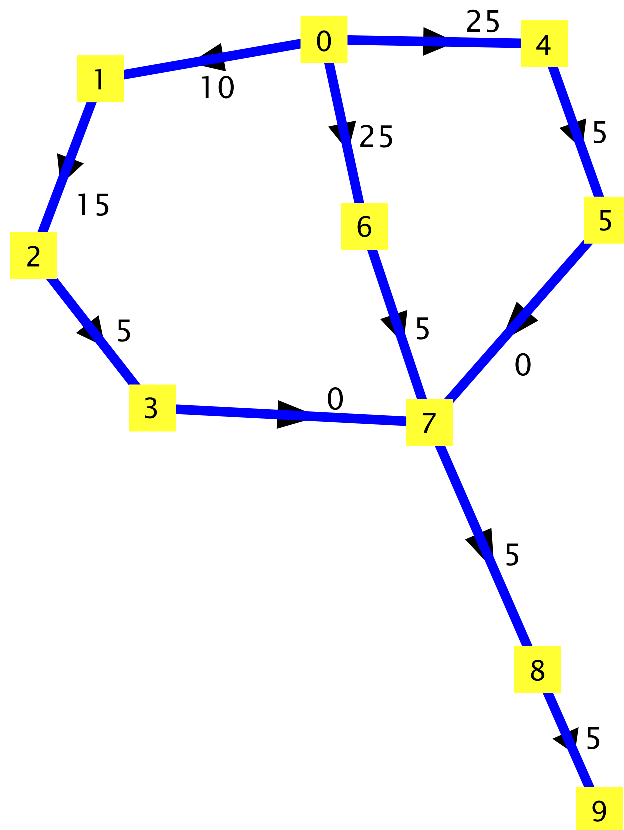
A példa esemény típusú gráfja (több, logikai kapcsolattal összekötött csomópont segítségével):

```
> Gtakaritasesemenytip := Graph({[[0, 1], 10], [[1, 2], 15], [[2, 3], 5], [[0, 4], 25], [[4,
    5], 5], [[0, 6], 25], [[6, 7], 5], [[6, 7], 5], [[3, 7], 0], [[5, 7], 0], [[7, 8], 5], [[8, 9],
```

5]});

Gtakaritasesemenytip := Graph 27: a directed weighted graph with 10 vertices and 11 arc(s) **(4.3)**

> *DrawGraph*(*Gtakaritasesemenytip*, *style* = *spring*);



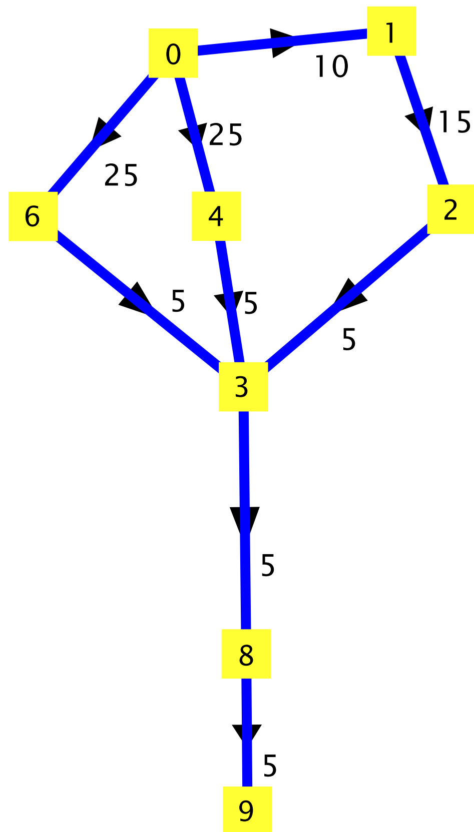
>

A példa esemény típusú gráfja (a logikai kapcsolattal azonossá tett csomópontok egy csomóponttal történ megjelenítésével):

> *Gtakaritasesemenytip2* := *Graph*({[[0, 1], 10], [[1, 2], 15], [[2, 3], 5], [[0, 4], 25], [[4, 3], 5], [[0, 6], 25], [[6, 3], 5], [[6, 3], 5], [[3, 8], 5], [[8, 9], 5]});

Gtakaritasesemenytip2 := Graph 28: a directed weighted graph with 8 vertices and 9 arc(s) **(4.4)**

> *DrawGraph*(*Gtakaritasesemenytip2*, *style* = *spring*);



Mj: Ilyen típusú átalakításra a gyakorlati életben (szerencsére) nincs szükség, mivel a projekteket kezel, leginkább elterjedt MS Projekt program automatikusan elkészíti számunkra tevékenység láncolatunk (tevékenység típusú) gráf reprezentációját. Ezen átalakítást csak a teljesség kedvéért adtuk meg.

A továbbiakban tehát a fenti esemény típusú (a tevékenységek az éleken jelennek meg) gráfokkal megadott tevékenység láncolatok elemzésével fogunk foglakozni.

A továbbiakban az alábbi módszereket használjuk majd a teljes tevékenység láncolat idejének kiszámítására:

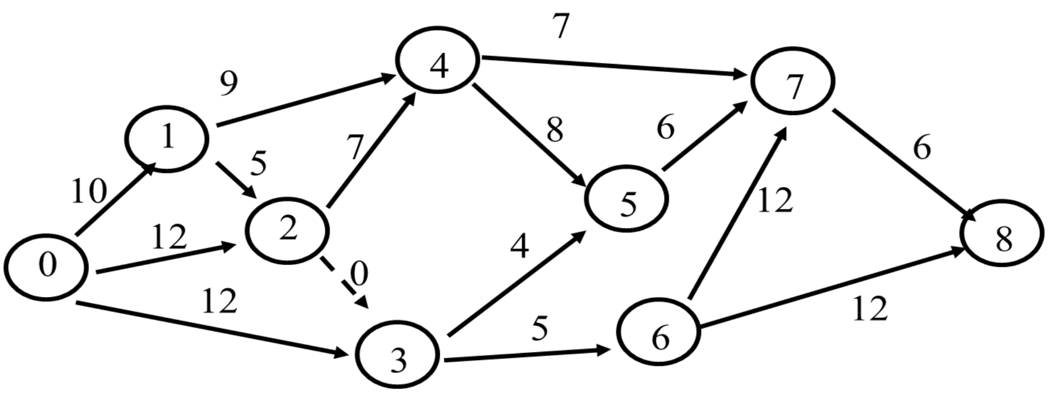
Kritikus út módszer
(angolul Critical Path Method rövidítve: CPM módszer)
A tevékenység láncolatban a legkisebb csomóponttal kezdve rendre meghatározza a csomópontok (események) bekövetkezési idejét és így jut el az utolsó (legmagasabb számú, vég,-) csomóponthoz, melynek bekövetkezési ideje egyben a tevékenység láncolat elvégzéséhez szükséges id (illetve idpont) is.

