



Modellezés és optimalizálás a gyakorlatban

GKNB-INTM0019

Edzőterem bevételének maximalizálása

Schöffer Fruzsina

TSH86V

Győr, 2020. január 13.

Tartalomjegyzék

1. Bevezetés	2
2. Edzésterv	3
2.1. Probléma megfogalmazása	3
2.2. Bemenő adatok	3
2.3. Változók	4
2.4. Korlátozások	5
2.5. Célfüggvény	7
3. Modell különböző datasettel	8
3.1. Alap data	8
3.2. Másik dataset	10
3.3. Nagy dataset	12
4. Összefoglalás és konklúzió	13
5. Mellékletek	14
5.1. Alapfeladat mod	14
5.2. Alapfeladat data	16
5.3. Módosított feladat data	18
5.4. Nagy data	19

1. Bevezetés

Egy adott edzőtermi létesítményt több Vendég is látogat, akiknek mind megvannak az egyéni preferenciáik azzal kapcsolatban, hogy melyik nap, összesen mekkora összegért, kivel igen és kivel nem mennének edzeni, hogy a minimális kalóriát a héten elégezzék. Az egészségük érdekében ugyanakkor az is meg van szabva mennyi kalóriát égethetnek el maximum egy héten.

A létesítményben több edzés közül lehet válogatni, amik mindegyike egy-két bizonyos izomcsoportot céloz meg. Fontos azonban, hogy ugyanarra a csoportra egy héten egynél többször nem lehet menni, így eszerint kell az edzésre a vendégeket beosztani. A létesítmény rendelkezik valamennyi teremmel, amelyiknek közül mindegyiknek van egy profitösszege, ha edzés történik benne. A célja a létesítménynek a lehető legnagyobb összeg befolyása a termék megfelelő beosztásával úgy, hogy a vendégek igényei is teljesülnek.

A modell célja, hogy többféle dat fájl esetén is képes legyen egy olyan megoldást találni, ami minden igényt kielégít. A modellnek meg kell oldania azokat az eseteket is, amikor bár lenne edzés, de egyik vendég sem akar elmenni rá. Ilyenkor például nem lesz terem foglalva az edzésre sem.

A dolgozat során több dat fájl is használatra került, amelyekben sok esetben az adatok sűrűlték a feladat megoldhatóságát, így is megnehezítve a jó megoldás megtalálását.

2. Edzésterv

2.1. Probléma megfogalmazása

Egy edzőtermeket biztosító létesítményt több vendég is látogat. A cél, hogy olyan heti beosztás kerüljön kialakításra, amivel a termék kiadásával a lehető legtöbb pénzt tudják kihozni, és egyszerre meg is felelni a vendégek igényeinek. Mindenki maximum a megengedett heti keretből költhet, és olyan napokon járhat edzésre, amikor ráér. A megadott keretből egy olyan edzéstervet kell kihozni, amivel a minimálisan preferált kalóriaszámot mindenki elégeti, és ugyanarra a testrészt nem edz egy héten egynél többször. Mindenki számára van olyan vendég akit kevésbé szívével, mint valaki mást, így természetesen ők nem szeretnék egy edzésre járni.

A létesítménynek limitált számú és férőhelyű terme van, ahol az edzések lezajlanak, így ennek megfelelően kell történnie a napi beosztásnak. Természetesen, ha egy terem aznap foglalt, vagy már megtelt a maximum résztvevők száma, akkor másikat kell keresni, vagy valakit átosztani egy másik napra, másik edzésre. A feladat végére a modell megadja, hogy ki, melyik nap, melyik edzést fogja látogatni, és hogy ez az edzés melyik nap, melyik teremben fog történni.

A probléma tehát az, hogy n darab Vendéget az adott paraméterekkel, hogyan osszunk be úgy m darab Edzésre, hogy k darab Teremben ne csak elférjenek, de a lehető legnagyobb bevételt generálják az elosztásból.

A modell esetében az első lépés egy lecsupaszított alapfeladat megvalósítása volt, amiben csak az adott vendégeknek kell az edzésekkel egy beosztást csinálni, és még nincsenek termek. Miután ez meg lett valósítva, a következő lépés az egyes termek bevezetése, és azok beosztása a paraméterek alapján.

2.2. Bemenő adatok

Adott előre megadott számú Vendég, Edzés, Terem és egy hét napjai, amin ezeket szét kell osztani, és ezek mindegyike egy egy halmazban kerül eltárolásra, sorrendben a Vendegek, Edzesek, Terem és Napok halmazban.

A Vendegek halmaz a nevéből adódóan az adott vendégeket tárolja el. A vendégeknek különböző paramétereik vannak, amik a következők.

