# Ádám és Éva együtt

Ádám és Éva megadta, hogy egy mely időszakokban érnek rá. Az időszakokat (K,V) intervallumokkal adjuk meg, ami azt jelenti, hogy az adott személy K és V óra között ér rá – két időszak biztos nem ér össze. Értelmezés: Ha 5 órától 7 óráig és 8 órától 9 óráig érek rá, akkor biztos foglalt vagyok 7 és 8 között.

Írj programot, amely megadja azon időszakokat, amikor Ádámmal és Évával egyszerre találkozhatunk!

## Bemenet

A *standard bemenet* első sorában a napon belüli utolsó időpont értéke szerepel (100≤P≤ 100000000). A második sorban Ádám elérhető időszakai száma van (1≤A≤100000). A következő A sor tartalmazza Ádám elérhető időszakait (1≤AKi<AVi≤P), időrendben (AKi>AVi-1). A következő sorban Éva elérhető időszakai száma van (1≤E≤100000). Az utolsó E sor tartalmazza Éva elérhető időszakait (1≤EKi<EVi≤P), időrendben (EKi>EVi-1).

## Kimenet

A *standard kimenet* első sorába azon időszakok K számát kell írni, amelyekben Ádámmal és Évával egyszerre találkozhatunk! A következő K sorba ezen időszakok kezdete és vége kerüljön, időrendben! Két szomszédos időszak nem érhet össze!

## Példa

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 100 3 8 10 11 14 18 19 3 9 13 15 17 19 20 | 2 9 10 11 13  Magyarázat: 9-től 10-ig mindketten ráérnek, 10-től 11-ig Ádám nem ér rá, majd 11-től 13-ig megint mindketten ráérnek. Utána pl. Ádám 18 és 19 között ér rá, Éva pedig 19 és 20 között, azaz egyszerre nem érnek rá. |

## Korlátok

Időlimit: 0.4 mp.

Memórialimit: 32 MB

## Pontozás

A tesztek 60%-ában P≤100000.

# Dinamit

Bájtország királya hintóján kirándulni szeretne egy dimbes-dombos területen. Az i. sor j. oszlopában ti;j magasságú domb található. A bal felső sarokból indul és a jobb alsóba szeretne érkezni jobbra és lefelé lépésekkel. A király kényelmetlensége megegyezik a meglátogatott mezők (beleértve az elsőt és utolsót is) dombjainak magasság összegével. Te vagy a királyi útegyengető szakosztály dinamit felelőse. Ha az i. sor j. oszlopában felrobbantasz egy dinamitot, akkor az ott lévő domb fele akkorává (kettővel osztás hányadosa) zsugorodik. Egy cellában többször is robbanthatsz. Mielőtt megérkezne a király a bal felső sarokba, tetszőleges mezőkben felrobbanthatsz összesen legfeljebb K dinamitot

Készíts programot, amely kiszámítja, hogy minimálisan mekkora kényelmetlenséget kell a királynak eltűrnie!

## Bemenet

A *standard bemenet* első sorában a terület sorai és oszlopai száma (1≤N,M≤40), valamint a dinamitok száma (0≤K≤80) szerepel. A következő N sorban soronként M domb magassága van (0≤ti,j≤1000000).

## Kimenet

A *standard kimenet* első és egyetlen sorában a király minimális kényelmetlensége álljon!

## Példa

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 5 5 5 **1 4 2** 6 9 2 4 **7** 6 3 9 8 **2** **4** 1 3 5 8 **1** 2 7 7 8 **1** **8** | 16  A példában az optimális útvonalon eredetileg sorban az 1; 4; 2; 7; 2; 4; 1; 1; 8 magasságú dombok voltak, a robbantások után összesen 1+2+2+3+2+2+1+1+2=16 kényelmetlenség lesz. |

## Korlátok

Időlimit: 0.4 mp.

Memórialimit: 64 MiB

## Pontozás

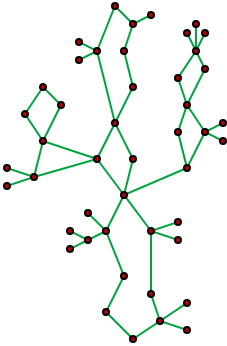
A pontok 20%-át érő tesztekben K=0.

A pontok további 10%-át érő tesztekben ti,j≤100, ha j=1 vagy i=N, egyébként ti,j=106.

A pontok további 20%-át érő tesztesetekben N,M≤10.