PERT módszer:
A PERT (Program Evaluation Review Technique) módszer annyiban különbözik a fenti CPM módszertl, hogy a tevékenység idket valószínségi változóknak tekinti, és várható értékeikkel dolgozik.

Kritikus út meghatározása: CPM (Critical Path Method) Módszerrel

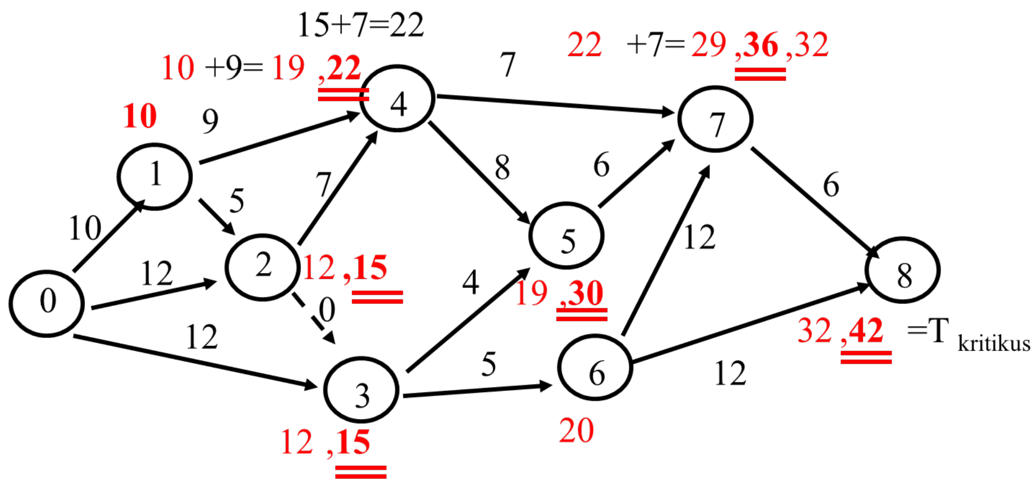
Tevékenység láncolatok esetén általában a teljes tevékenységlánc idejére vagyunk kíváncsiak. Ennek kiszámítására alkalmas az úgynevezett kritikus út módszer (angolul: Critical Path Method, CPM).
Kézi számítással a módszer a következ. Minden csomóponton megjelenítjük az adott csomópont a bekövetkezési idpontját. Bekövetkezési idponton értjük azt értjük az összes ide befutó tevékenység befejezésének idpontját - mivel csak ezek végeztével lehet az innen kiinduló tevékenységeket megkezdeni. Ez tulajdonképpen a bejöv utakon számítható idk maximuma.

Az alábbi rajz ezen számításra mutat példát.
Az éleken az adott élhez tartozó tevékenység ideje szerepel, napban.



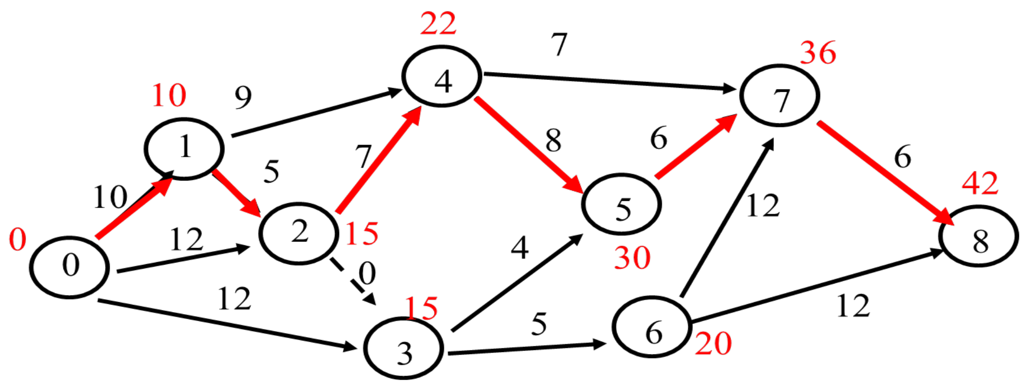
Számítsuk most ki rendre a csomópontok bekövetkezési idpontjait úgy, hogy a tevékenység nyíl kiindulópontjában lév csomópont bekövetkezési idejéhez
adjuk hozzá a nyílon lév tevékenység idt. Pl. 0-2 nyíl, a 0-ás kezd esemény bekövetkezési idpontja a nulladik nap, mivel ez a projekt kezdetekor azonnal "bekövetkezik". A (0-2) tevékenység ideje 12 nap így a 2-es csomópont bekövetkezési ideje (ezen az úton) 12. Azonban a 0-1 tevékenység miatt az 1-es csomópont bekövetkezési ideje 10, az 1-2 tevékenység miatt pedig a 2 csomópontté 10+5 = 15. Mivel az innen kiinduló tevékenységek megkezdéséhez minden ide befutó tevékenységnek be kell fejezdnie, ezért a 2. csomópont bekövetkezési ideje (amikor meg lehet kezdeni az innen kilép tevékenységeket) az 15. amit nemcsak félkövérrel emeltünk ki, de ketts aláhúzással is.
Nulla tevékenység idvel a logikai kapcsolatot jelezzük, vagyis hogy a 3-as csomópontból kiinduló tevékenységek csak a kettesbe befutók után kezdhetk meg. A 3-as csomópont azonban nem lehet azonos a 2-essel, mivel akkor a 0-ásból két él vezetne - amit pedig a hálóink megadásában nem tudunk kezelni. Ezért kellett "széthúznunk" ezt a csomópontot és a logikai kapcsolatot jelent 0-ás