- $preferaltIdopont_{v,n} \quad \forall v \in Vendegek, n \in Napok$
Ez a paraméter azt adja meg, ki melyik nap tud menni edzeni.
- $maxPenz_v \quad \forall v \in Vendegek$
A paraméter az egyes emberek maximum pénzét adja meg egy hétre.
- $maxkaloria_v \quad \forall v \in Vendegek$
A paraméter az egyes emberek heti maximum elégethető kalóriáját mutatja egy hétre.
- $preferaltKaloria_v \quad \forall v \in Vendegek$
A paraméter az egyes emberek minimum elégetendő kalóriáját adja meg egy hétre.

A feladat megoldásához szükség van további halmazra, ami az egyes edzésekhez kötődik. Ez névlegesen a Terem halmaz. Ennek a következő paramétereik vannak.

- $kapacitas_t \quad \forall t \in Terem$
A paraméter az egyes termek maximum befogadóképességét adja meg.

- $penz_t \quad \forall t \in Terem$
A paraméter az egyes termek foglalásából származó összeget adja meg.
- $szabad_{t,n} \quad \forall t \in Terem, n \in Napok$
Ez a paraméter azt adja meg, melyik nap melyik terem szabad.

A Terem esetében a szabad és a kapacitás paraméterek összevonhatóak is lehetnének egy közös kétdimenziós mátrixba, ahol, ha a kapacitás 0, akkor aznap a terem nem is szabad, és amennyiben valami más értékű, mint 0, akkor pedig szabad, mégpedig a meghatározott főre. Ezzel így kivehető a szabad paraméter. Mivel viszont a modellben nem állt szándékomban napi szintre lebontva a termék kapacitását változtatni, és alapvetően egy összevont paraméterben ezt időigényesebb átírni, így a két változó megtartásánál maradtam.

Az Edzések halmaz több paramétert is igényel, amik a következők.

- $dij_e \quad \forall e \in Edzések$
A paraméter az egyes edzések összegét mutatja.
- $kaloria_e \quad \forall e \in Edzések$
A paraméter az egyes edzésekkel elégethető kalóriát adja meg.
- $testresze_{e,tr} \quad \forall e \in Edzések, tr \in Testreszek$
Ez a paraméter azt adja meg, hogy melyik edzés melyik izomcsoportra hat.
- $maxfo_{e,n} \quad \forall e \in Edzések, n \in Napok$
Ez a paraméter azt adja meg, hogy melyik nap hány ember mehet edzésre. Ha ez a szám 0, akkor az azt jelenti, hogy nincs edzés.

További halmaz még a Testresz halmaz, ami a paraméterével azt tárolja el, hogy melyik edzés melyik testrészt/izomcsoportot érinti. A fenti paraméterek alapján tehát már körvonalazódnak a lehetséges korlátozások is.

2.3. Változók

A feladat megoldásához a modell két 3 dimenziós bináris változót használ. Az egyik változó azt hivatott megoldani, hogy melyik vendég mikor és melyik edzésre megy, míg a másik változó azt, hogy melyik edzés melyik nap és melyik teremben van. Ezek együttes használatával pedig megállapítható a feladat optimális megoldása, hiszen ha mindkettő igaz, akkor megadható, hogy a vendégek hol és mit eddzenek az adott napon.

- $mit_{v,e,n} \quad \forall v \in Vendegek, e \in Edzések, n \in Napok$
Ez a változó azt határozza meg, hogy melyik vendég, melyik edzésre melyik nap megy el. A változó 3 dimenzióból áll. Mivel minden vendégre meg kell határozni, hogy melyik edzésre megy, viszont vannak preferenciáik az adott nappal kapcsolatban és az edzéseknek is meg van határozva melyik nap hány fővel vannak megtartva, így a változó a Napok halmazt is használja, hogy ezek eldöntésre kerüljenek. A változó a korlátozások javarésznél használatban van.
- $hol_{e,t,n} \quad \forall e \in Edzések, t \in Terem, n \in Napok$
A változó az Edzések, Terem és a Napok halmazokat használja. Hogy a mit változóval együttesen meg lehessen kapni a megfelelő eredményt, ennek a változónak is szüksége volt több dimenzióra, így került használatba az Edzések és Napok halmaz is. A változó ugyanúgy bináris, így megadja, hogy egy adott napon az adott edzés az adott teremben történik-e. Természetesen ha nem, akkor a változó értéke 0, ha igen, akkor pedig 1.

2.4. Korlátozások

Az első korlátozások alapvetően a Termék nélküli modellt, és a vendégek igényeit kielégítő feltételeket valósítják meg.

1. Kódrészlet. Max létszám

```
1 s.t. MaxLetszam{e in Edzesek, n in Napok: maxfo[e,n]!=0}:
2      sum{v in Vendegek} mit[v,e,n]<=maxfo[e,n];
```

Ez a korlátozás meghatározza, hogy egy adott edzésen csakis a megadott maximum fő vehet részt. Ha ez a paraméter 0, akkor aznap nincsen edzés. Ezzel elkerülhető egy fölösleges paraméter bevezetése és használata, ami azt indikálná, hogy melyik napon van edzés.

