

# Modellezés és optimalizálás a gyakorlatban

## SZÖVEGES FELADATOK

- 1) Egy üzemben két terméket (T1, T2) gyártanak, megmunkálásuk két gépen (G1,G2) történik. A T1 termék egy darabjának megmunkálása a G1 gépen 3 órát, a G2 gépen 4 órát vesz igénybe, a T2 termék esetén 4 és 6 óra ez a két adat. A G1 gép kapacitása 130, a G2 gép kapacitása 200 óra. A T1 termék szerelési ideje 1 óra, a T2 terméké 2 óra, a szerelde kapacitása 60 óra. A T1 termék eladási egységára 8, a T2 termék eladási egységára 10 pénzegység. Milyen termékösszetételben érdemes gyártani, ha maximális árbevételre törekszünk, úgy, hogy a gépek kapacitását nem lépjük túl és a szerelde teljes kapacitással dolgozik (azaz egyenlőség van a szereldére vonatkozó feltételben)?

```
var T1 >= 0;
var T2 >= 0;

s.t. gep1: 3 * T1 + 4 * T2 <= 130;
s.t. gep2: 4 * T1 + 6 * T2 <= 200;
s.t. szerelesi_ido: 1 * T1 + 2 * T2 = 60;

maximize koltseg: 8 * T1 + 10 * T2;
end;
```

- 2) Egy anyuka gyermeke születésnapjára zsúrára kétféle szendvicset készít. Egy szalámis szendvicse 3 gramm vaját, 3 karika tojást és 2 szelet szalámit tesz. A sonkás szendvicshöz 4 gramm vaj, 2 karika tojás és 1 szelet sonka szükséges. A szendvicsek készítéséhez rendelkezésre áll 100 szelet szalámi, 40 szelet sonka, 170 karika tojás és 220 gramm vaj (kenyérszeletből nincs korlátozás, abból elegendően sok van). Melyik szendviczből hány darabot kell készítenie az anyukának, ha azt akarja, hogy a rendelkezésre álló készletekből összességében a lehető legtöbb szendvics készüljön el?

```
var sonkas >= 0;
var szalamis >= 0;

s.t. vaj: 3 * szalamis + 4 * sonkas <= 220;
s.t. tojas: 3 * szalamis + 2 * sonkas <= 170;
s.t. sonka: 1 * sonkas <= 40;
s.t. szalami: 2 * szalamis <= 100;

maximize szendoka: szalamis + sonkas;
end;
```

- 3) Egy hátizsákgyártó cégnél kétféle modellt gyártanak: a Túra és a Mini modelleket. Mindkettő ugyanabból a poliészterből készül. A cég a poliészter beszállítójával hosszútávú szerződést kötött, aminek keretében hetente 5400 m<sup>2</sup> anyagot dolgozhat fel. A Túra modell egy darabjához 3 m<sup>2</sup>, a Mini modell egy darabjához 2 m<sup>2</sup> anyagra van szükség. Piackutatási eredmények szerint a Túra modellből legfeljebb 1000, a Mini-ből legfeljebb 1200 darabot tudnak hetente értékesíteni. A Túra hátizsák elkészítése 45 perc, a Mini 40 perc munkaidőt igényel. Egy darab Túra hátizsákon 32\$ hasznuk van, míg egy darab Mini-n 24\$. A cégnél 35 dolgozó foglalkozik közvetlenül a hátizsák készítéssel, fejenként heti 40 óra munkaidőben. Adjunk tanácsot a cégvezetésnek, hogy mennyit érdemes az egyes hátizsákokból hetente elkészíteni!

```
var tura >= 0, integer;
var mini >= 0, integer;

s.t. anyag: 3 * tura + 2 * mini <= 5400;
s.t. munkaido: 45 * tura + 40 * mini <= 35 * (40 * 60);
s.t. turamodell: tura <= 1000;
s.t. minimodell: mini <= 1200;

maximize haszon: 32 * tura + 24 * mini;
end;
```

- 4) Egy építési vállalkozó kétféle üzemcsarnok építésével foglalkozik. Az A típusú csarnok 4000 m<sup>2</sup> lemez, 4 tonna acél, 300 m<sup>2</sup> tetőfedő anyag és 200 m<sup>3</sup> beton felhasználásával készül el. A B típusú csarnok felépítéséhez 5000 m<sup>2</sup> lemez, 3 tonna acél, 200 m<sup>2</sup> tetőfedő anyag és 100 m<sup>3</sup> beton szükséges. Az alapanyagok csak korlátozott mennyiségben állnak rendelkezésre: 32000 m<sup>2</sup> lemez, 24 tonna acél, 2000 m<sup>2</sup> tetőfedő anyag és 1600 m<sup>3</sup> beton használható fel az adott szezonban. Egy A típusú csarnok felépítése 4000 \$, a B típusú csarnok felépítése 5000 \$ nyereséget hoz. Milyen összetételben vállaljon a kétféle csarnokra vonatkozó megrendelésekből a vállalkozó, ha maximalizálni szeretné a nyereségét?

```
var A >= 0, integer;
```

```
var B >= 0, integer;
```

```
s.t. lemez: 4000 * A + 5000 * B <= 32000;
```

```
s.t. acél: 4 * A + 3 * B <= 24;
```

```
s.t. tetofedo: 300 * A + 200 * B <= 2000;
```

```
s.t. beton: 200 * A + 100 * B <= 1600;
```

```
maximize nyereseg: 4000 * A + 5000 * B;
```

```
end;
```

