ÚJ SZÉCHENYI TERV

Ég és Föld vonzásában – a természet titkai

Informatikai tehetséggondozás:

Visszalépéses maximumkiválasztás

TÁMOP-4.2.3.-12/1/KONV

MAGYARORSZÁG MEGÚJUL

A projektek az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósulnak meg.

1. Munkásfelvétel: N állás – N jelentkező

Egy vállalkozás N különböző állásra keres munkásokat. Pontosan N jelentkező érkezett, ahol minden jelentkező megmondta, hogy mely munkákhoz ért, illetve amihez ért, arra mennyi fizetést kérne.

Legyen `F(i,j)` értéke 0, ha az `i`. jelentkező a j. munkához nem ért, illetve `F(i,j)=A>0`, ha ért hozzá és `A` forintot szeretne kapni érte.

Állások:	1.	2.	3.	4.	5.
1. jelentkező:	100	0	0	**100**	0
2. jelentkező:	0	**200**	0	0	0
3. jelentkező:	**200**	100	0	0	0
4. jelentkező:	0	0	**200**	0	400
5. jelentkező:	500	0	400	0	**200**

A vállalkozás vezetője azt szeretné, ha az összes állást betöltenék, de úgy, hogy számára ez a lehető legkisebb költséggel járna.

A következőt gondolta ki: első lépésként állítsuk elő az összes lehetséges állásbetöltést (ha van egyáltalán olyan). Ez azt jelenti, hogy minden jelentkezőhöz hozzárendelünk egy állás sorszámot két feltétellel:

- olyan állást választhatunk számára, amihez ért (`F(i,j)>0`);
- olyan állást választhatunk számára, amit még nem adtunk másnak (`volt(i,j)` függvény értéke hamis).

```
Megoldás(N):
  Összes állás(N,1)
Eljárás vége.
```

Az `Összes állás` eljárás első paramétere a jelentkezők (és egyben állások) `N` száma, a második paramétere pedig az aktuálisan vizsgált jelentkező `i` sorszáma legyen!

A megoldásban használt fontos változók:

- `X` – az aktuálisan számolt megoldás: `X(i)` az `i`. jelentkezőnek adott munka sorszáma
- `db` – a megoldások száma;
- `Y` – a megoldásokat tartalmazó vektor.

```
Összes állás(N,i):
  Ha i>N akkor db:=db+1; Y(db):=X
  különben Ciklus j=1-től N-ig
             Ha nem volt(i,j) és F(i,j)>0
                akkor X(i):=j; Összes állás(N,i+1)
           Ciklus vége
  Elágazás vége
Eljárás vége.
```

```
volt(i,j):
  k:=1
  Ciklus amíg k<i és X(k)≠j
    k:=k+1
  Ciklus vége
  volt:=(k<i)
Függvény vége.
```

Ezután nincs más hátra, mint az Y vektorban összegyűlt megoldásokból kiválasztani a vállalkozó számára leggazdaságosabbat. Nagyon hamar kiderülhet azonban, hogy túlságosan sok elem lehet az Y vektorban.

A megoldási ötlet: felesleges az összes lehetséges megoldást tárolni, elég csupán minden lépés után a leggazdaságosabbat.

A megoldásban használt fontos változók:

- X – az aktuálisan számolt megoldás: X(i) az i. jelentkezőnek adott munka sorszáma
- db – a megoldások száma;
- Y – a legjobb megoldás.

```
Legjobb állás(N,i):
  Ha i>N akkor Ha költség(X)<költség(Y) akkor Y:=X
               Elágazás vége
  különben Ciklus j=1-től N-ig
             Ha nem volt(i,j) és F(i,j)>0
                akkor X(i):=j; Legjobb állás(N,i+1)
           Ciklus vége
  Elágazás vége
Eljárás vége.
```

```
költség(X):
  S:=0
  Ciklus i=1-től N-ig
    S:=S+F(i,X(i))
  Ciklus vége
Függvény vége.
```

Itt egy kicsi probléma léphet fel: az első megoldást mivel hasonlítjuk? Vegyünk fel egy új változót, ami az eddigi legjobb megoldás költségét tartalmazza! Állítsuk ennek az értékét a program elején a legnagyobb egész számra! Ha egy megoldást találunk, akkor az ennél biztosan jobb lesz, azaz lecserélhetjük rá.

```
Megoldás(N):
  maxkölt:=+∞
  Összes állás(N,1)
Eljárás vége.
```