(szaggatottal jelölt) technikát használnunk. A továbbiakban az csomóponton az összes oda vezet úton számítható bekövetkezési idpont - pirossal - és kiemelve, kétszer aláhúzva ezek maximuma. Nem tudtuk az összes csomópont összes megközelítésénél lév számok elállítását is megjeleníteni, de reméljük az alábbi gráf is elégséges lesz a módszer megértéséhez.
Mj: Sajnos a Maple nem ad (a szerz által ismert) lehetséget animáltan történ megjelenítésre, az (a sokkal elterjedtebb) MS PowerPoint programmal sokkal könnyebben megjeleníthet. Ez el is készült és a hallgatók rendelkezésére is áll.)



Tehát a fenti tevékenység láncolat elvégzéséhez szükséges id 42 nap - mivel ez a végcsomópont bekövetkezési ideje.
Ezen idt kritikus idnek szoktuk nevezni, mivel ez a legrövidebb id amely alatt az egész projekt elvégezhet ami miatt minimumnak lehetne neveznünk. Azonban ez a maximuma a kezd csomópontból a végcsomópontig vezet utaknak - ami miatt pedig maximális idnek kellene neveznünk. Ezek miatt (maximum és minimum is egyszerre) nevezzük kritikus idnek.
Fontos még hogy mely tevékenységek miatt alakul ki ez az id?
Ezt "visszafelé" a legkönnyebb meghatározni. Mely befutó tevékenység alakította ki a 42-es értéket? Válasz: a 7-8. Mely tevékenység miatt lett 36 a 7 csomópont bekövetkezési ideje? Válasz: 5-7. és így tovább. A pirossal jelöltük az úgynevezett kritikus utat, mely a kritikus idt adó tevékenység láncolat.

A alábbi rajzon csak a csomópontok bekövetkezési idpontja került megjelenítésre, pirossal, valamint - szintén pirossal - a kritikus út, mely a kritikus idt adó tevékenységek sorozata.



Mj: A logikai kapcsolatot jelent nullás idej tevékenység is lehetne kritikus út része - csak esetünkben nem az.

Tevékenység láncolatok kritikus idejének és tartalék idejének számítása küldönböz háló megadások esetén

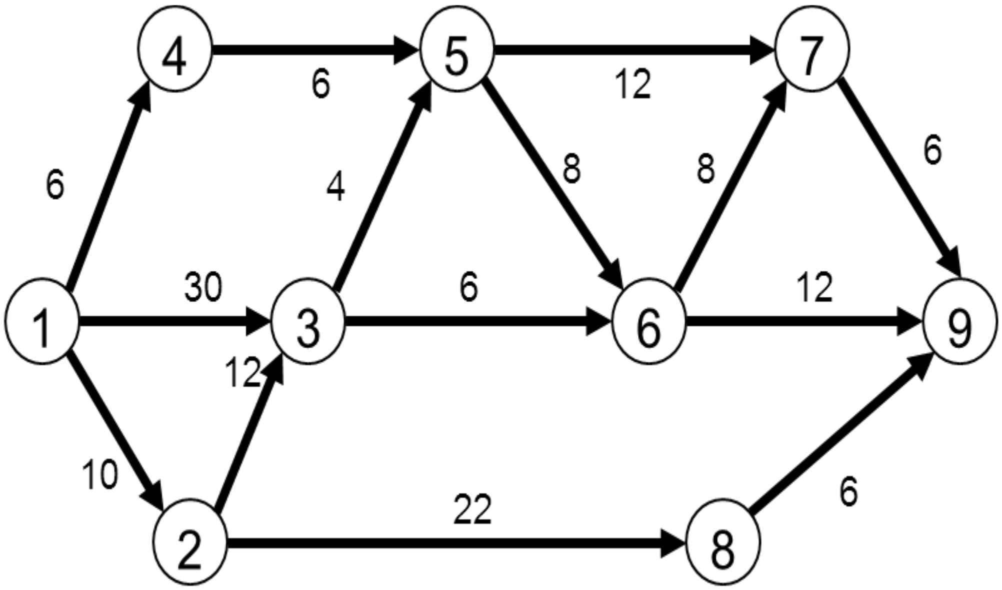
A fenti módszer szemléletes a már megrajzolt tevékenységláncolati gráfon. Azonban a gráf nem mindig áll rajzként rendelkezésünkre. A gráfoknak sok más területen történ alkalmazásához, más modellek megértéséhez is jó gyakorlat táblázattal, vagy mátrixszal megadott gráf grafikus reprezentációjának felrajzolása.

A gráfok megadásának - legtöbbször elforduló - lehetségeit a gráf elméleti részben a "Gráfok megadás", második fejezet tárgyalja.
A számunkra elforduló reprezentációk az "Adjecancia lista irányított gráfra" és a "Szomszédossági vagy adjecencia mátrix".
Nézzünk most példát ezekre.