- 5) A távoli jövőben az optimalizálás vizsgára különböző ízesítésű tudástabletták bevitelével is fel lehet készülni. Egy spenótos tabletta elfogyasztása 10%-kal, egy eperízű tabletta elfogyasztása 7%-kal növeli meg a hallgató tudását. (Tehát 4 spenótos és 5 eperes tabletta esetén a tananyag 75%-ának lennének birtokában). Egy spenótos tabletta 2000, egy eperízű tabletta 1500 forintba kerül. A túladagolás elkerülése érdekében a két tablettából összesen legfeljebb 8 darab vehető be, és arra is ügyelnünk kell, hogy legalább annyi eprest vegyünk be, mint spenótost. Milyen összetételben vásároljunk a tablettákból, ha a lehető legkevesebbet szeretnénk költeni és 51% a sikeres vizsga teljesítésének feltétele?

```
var spenottabi >= 0, integer;
```

```
var eprestabi >= 0, integer;
```

```
s.t. mertek: spenottabi + eprestabi <= 8;
```

```
s.t. osszetetel: spenottabi <= eprestabi;
```

```
s.t. siker: 10*spenottabi + 7*eprestabi >= 51;
```

```
minimize koltseg: 2000*spenottabi + 1500*eprestabi;
```

```
end;
```

- 6) Mikulásra csomagokat akarunk összeállítani. A csomagok tartalma: csoki, narancs, mogyoró, virgács, kindertojás, gumicukor. Az 1. csomag, ára 5 pénzegység és narancsot, mogyit tartalmaz. A 2. csomag ára 10 egység, van benne gumicukor, mogyi és csoki. A 3. csomag 3 pénzegység, virgácsot, kindert és gumicukrot tartalmaz. A 4. csomag ára 4 pénzegység, narancsot, kindert és csokit tartalmaz. Az 5. csomag ára 6 pénzegység, virgácsot és csokit tartalmaz. Milyen csomagokat vegyünk a Mikulás csomagunk összeállításához, ha minimálisan szeretnénk költekezni?

```
var cs1, binary;
```

```
var cs2, binary;
```

```
var cs3, binary;
```

```
var cs4, binary;
```

```
var cs5, binary;
```

```
s.t. csoki: cs2 + cs4 + cs5 >= 1;
```

```
s.t. narancs: cs1 + cs4 >= 1;
```

```
s.t. mogyi: cs1 + cs2 >= 1;
```

```
s.t. virgacs: cs3 + cs5 >= 1;
```

```
s.t. kinder: cs3 + cs4 >= 1;
```

```
s.t. gumicukor: cs2 + cs3 >= 1;
```

#Ezekben a csomagokban van csoki. Legalább 1-et meg kell vásároljak közülük ahhoz, hogy biztos legyen csokim.

```
minimize koltseg: 5*cs1 + 10*cs2 + 3*cs3 + 4*cs4 + 6*cs5;
```

```
end;
```

- 7) Ha valamiből megmarad és felesleg lesz, azért büntetés jár. A csokiért 2, a narancsért 3, a mogyiért 1, a virgácsért 4, a kinderért 5, míg a gumicukorért 6 pénzegység büntetés jár. Milyen csomagokat vegyünk a Mikulás csomagunk összeállításához, ha minimálisan szeretnénk költekezni és minimális büntípontot szeretnénk csak kapni?

```
var cs1, binary;  
var cs2, binary;  
var cs3, binary;  
var cs4, binary;  
var cs5, binary;
```

```
var y1 >= 0, integer;  
var y2 >= 0, integer;  
var y3 >= 0, integer;  
var y4 >= 0, integer;  
var y5 >= 0, integer;  
var y6 >= 0, integer;
```

```
s.t. bcsoki: cs2 + cs4 + cs5 - y1 = 1;  
s.t. bnarancs: cs1 + cs4 - y2 = 1;  
s.t. bmogyi: cs1 + cs2 - y3 = 1;  
s.t. bvirgacs: cs3 + cs5 - y4 = 1;  
s.t. bkinder: cs3 + cs4 - y5 = 1;  
s.t. bgumicukor: cs2 + cs3 - y6 = 1;
```

```
minimize költség: (5*cs1 + 10*cs2 + 3*cs3 + 4*cs4 + 6*cs5) + (2*y1 + 3*y2 + 1*y3 + 4*y4 + 5*y5 + 6*y6);  
end;
```