A pontok további 10%-át érő tesztekben N,M,K≤20.

# Kaktuszgráf

Egy kaktuszgráf olyan gráf, amelynek minden éle legfeljebb egy körhöz tartozik (mellékelt ábra a wikipédiából), másképp fogalmazva: bármely két körének legfeljebb egy közös csúcsa van. A feladatban feltesszük, hogy a kaktuszgráfban legalább 1 kör van.

Írj programot, amely kiszámítja egy kaktuszgráf leghosszabb körének hosszát!

## Bemenet

A *standard bemenet* első sorában a gráf pontjai száma (3≤N≤1000) és az élei száma (1≤M≤10000) van. A következő M sorban egy-egy él két végpontja szerepel (1≤Ai≠Bi≤N).

## Kimenet

A *standard kimenet* első sorába a leghosszabb kör hosszát kell írni!

## Példa

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 13 15 1 3 1 4 2 3 3 5 4 5 4 12 12 13 13 4 5 7 7 6 7 11 6 8 11 10 10 9 8 9 | 6  A leghosszabb kör pontjai: 7, 6, 8, 9, 10, 11. |

## Korlátok

Időlimit: 0.1 mp.

Memórialimit: 32 MB

## Pontozás

A tesztek 50%-ában N≤100.

# Az óvodai lét elviselhetetlen könnyűsége

A nagycsoportosok óvónénije, Nóra néni komoly problémával küzd, és hozzánk fordul segítségért. A csoportjával karácsonyi színdarabot szeretnének előadni, melyben a különböző szerep fordul elő, de nem biztos, hogy annyi, ahány gyerek van a csoportban. Mivel mindenki szerepelni szeretne, így úgy döntött, lesznek olyan szerepek, melyeket több gyerek is megkap majd – ehhez Nóra néni előre meghatározta, hogy melyik szerepből legfeljebb hányat tudnak bevenni a darabba úgy, hogy a történet ne veszítse értelmét. Annak is teljesülnie kell, hogy minden szerepet legalább egy gyereknek el kell játszania.

Minden gyereknek határozott elképzelése van arról, hogy ő milyen szerepet szeretne magának, így Nóra néninek nincs más választása, mint egyes gyerekekre más szerepet osztani, mint amit szerettek volna. Azonban szeretné ezt úgy megtenni, hogy összességében a lehető legkevesebb fájdalmat okozza. A szerepeket 1-től K-ig, a gyerekeket 1-től N-ig sorszámozzuk. Nóra néni mindegyik gyerekről tudja, hogy hányas számú szerepet szeretné magának, valamint azt is, hogy az egyes gyerekek hány percet fognak kínkeserves sírással tölteni, ha végül nem azt kapják, amit szerettek volna.

A feladatod segíteni Nóra néninek kiválasztani, melyik gyerekek melyik szerepeket kapják meg úgy, hogy a sírással töltött perceik összege minimális legyen, miközben minden szerepet legalább egy gyerek eljátszik, és egyik szerepet sem kapják többen, mint a Nóra néni által megadott maximális érték.

## Bemenet

A *standard bemenet* első sorában a gyerekek száma és a szerepek száma található (1≤K≤N≤100000). A második sor i. száma azt jelenti, hogy az i. szerepet maximum hány gyerek kaphatja meg (1≤Mi≤N, ahol ∑Mi≥N). A harmadik i. száma az i. gyerek által választott szerep sorszáma (1≤Si≤K). A negyedik sor i. száma az i. gyerek által sírással töltött idő, ha nem a választott szerepet kapja (1≤Ti≤10000).

## Kimenet

A *standard kimenet* első sorába a gyerekek által összesen sírással töltött percek számának legkisebb lehetséges értékét kell írni! A második sor i. száma az i. gyerek szerepének sorszáma legyen egy (az összes feltételt teljesítő) optimális esetben!

## Példa

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 6 4 2 2 2 2 1 2 4 1 2 2 1 4 2 1 2 3 | 2 1 2 4 1 3 2 |

## Korlátok

Időlimit: 0.4 mp.

Memórialimit: 32 MB

## Pontozás

A pontok 32%-a szerezhető olyan esetekben, amikor N≤10.

A pontok 50%-a szerezhető olyan esetekben, amikor N≤1000.

