

Ideális város

Leonardo egy ideális várost tervezett.

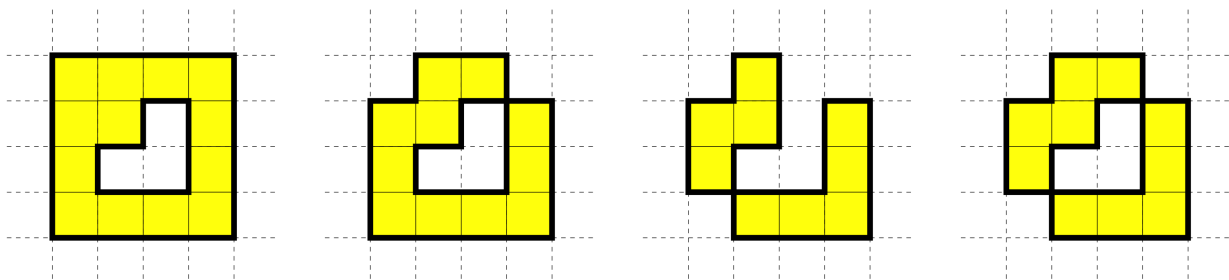
Az ideális város

A város N mezőt foglal el egy négyzetrácson. A mezőket (sor,oszlop) koordinátákkal azonosítjuk. A $(0,0)$ mező a bal felső sarok. Az (i,j) mező szomszédai az $(i-1,j)$, az $(i+1,j)$, az $(i,j-1)$, és az $(i,j+1)$. A város minden blokkja pontosan 1 mezőt foglal el, az (i,j) mezőn akkor és csak akkor lehet blokk, ha $1 \leq i, j \leq 2^{31} - 2$. Két blokk akkor szomszédos, ha szomszédos mezőn vannak. Egy ideális városban a blokkok úgy vannak összekapcsolva, hogy nincs lyuk a határukon belül, azaz

- Bármely két üres mező között létezik legalább egy, szomszédos üres mezőkből álló sorozat, amely összeköti őket.
- Bármely két nem üres mező között létezik legalább egy, szomszédos nem üres mezőkből álló sorozat, amely összeköti őket.

1. példa

Egyik ábrán sincs ideális város. Az első kettő az első feltételt nem teljesíti, a harmadik a második feltételt nem teljesíti, a negyedik pedig egyik feltételt sem teljesíti.

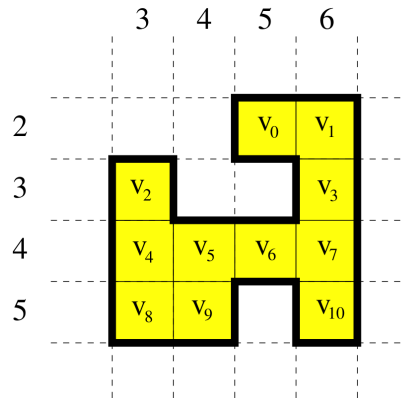


Távolság

A várost szomszédos blokkokon lépkedve járhatjuk be, üres mezőre nem léphetünk. Legyenek v_0, v_1, \dots, v_{N-1} a város N blokkja koordinátái. Egy v_i és v_j blokk távolsága $d(v_i, v_j)$ a lépések legkisebb száma, amivel eljuthatunk egyikről a másikra.

2. példa

Az ideális város $N = 11$ blokkjai: $v_0 = (2, 5)$, $v_1 = (2, 6)$, $v_2 = (3, 3)$, $v_3 = (3, 6)$, $v_4 = (4, 3)$, $v_5 = (4, 4)$, $v_6 = (4, 5)$, $v_7 = (4, 6)$, $v_8 = (5, 3)$, $v_9 = (5, 4)$, és $v_{10} = (5, 6)$. Ekkor: $d(v_1, v_3) = 1$, $d(v_1, v_8) = 6$, $d(v_6, v_{10}) = 2$, és $d(v_9, v_{10}) = 4$.



Feladat

Írj programot, amely kiszámítja az összes v_i and v_j blokk távolságának összegét ($i < j$ -re)! Azaz az alábbi szummát kell kiszámolnod

$$\sum d(v_i, v_j), \text{ ahol } 0 \leq i < j \leq N - 1$$

Írd meg a `DistanceSum(N, X, Y)` eljárást, amely a várost leíró N értékre és X, Y tömbökre kiszámolja a fenti formulát! X és Y N elemű, az i . blokk koordinátái: $(X[i], Y[i])$ $0 \leq i \leq N - 1$, és $1 \leq X[i], Y[i] \leq 2^{31} - 2$. Mivel az eredmény túl nagy is lehet (nem fér el 32 bites változóban), ezért moduló $1\,000\,000\,000$ kell kiírni!

A 2. példában $11 \times 10 / 2 = 55$ pár távolságát kell összeadni, ami 174.

1. részfeladat [11 pont]

$N \leq 200$.

2. részfeladat [21 pont]

$N \leq 2\,000$.

3. részfeladat [23 pont]

$N \leq 100\,000$.

További feltétel: bármely i és j nem üres mezőre, ahol $X[i] = X[j]$, teljesül, hogy közöttük csak nem üres mezők vannak, valamint bármely i és j nem üres mezőre, ahol $Y[i] = Y[j]$, teljesül, hogy közöttük csak nem üres mezők vannak.

4. részfeladat [45 pont]

$N \leq 100\,000$.

Megvalósítás

Egyetlen file-t kell beküldened: `city.c`, `city.cpp` vagy `city.pas`. Ebben kell megvalósítanod a Distance eljárást!

C/C++ program

```
int DistanceSum(int N, int *X, int *Y);
```

Pascal program

```
function DistanceSum(N : LongInt; var X, Y : array of LongInt) : LongInt;
```

Más eljárásokat is írhatsz, standard inputot, outputot és file-okat nem használhatsz.

Minta értékelő

A minta értékelő a bemenetet az alábbi formában várja:

- 1. sor: N ;
- 2, ..., $N + 1$. sorok: $X[i]$, $Y[i]$.

Idő és memória limitek

- Időlimit: 1 másodperc.
- Memórialimit: 256 MiB.