2. Kódrészlet. Preferált nap

```
1 s.t. PreferaltNap{v in Vendegek, e in Edzesek, n in Napok:
   preferaltIdopont[v,n]=0}:
2      mit[v,e,n]=0;
```

Minden vendég rendelkezik olyan preferenciával, hogy melyik nap ér, vagy nem ér rá. A korlátozással a vendégek nem mennek olyan edzésre, amikor nem érnek rá.

3. Kódrészlet. Ne eddzen ha nem szereti

```
1 s.t. NeEddzenHaNemSzereti{e in Edzesek, n in Napok, v in Vendegek,
   v2 in Vendegek: barátok[v,v2]=0}:
2      mit[v,e,n]+mit[v2,e,n]<=1;
```

Egyes vendégek nem éppen szívelelik egymást, és szeretnék elkerülni a közös edzést. A korlátozással a modell már akkor sem enged két személyt egymással edzeni, ha csak az egyik nem kedveli a másikat, viszont a másikinál ez nem kölcsönös.

4. Kódrészlet. Csak amikor van edzés

```
1 s.t. csakAmikorVanEdzes{v in Vendegek, e in Edzesek, n in Napok:
   maxfo[e,n]=0}:
2      mit[v,e,n]=0;
```

Ez a korlátozás meghatározza, hogy amikor nincsen edzés, vagyis a maximum létszám 0, akkor ne is eddzen senki.

5. Kódrészlet. Menjen edzeni eleget

```
1 s.t. MenjenedzeniEleget{v in Vendegek}:
2      sum{n in Napok, e in Edzesek} mit[v,e,n]*kaloria[e]>=
   preferaltKaloria[v];
```

Mivel minden vendég egy héten minimum kalóriát akar elégetni, így olyan edzéseket kell választania, amikkel eléri ezt a számot. Ez a korlátozás ezt állítja be.

6. Kódrészlet. Ne hajszolja túl magát

```
1 s.t. NehajszoljaTulmagat{v in Vendegek}:
2      sum{n in Napok, e in Edzesek} mit[v,e,n]*kaloria[e]<=
   maxkaloria[v];
```

A célja a modellnek az, hogy a lehető legtöbb kalória legyen elégetve, viszont a határ nem a csillogos ég, hiszen az emberi test sem képes mindenre. Ebből kifolyólag minden vendégnek van egy maximum kalória száma, amit nem haladhat meg a héten az egészsége érdekében. Ez a korlátozást ezt teszi lehetővé.

7. Kódrészlet. Maxpénz

```
1 s.t. MaxPenz{v in Vendegek}:
2      sum{n in Napok, e in Edzesek} mit[v,e,n]* dij[e] <= maxPenz[v];
```

Minden edzésnek van egy díja, és minden vendégnek van egy heti kerete, amibe bele kell férnie a választott edzésekkel. Ezt biztosítja ez a korlátozás.

8. Kódrészlet. Max egy testrészt

```
1 s.t. MaxegyTestresz{t in Testresz, v in Vendegek}:
2      sum{n in Napok, e in Edzesek} mit[v,e,n]* testresz[e,t] <= 1;
```

Hogy az edzések effektívek legyenek, egy testrészt egy héten nem lehet egyénél többször edzeni, és ezt oldja meg ez a korlátozás.

9. Kódrészlet. Egy nap egy edzés

```
1 s.t. egyNapMaxEgy{n in Napok, v in Vendegek}:
2      sum{e in Edzesek} mit[v,e,n] <= 1;
```

A vendégek egy nap csakis egy edzésen vehetnek részt, amire ez a korlátozás ad megoldást.

A következő korlátozások segítenek a termék beosztásában, és a kapacitások megtartásában.

10. Kódrészlet. Egy nap egy edzés

```
1
2 s.t. EgyNapEgyEdzes{n in Napok, e in Edzesek}:
3      sum{t in Terem} hol[e,t,n] <= 1;
```

Ez a korlátozás azt szabja meg, hogy egy adott edzés egy nap csakis egy teremben történhet.

11. Kódrészlet. Nincs edzés ha

```
1
2 s.t. HaNemszabadVagyNincsedzesNemkellterem{n in Napok, t in Terem,
3      e in Edzesek: maxfo[e,n]=0 || szabad[t,n]=0}:
      hol[e,t,n]=0;
```

Ez a korlátozás megadja, hogy ha adott napon nem szabad a terem, vagy alapvetően nincs edzés, akkor ne is legyen terem lefoglalva, mivel senki se fog fizetni érte.