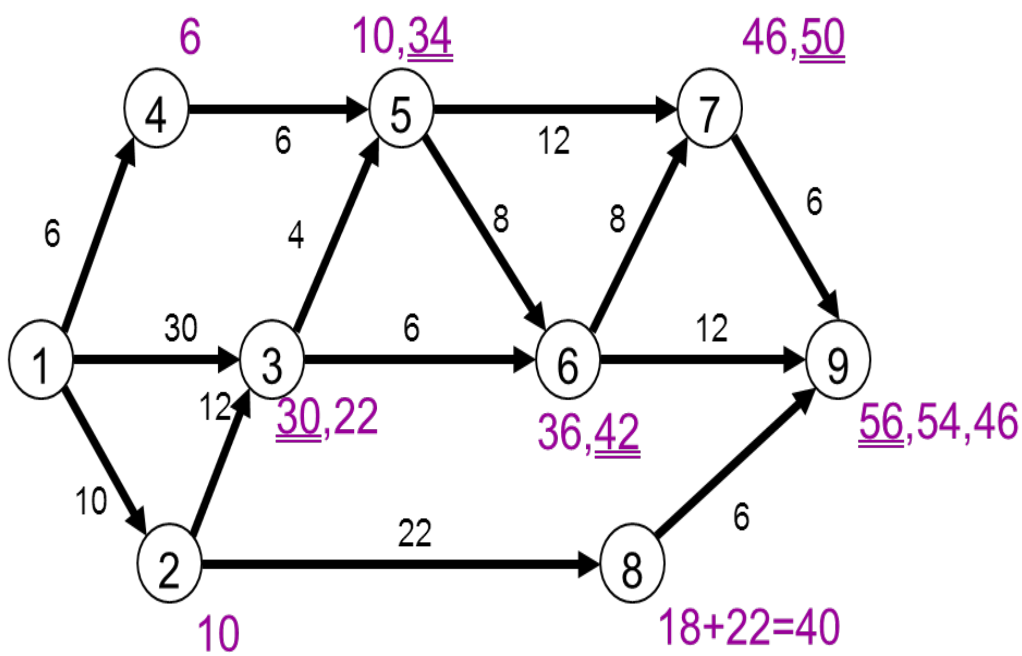
Rajzoljuk fel az alábbi táblázattal, vagyis irányított gráfra megadott adjacencia listával megadott - irányított súlyozott - gráfot vagy az ezen a területen szokásos szóhasználattal a hálót.

i	1	1	1	2	2	3	3	4	5	5	6	6	7	8
j	2	3	4	3	8	5	6	5	6	7	7	9	9	9
t_i	18	4	6	12	22	4	6	6	8	1\	8	1\	6	6

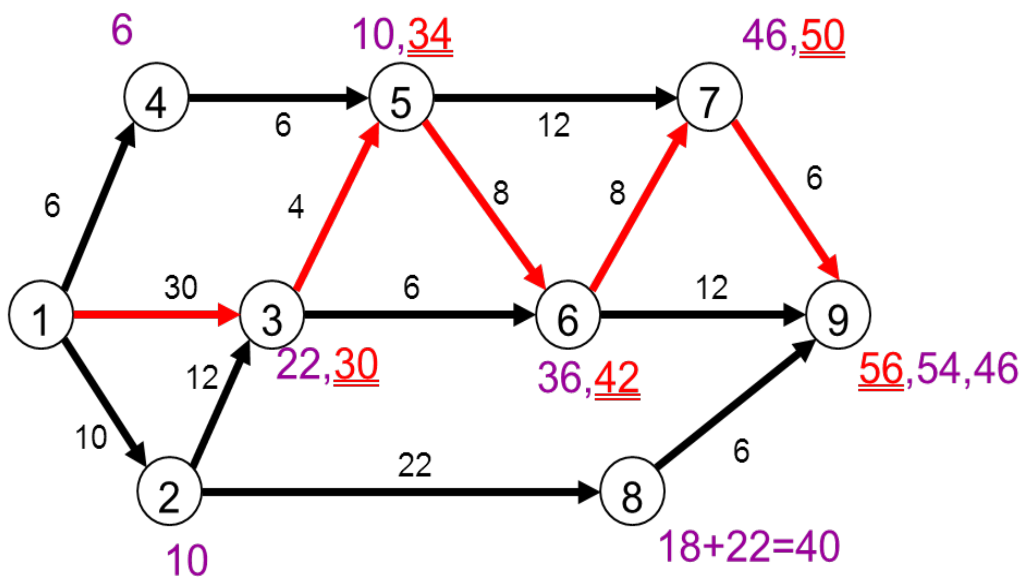
Ha elsre nem sikerül jól elhelyezni a csomópontokat célszer lehet újrarajzolni a hálót.



Most már alkalmazható a kritikus út számolásának az elz fejezetben megismert módszere. A csomópontokon rendre a bekövetkezési idpontokat jelenítettük meg, aláhúzással pedig ezek maximumát jelöltük. (Mivel csak ha minden bejöv élen végzett tevékenység befejezdik, akkor lehet megkezdeni a kimen éleken szerepl tevékenységeket.)



A kritikus út visszakeresése pedig:



A kritikus út: 1 - 3 - 5 - 6 - 7 - 9.

A kritikus úton szerepl tevékenységeknek nem szabad késnie, mert ha igen akkor teljes tevékenység ideje (kritikus id) késedelmet szenvedne.
Fontos kérdés lehet számunkra, hogy a többi tevékenység késedelme hogyan befolyásolja a végbefejezést. Azt, hogy egy adott tevékenység mennyit késhet a nélkül, hogy a végbefejezést késleltetné, tartalék idnek hívjuk.
Számítsuk ki az 6-9 tevékenység tartalék idejét!
Ez egyszer mivel biztosan be kell fejeznünk a 6-9 tevékenységet az 56. napra. Elkezdeni pedig csak a 42. napon lehet. Vagyis összesen (56-42) = 14 nap áll rendelkezésünkre a 12 nap tevékenység idej tevékenységünk megvalósítására. Tartalék, késlekedési idnk tehát csak 2 nap marad.
Hasonlóan a 3-6 tevékenységre:
Ahhoz, hogy a végbefejezést ne késleltessük, a 42. napra be kell fejeznünk. (Úgy is számolhatunk, hogy 56 - (6+8) = 42 vagyis, hogy a végbefejezés eltt hagynunk kell idt az utánunk következ tevékenységeknek, de mivel a kritikus út tevékenységei nem késhetnek így számítási technikaként javasolt inkább a kritikus úton lév csomóponton szerepl szám használata, mivel a "visszaszámolás" elszámolási lehetséget rejthet magában.) Elkezdeni a 30. napon kezdhetjük - mivel ekkorra fejezdik a szükséges megelz tevékenység. Így (42 - 30) = 12 nap áll rendelkezésünkre a 6 napot igényl tevékenységünkhöz. Ezért tartalékidnként 6 nap áll rendelkezésünkre.