A megoldásban használt fontos változók:

- `X` – az aktuálisan számolt megoldás: `X(i)` az `i`. jelentkezőnek adott munka sorszáma
- `db` – a megoldások száma;
- `maxkölt` – az eddigi legjobb megoldás költsége;
- `Y` – a legjobb megoldás.

```
Legjobb állás(N,i):
  Ha i>N akkor Ha költség(X)<maxkölt akkor Y:=X
                                           maxkölt:=költség(X)
             Elágazás vége
  különben Ciklus j=1-től N-ig
             Ha nem volt(i,j) és F(i,j)>0
                akkor X(i):=j; Legjobb állás(N,i+1)
           Ciklus vége
  Elágazás vége
Eljárás vége.
```

Előfordulhat természetesen, hogy a feladatnak egyáltalán nincs megoldása, azaz legjobb megoldás sincs:

Állások:	1.	2.	3.	4.	5.
1. jelentkező:	0	0	100	**100**	0
2. jelentkező:	0	200	0	0	**100**

3. jelentkező:	200	**100**	0	0	0
4. jelentkező:	0	0	**200**	400	0
5. jelentkező:	0	0	400	200	0

Már csak egy apróságra gondolhatunk: ha van egy megoldásunk és a most készülő megoldásról látszik, hogy már biztosan rosszabb lesz – többe fog kerülni –, akkor azt már nem érdemes tovább vinni.

Legyen az eljárás paramétere az eddigi költség, s az eljárást csak akkor folytassuk, ha még nem érjük el a korábban kiszámolt maximális költséget. Emiatt nem a megoldások elkészültekor kell számolni költséget, hanem menet közben, folyamatosan.

A megoldást is módosítanunk kell, a `Legjobb állás` eljárásnak új paramétere lesz, az aktuális előtt választások `költ` költsége.

A megoldásban használt fontos változók:

- `X` – az aktuálisan számolt megoldás: `X(i)` az `i`. jelentkezőnek adott munka sorszáma
- `költ` – az aktuálisan számolt megoldás költsége;
- `db` – a megoldások száma;
- `maxkölt` – az eddigi legjobb megoldás költsége;
- `Y` – a legjobb megoldás.

```
Megoldás(N):
  maxkölt:=+∞
  Legjobb állás(N,1,0)
Eljárás vége.
```

```
Legjobb állás(N,i,költ):
  Ha i>N akkor Ha költ<maxkölt akkor Y:=X
                                     maxkölt:=költ
               Elágazás vége
  különben Ciklus j=1-től N-ig
              Ha nem volt(i,j) és F(i,j)>0
                 és költ+F(i,j)<maxkölt
                 akkor X(i):=j
                       Legjobb állás(N,i+1,költ+F(i,j))
            Ciklus vége
  Elágazás vége
Eljárás vége.
```

2. Munkásfelvétel: N állás – M jelentkező

Egy vállalkozás N különböző állásra keres munkásokat. A hirdetésre M jelentkező érkezett (M<N), ahol minden jelentkező megmondta, hogy mely munkákhoz ért, illetve amihez ért, arra mennyi fizetést kérne. Legyen `F(i,j)` értéke 0, ha az `i`. jelentkező a `j`. munkához nem ért, illetve `F(i,j)=A>0`, ha ért hozzá és `A` forintot szeretne kapni érte.

Állások:	1.	2.	3.	4.	5.
1. jelentkező:	100	0	0	**100**	0
2. jelentkező:	0	**200**	0	0	0
3. jelentkező:	**200**	100	0	0	0
4. jelentkező:	0	0	**200**	0	400

A vállalkozás vezetője azt szeretné, ha az összes jelentkezőt fel tudná venni, de úgy, hogy számára ez a lehető legkisebb költséggel járna.

A megoldás nagyon hasonló az előzőhöz: itt nem akkor van kész egy megoldás, ha N jelentkezőnek adtunk munkát, hanem akkor, ha az M jelentkezőnek adtunk munkát:

```
Megoldás(N,M):
  maxkölt:=+∞
  Legjobb állás(N,M,1,0)
Eljárás vége.

Legjobb állás(N,M,i,költ):
  Ha i>M akkor Ha költ<maxkölt akkor Y:=X
                                     maxkölt:=költ
               Elágazás vége
  különben Ciklus j=1-től N-ig
              Ha nem volt(i,j) és F(i,j)>0
                 és költ+F(i,j)<maxkölt
                 akkor X(i):=j
                       Legjobb állás(N,M,i+1,költ+F(i,j))
           Ciklus vége
  Elágazás vége
Eljárás vége.
```

Egy vállalkozás N különböző állásra keres munkásokat. A hirdetésre M jelentkező érkezett (M>N), ahol minden jelentkező megmondta, hogy mely munkákhoz ért, illetve amihez ért, arra mennyi fizetést kérne.