12. Kódrészlet. Nincs ember nincs terem

```
1 s.t. HaNincsEmberEdzesreAkkorNelegyenTerem{ t in Terem, e in
      Edzesek, n in Napok}:
2      hol[e,t,n] <= sum{v in Vendegek} mit[v,e,n];
```

Ez a korlátozás megvalósítja, hogy ha adott nap lenne edzés, de az egyes vendégek a személyi preferenciák miatt nem mennek el, akkor fölöslegesen ne legyen terem emiatt lefoglalva.

13. Kódrészlet. Edzés csakis teremben

```

1 s.t. TeremFoglalas{v in Vendegek, e in Edzesek, n in Napok}:
2     mit[v,e,n]<=sum{t in Terem} hol[e,t,n];

```

Ez a korlátozás megvalósítja, hogy edzés csakis teremben történhet. Ha valamiért nincs terem már aznap, vagy egyáltalán, akkor nem történhet edzés se.

14. Kódrészlet. Egy teremben egy edzés

```

1 s.t. EgyNapEgyTerembenEgyEdzes{n in Napok, t in Terem}:
2     sum{e in Edzesek} hol[e,t,n]<=1;

```

Ez a korlátozás biztosítja azt, hogy adott napon egy teremben csakis egy edzés történhessen.

2.5. Célfüggvény**15. Kódrészlet.** Célfüggvény

```

1 maximize Bevetel:
2     sum{t in Terem, n in Napok, e in Edzesek} hol[e,t,n]*penz[t];

```

A célfüggvény feladata, hogy maximalizálja az adott heti bevételt azzal, hogy minél több és olyan teremre vannak kiosztva az edzések, hogy a lehető legnagyobb legyen a bevétel. Ehhez a hol változó lesz megszorozva az egyes termék kiadásának profitjával.

3. Modell különböző datasettel

3.1. Alap data

Az alap modell elkészítéséhez alapvetően egy kisebb méretű data fájlt használtam, amivel folyamatosan tudtam ellenőrizni az egyes kikötések teljesülését. Az adatok ebben úgy vannak megadva, hogy a modell számára megadott korlátozások egyszerűen teljesíthetők legyenek, és ezzel együtt kézzel is ellenőrizhető legyen a megadott megoldás helyessége.

Ebben a data fájlban mindösszesen 4 edzést lehet 3 emberre elosztani 2 terembe, és mindegyik ember hetente 2-3-szor ér rá edzeni. Ha valaki nem kedveli a másikat, akkor nem is akar vele egy edzésen lenni, ezért természetesen külön edzéseken fognak részt venni. Azonban, ha majdnem mindenki kedveli a másikat, már kerülhetnek közös edzésre is, viszont ez akkor kevesebb termet, és kisebb profitot is jelent. A modellnek ebben az esetben a maximum profit érdekében illendő még mindig nem egymással összerakni az embereket.

Mindkét terem a hét majdnem minden napján egyszerre elérhető, így, ha ténylegesen jól működik a modell akkor a drágábbba fogja az embereket beosztani, maximum profitot generálva. A fenti adatokkal és kikötésekkel, úgy, hogy az egyik terem 3000 Ft a másik 4000 Ft összeget hoz a konyhára alkalmanként, mindösszesen 23.000 Ft bevételt tud a létesítmény szerezni.

Ezekből az adatokból néhány, amely a terembeosztással kapcsolatos, az alább található.

16. Kódrészlet. Adatok a modellhez

```

1 set Edzesek := A B C D;
2 set Vendegek:= Zsuzsi Barbi Fruzsi;
3 set Napok:= Hetfo Kedd Szerda Csut Pentek Szombat Vasarnap;
4 set Testresz:=Vadli Kar Vall Fenek Hat Has Comb;
5 set Terem:=Foterem Kisterem;
6
7 param szabad:
8           Hetfo Kedd Szerda Csut Pentek Szombat Vasarnap :=
9   Foterem      1      1      1      1      1      1      0
10  Kisterem     1      1      1      1      1      0      1
11  ;
12
13 param kapacitas:=
14 Foterem 3
15 Kisterem 2
16 ;
17
18 param penz:=
19 Foterem 3000
20 Kisterem 4000
21 ;
22
23 param preferaltIdopont:
24           Hetfo Kedd Szerda Csut Pentek Szombat Vasarnap :=
25   Zsuzsi      0      1      1      1      0      0      1
26   Barbi       0      0      1      0      0      1      0
27   Fruzsi      1      0      0      1      1      0      0
28  ;
29
30 param barátok:
31           Zsuzsi Barbi Fruzsi:=
32   Zsuzsi      1      1      0
33   Barbi       1      1      1
34   Fruzsi      1      1      1
35  ;
36
37 param maxfo:
38           Hetfo Kedd Szerda Csut Pentek Szombat Vasarnap :=
39   A           0      2      0      3      3      1      3
40   B           2      1      2      0      0      3      0
41   C           0      1      0      2      1      0      0
42   D           0      3      3      0      0      2      0
43  ;
44
45 end;

```

A következő ábra az ezzel az adathalmazzal generált beosztást mutatja. Látható, hogy a profit

	Hétfő	Kedd	Szerda	Csütörtök	Péntek	Szombat	Vasárnap
Főterem						Edzés A (Barbi)	
Kisterem		Edzés D (Zsuzsi)	Edzés D (Barbi)	Edzés A (Fruzi)	Edzés C (Fruzi)		Edzés A (Zsuzsi)

1. ábra. Modell alap adathalmazzal

Forrás: saját

érdekében mindenki külön edzésen van, és amikor a Kisterem elérhető, csakis oda vannak az emberek beosztva.

