

CEOI 2026



2. nap
(Magyar)

Day 2
(Hungarian)

Virág metszés

Időlimit: 4 s Memórialimit: 256 MiB

A CEOI közösségi kertjében különleges virágok egy gyűjteményét neveljük. Ezen virágok jellemzője, hogy szorosan összefonják a gyökereiket. Ha elvágjuk a gyökereket, azok újránőnek, feltéve, hogy nem vágunk túl agresszíven, és nem okozunk helyrehozhatatlan kárt a virágokban.

Ha egy virágpár, a és b , összefonja a gyökereit, akkor azt mondjuk, hogy **kapcsolódnak**. Ellenkező esetben **nem kapcsolódnak**. A gyökerek a következő szabály szerint nőnek: Legyen a és b két nem kapcsolódó virág. Ha létezik legalább két másik virág, c és d , amelyekre igaz, hogy mind a , mind b kapcsolódik c -hez is és d -hez is, akkor gyökerek nőnek a és b között, és kapcsolódóvá válnak.

A gyökerek már régóta növekednek, és minden olyan gyökér, amely a fenti szabály szerint kinőhetett, mostanra kinőtt. Más szóval, ha a és b két virág, amelyek mindketten kapcsolódnak valamely c és d virágokhoz, akkor garantált, hogy a és b is kapcsolódnak egymáshoz.

Most fel kell ásnunk a kertet, és át kell költöztetnünk a gyűjteményt a következő CEOI helyszínére. A költöztetés egyszerűsítése érdekében szeretnénk elvágni a lehető legtöbb gyökeret. Ugyanakkor azt is szeretnénk, hogy a gyökerek visszánőjenek a jelenlegi állapotukba az átköltöztetés után.

Mi a kapcsolódó virágpárok maximális száma, amelyek közötti gyökereket elvághatjuk úgy, hogy azok később visszánőjenek a jelenlegi állapotukba? (Nem számít, hogy ehhez mennyi ideig kell növekedniük.)

Bemenet

A bemenet első sora két egész számot tartalmaz, a virágok n számát és a köztük lévő kapcsolatok m számát.

A következő m sor mindegyike két egész számot tartalmaz, a_i -t és b_i -t, ahol a_i és b_i kapcsolódó virágok. A virágokat az $1 \dots n$ számokkal azonosítjuk.

Garantált, hogy a bemenet megfelel a feladatleírásban szereplő növekedési szabálynak.

Korlátok

- $1 \leq n \leq 1000$
- $1 \leq m \leq 100\,000$

Részfeladatok

1. részfeladat (20 pont): $n \leq 10$ és $m \leq 20$
2. részfeladat (14 pont): $m = \frac{n \cdot (n-1)}{2}$
3. részfeladat (15 pont): Minden virág legfeljebb 7 másik virághoz kapcsolódik.
4. részfeladat (15 pont): $n \leq 50$ és $m \leq 1000$
5. részfeladat (36 pont): Nincsenek további korlátozások.

Kimenet

A kimenet egyetlen egész szám — a vágások maximális száma.

Példák

Bemenet

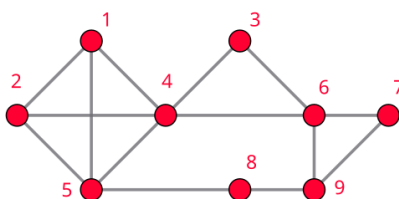
9 14
1 2
1 4
1 5
2 4
2 5
3 4
4 5
3 6
4 6
6 7
6 9
7 9
8 9
5 8

Kimenet

2

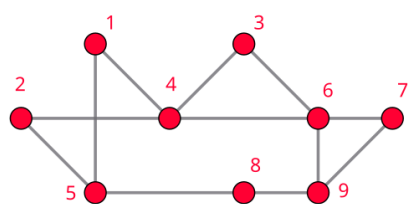
Magyarázat

Az alábbi ábra a virágokat és a köztük lévő kapcsolatokat ábrázolja a vágások előtt. Ellenőrizhető, hogy a feladateleírásban szereplő növekedési eljárás alapján nem alakulhatnak ki további kapcsolatok.