Tekintsük most a 8-9 tevékenységet. Ezt is - triviális módon - az 56. napra kell befejeznünk.
Elkezdeni pedig a 40. napon lehet. Így (56-40) = 16 nap áll rendelkezésünkre a 6 nap idej tevékenységünk elvégzéséhez - ami 10 nap tartalék idt jelent.

Algoritmikusan megjelenítve a tartalék id:

Tartalék id = Legkésbbi befejezés - Legkorábbi kezdés - Tevékenység id

Esetünkben: **pm (6-9)** = 56 - 40 - 6 = 10 nap

Nézzünk még nem ennyire triviális esetet:

pm (2-8) = ?

Itt már a tevékenység befejez csomópontja sem nem a végs csomópont sem nem része a kritikus útnak. Az elv azonban itt is alkalmazható. A legkésbbi (olyan befejezés, mely nem késlelteti a teljes projekt befejezését) úgy számítható ki, ha megnézzük, hogy tevékenységünk után még milyen tevékenységet kell elvégezni - amelynek "idt kell hagynunk". Esetünkben ez a (8-9) tevékenység melynek ideje 6 nap. Vagyis legkésbben az (56-6).=50. napon be kell fejeznünk a (2-8) tevékenységet. Elkezdeni - a kezd csomóponton álló számnak megfelelen - a 10. napon lehet. Tevékenységünk ideje pedig 22 nap. Vagyis:

pm (2-8) = (56 - 6) - 10 - 22 = 18 nap

pm (2-3) = ?

Végcsomópontja kritikus útra esik bekövetkezési ideje a 30. nap. Kezdeni a 10 napon lehet. Tevékenység ideje: 12 nap Vagyis:

pm (2-3) = (30) - 10 - 12 = 8 nap

Még kevésbé triviális az (1-2) tevékenység tartalék ideje.

pm (1-2) =?

Végcsomópontja nem esik kritikus útra, ezért az ezt követ tevékenységeknek idt kell hagynunk. De melyikeknek? A (2-3)-as ágon becsatlakozva a kritikus útba és úgy tovább a végbefejezésig, vagy a 2-8-9 ágon? általános elv, hogy mindegyiknek, mivel egyetlen "ágon", úton sem késleltethetjük a végbefejezést. Vagyis amelyik "visszaszámolás" kisebb értéket ad azt kell tekintenünk legkorábbi befejezésnek. megjegyezzük azért, hogy amely "ág" korábban lép be a kritikusútba az szokott kisebb értéket szolgáltatni, mivel ez kisebbet biztosan nem szolgáltathat.
Tehát a legkésbbi befejezés idpontja: (30 - 12) = 18 a legkorábbi kezdés a 0. nap a tevékenység id pedig 10 nap. Így a tartalék id:

pm (1-2) = (30-12) - 0 - 10 = 8 nap

Ez láthatóan megegyezik a (2-3) tevékenység tartalék idejével! Ez érthet is hiszen ha az (1-2) tevékenység késik például 4 napot akkor a (2-3) már annyival késbben tud csak kezdeni, vagyis számára már csak 4 nap késedelmi lehetség marad - a közös 8 napból. Vagyis a tartalék id "áganként" értend. Egy a kritikus útból kilép majd oda visszatér ágnak közös tartalékideje van, melybl, ha az egyik tevékenység elhasznál valamennyit a többinek már csak annyival kevesebbre lesz lehetsége. Úgy is fogalmazhatunk, hogy a tartalékid számítása a (már jól ismert) "Ceteris Paribus" elv alapján zajlik, vagyis úgy számolunk egy tevékenységre tartalék idt, hogy azt feltételezzük, hogy egyetlenegy másik tevékenység sem fog késlekedni.

Ez alapján elégséges csak a (4-5) tevékenység tartalékidejét kiszámítani az (1-4) re ugyanaz az érték adódik. (Csak a kivonandó számok csoportosítása más.)

pm (4-5) = 34 - 6 - 10 = 18 = pm (1-4) = (34-10) - 0 - 6 = 18

A kritikus id és a kritikus út meghatározása Maple eljárások segítségével

Nézzük most, milyen lehetségeket kínál a Maple ezen számítások kiküszöbölésére, ellenrzésére. Alkalmas input ablakban (adott struktúrában megadott) háló felrajzolására és (amennyiben felsháromszög mátrixként van megadva a gráf) a kritikus id kiszámítására a kritikus út felrajzolására.
(Ezen eljárások jelen pályázat keretében kerültek kifejlesztésre.)

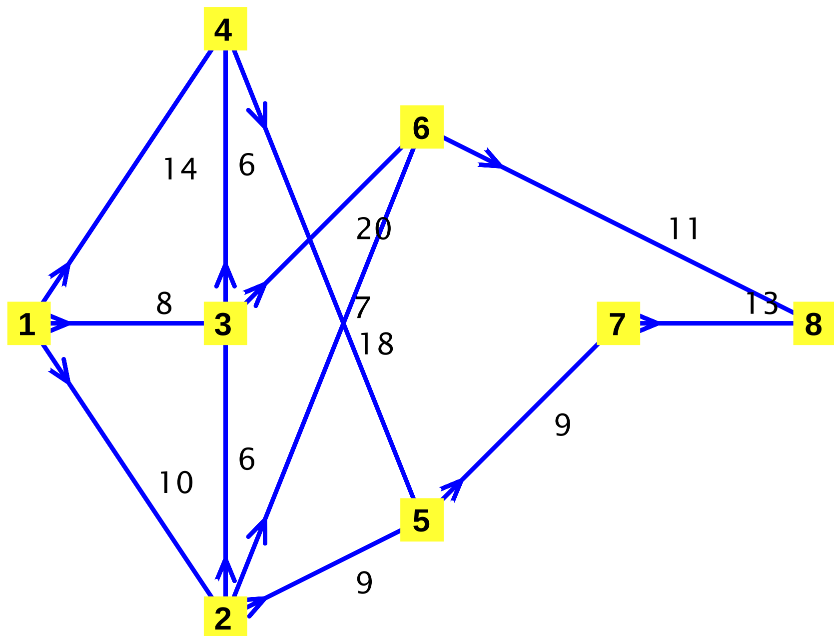
A hálózat megadása:
Az alábbi példán látható módon adható meg egy él: szögletes zárójelek közé az induló és az érkezési csomópont számát, és ismét vesszvel elválasztva az élen szerepl tevékenység idt, és az egészet ismét szögletes zárójelbe tenni. A következ él vesszvel elválasztva hasonló módon következik. A legutolsó él után nem szabad vesszt tenni!