3.2. Másik dataset

Az előző datasetnél a modell alapvetően szépen elosztotta az embereket külön napokra, annak ellenére is, hogy kedvelték egymást, és semmi akadálya nem volt annak, hogy egy terembe tegye őket. Hogy meggyőződjek arról, hogy biztosan a profitot akarja maximalizálni, és nem pedig baráti kapcsolatokat ápolni, ezért az adatokon a következőleg módosítottam.

Mindegyik vendégnek nagyjából ugyanaz a nap jó, mindenki szereti a másikat, és ezeken a napokon mindig mindkét terem szabad maximális 3 fős kapacitással, tehát egyszerűen beférnének egymás mellé. A modellnek gyakorlatilag minden oka meg lenne arra, hogy egymás mellé ossza be az embereket, viszont ez nem termelne akkor profitot. A futtatáskor kis gondolkozás után (2.6s) meg is kapjuk az eredményt, ami 21.000 Ft maximum bevételt hoz ki.

A modell az embereket az alábbi szerint osztotta be, vagyis minden olyan nap, amikor közösen mennek edzeni a vendégek, mégis külön terembe kerülnek, hogy maximális legyen a bevételünk.

	Hétfő	Kedd	Szerda	Csütörtök	Péntek	Szombat	Vasárnap
Főterem		Edzés C (Barbi)		Edzés C (Zsuzsi)		Edzés A (Zsuzsi)	
Kisterem		Edzés A (Fruzi)		Edzés A (Barbi)		Edzés B (Fruzi)	

2. ábra. Modell szoros napos adathalmazzal

Forrás: saját

A használt adatok, ugyancsak a terembeosztáshoz legfontosabb adatokkal az alábbiakban találhatóak.

17. Kódrészlet. Adatok a modellhez

```

1  set Edzesek := A B C D;
2  set Vendegek:= Zsuzsi Barbi Fruzsi;
3  set Napok:= Hetfo Kedd Szerda Csut Pentek Szombat Vasarnap;
4  set Testresz:=Vadli Kar Vall Fenek Hat Has Comb;
5  set Terem:=Foterem Kisterem;
6
7  param szabad:
8      Hetfo Kedd Szerda Csut Pentek Szombat Vasarnap :=
9      Foterem      0      1      0      1      1      1      0
10     Kisterem     0      1      0      1      1      1      1
11     ;
12 param kapacitas:=
13 Foterem 3
14 Kisterem 3
15 ;
16 param penz:=
17 Foterem 3000
18 Kisterem 4000
19 ;
20 param preferaltIdopont:
21     Hetfo Kedd Szerda Csut Pentek Szombat Vasarnap :=
22     Zsuzsi  0      1      0      1      0      1      0
23     Barbi   0      1      0      1      0      1      0
24     Fruzsi  0      1      0      1      0      1      0
25     ;
26 param barátok:
27     Zsuzsi Barbi Fruzsi:=
28     Zsuzsi  1      1      1
29     Barbi   1      1      1
30     Fruzsi  1      1      1
31     ;
32 param maxfo:
33     Hetfo Kedd Szerda Csut Pentek Szombat Vasarnap :=
34     A      0      2      0      3      3      1      3
35     B      2      1      2      0      0      3      0
36     C      0      1      0      2      1      0      0
37     D      0      3      3      0      0      2      0
38     ;
39 param testresz:
40     Vadli Kar Vall Fenek Hat Has Comb :=
41     A      1      0      0      0      0      0      1
42     B      0      1      0      0      0      1      0
43     C      0      0      0      1      1      0      0
44     D      0      0      1      0      0      0      0
45     ;
46
47 end;

```

3.3. Nagy dataset

A kicsi adatok mellett szükségesnek éreztem még egy nagyobb adathalmazzal való kipróbálását is a modellnek. Ebben mindösszesen 13 ember van, 6 teremmel. Hogy minél nagyobb legyen az elérhető profit, majdnem minden ember majdnem minden nap ráér, majdnem mindegyik edzés több embert képes befogadni, mint ahány ember van, és majdnem mindenki kedveli a másikat.

A modell több embernél természetesen jóval nagyobb összeget is generált, ami 121 000 Ft volt.