- 8) Egy háziasszony eladásra lekvárt és befőttet készít. Összesen 120 kg gyümölcs és 80 kg cukor áll rendelkezésre. Egy üveg lekvárhoz 80 dkg gyümölcsre és 40 dkg cukorra, egy üveg befőtt elkészítéséhez 40 dkg gyümölcsre és 40 dkg cukorra van szüksége. A nyeresége egy üveg lekváron 100 Ft, egy üveg befőttön 80 Ft. Hány üveg lekvárt és befőttet kell elkészítenie, hogy maximális nyereséget érjen el és mekkora ez a maximális nyereség?

```
var lekvár >= 0, integer;  
var befött >= 0, integer;
```

```
s.t. gyumolcs: 80*lekvár + 40*befött <= 12000;  
s.t. cukor: 40*lekvár + 40*befött <= 8000;
```

```
maximize nyereseg: 100*lekvár + 80*befött;  
end;
```

- 9) Egy kis szabóságban ingeket és szoknyákat varrnak. Egy ingen 4\$, egy szoknyán 3\$ nyeresége van a szabóságnak. Egy ing 3 m anyagból 5 óra alatt, egy szoknya 4 m anyagból 2 óra alatt készül el. A cég vezetője azt várja el a varrónőktől, hogy a napi 10 órás munkaidő alatt legalább 4 ruhadarabot készítsenek el, legfeljebb 12 m anyag felhasználásával. Hány ing és hány szoknya megvarrásával érhető el egy varrónő esetén a maximális profit?

```
var ing >= 0, integer;  
var szoknya >= 0, integer;
```

```
s.t. anyagszukseglet: 3*ing + 4*szoknya <= 12;  
s.t. munkaido: 5*ing + 2*szoknya = 10;  
s.t. ruhadarab: ing + szoknya >= 4;
```

```
maximize profit: 4*ing + 3*szoknya;  
end;
```

- 10) Egy fogyókúrázó napi C-vitamin és vas szükségletét kétféle gyümölcs, citrom és alma fogyasztásával szeretné fedezni. Tegyük fel, hogy a napi C-vitamin szükséglet 500 egység, és 100g citromban 300 egységnyi, 100g almában 100 egységnyi C-vitamin van. Vasból 1.2mg a napi szükséglet, és 100g citrom 0.2mg, 100g alma pedig 0.5mg vasat tartalmaz. A fogyókúrázó a gyümölcsök fogyasztásával legfeljebb 400 kalóriát szeretne bevinni. 100g citromban 50, 100g almában 80 kalória van. A piacon 1kg citrom 600 Ft-ba, 1kg alma 400 Ft-ba kerül. Mennyit vegyen az egyes gyümölcsökből, ha azt szeretné, hogy a lehető legkevesebbet költsön, ne lépje túl az előírt kalóriamennyiséget, de a napi C-vitamin és vasszükségletét fedezze a megvásárolt gyümölcs?

```
var citrom >= 0;
var alma >= 0;

s.t. Cvitamin: 300*citrom + 100*alma >= 500;
s.t. vas: 0.2*citrom + 0.5*alma >= 1.2;
s.t. kcal: 50*citrom + 80*alma <= 400;

minimize költség: 600*citrom + 400*alma;
end;
```

- 11) Egy üzem háromféle termék előállításával foglalkozik. A termékek megmunkálását két gépen végzik. Az esztergagép kapacitása 81 óra/hét, a marógép kapacitása 70 óra/hét. Az egyes termékfajták egységének megmunkálásához rendre 2, 1, 1 óra szükséges az esztergagépen, a marógépen mindhárom termékre 1 óra ez az adat. A termékek eladási egységára 3, 1 és 2 pénzegység. A termékegységre jutó anyagköltségek rendre 0,8, 0,2 és 0,5 pénzegység. A cél olyan heti termelési terv meghatározása, mellyel legalább 100 pénzegységnyi bevétel érhető el, de az anyagköltségek összege minimális.

```
var t1 >= 0, integer;
var t2 >= 0, integer;
var t3 >= 0, integer;

s.t. esztergagep: 2*t1 + 1*t2 + 1*t3 <= 80;
s.t. marogep: 1*t1 + 1*t2 + 1*t3 <= 70;
s.t. bevetel: 3*t1 + 1*t2 + 2*t3 >= 100;

minimize anyagköltség: 0.8*t1 + 0.2*t2 + 0.5*t3;
end;
```

- 12) Az üzemben T1, T2 termék készül G1, G2 géppel. Az összes munkás max 60 órát dolgozik. Melyik termékből mennyit gyártsunk a maximális bevételhez?

	T1	T2	Összkapacitás
G1	3	4	130
G2	4	6	200
Ár	8	10	

```
var t1 >= 0;
var t2 >= 0;

s.t. g1_kapacitas: 3*t1 + 4*t2 <= 130;      #1. gép kapacitása
s.t. g2_kapacitas: 4*t1 + 6*t2 <= 200;     #2. gép kapacitása
s.t. munkas: 1*t1 + 2*t2 = 60;             #A munkás teljes kapacitással dolgozik, ezért =

maximize bevetel: 8*t1 + 10*t2;
end;
```

13) Hogyan rendeljük össze a munkát, hogy a lehető legkevesebbet fizessük a munkásoknak?