[[1, 2], 10], [[1, 3], 8], [[1, 4], 14], [[2, 3], 6], [[2, 5], 9], [[3, 4], 6], [[2, 6], 18], [[4, 5], 7], [[3, 6], 20], [[5, 7], 9], [[6, 8], 11], [[7, 8], 13]

A rendelkezésre álló eljárások:

InputGraph1( ); - beolvasás a fenti MathContNet1 nev input ablakból ahol a fenti módon kell megadni a hálót
CheckMatrix( ); - ellenrzi, hogy megfelelen került-e megadás ra mátrix, (csak fels háromszög mátrixként megadott hálóval tud dolgozni az eljárás, vagyis csak kisebb csomópontból nagyobb felé mutathat nyíl)
CPM( ); - a kritikus út meghatározó eljárás (megadja a kritikus idt, a kritikus út csomópontjait (visszafelé) és a kritikus út éleit "elre"
DrawCPM( ); - a gráf és a kritikus út kirajzoltató eljárása (mind a kritikus út csomópontjai mind az élei pirossal kerülnek kiemelésre)

Beolvasás:
InputGraph1( );



Ellenrzés: (fels háromszög mátrix-e)

CheckMatrix( );

"Jó a háló megadása! Fels háromszg a mátrixa" **(7.1)**

Kritikus út meghatározás:

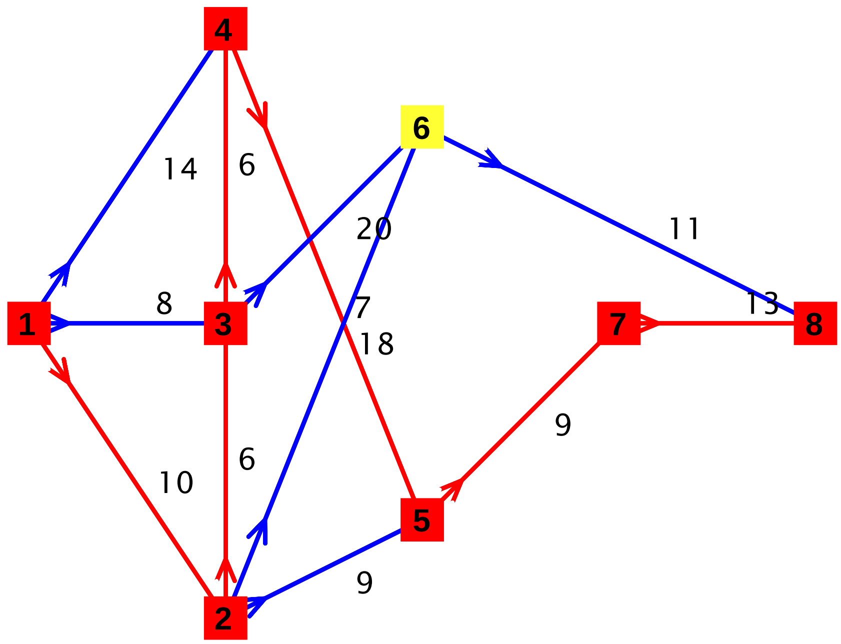
CPM( );

"Kritikus idő:", S_8, " Kritikus út csomópontjai:" [8, 7, 5, 4, 3, 2, 1], **(7.2)**

" Kritikus út élei:", {[1, 2], [2, 3], [3, 4], [4, 5], [5, 7], [7, 8]}

Háló és benne a kritikus út felrajzolása:

DrawCP( );



Tartalék id számításra - egyenlre - nem készült eljárás.

Összefoglalás

A hálótervezés témakörben megismerkedtünk egyrészt néhány gráf elméleti alapfogalommal, gráfok megadásának legfontosabb lehetségeivel - mivel ezen ismeretek szükségesek voltak a tevékenységláncolatok modellezéséhez.
Példán keresztül láttuk a tevékenység és az esemény típusú gráfok megkonstruálását adott feladatra.
Szintén példán keresztül, - sajnos ugyan nem animáltan - láttuk esemény típusú háló esetére a kritikus id (teljes tevékenységláncolat ideje) és kritikus út (az ezen idt meghatározó tevékenység láncolat) meghatározásának a csomópontok bekövetkezési idejére épül módszerét.
Definiáltuk a tartalék id fogalmát, mely azon idtartam mellyel megnövelve egy adott tevékenység idejét még nem késlelteti a végbefejezést. Megállapítottuk, hogy ez a kritikus úton - triviálisan - nulla. Példát mutattunk tartalék id számításra
Végezetül (ezen pályázat keretében kifejlesztett) Maple eljárásokat mutattunk be input ablakban - a

Maple gráf megadási szintaktikájának megfelelen - megadható hálók felrajzolására, a kritikus út és kritikus id kiszámítására. Ehhez szükség volt még a háló megadás fels háromszög voltának ellenrzésére is.
Tartalék id számításra - egyenlre - nem áll rendelkezésre eljárás.
Ezen eljárások haszonnal alkalmazhatóak oktatóknak feladat készítéskor illetve hallgatók számára kézzel készített megoldásaik ellenrzésére.