4. Összefoglalás és konklúzió

A fent bemutatottak alapján látható, hogy a modell többféle adattal is képes megoldani a feladatot és helyes maximális bevételt generálni a különböző korlátozások betartásával.

Bár alapvetően egy nagyobb adathalmaznál a futási időről azt hinné az ember, hogy tovább tart, mint egy kisebbnél, a modell esetében a második adathalmaznál, ahol a megadott adatokkal nagyon kevés egyáltalán jó megoldás tudott születni, mégis tovább tartott a megoldás.

A modellt természetesen lehetne tovább bővíteni, olyan további paraméterekkel és halmazokkal mint edzők, akik bizonyos összegért, bizonyos edzéseket vállalnak. A dokumentációban látható modell elkészítése előtt ennek ugyanakkor más változatai is születtek.

Egy ilyen volt például, amikor nem a terembeosztással megszerezhető profitot akartuk maximalizálni, hanem az egyes emberek által elégetett kalóriaszámot. Terembeosztás már ennél a megvalósításnál is történt, a végső modellhez pedig csak a célfüggvényt kellett átalakítani, illetve egy további paramétert bevezetni.

Összefoglalva, a modell teljesíti az eleinte megfogalmazott feltételeket többféle adathalmazzal is, és könnyedén tovább-, vagy visszalakítható.

5. Mellékletek

5.1. Alapfeladat mod

```

set Napok;
set Edzesek;
set Testresz;
set Terem;

param kapacitas{Terem};
param penz{Terem};
param szabad{Terem, Napok};

param dij{Edzesek};
param kaloria{Edzesek};
param testresz{Edzesek, Testresz};
param maxfo{Edzesek, Napok};

set Vendegek;
param preferaltIdopont{Vendegek, Napok};
param maxPenz{Vendegek};
param preferaltKaloria{Vendegek};
param maxkaloria{Vendegek};
param barátok{Vendegek, Vendegek};

var mit{Vendegek, Edzesek, Napok} binary;
var hol{Edzesek, Terem, Napok} binary;

s.t. MaxLetszam{e in Edzesek, n in Napok: maxfo[e,n]!=0}:
    sum{v in Vendegek} mit[v,e,n] <= maxfo[e,n];

s.t. PreferaltNap{v in Vendegek, e in Edzesek, n in Napok:
preferaltIdopont[v,n]=0}:
    mit[v,e,n]=0;

s.t. NeEddzenHaNemSzereti{e in Edzesek, n in Napok, v in Vendegek,
v2 in Vendegek: barátok[v,v2]=0}:
    mit[v,e,n] + mit[v2,e,n] <= 1;

s.t. csakAmikorVanEdzes{v in Vendegek, e in Edzesek, n in Napok:
maxfo[e,n]=0}:
    mit[v,e,n]=0;

```

```

s.t. NehajszoljaTulmagat{v in Vendegek}:
    sum{n in Napok, e in Edzesek}mit[v,e,n]*kaloria[e]<=
    maxkaloria[v];

s.t. MaxPenz{v in Vendegek}:
    sum{n in Napok, e in Edzesek}mit[v,e,n]*dij[e]<=maxPenz[v];

s.t. egyNapMaxEgy{n in Napok, v in Vendegek}:
    sum{e in Edzesek}mit[v,e,n]<=1;

s.t. MaxegyTestresz{t in Testresz, v in Vendegek}:
    sum{n in Napok,e in Edzesek}mit[v,e,n]*testresz[e,t]<=1;

s.t. MenjenedzeniEleget{v in Vendegek}:
    sum{n in Napok, e in Edzesek}mit[v,e,n]*kaloria[e]>=
    preferaltKaloria[v];

s.t. EgyNapEgyEdzes{n in Napok, e in Edzesek}:
    sum{t in Terem}hol[e,t,n]<=1;

s.t. HaNemszabadVagyNincsedzesNemkellterem{n in Napok, t in Terem,
e in Edzesek:maxfo[e,n]=0 || szabad[t,n]=0}:
    hol[e,t,n]=0;

s.t. TeremFoglalas{v in Vendegek,e in Edzesek, n in Napok}:
    mit[v,e,n]<=sum{t in Terem}hol[e,t,n];

s.t. HaNincsEmberEdzesreAkkorNelegyenTerem{ t in Terem,
e in Edzesek, n in Napok}:
    hol[e,t,n]<=sum{v in Vendegek}mit[v,e,n];

s.t. EgyNapEgyTerembenEgyEdzes{n in Napok, t in Terem}:
    sum{e in Edzesek}hol[e,t,n]<=1;

maximize Bevetel:
    sum{t in Terem,n in Napok,e in Edzesek}(hol[e,t,n]*penz[t]);

solve;

```



```

printf "Mikor mire megy: \n";

for{n in Napok, v in Vendegek, e in Edzesek,
  t in Terem: mit[v,e,n]=1 && hol[e,t,n]=1}
    printf "%s : %s ezt edzette: %s a
    %s-ben \n\n",n,v,e,t;
    printf "\n";