	1	2	3
A	2	4	3
B	10	5	6
C	4	2	1

```
var a1, binary;  
var a2, binary;  
var a3, binary;  
var b1, binary;  
var b2, binary;  
var b3, binary;  
var c1, binary;  
var c2, binary;  
var c3, binary;
```

```
s.t. munka1: a1 + a2 + a3 = 1;      #Munkára vonatkozó korlátozások  
s.t. munka2: b1 + b2 + b3 = 1;  
s.t. munka3: c1 + c2 + c3 = 1;  
s.t. munkas1: a1 + b1 + c1 = 1;     #Munkásra vonatkozó korlátozások  
s.t. munkas2: a2 + b2 + c2 = 1;  
s.t. munkas3: a3 + b3 + c3 = 1;
```

```
minimize költség: 2*a1 + 10*b1 + 4*c1 + 4*a2 + 5*b2 + 2*c2 + 3*a3 + 6*b3 + 1*c3;  
end;
```

14) Minimális költséggel mennyit szállítson az A, B, C gyár?

```
var xA1 >= 0, integer;  
var xA2 >= 0, integer;  
var xA3 >= 0, integer;  
var xA4 >= 0, integer;  
var xB1 >= 0, integer;  
var xB2 >= 0, integer;  
var xB3 >= 0, integer;  
var xB4 >= 0, integer;  
var xC1 >= 0, integer;  
var xC2 >= 0, integer;  
var xC3 >= 0, integer;  
var xC4 >= 0, integer;
```

```
s.t. sor1: xA1 + xA2 + xA3 + xA4 = 100;  
s.t. sor2: xB1 + xB2 + xB3 + xB4 = 100;  
s.t. sor3: xC1 + xC2 + xC3 + xC4 = 200;  
s.t. oszlop1: xA1 + xB1 + xC1 = 100;  
s.t. oszlop2: xA2 + xB2 + xC2 = 100;  
s.t. oszlop3: xA3 + xB3 + xC3 = 100;  
s.t. oszlop4: xA4 + xB4 + xC4 = 100;
```

```
minimize költség: 2*xA1 + 4*xA2 + 1*xA3 + 5*xA4 + 4*xB1 + 3*xB2 + 10*xB3 + 6*xB4 + 8*xC1 + 2*xC2 + 7*xC3 +  
6*xC4;  
end;
```

15) Minden sorban és oszlopban legyen 1200 a számok összege.

```
var A1 = 897; #Megadott adat
var A2 >= 0;
var A3 >= 0;
var B1 >= 0;
var B2 >= 0;
var B3 = 762;
var C1 >= 0;
var C2 = 832;
var C3 >= 0;

s.t. Aoszlop: A1 + A2 + A3 = 1200;
s.t. Boszlop: B1 + B2 + B3 = 1200;
s.t. Coszlop: C1 + C2 + C3 = 1200;
s.t. sor1: A1 + B1 + C1 = 1200;
s.t. sor2: A2 + B2 + C2 = 1200;
s.t. sor3: A3 + B3 + C3 = 1200;

solve;
end;
```

16) Napi termelés meghatározása az egyes termékekből a maximális árbevételhez.

```
var T1 >= 0;
var T2 >= 0;

s.t. g1: (1*1+0*0+2*1)*T1 + (0*1+1*0+1*1)*T2 <= 240;    #1. gép kapacitása (a szükséges alkatrészek számát
s.t. g2: (1*7+0*1+2*1)*T1 + (0*7+1*1+1*1)*T2 <= 630;    szoroztam a megmunkáláshoz szükséges idővel)
s.t. szerelde: 2*T1 + 1*T2 <= 220;                        #2. gép kapacitása
                                                         #szerelde kapacitása

maximize arbevétel: 27*T1 + 8*T2;
end;
```

17) 4 terem van a gyárban. Hangosbemondót szereljük fel úgy, hogy a lehető legkevesebb helyre kelljen tenni.

```
var t1, binary;
var t2, binary;
var t3, binary;
var t4, binary;

s.t. terem1: t1 + t2 + t4 >= 1;    #Ha az egyes terembe tesszük, hallható az 1,2,4-es teremben. >=, hiszen több
s.t. terem2: t1 + t2 + t3 >= 1;    terembe is rakhatunk.
s.t. terem3: t2 + t3 >= 1;
s.t. terem4: t1 + t4 >= 1;

minimize keszulek: t1 + t2 + t3 + t4;
end;
```

- 18) Melyik kurzusok tarthatók meg úgy, hogy a lehető legtöbb óra meg legyen tartva?  
 #Nem lehet ütközés, illetve nem muszáj mindennap kurzusnak lennie.

```

var x12, binary;
var x24, binary;
var x3, binary;
var x57, binary;
var x45, binary;
var x78, binary;

s.t. nap1: x12 <= 1;
s.t. nap2: x12 + x24 <= 1;
s.t. nap3: x24 + x3 <= 1;
s.t. nap4: x24 + x45 <= 1;
s.t. nap5: x57 <= 1;
s.t. nap6: x57 + x45 <= 1;
s.t. nap7: x57 + x78 <= 1;
s.t. nap8: x78 <= 1;

maximize kurzusok: x12 + x24 + x3 + x57 + x45 + x78;
end;