Legyen `F(i,j)` értéke 0, ha az `i.` jelentkező a `j.` munkához nem ért, illetve `F(i,j)=A>0`, ha ért hozzá és `A` forintot szeretne kapni érte.

Állások:	1.	2.	3.	4.
1. jelentkező:	100	0	0	**100**
2. jelentkező:	0	**200**	0	0
3. jelentkező:	**200**	100	0	0
4. jelentkező:	0	0	**200**	0
5. jelentkező:	500	0	400	0

A megoldási ötlet: ne jelentkezőhöz keressünk állást, hanem álláshoz jelentkezőt! Így a feladat megoldása az előzőével majdnem megegyezik, csupán a két index szerepét kell felcserélni.

```
Legjobb állás(N,M,i,költ):
  Ha i>N akkor Ha költ<maxkölt akkor Y:=X
                                     maxkölt:=költ
               Elágazás vége
  különben Ciklus j=1-től M-ig
              Ha nem volt(j,i) és F(j,i)>0
                 és költ+F(j,i)<maxkölt
                 akkor X(j):=i
                       Legjobb állás(N,M,i+1,költ+F(i,j))
           Ciklus vége
  Elágazás vége
Eljárás vége.
```

3. Munkásfelvétel: N állás – N jelentkező, nem lehet mind felvenni

Egy vállalkozás N különböző állásra keres munkásokat. Pontosan N jelentkező érkezett, ahol minden jelentkező megmondta, hogy mely munkákhoz ért, illetve amihez ért, arra menynyi fizetést kérne. Legyen `F(i,j)` értéke 0, ha az `i.` jelentkező a `j.` munkához nem ért, illetve `F(i,j)=A>0`, ha ért hozzá és `A` forintot szeretne kapni érte.

Állások:	1.	2.	3.	4.	5.
1. jelentkező:	100	0	0	**100**	0
2. jelentkező:	0	**200**	0	0	0
3. jelentkező:	**200**	100	0	0	0
4. jelentkező:	0	0	**200**	0	400
5. jelentkező:	500	400	0	0	0

A vállalkozás vezetője azt szeretné, ha az összes állást betöltenék, de úgy, hogy számára ez a lehető legkisebb költséggel járna. Nem biztos azonban, hogy ez lehetséges. Akkor is érdekes lehet azonban egy olyan megoldás, amiben a lehető legtöbb munkát adhatjuk ki.

A megoldás ötlete: Vezessünk be egy N+1. fiktív állást, amihez azt gondoljuk, hogy mindenki ért! Legyen ennek az ára nagyobb, mint minden más összeg a táblázatunkban (`maxért`)! Az N+1. állásra engedjük meg, hogy többen is válasszák!

Állások:	1.	2.	3.	4.	5.	6.
1. jelentkező:	100	0	0	**100**	0	1000
2. jelentkező:	0	**200**	0	0	0	1000
3. jelentkező:	**200**	100	0	0	0	1000
4. jelentkező:	0	0	**200**	0	400	1000
5. jelentkező:	500	400	0	0	0	**1000**

Belátható, hogy a leggazdaságosabb megoldásban ekkor a lehető legkevesebb fiktív állás lesz, azaz a legtöbb állást tudjuk betölteni.

```
Legjobb állás(N,i,költ):
  Ha i>N akkor Ha költ+maxért<maxkölt
                 akkor Y:=X; maxkölt:=költ+maxért
              Elágazás vége
  különben Ciklus j=1-től N-ig
             Ha nem volt(i,j) és F(i,j)>0
                és költ+F(i,j)<maxkölt
                akkor X(i):=j
                      Legjobb állás(N,i+1,költ+F(i,j))
           Ciklus vége
  Elágazás vége
Eljárás vége.
```

4. L üzlet – K pékség

Egy üzletlánc L üzlete K pékségtől rendelhet kenyeret. Megadjuk, hogy az egyes üzletek mennyi kenyérre tartanak igényt, és azt, hogy az egyes pékségek mennyit sütnek naponta. Továbbá adott az is, hogy az egyes üzletek mely pékségekkel állnak kapcsolatban. Az üzletek csak egyetlen egy pékségtől rendelhetnek (az adott napon). Ha ismerjük azt is, hogy az egyes pékségek hány forintért adják a kenyeret, akkor megadhatjuk, hogy hova honnan szállítsák a kenyeret úgy, hogy az az üzletláncnak a lehető legkevesebbe kerüljön!

