

**9. feladat: Játék (50 pont)**

Ádám és Éva kétszemélyes játékot játszik egy olyan táblán, amelyen van  $N$  pont és bizonyos pontpárokat vonal köt össze. Egy pontnak azok a szomszédjai, amelyek vonallal vannak összekötve vele. Minden pontnak legfeljebb 3 szomszédja van. A pontokat az  $1, \dots, N$  számokkal azonosítják, az 1. pont kitüntetett, ennek pontosan 2 szomszédja van. Ádám kék, Éva pedig piros korongokat helyezhet a tábla pontjaira. A játék kezdetén Ádám egy kék korongot rak az 1. pontra. Ezután a játékosok felváltva lépnek, tehát Éva következik.

1. Ádám korongot csak olyan pontra rakhat, amelyen nincs korong és amelynek legalább egy szomszédján saját, azaz kék korong van.
2. Éva bármely pontra rakhatja korongját, feltéve, hogy ott nincs korong.

A játék akkor ér véget, ha az aktuális játékos nem tud lépni. Ádám célja a játékban az, hogy a legtöbb pontra helyezzen kék korongot, Éva célja pedig, hogy akadályozza Ádámot.

Készíts programot (**jatek.pas**, **jatek.c**, ...), amely kiszámítja, hogy hány helyre tud Ádám kék korongot rakni a legjobb esetben! Tehát van olyan játékmenet, amelyben ennyi korongot helyez el, de ennél többet nem tud (mert Éva okosan játszik).

A **jatek.be** szöveges állomány első sora egy egész számot tartalmaz,  $N$  ( $N \leq 1000$ ) a táblán lévő ponthelyek számát. A következő  $N$  sor mindegyike egy pont szomszédjait tartalmazza, a felsorolást a 0 szám zárja. Az  $i$ -edik pont szomszédjait az állomány  $i+1$ -edik sora tartalmazza. Természetesen, ha  $U$ -nak szomszédja  $V$ , akkor  $V$ -nek szomszédja  $U$ . Bármely két pont között legfeljebb egy vonal van.

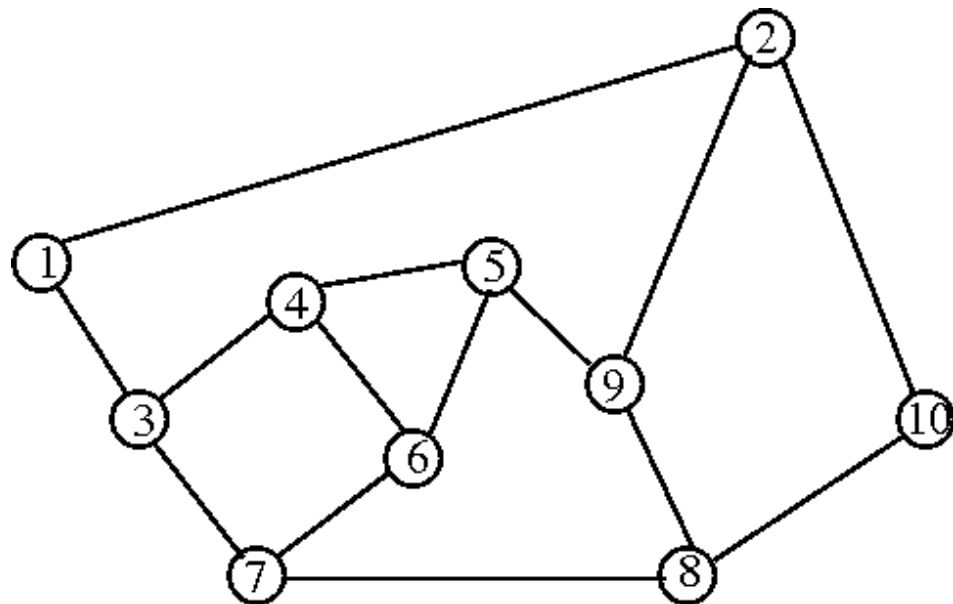
A **jatek.ki** szöveges állomány első és egyetlen sora egy egész számot tartalmazzon, azon korongok számát, amennyit Ádám legjobb esetben el tud helyezni a táblán!

**Példa:****jatek.be**

```
10
2 3 0
1 9 10 0
1 4 7 0
3 5 6 0
4 6 9 0
4 5 7 0
3 8 6 0
7 9 10 0
8 5 2 0
2 8 0
```

**jatek.ki**

3



**10. feladat:** Villamos (50 pont)

Egy városban  $N$  villamosmegálló van, a megállókat 1 és  $N$  közötti sorszámukkal azonosítjuk. A városban  $M$  villamos-vonal van, minden villamos adott megállókon keresztül közlekedik. Egyes megállók között újabb vágányokat lehet építeni, ezzel megoldva villamosvonalak összekötését.

Készíts programot (**villamos.pas, villamos.c, ...**), amely a villamosvonalak és a kiépítendő vágányok költségének ismeretében megadja, hogy mely vágányokat kell kiépíteni ahhoz, hogy a villamoshálózat összefüggő legyen!

A **villamos.be** állomány első sorában a megállók száma ( $1 \leq N \leq 1000$ ) a villamosvonalak száma ( $1 \leq M \leq 100$ ), valamint a megépíthető vágányok száma ( $1 \leq K \leq 10000$ ) van, egy-egy szóközzel elválasztva. A következő  $M$  sor egy-egy villamos-vonal leírását tartalmazza. Minden egyes sor első száma a villamos-vonal megállói száma ( $2 \leq V_i \leq N$ ), a következő  $V_i$  szám pedig a vonal egyes állomásai sorszáma ( $1 \leq \text{sorszám} \leq N$ ), egy-egy szóközzel elválasztva. Az ezt követő  $K$  sorban a megépíthető vágányok száma. Mindegyik sor első két száma a vágány két végpontjának állomás sorszáma ( $1 \leq \text{sorszám} \leq N$ ), a harmadik pedig a kiépítés költsége ( $1 \leq \text{költség} \leq 32767$ ).

A **villamos.ki** állomány első sorába a villamoshálózat összefüggővé tételének minimális költsége kerüljön! A második sorba a megépítendő vágányok  $S$  számát kell írni, a következő  $S$  sorba pedig a megépítendő vágányokat megadó két-két állomás sorszámot! Ha a feladat nem oldható meg, akkor az egyetlen sorba egy -1-et kell kiírni!

**Példa:****villamos.be**

```
11 4 5
3 1 2 3
4 8 2 3 4
3 5 6 7
3 11 10 9
2 6 10000
3 11 5000
11 7 8000
5 9 6000
8 4 100
```

**villamos.ki**

```
11000
2
3 11
5 9
```

