



Pac-Man (pacman)

Alessandro a Pac-Man új, 3 dimenziós változatát kódolja. A játéktáblát egy háromdimenziós rács írja le, egyes cellák blokkolva vannak, a többi szabad. Pac-Man és a szellemek bármelyik szabad cellába léphetnek, amelynek az aktuális cellával van közös lapja.

Egy szellem mozgására Alessandro egy egyszerű szkriptet írt. Ha a szellem az A cellából a B cellába akar eljutni (ahol az A cellát az (A_x, A_y, A_z) , a B cellát a (B_x, B_y, B_z) koordináták írják le), akkor a szellem addig ismétli az alábbi eljárást, amíg el nem éri a célját, vagy el nem bukik:


- Ha a szellem képes csökkenteni az x tengely mentén a távolságot (azaz $|A_x - B_x| - t$), akkor ezt úgy fogja megtenni, hogy egy cellát mozog a x tengely mentén.
- Ellenkező esetben, ha a szellem az y tengely mentén csökkentheti a távolságot (azaz $|A_y - B_y| - t$), akkor ezt úgy fogja megtenni, hogy egy cellát mozog a y tengely mentén.
- Egyébként, ha a z tengely mentén tudja csökkenteni a távolságot (azaz $|A_z - B_z| - t$), akkor ezt úgy teszi, hogy egy cellát mozog a z tengely mentén.
- Ellenkező esetben nem tud lépni és elbukik.



1. ábra. Alessandro egy klasszikus Pac-Man játékot játszik.

Alessandro verziójában az N szabad cellát az (X_i, Y_i, Z_i) koordinátákkal megadott mezők írják le (amelyek 0-tól $N - 1$ -ig vannak indexelve). Szeretné tudni, hogy a stratégiája elég okos-e ahhoz, hogy a szellemeket irányítsa.

Segíts neki egy olyan program megírásával, amely meghatározza, hogy minden A és B szabad cellapár esetén a szellemnek sikerül-e az A cellából kiindulva elérnie a B cellát!

 Az értékelő rendszerből letölthető csatolmányok közt találhatsz `pacman.*` nevű fájlokat, melyek a bemeneti adatok beolvasását valósítják meg az egyes programnyelveken. A megoldásodat ezekből a hiányos minta implementációkból kiindulva is elkészítheted.

Bemenet

A bemenet a következőkből áll:

- az első sor egy egész N számot tartalmaz.
- a második sor az X_0, \dots, X_{N-1} egész számokat tartalmazza.
- a harmadik sor az Y_0, \dots, Y_{N-1} egész számokat tartalmazza.
- a negyedik sor a Z_0, \dots, Z_{N-1} egész számokat tartalmazza.

Kimenet







Egyetlen szót kell kiírnod. Azt, hogy YES, ha a szellemek bárholnan bárhová el tudnak jutni a fenti programmal. Ha pedig ez nem lehetséges, akkor azt írd ki, hogy NO.

Korlátok

- $1 \leq N \leq 100\,000$.
- $0 \leq X_i, Y_i, Z_i < 100\,000$ minden $i = 0 \dots N - 1$ esetén.

Pontozás

A megoldásodat sok különböző tesztesetre lefuttatjuk. A tesztesetek részfeladatokba vannak csoportosítva. Egy-egy részfeladatot akkor tekintünk megoldottnak, ha volt legalább egy olyan beadásod, amely az adott részfeladat minden tesztesetére helyes megoldást adott. A feladat összpontszámát a megoldott részfeladatokra kapott pontszámok összege adja.

- **1. Részfeladat** (0 pont) Példák.

- **2. Részfeladat** (18 pont) $N \leq 100$ és $X_i, Y_i, Z_i < 100$.

- **3. Részfeladat** (19 pont) $N \leq 7500$ és $X_i, Y_i, Z_i < 100$.

- **4. Részfeladat** (24 pont) $N \leq 1000$ és $Z_i = 0$.

- **5. Részfeladat** (22 pont) $X_i, Y_i, Z_i < 100$.

- **6. Részfeladat** (17 pont) Nincsenek további megkötések.


Példák

input	output
4 0 0 1 1 0 1 1 2 0 0 0 0	YES
8 0 1 2 2 2 1 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 1 1	NO
5 0 0 1 1 2 0 1 1 0 2 0 0 0 0 2	NO

Magyarázat

Az **első példában** a szellem bármely szabad cellából indulva tetszőleges célállomást el tud érni. Például a szellem az $A = (0, 0, 0)$ cellából a $B = (1, 2, 0)$ cellába a $(0, 0, 0) \rightarrow (0, 1, 0) \rightarrow (1, 1, 0) \rightarrow (1, 2, 0)$ útvonalon haladva jut el.

A **második példában** a szellem nem tud az $A = (1, 0, 0)$ cellából a $B = (1, 1, 1)$ cellába eljutni a megadott módon.