```

- 19) Celok

	P1	P2	P3	Cél	Büntetés
Profit	12	9	15	125 (legalább)	5 ('M hiány esetén)
Munkaerő	5	3	4	4000 (pontosan)	+ :2 - :4 (100 emberenként 4M)
Tőke	5	7	8	55 (legfeljebb)	3 (1M túllépésnél)

```

var x1 >= 0, integer;
var x2 >= 0, integer;
var x3 >= 0, integer;
var y1 >= 0;
var y2 >= 0;
var y3 >= 0;
var y4 >= 0;

s.t. profit: 12*x1 + 9*x2 + 15*x3 + y1 = 125;
s.t. munkaero: 5*x1 + 3*x2 + 4*x3 + y2 - y3 = 40;
s.t. toke: 5*x1 + 7*x2 + 8*x3 - y4 = 55;

minimize buntetes: 5*y1 + 4*y2 + 2*y3 + 3*y4;
end;

```

- 20) Pizzas - Bevétel maximalizálása

```

var x1 >= 0;
var x2 >= 0;

s.t. sajt: 10*x1 + 5*x2 <= 550;
s.t. sonka: 2*x1 + 4*x2 <= 150;
s.t. ananasz: 3*x2 <= 120;

maximize koltseg: 600*x1 + 800*x2;
end;

```

21) Diéta - Költség minimalizálása a tápértékeknek megfelelően

```
var x1 >=0, integer;
var x2 >=0, integer;
var x3 >=0, integer;
var x4 >=0, integer;

s.t. fehérje: 50*x1+100*x2+30*x3+10*x4 >= 120;
s.t. szénhidrát: 100*x1+10*x2+90*x3+100*x4 >= 200;
s.t. zsír: 5*x1+10*x2+6*x3+10*x4 >= 50;

minimize költség: 500*x1+1500*x2+400*x3+90*x4;
end;
```

22) 6273 Ft kifizetése a lehető legkevesebb címlet felhasználásával - címletek

```
var x1 >= 0, integer;
var x2 >= 0, integer;
var x3 >= 0, integer;
var x4 >= 0, integer;

s.t. összeg: 516 * x1 + 42 * x2 + 33 * x3 + 2 * x4 = 6273;

minimize darabszám: x1 + x2 + x3 + x4;
solve;

#Kiírás
printf "Szükséges érmék száma: %d\n", darabszám;
printf "516 forintosból %d db kell\n", x1;
printf "42 forintosból %d db kell\n", x2;
printf "33 forintosból %d db kell\n", x3;
printf "2 forintosból %d db kell\n", x4;
end;

címletek2

var x1 >= 0, integer;
var x2 >= 0, integer;
var x3 >= 0, integer;
var x4 >= 0, integer;
var x5 >= 0, integer;

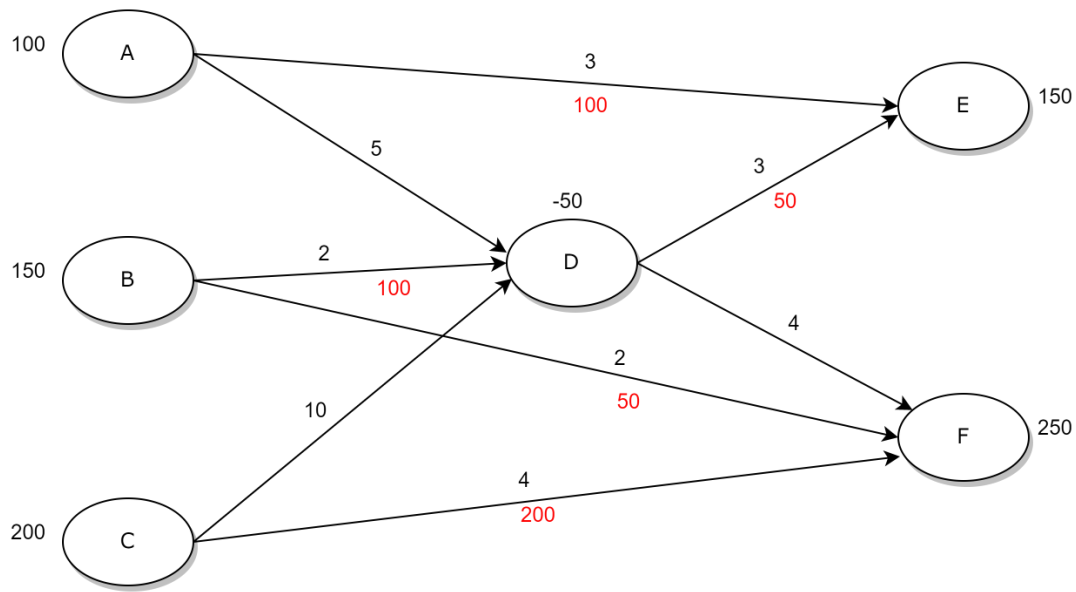
s.t. összeg: x1+3*x2+7*x3+18*x4+42*x5 = 1234;

minimize darabszám: x1 + x2 + x3 + x4;
end;
```