Kettő vágást végeztünk az 1 és 2 virágok között, továbbá a 4 és 5 virágok között. Az alábbi ábra a vágások utáni kapcsolatokat ábrázolja. Az elvágott gyökök újra nőhetnek, mivel 1 és 2 virágok mindketten kapcsolódnak a 4 és 5 virágokhoz, és hasonlóan a 4 és 5 virágok mindketten kapcsolódnak az 1 és 2 virágokhoz.

Ellenőrizhető, hogy kettőnél több vágást nem lehet elvégezni.



Tornyok

Időlimit: 1 s Memórialimit: 256 MiB

Egy egyenes vonal mentén n számítógép és m torony helyezkedik el, mind különböző pozíciókban. A számítógépeket **párokba kell rendezned** és kábelek segítségével összekötnöd úgy, hogy minden kábel egy számítógépnél kezdődik, keresztül halad néhány (0 vagy több) tornyon, majd egy másik számítógépnél végződik. A számítógépek száma páros, és minden számítógépet össze kell kötni valamelyik másik számítógéppel. Nem szabad a számítógépeket önmagukkal összekötni.

Minden kábel tetszőleges tornyokon keresztül haladhat (nem csak olyanokon, amik a két összekötött számítógép között helyezkednek el az egyenesen), tetszőleges sorrendben. A kábel el is haladhat tornyok mellett anélkül, hogy áthaladna rajtuk, illetve ugyanazon a tornyon többször is áthaladhat.

Legyen a és b két egymással összekötött számítógép pozíciója, és legyenek $x_1, \dots, x_k$ azoknak a tornyoknak a pozíciói, amelyeken a kábel keresztül halad. A kábel **hossza** ekkor az alábbi összeg:

$$|a - x_1| + |x_1 - x_2| + \dots + |x_{k-1} - x_k| + |x_k - b|.$$

A kábel **pontszámát** az $f \cdot u - l$ értéként definiáljuk, ahol l a kábel hossza, f egy rögzített egész érték, u pedig azon **különböző** tornyok száma, amelyeken a kábel keresztül halad. Ugyanazon a tornyon több kábel is keresztül haladhat, ekkor a torony mindegyik ilyen kábel pontszámához hozzájárul.

A számítógépek egy párosításának **pontszáma** a párokat összekötő kábelek pontszámainak összege. Írj programot, ami meghatározza a legmagasabb pontszámú párosítás pontszámát.

Bemenet

Az első sor a tesztesetek T számát tartalmazza. Ezt T teszteset leírása követi.

Minden teszteset három sorból áll. Az első sor három egész számot tartalmaz: a számítógépek n számát, a tornyok m számát és az f egész értéket. A második sor n egész számot tartalmaz, a számítógépek $a_1, a_2, \dots, a_n$ pozícióit. A harmadik sor m egész számot tartalmaz, a tornyok $b_1, b_2, \dots, b_m$ pozícióit.

Korlátok

Legyen N és M rendre az n és m értékek összege a tesztesetekben.

- $1 \leq T \leq 10\,000$
- $1 \leq N, M \leq 200\,000$
- n páros.
- $0 \leq f \leq 10^9$
- $1 \leq a_i, b_i \leq 10^9$
- A számítógépek és a tornyok pozíciói mind különbözőek (egy adott teszteseten belül).

Részfeladatok

1. részfeladat (5 pont): $N \leq 5000$, $m = 1$
2. részfeladat (10 pont): $T \leq 20$, $n \leq 10$, $m \leq 100$
3. részfeladat (27 pont): $N, M \leq 5000$
4. részfeladat (21 pont): $N \leq 5000$
5. részfeladat (37 pont): Nincsenek további korlátozások.

Kimenet

A kimenetre T sort kell kiírni, soronként egy egész értékkel. Az i -edik sor a maximális pontszámú párosítás pontértékét tartalmazza az i -edik tesztesetben.

Példák**Bemenet**

```
4
2 1 100
1 10
11
4 1 10
2 4 6 8
20
4 1 10
2 4 6 8
5
6 3 10
2 13 4 8 6 10
5 1 9
```

Kimenet

```
89
-4
12
51
```

Vim

Időlimit: 1 s Memórialimit: 256 MiB

Nehéz lenne *kockább* dolgot találni a régi öreg Vimnél. A Vim egy szövegszerkesztő, amiben olyan dolgokat tehetünk meg, amelyekről a „normál” szerkesztőkben (például ChaIDE) *még* csak álmodhatunk. Persze csak akkor, ha hajlandóak vagyunk egy hetet tanulással és egy hónapot gyakorlással tölteni, hogy a Vim által kínált szokatlan, de hatékony parancsok „izommemóriává” váljanak.