Feladatok

1. Az alábbi hálótervezési feladatra adja meg a kritikus idt és a kritikus uta(ka)t!
Határozza meg a 5 – 6 , 5 - 9 és a 6 – 8 tevékenységek tartalék időit!

i	1	1	2	3	3	4	4	4	5	5	6	6	7	8	9
j	2	3	5	4	5	5	6	7	6	9	8	9	8	10	10
$t_{i,j}$	6	6	2	2	6	0	14	18	8	10	8	10	10	10	2

2. Az alábbi hálótervezési feladatra adja meg a kritikus idt és a kritikus uta(ka)t!
Határozza meg a 3 – 5 , 7-8 és a 7 – 9 tevékenységek tartalék időit!

i	1	1	1	2	3	4	5	5	6	7	7	8	9
j	2	3	4	4	5	6	6	7	7	8	9	10	10
$t_{i,j}$	4	10	10	6	0	8	8	2	4	6	2	4	6

3. Adja meg az alábbi hálótervezési feladatban a kritikus idt és kritikus uta(ka)t, valamint az alábbi tevékenységek tartalék idit!

pm1,2=? pm 3,5=? pm5,6=? pm6,7=?

4. Készítsük el a mátrixszal megadott hálótervezési feladat gráfos reprezentációját, majd számítsuk ki a kritikus idt, és adjuk meg a kritikus utat. Határozzunk meg tartalék idt a 2-8 tevékenységre!

	1	2	3	4	5	6	7	8	9	10
1		6	3	5						
2					4			5		
3						7				
4						4	6			
5								3		
6								4	3	

7									2	
8										6
9										5

5. Határozza meg az alábbi hálótervezési feladatra a kritikus idt és a kritikus uta(ka)t!
Adja meg a 4 – 6 , 5 - 9 és a 6 – 8 tevékenységek tartalék időit!

i	1	2	2	3	3	4	4	4	5	5	6	6	7	8	9
j	2	3	5	4	5	5	6	7	6	9	8	9	8	10	10
$t_{i,j}$	6	6	4	4	16	1	4	8	6	11	7	5	7	9	5

6. Határozza meg az alábbi hálótervezési feladatra a kritikus idt és a kritikus uta(ka)t!
Adja meg a 4 – 6 , 5 - 9 és a 6 – 8 tevékenységek tartalék időit!

i	1	1	2	3	3	3	4	4	4	5	5	6	6	6	7	8	9
j	2	4	4	4	5	7	5	6	7	6	9	9	8	9	8	10	10
$t_{i,j}$	6	6	4	16	3	16	1	4	8	6	11	2	5	11	9	9	5

7. Határozza meg az alábbi hálótervezési feladatra a kritikus idt és a kritikus uta(ka)t!
Adja meg a 3 – 4 , 5 - 6 és a 6 – 9 tevékenységek tartalék időit!

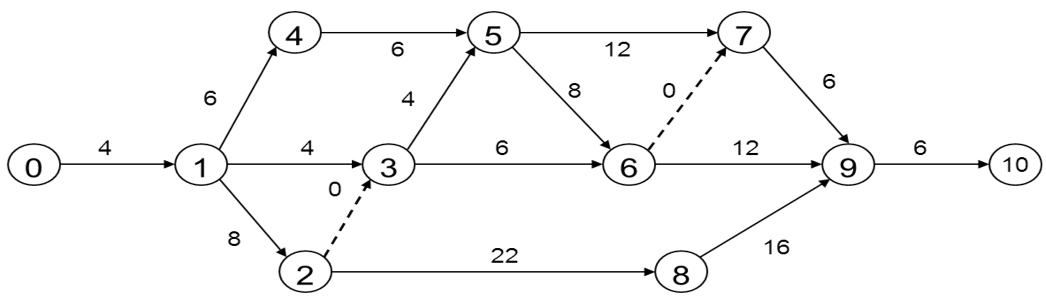
i	1	1	2	3	3	3	4	4	4	5	5	6	6	6	7	8	9
j	2	4	4	4	5	7	5	6	7	6	9	9	8	9	8	10	10
$t_{i,j}$	6	6	4	4	16	16	1	4	8	6	11	2	5	11	9	9	5

8. Határozzuk meg az alábbi a mátrixszal megadott hálótervezési feladatban - a gráfos reprezentáció megrajzolása nélkül - a kritikus idt, és adjuk meg a kritikus utat (utakat)!

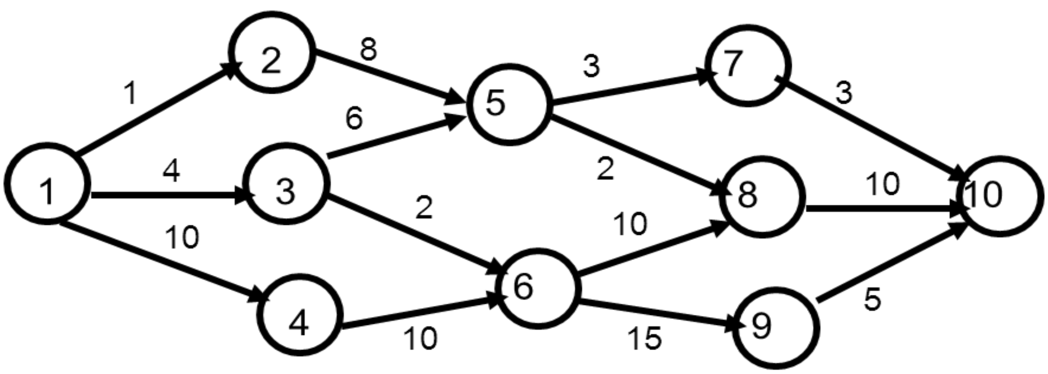
	1	2	3	4	5	6	7	8	9	10
1		2		4						
2			4	6		2		5		
3					7	7				
4						4	6			
5							4	11		

6								3	3	
7									2	
8										8
9										12

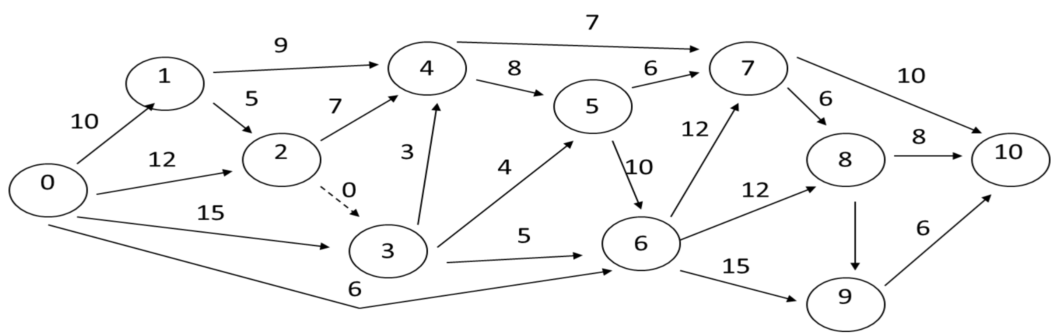
9. Adja meg az alábbi - gráffal megadott - hálótervezési feladatban a kritikus idt és kritikus uta(ka)t, valamint a 1- 2 , 3 – 5 , 5 - 7 , 9 – 10 , 3 – 5 , 1 – 3 tevékenységek tartalék időit!



