Feladat 1 - Energia 100pont

Nagyváradon egy új keréparkölcsonző nyílt és új kerékpárutak épültek. Az újonnan megépített kerékpárutak összekötik a város fontosabb pontjait és ezeken az utakon a kerékpárosok mindkét irányban haladhatnak. Egyes pontok között a kerékpárút egy sík felületen keresztül halad át, más pontok között az út emelkedik vagy ereszkedik a menetiránytól függően. Emiatt két pont között az energiafogyasztás is változik. Ha két pontot egy sík kerékpárút köt össze, akkor azt a távot **1** egység energiafogyasztással lehet megtenni, ha a két pont között az út emelkedik a fogyasztás **2** egység energia, ha viszont két pont között az út ereszkedik az energiafogyasztás **0.** Például ha **x** és **y** pontok között az út emelkedik a kerékpáros **2** egység energiafogyasztása után teszi meg a távot, viszont értelemszerűen **y** és **x** pontok között az út ereszkedik, tehát ezt az útat **0** energiafogyasztással teszi meg.

Követelmény

Ismerve az ***n***-et, város azon pontjainak számát, amelyek között létezik kerékpárút, ***m***-et, a sík kerékpárutak számát, a sík kerékpárutak kiinduló és végponját, valamint ***k***-t, az ereszkedő kerékpárutak számát és az ereszkedő kerékpárutak kezdő és végpontját, írjunk programot, amelyik meghatározza:

a) Melyik pontban keresztezi egymást a legtöbb sík kerékpárút. Ha több megoldás létezik, a legkisebb értéket kell kiírni.
b) Két ***x*** és ***y***-nal jelölt pont között mennyi a minimális energiafogyasztás.

Bemeneti adatok

Az **`energia.in bemeneti`** állomány első sora egy ***p (1 ≤ p ≤ 2)***, természetes számot tartalmaz, amelyik a megoldásra váró követelmény sorszámát jelöli. Az állomány második sora három természetes számot tartalmaz, ***n, m*** és ***k***, egy-egy szóközzel elválasztva, a leírásban megadott jelentéssel. A következő ***m*** sor mindegyikében ***két*** természetes szám található egymástól szóközzel elválasztva, a **sík** kerékpárutak kiinduló, illetve végpontja. A következő ***k*** sor mindegyikében ***két*** természetes szám található egymástól szóközzel elválasztva, az **ereszkedő** kerékpárutak kiinduló, illetve végpontja. Az állomány utolsó sora két természetes számot tartalmaz, x és y.

Kimeneti adatok

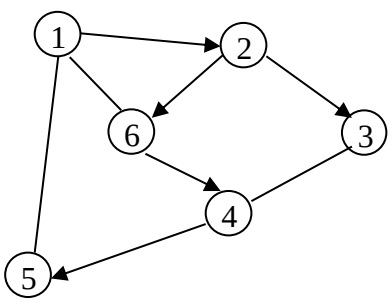
Ha **`p=1`** akkor **csak a követelmény a) pontját** kell megoldani. Az **`energia.out`** kimeneti állomány egy természetes számot tartalmaz, annak a pontnak a sorszámát amelyikben a legtöbb sík kerékpárút találkozik.

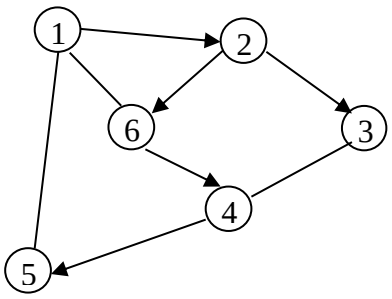
Ha **`p=2`** akkor **csak a követelmény b) pontját** kell megoldani. Az **`energia.out`** kimeneti állomány egy természetes számot tartalmaz, a minimális energiafogyasztást az x-szel és y-nal jelölt pontok között.

Megszorítások és pontosítások

- a város kerékpárutakkal összekötött pontjait 1-től n-ig természetes számokkal jelöljük;
- $1 \leq \mathbf{n} \leq 200$, $1 \leq \mathbf{m} + \mathbf{k} \leq 19900$;
- Ha két, a-val és b-vel jelölt pont között az út emelkedik, akkor b és a között az út ereszkedik;
- két pont között az út vagy csak sík, vagy csak emelkedi, vagy csak ereszkedik.
- a város bármelyik két pontja össze van kötve egy kerékpárúttal (direkt vagy indirekt módon).
- a teszteknek a 30%-án a bemeneti állományok első sorában az 1-es érték található, a többi tesztállományok első sorában a 2-es érték található.

Példa

`energia.in`	`energia.out`	`Magyarázat`
1 6 3 5 1 5 1 6 3 4 1 2 2 6 2 3 4 5 6 4 1 5	1	 Csak a követelmény a) pontját kell megoldani. Az 1-es jelölt pontban találkozik a legtöbb sík kerékpárút.

2 6 3 5 1 5 1 6 3 4 1 2 2 6 2 3 4 5 6 4 1 5	0		Csak a követelmény b) pontját kell megoldani. Az 1→2→6→4→5 útvonalon az energiafogyasztás minimilis. Ennek értéke 0.

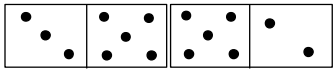
Maximális futási idő: 1 mp/teszt.
Rendelkezésre álló memória: 8 MB, amelyből 4 MB a veremnek.
A forráskód maximális mérete: 20 KB.

Feladat 2 – Dominó 100pont

Adott **n** darab dominó. Ezekből a dominókból különböző dominósorokat lehet építeni, a következő szabályok betartásával:

- Az első dominó kötelező módon a sor része.
- A következő dominókat a megjelenésük sorrendjében kell figyelembe venni (minden alkalommal el kell dönteni, hogy felhasználjuk-e a sorban vagy sem).
- Minden dominót a meglévő sor végéhez kell illeszteni a megadott pozicióban illetve 180°=kal elforditva vagy az adott domiót félrerakjuk, és később sem vehetjük figyelembe.
- Egy dominó akkor illeszkedik a sor végéhez, ha a sor végén talalható dominó szabadoldalán a pöttyök száma megegyezik az illeszkedő dominó egyik felén található pöttyök számával.

Példa:



Követelmény

Készíts programot, amely meghatározza a kirakható leghosszabb illeszkedő dominósor hosszát.

Bemeneti adatok

A **DOMINO.IN** bemeneti állomány első sora egy **n** természetes számot tartalmaz, a rendelkezésre álló dominók számát. A következő **n** sor mindenikében két természetes szám található **x** és **y,** egy-egy szóközzel elválasztva, amelyek egy dominó két felén található pöttyök számát jelentik.

Kimeneti adatok

A **DOMINO.OUT** kimeneti állomány egyetlen természetes számot fog tartalmazni, a leghosszabb kirakható dominósor elemeinek számát, a fenti szabályok betartásával.

Megkötések és pontosítások

- $1 \leq \mathbf{n} \leq 100000$;
- $0 \leq \mathbf{x}, \mathbf{y} \leq 9$;
- a dominókat lehet forgatni miellőt a sor végéhez elhelyezzük.

Exemplu

DOMINO.IN	**DOMINO.OUT**
6 1 2 1 6 2 3 1 4 2 3 4 3	5

Maximális futásidő: **1** másodperc/teszt.
Rendelkezésre álló memória 10 MB.
A forráskód maximális mérete: **5** KB.