```

5.2. Alapfeladat data

```

set Edzesek := A B C D;
set Vendegek:= Zsuzsi Barbi Fruzsi;
set Napok:= Hetfo Kedd Szerda Csut Pentek Szombat Vasarnap;
set Testresz:=Vadli Kar Vall Fenek Hat Has Comb;
set Terem:=Foterem Kisterem;

param szabad:
      Hetfo Kedd Szerda Csut Pentek Szombat Vasarnap :=
Foterem      1      1      1      1      1      1      0
Kisterem      1      1      1      1      1      0      1
;

param kapacitas:=
Foterem 3
Kisterem 2
;

param penz:=
Foterem 3000
Kisterem 4000
;

param preferaltIdopont:
      Hetfo Kedd Szerda Csut Pentek Szombat Vasarnap :=
Zsuzsi      0      1      1      1      0      0      1
Barbi       0      0      1      0      0      1      0
Fruzsi      1      0      0      1      1      0      0
;

param barátok:
      Zsuzsi Barbi Fruzsi:=
Zsuzsi      1      1      0
Barbi       1      1      1
Fruzsi      1      1      1

```

```

;

param dij :=
A 2000
B 3000
C 2000
D 1500
;

param maxPenz :=
Zsuzsi 5000
Barbi 4700
Fruzszi 6000
;

param preferaltKaloria :=
Zsuzsi 600
Barbi 450
Fruzszi 450
;

param maxkaloria :=
Zsuzsi 2500
Barbi 2200
Fruzszi 1500
;

param kaloria :=
A 300
B 200
C 400
D 350
;

param maxfo:
      Hetfo  Kedd  Szerda  Csut  Pentek  Szombat  Vasarnap :=
A      0      2      0      3      3      1      3
B      2      1      2      0      0      3      0
C      0      1      0      2      1      0      0
D      0      3      3      0      0      2      0
;

param testresz:
      Vadli  Kar  Vall  Fenek  Hat  Has  Comb :=
A      1      0      0      0      0      0      1
B      0      1      0      0      0      1      0
C      0      0      0      1      1      0      0
D      0      0      1      0      0      0      0
;

```

```
end;
```

5.3. Módosított feladat data

```

set Edzesek := A B C D;
set Vendegek:= Zsuzsi Barbi Fruzsi;
set Napok:= Hetfo Kedd Szerda Csut Pentek Szombat Vasarnap;
set Testresz:=Vadli Kar Vall Fenek Hat Has Comb;
set Terem:=Foterem Kisterem;

param szabad:
      Hetfo Kedd Szerda Csut Pentek Szombat Vasarnap :=
      Foterem      0      1      0      1      1      1      0
      Kisterem      0      1      0      1      1      1      1
      ;
param kapacitas:=
Foterem 3
Kisterem 3
;
param penz:=
Foterem 3000
Kisterem 4000
;
param preferaltIdopont:
      Hetfo      Kedd Szerda Csut Pentek Szombat Vasarnap :=
      Zsuzsi      0      1      0      1      0      1      0
      Barbi      0      1      0      1      0      1      0
      Fruzsi      0      1      0      1      0      1      0
      ;
param barátok:
      Zsuzsi Barbi Fruzsi:=
      Zsuzsi 1      1      1
      Barbi  1      1      1
      Fruzsi  1      1      1
      ;

param dij :=
A 2000
B 3000
C 2000
D 1500
;

param maxPenz :=
Zsuzsi 5000
Barbi 4700

```

```

Fruzsi 6000
;

param preferaltKaloria :=
Zsuzsi 600
Barbi 450
Fruzsi 450
;

param maxkaloria :=
Zsuzsi 2500
Barbi 2200
Fruzsi 1500
;
param kaloria :=
A 300
B 200
C 400
D 350
;

param maxfo:
      Hetfo  Kedd  Szerda  Csut  Pentek  Szombat  Vasarnap :=
A      0      2      0      3      3      1      3
B      2      1      2      0      0      3      0
C      0      1      0      2      1      0      0
D      0      3      3      0      0      2      0
;

param testresz:
      Vadli  Kar  Vall  Fenek  Hat  Has  Comb :=
A      1      0      0      0      0      0      1
B      0      1      0      0      0      1      0
C      0      0      0      1      1      0      0
D      0      0      1      0      0      0      0
;

end;

```

5.4. Nagy data

```

set Edzesek := A B C D E F G;
set Vendegek:= Zsuzsi Barbi Fruzszi Reba Bogi Csim Andi Angi
Kitti Anna Aliz Judit Eszter;
set Napok:= Hetfo Kedd Szerda Csut Pentek Szombat Vasarnap;
set Testresz:=Vadli Kar Vall Fenek Hat Has Comb;
set Terem:=Foterem Mellekterem Kisterem Foterem2 Mellekterem2
Kisterem2;