10. Adja meg az alábbi - gráffal megadott - hálótervezési feladatban a kritikus idt és kritikus uta(ka) t, valamint a 1- 2 , 2 – 5 , 5 - 7 , 7 – 10 , 3 – 5 , 8 - 10 tevékenységek tartalék időit!



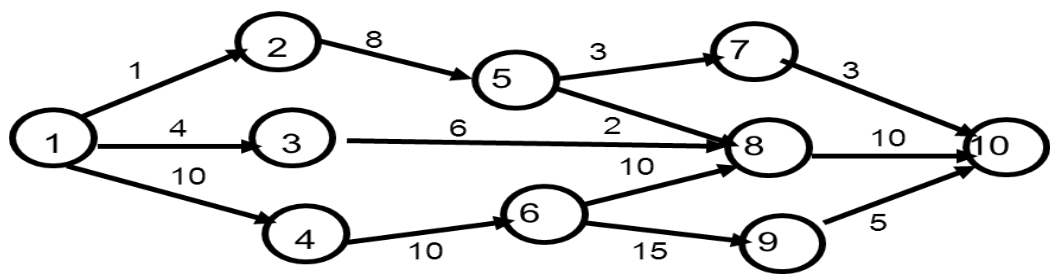
11. Adja meg az alábbi - gráffal megadott - hálótervezési feladatban a kritikus idt és kritikus uta(ka) t, valamint a 1- 2 , 2 – 4 , 5 - 7 , 7 – 10 , 3 – 5 , 6 - 9 tevékenységek tartalék időit!



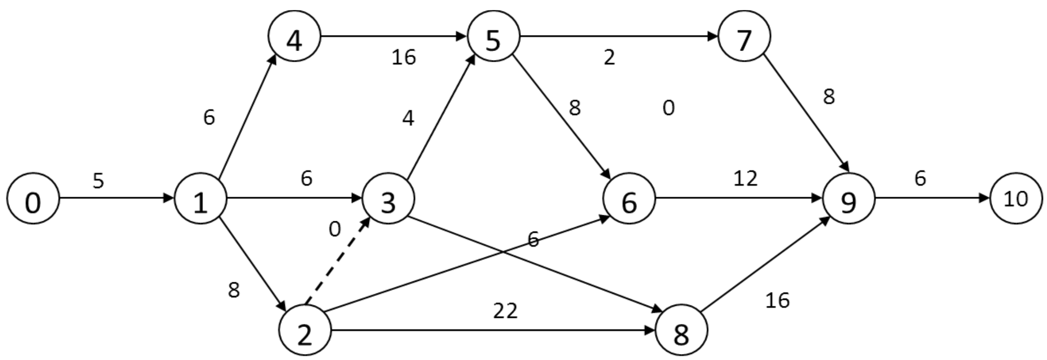
12. Határozza meg az alábbi hálótervezési feladatra a kritikus idt és a kritikus uta(ka)t!
Adja meg a 1 – 3 , 4 - 7 és a 6 – 8 tevékenységek tartalék időit!

i	1	1	1	2	3	4	4	4	5	5	6	6	7	8	9
j	4	2	3	4	5	5	6	7	6	9	8	9	8	10	10
$t_{i,j}$	6	8	5	8	15	1	11	6	4	8	2	4		9	5

13. Adja meg az alábbi - gráffal megadott - hálótervezési feladatban a kritikus idt és kritikus uta(ka) t, valamint a 1- 2 , 2 – 5 , 5 - 8 , 8 – 10 , 3 – 8 , 6 - 9 tevékenységek tartalék időit!



14. Adja meg az alábbi - gráffal megadott - hálótervezési feladatban a kritikus idt és kritikus uta(ka) t, valamint a 0 - 1 , 2 – 6 , 5 - 6 , 8 – 9 , 3 – 8 , 6 - 9 tevékenységek tartalék időit!



15. Határozza meg az alábbi hálótervezési feladatban a kritikus idt és a kritikus uta(ka)t!
Adja meg a 3 – 4 , 5 - 6 és a 7 – 8 tevékenységek tartalék időit!

i	1	1	2	3	3	3	4	4	4	5	5	6	6	6	7	8	9
j	2	4	4	4	5	7	5	6	7	6	9	9	8	9	8	10	10
$t_{i,j}$	6	2	2	9	5	1	1	2	8	6	11	2	5	11	9	9	5

Irodalom jegyzék

Felhasznált irodalom:

Csernyák László - Jánosa András: Operációkutatás II. A gazdasági optimalizálás módszerei II. Budapest, 2004, Nemzeti Tankönyvkiadó Rt.

Frederick S. Hillier - Gerard J. Liebermann: Bevezetés az operációkutatásba LSI Oktatóközpont, Budapest, 1994.

K. Sydsaeter - P. Hammond: Matematika Közgazdászoknak, Aula Kiadó, Budapest, 1998.

Ferenczi Zoltán: Operációkutatás, Készült a HEFOP 3.3.1-1.-P.-2004-09-0102/1. pályázat támogatásával, 2006.

Példatár az operációkutatás II. tananyaghoz (Egyéni tanulást segítkidolgozott feladatok) Pénzügyi és Számviteli Fiskola Budapest 1996. F. Sz.: 243.

Programok

► Minta input adatok