## GRÁFOK

- 1) Melyik szakaszon mennyit szállítson a gyár úgy, hogy a lehető legolcsóbb legyen a szállítás?



```

var AD >= 0;
var AE >= 0;
var BD >= 0;
var BF >= 0;
var CD >= 0;
var CF >= 0;
var DE >= 0;
var DF >= 0;

```

```

s.t. A: AD + AE = 100;
s.t. B: BD + BF = 150;
s.t. C: CD + CF = 200;
s.t. D: AD + BD + CD - 50 = DE + DF;      #D állomáson 50 egységet elvesznek
s.t. E: AE + DE = 150;
s.t. F: DF + BF + CF = 250;

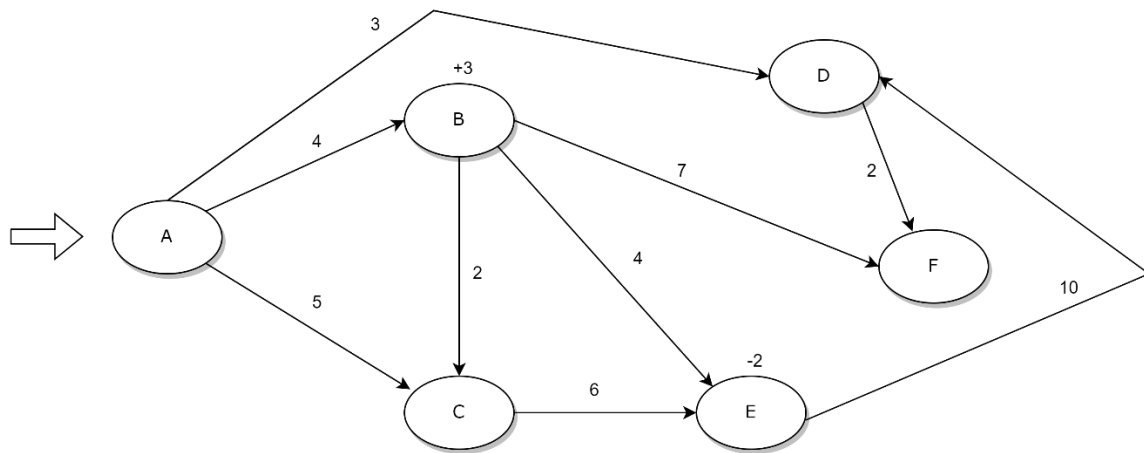
```

```

minimize szallitas: 5*AD + 3*AE + 2*BD + 2*BF + 10*CD + 4*CF + 3*DE + 4*DF;
end;

```

2) A-ból az F-be a lehető legtöbb mennyiségű olaj jusson el.



```

var xAB >= 0;
var xAC >= 0;
var xAD >= 0;
var xBF >= 0;
var xBE >= 0;
var xBC >= 0;
var xCE >= 0;
var xDF >= 0;
var xED >= 0;

```

```

s.t. fh1: xAB <= 4;
s.t. fh2: xAC <= 5;
s.t. fh3: xAD <= 3;
s.t. fh4: xBF <= 7;
s.t. fh5: xBE <= 4;
s.t. fh6: xBC <= 2;
s.t. fh7: xCE <= 6;
s.t. fh8: xDF <= 2;
s.t. fh9: xED <= 10;

```

```

s.t. B: 3 + xAB = xBF + xBE + xBC;
s.t. C: xCE = xAC + xBC;
s.t. E: xED = xCE + xBE - 2;
s.t. D: xDF = xAD + xED;

```

```

maximize olaj: xAD + xAB + xAC;
end;

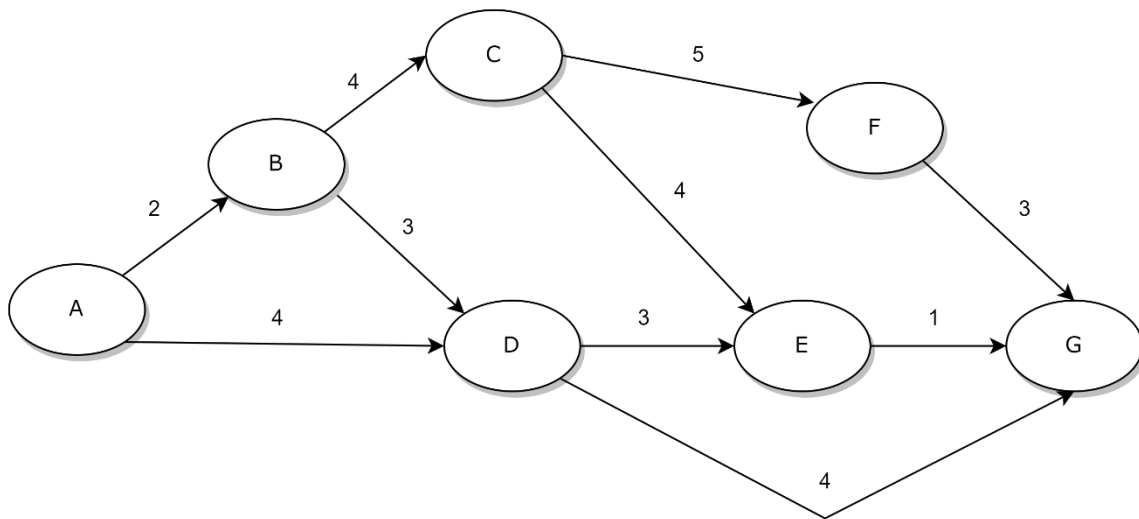
```