# Tevefarm

Bittisztán országának városait kétirányú utak kötik össze úgy, hogy bármely városból bármely másik városba pontosan egyféleképpen lehet eljutni. Az emberek az ország minden városában tevéket nevelnek, minden hónapban fix számút, melyet a hónap végén elvisznek a fővárosba. Az ország királya szeretne minél több tevét magának, ezért megpróbál minél több várost bevonni az Országos Tevenevelő Programba. Hogy a kiválasztott városok igazán különlegesnek érezzék magukat, a királyi tevékért felelős miniszter tanácsára úgy választják ki a városokat, hogy amikor a hónap végén egy kiválasztott városból elviszik a fővárosba a tevéket, ne haladjanak keresztül olyan városon, ami szintén részt vesz az Országos Tevenevelő Programban.

Írj programot, ami megadja, hogy mely városokat kell bevenni az Országos Tevenevelő Programba, hogy a király a lehető legtöbb tevéhez jusson havonta!

## Bemenet

A *standard bemenet* első sorában a városok száma (1≤N≤100000) szerepel. A főváros az 1-es sorszámú. A második sor tartalmazza, hogy az egyes városokban havonta hány tevét nevelnek (1≤Ti≤109). A következő N-1 sor annak a városnak a sorszámát tartalmazza, amely az i+1-edik városból a fővárosba vezető úton az első város (1≤Ai≤i).

## Kimenet

A *standard kimenet* első sorába a király által havonta megszerezhető tevék maximális számát kell írni! A második sorba az ehhez szükséges kiválasztott városok számát kell írni! A harmadik sorba számot kell írni, a kiválasztott városok sorszámait, tetszőleges sorrendben! Több megoldás esetén bármelyik megadható.

## Példa

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 8  5 5 7 1 2 3 2 4  1  1  2  2  2  3  3 | 13 4  A képen szöveg, óra, karóra látható  Automatikusan generált leírás3 4 5 6 |

## Korlátok

Időlimit: 0.5 mp.

Memórialimit: 32 MB

## Pontozás

A pontok 50%-a szerezhető, ha helyes az első sor.

# Útadó

Binárisország N városból áll, amelyek N-1 úttal vannak összekötve. Azt is tudjuk, hogy az utak hálózata bináris fa alakú és az ország fővárosa az 1-es sorszámú város. Mivel Binárisország válságba került, az ország elnöke útadót szeretne kivetni az összes útra. Gazdasági tanácsadói már javasoltak neki N-1 adómennyiséget (W1,W2,...,WN-1), amelyeket az N-1 út között osztanának el valahogy (minden lehetséges adómennyiséget egy úthoz rendelve és minden úthoz egy lehetséges adómennyiséget rendelve). Neked mint az ország programozásügyi miniszterének, meg kell határoznod hogy maximum mennyi bevétele lehet az államháztartásnak az útadókból! Egy elosztás esetén az ország bevétele a következőképpen értendő: vegyünk minden városban N-1 embert, akik az összes többi városba elutaznak, az összes, az utazások során befizetett útadó lesz ezen elosztás bevétele (minden ember, amikor áthalad egy úton. az ahhoz hozzárendelt adómennyiséget befizeti).

Írj programot, amely meghatározza a lehetséges maximum bevételt!

## Bemenet

A *standard bemenet* első sorában Binárisország városainak a száma található (1≤N≤ 50000). A következő N-1 sor mindegyikében egy-egy út két végpontja található (1≤ai≠bi≤N, ahol ai közelebb van a fővároshoz mint bi). Az utolsó sorban a lehetséges adómennyiségek vannak (1≤Wi≤109).

## Kimenet

A *standard kimenet* első sorába a maximum bevétel 32609-tel vett osztási maradékát kell írni! Minden további sorban 3 szám legyen egy sorban, az első kettő egy út két végpontja (minden út pontosan egyszer szerepeljen), a harmadik ezen maximális bevételű elosztás esetén az ehhez az úthoz tartozó adó mennyisége! Több lehetséges megoldás esetén bármelyik megadható.

## Példa

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 5 1 2 2 3 1 4 2 5 5 6 3 1 | 144 5 2 5 1 2 6 2 3 3 4 1 1 |

## Korlátok

Időlimit: 0.7 mp.

Memórialimit: 64 MiB

## Pontozás

A pontok 50%-a jár helyes maximum értékre, 50%-a jár helyes maximum bevételhez tartozó elosztásra.

A pontok 20%-a jár olyan tesztekre, amelyekben N≤100.

A pontok 16%-a jár olyan tesztekre, amelyekben minden csúcs foka legfeljebb 2.

A pontok 20%-a jár olyan tesztekre, amelyekben N≤1000.