```

param szabad:

	Hetfo	Kedd	Szerda	Csut	Pentek	Szombat	Vasarnap	:=
Foterem	1	1	1	1	1	1	1	
Mellekterem	1	1	1	1	1	1	1	
Kisterem	1	1	1	1	1	1	1	
Foterem2	1	1	1	1	1	1	1	
Mellekterem2	1	1	1	1	1	1	1	
Kisterem2	1	1	1	1	1	1	1	
;								

param kapacitas:=

Foterem 6

Mellekterem 7

Kisterem 7

Foterem2 3

Mellekterem2 3

Kisterem2 3

;

param penz:=

Foterem 6000

Mellekterem 7000

Kisterem 7000

Foterem2 3000

Mellekterem2 3000

Kisterem2 3000

;

param preferaltIdopont:

	Hetfo	Kedd	Szerda	Csut	Pentek	Szombat	Vasarnap	:=
Zsuzsi	1	1	1	1	1	1	1	
Barbi	1	1	1	0	1	1	1	
Fruzsi	1	1	1	1	1	1	1	
Reba	0	1	1	1	1	1	1	
Bogi	1	0	1	1	1	1	1	
Csim	1	1	1	1	1	1	1	
Andi	1	1	1	1	1	1	1	
Angi	0	1	1	1	1	1	1	
Kitti	1	1	1	1	1	1	1	
Anna	1	1	1	1	1	1	1	
Aliz	1	1	1	1	1	1	1	
Judit	1	1	1	1	1	1	1	
Eszter	1	1	0	1	1	1	1	
;								

param barátok:

	Zsuzsi	Barbi	Fruzszi	Reba	Bogi	Csim	Andi	Angi
	Kitti	Anna	Aliz	Judit	Eszter :=			
Zsuzsi	1 0	1	1	1	1	1	0	1
1	1	1	0					
Barbi	0 1	1	1	1	1	1	0	1
1	1	1	0					
Fruzszi	1 1	1	0	1	1	1	1	1
1	1	1	1					
Reba	1 0	0	1	1	0	1	1	1
1	1	1	1					
Bogi	1 1	0	1	1	1	0	1	1
1	1	1	1					
Csim	0 1	1	0	1	1	1	1	1
1	1	1	1					
Andi	1 0	1	1	1	0	1	0	1
1	1	1	1					
Angi	0 1	1	1	0	1	1	1	1
1	1	1	1					
Kitti	1 1	0	1	1	0	0	1	1
1	1	1	1					
Anna	0 1	0	0	0	1	1	0	0
1	1	1	1					
Aliz	1 0	1	0	1	0	1	1	1
1	1	1	1					
Judit	1 1	1	1	1	1	1	1	1
1	1	1	1					
Eszter	1 0	0	0	1	1	1	1	1
0	1	1	1					
;								

param dij :=

A 2000

B 3000

C 4000

D 1500

E 1300

F 1300

G 2500

;

param maxPenz :=

Zsuzsi 17000

Barbi 8700

Fruzszi 15000

Reba 7600

Bogi 16000

Csim 14000

Andi 17000

```
Angi 17000
Kitti 17000
Anna 17000
Aliz 17000
Judit 17000
Eszter 17000
;

param preferaltKaloria :=
Zsuzsi 600
Barbi 894
Fruzszi 700
Reba 560
Bogi 700
Csim 450
Andi 550
Angi 550
Kitti 550
Anna 550
Aliz 550
Judit 550
Eszter 550
;

param maxkaloria :=
Zsuzsi 4500
Barbi 4200
Fruzszi 5500
Reba 5800
Bogi 5900
Csim 5650
Andi 5700
Angi 5700
Kitti 5700
Anna 5700
Aliz 1700
Judit 1700
Eszter 1700
;
param kaloria :=
A 300
B 300
C 400
D 350
E 210
F 330
G 350
;
```

```
param maxfo :
```

	Hetfo	Kedd	Szerda	Csut	Pentek	Szombat	Vasarnap	:=
A	15	0	0	10	0	0	10	
B	10	0	16	0	13	0	0	
C	0	13	0	8	0	0	0	
D	8	0	13	0	0	0	10	
E	10	0	0	0	16	0	0	
F	0	10	0	8	0	0	0	
G	10	0	10	0	0	16	16	

```
;
```

```
param testresz :
```

	Vadli	Kar	Vall	Fenek	Hat	Has	Comb	:=
A	1	0	0	0	0	0	0	
B	0	0	0	0	0	1	0	
C	0	0	0	1	0	0	0	
D	0	0	1	0	0	0	0	
E	0	0	0	0	0	0	1	
F	0	0	0	0	0	1	0	
G	0	1	0	0	0	0	0	

```
;
```

```
end ;
```