#A kétoldali korlátozás másik része (felső határ)

#a B pontban 3 egység olaj hozzáadódik az eddigiekhez

#az E pontban 2 egység olajat elvesznek

3) A-ból G-be a lehető legrövidebb úton eljutni.



```

var xAB, binary;
var xAD, binary;
var xBC, binary;
var xBD, binary;
var xCF, binary;
var xCE, binary;
var xDE, binary;
var xDG, binary;
var xEG, binary;
var xFG, binary;

```

```

s.t. indulas: xAB + xAD = 1;
s.t. B: xAB = xBC + xBD;
s.t. C: xBC = xCF + xCE;
s.t. D: xAD + xBD = xDE + xDG;
s.t. E: xCE + xDE = xEG;

```

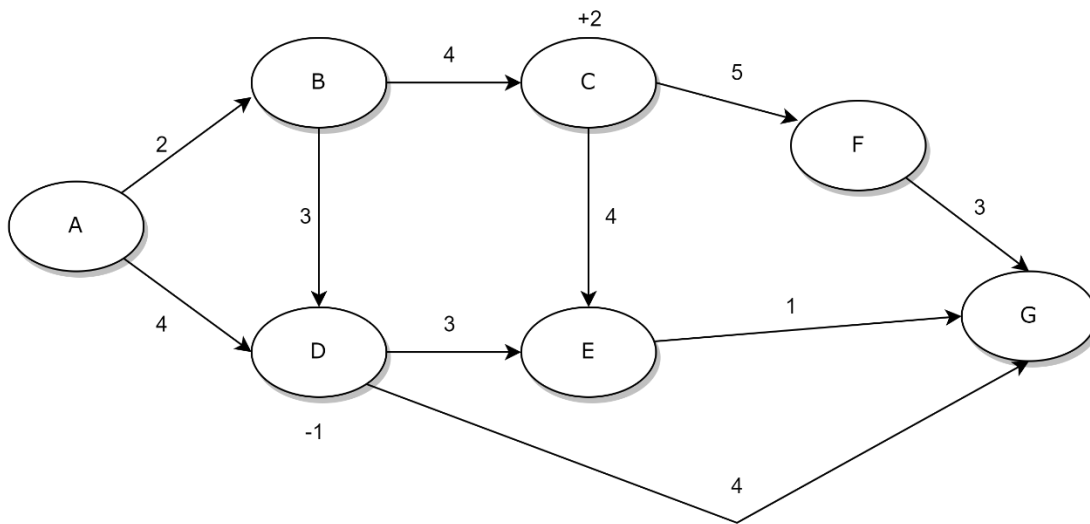
#vagy AB-n vagy AD-n indul el (azt állítja be 1-re, másik 0 lesz)

```

minimize ut: 2*xAB + 4*xAD + 4*xBC + 3*xBD + 5*xCF + 4*xCE + 3*xDE + 4*xDG + 1*xEG + 3*xFG;
end;

```

4) A lehető legtöbb sört szállítsuk el A-ból a G-be.



```

var xAB >= 0;
var xAD >= 0;
var xBC >= 0;
var xBD >= 0;
var xCF >= 0;
var xCE >= 0;
var xDE >= 0;
var xDG >= 0;
var xEG >= 0;
var xFG >= 0;

```

```

s.t. fh1: xAB <= 2;
s.t. fh2: xAD <= 4;
s.t. fh3: xBC <= 4;
s.t. fh4: xBD <= 3;
s.t. fh5: xCF <= 5;
s.t. fh6: xCE <= 4;
s.t. fh7: xDE <= 3;
s.t. fh8: xDG <= 4;
s.t. fh9: xEG <= 1;
s.t. fh10: xFG <= 3;

```

```

s.t. B: xAB = xBC + xBD;

```

```

s.t. C: xBC + 2 = xCF + xCE;

```

```

s.t. D: xAD + xBD - 1 = xDE + xDG;

```

```

s.t. E: xCE + xDE = xEG;