Pékségek:	1.	2.	3.	4.	Igény
1. üzlet:	**igaz**	hamis	hamis	hamis	200
2. üzlet:	**igaz**	hamis	hamis	hamis	100
3. üzlet:	hamis	**igaz**	hamis	hamis	300
4. üzlet:	igaz	hamis	**igaz**	hamis	200
5. üzlet:	hamis	hamis	hamis	**igaz**	200
6. üzlet:	hamis	hamis	igaz	**igaz**	100
7. üzlet:	**igaz**	hamis	igaz	hamis	100

Van	400	600	400	400
Ár	200	300	250	100

Ez a feladat is visszalépéses maximumkiválasztás, első lépésként azonban próbáljuk megfogalmazni az összes megoldás előállítását!

Adatok:

- `Kapcs(i,j)` – igaz, ha az i. üzletbe a j. pékségből lehet szállítani
- `Van(j)` – a j. pékségben előállított kenyérmennyiség
- `Igény(i)` – az i. üzlet kenyérigénye
- `Ár(j)` – a j. pékségben előállított kenyerek ára

A megoldásban az `i.` üzletbe lehet a `j.` üzletből szállítani, ha van közöttük kapcsolat és a `j.` pékségben a kisebb sorszámú üzletek által elvitt kenyérmennyiség után még maradt annyi, amennyi az `i.` üzletnek kell.

```
Megoldás(L,K):
  Összes rendelés(L,K,1)
Eljárás vége.

Összes rendelés(L,K,i):
  Ha i>L akkor db:=db+1; Y(db):=X
  különben Ciklus j=1-től K-ig
             Ha vankenyér(i,j) és Kapcs(i,j)
                akkor X(i):=j; Összes rendelés(L,K,i+1)
           Ciklus vége
  Elágazás vége
Eljárás vége.

vankenyér(i,j):
  s:=Van(j)
  Ciklus k=1-től i-1-ig
    Ha Y(k)=j akkor s:=s-Igény(k)
  Ciklus vége
  vankenyér:=s≥Igény(i)
Eljárás vége.
```

Ezt ezután már nem túl nehéz átfogalmazni maximumkiválasztásra:

```
Megoldás(L,K):
  maxkölt:=+∞
  Legjobb rendelés(L,K,1)
Eljárás vége.

Legjobb rendelés(L,K,i):
  Ha i>L akkor Ha költség(X)<maxkölt akkor Y:=X
                                          maxkölt:=költség(X)
               Elágazás vége
  különben Ciklus j=1-től K-ig
             Ha vankenyér(i,j) és Kapcs(i,j)
                akkor X(i):=j; Legjobb rendelés(L,K,i+1)
           Ciklus vége
  Elágazás vége
Eljárás vége.
```

A költségszámítás egy összegzés – ha az `i`. üzlet az `X(i)`. pékségtől rendel kenyeret, akkor annak az ára `Ár(X(i))*Igény(i)`.

```
költség(X):
  S:=0
  Ciklus i=1-től L-ig
    S:=S+Ár(X(i))*Igény(i)
  Ciklus vége
Függvény vége.
```

A költségszámítást itt is elvégezhetnénk menet közben, megtakarítva ezzel a költség függvény megírását:

```
Megoldás(L,K):
  maxkölt:=+∞
  Legjobb rendelés(L,K,1,0)
Eljárás vége.

Összes rendelés(L,K,i,költ):
  Ha i>L akkor Ha költ<maxkölt akkor Y:=X
                                     maxkölt:=költség(X)
               Elágazás vége
  különben Ciklus j=1-től K-ig
             Ha vankenyér(i,j) és Kapcs(i,j)
                akkor X(i):=j; z:=költ+Ár(j)*Igény(i)
                      Összes rendelés(L,K,i+1,z)
           Ciklus vége
  Elágazás vége
Eljárás vége.
```