Hasonlóan más szerkesztőkhöz, a Vimben is van egy *kurzor*, amely mindig az egyik karakteren áll. Kezdetben ez a karakter a szöveg első karaktere. Továbbá, ahogy más szerkesztőkben, a Vimben is van egy *vágólap* (clipboard), amely szövegrészek tárolására és közvetve másolására, mozgatására szolgál. A vágólap kezdetben üres.

Ebben a feladatban feltesszük, hogy kezdetben pontosan egy mínusz ('-') karakter van a szerkesztőben, és a célunk annak az elérése, hogy pontosan n egymást követő mínuszjel legyen a szerkesztőben. Ehhez négy utasítást fogunk használni:

- **h**: Ha a kurzor az első karakteren áll, akkor ez az utasítás nem csinál semmit, egyébként a kurzort egy karakterrel balra mozgatja.
- **l**: Ha a kurzor az utolsó karakteren áll, akkor ez az utasítás nem csinál semmit, egyébként a kurzort egy karakterrel jobbra mozgatja.
- **Y**: A kurzor pozíciójától (azaz attól a karaktertől, amin a kurzor áll) a szöveg végéig terjedő karaktersorozat a vágólapra másolódik, felülírva a vágólap korábbi tartalmát. (Ha a kurzor a szöveg első karakterén áll, akkor a teljes szöveg kerül a vágólapra.)
- **P**: A vágólapon tárolt szöveg egy másolata beszúrásra kerül az elé a karakter elé, amin a kurzor áll, majd a kurzor az utoljára beszúrt karakterre lép. A vágólap tartalma nem változik. Ha a vágólap üres, akkor nem történik semmi.

Írj programot, ami beolvassa az n értéket, és kiírja, hogy legkevesebb hány utasítás szükséges ahhoz, hogy pontosan n egymást követő '-' karakter legyen a szerkesztőben. A programodnak egy megfelelő utasítássorozatot is meg kell adnia.

Bemenet

Az első sor a tesztesetek t számát tartalmazza. A következő t sor mindegyike egy teszteset leírását tartalmazza: az i -edik sor az i -edik tesztesetben előállítandó karaktersorozat n hosszát adja meg.

Korlátok

- $1 \leq t \leq 100$
- $1 \leq n \leq 10\,000\,000$

Részfeladatok

1. részfeladat (20 pont): $n \leq 100$

2. részfeladat (8 pont): $n \leq 1000$
3. részfeladat (18 pont): $n \leq 10\,000$
4. részfeladat (18 pont): $n \leq 100\,000$
5. részfeladat (18 pont): $n \leq 1\,000\,000$
6. részfeladat (18 pont): Nincsenek további korlátozások.

Ebben a feladatban **részpontokat** lehet szerezni. Ha a programod csak a szükséges utasítások számát írja ki helyesen, de nem ad példát megfelelő utasítássorozatra, vagy a megadott utasítássorozat hibás, akkor az adott **részfeladat pontjainak felét** kapod meg.

Kimenet

Minden tesztesethez egy sort kell kiírni, ami a szükséges utasítások számát és egy megfelelő utasítássorozatot tartalmaz.

Példák

Bemenet

2
21
2

Kimenet

10 YPYPhPYPPP
2 YP

Magyarázat

Az alábbi táblázat szemlélteti, hogy a példa első tesztesetében a YPYPhPYPPP utasítássorozat helyes eredményt ad. A Szerkesztő oszlopban a '=' karakter azt a '-' karaktert jelöli, amin a kurzor áll.

Lépés	Utasítás	Szerkesztő	Vágólap
0		=	(üres)
1	Y	=	-
2	P	=-	-
3	Y	=-	--
4	P	=---	--
5	h	=----	--
6	P	=-----	--
7	Y	=-----	-----
8	P	-----=-----	-----
9	P	-----=-----	-----
10	P	-----=-----	-----