```

```

maximize sor: xAB + xAD;
end;

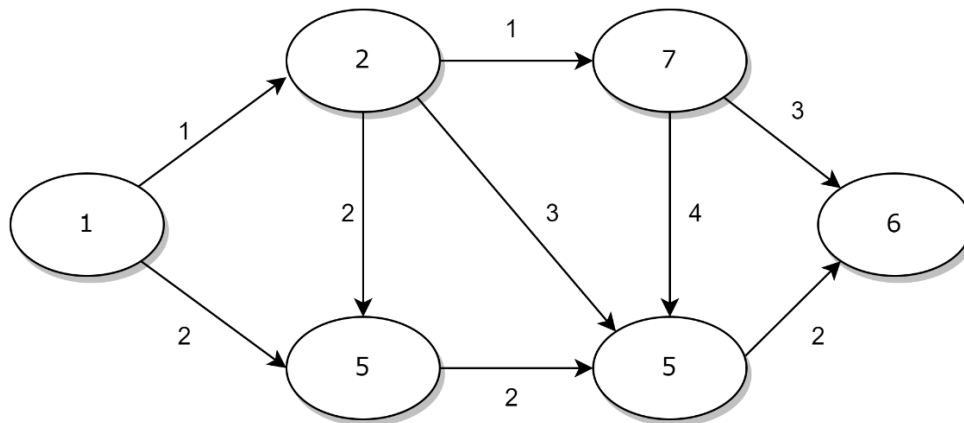
```

#A kétoldali korlátozás másik része (felső határ)

#a C pontban 2 egység sör hozzáadódik az eddigiekhez

#a D pontban elvesznek 1 egység sört

5) Legróvidebb úton eljutni 1-ből 6-ba.



```

var x12, binary;
var x13, binary;
var x24, binary;
var x25, binary;
var x23, binary;
var x35, binary;
var x45, binary;
var x46, binary;
var x56, binary;

```

```

s.t. indulas: x12 + x13 = 1;
s.t. csp1: x12 = x24 + x25 + x23;
s.t. csp2: x13 + x23 = x35;
s.t. csp3: x24 = x45 + x46;
s.t. csp4: x25 + x45 + x35 = x56;

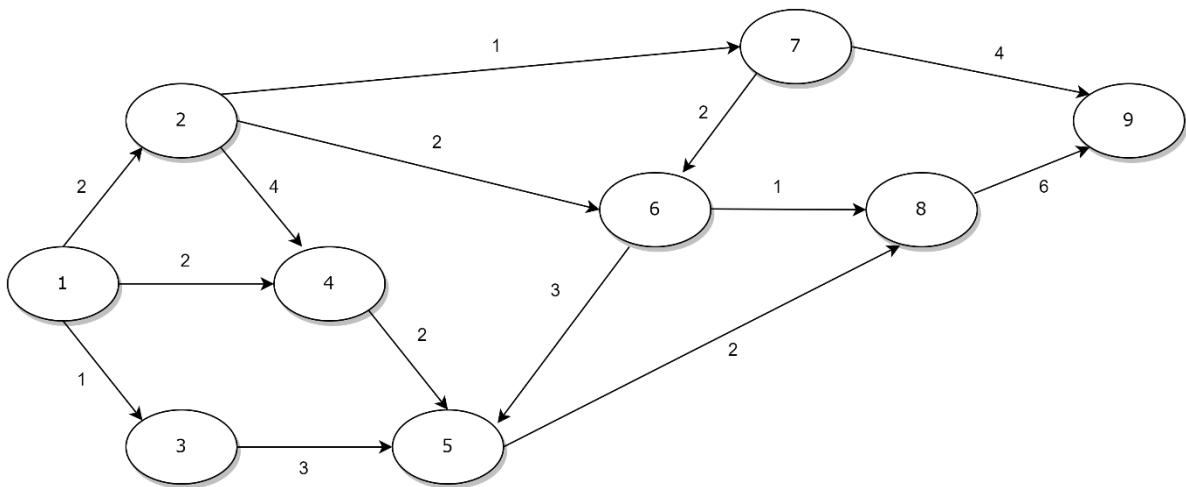
```

```

minimize ut: x12 + 2*x13 + x24 + 3*x25 + 2*x23 + 2*x35 + 4*x45 + 3*x46 + 2*x56;
end;

```

7) Lehető legkevesebb lépéssel eljutni az 1-esből a 9-esbe.



```

var x12, binary;
var x14, binary;
var x13, binary;
var x27, binary;
var x26, binary;
var x24, binary;
var x35, binary;
var x45, binary;
var x58, binary;
var x65, binary;
var x68, binary;
var x76, binary;
var x79, binary;
var x89, binary;

```

s.t. indulas:  $x_{12} + x_{14} + x_{13} = 1$ ; #Egy élen kell elindulni az 1-esből. Amelyiken elindulunk 1 lesz, többi 0.

```

s.t. csp1:  $x_{12} = x_{27} + x_{26} + x_{24}$ ;
s.t. csp2:  $x_{14} + x_{24} = x_{45}$ ;
s.t. csp3:  $x_{13} = x_{35}$ ;
s.t. csp4:  $x_{27} = x_{76} + x_{79}$ ;
s.t. csp5:  $x_{26} + x_{76} = x_{65} + x_{68}$ ;
s.t. csp6:  $x_{35} + x_{45} + x_{65} = x_{58}$ ;
s.t. csp7:  $x_{58} + x_{68} = x_{89}$ ;

```

```

minimize ut:  $x_{12} + x_{14} + x_{13} + x_{27} + x_{26} + x_{24} + x_{35} + x_{45} + x_{58} + x_{65} + x_{68} + x_{76} + x_{79} + x_{89}$ ;
end;

```