

## Tobulas miestas

Leonardo, kaip ir daugelis kitų to paties laikotarpio italų mokslininkų ir menininkų, labai domėjosi miestų planavimu. Jis norėjo suplanuoti tobulą miestą: patogų, talpų ir racionaliai naudojančią išteklius, visiškai priešingą nei siauri, klaustrofobiški viduramžių miestai.

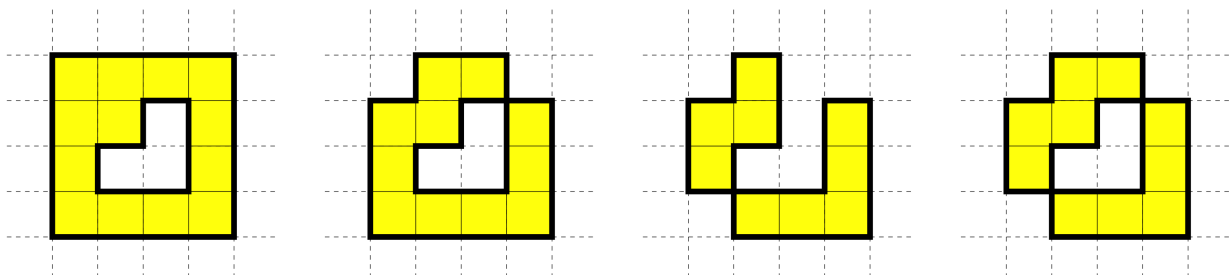
### Tobulas miestas

Miestą sudaro  $N$  blokų, išdėstytų begalinėje lentelėje. Kiekvieną lentelės kvadratėlį apibūdina koordinatų pora (eilutė, stulpelis). Kvadratėliui  $(i, j)$  gretimi kvadratėliai yra  $(i - 1, j)$ ,  $(i + 1, j)$ ,  $(i, j - 1)$  ir  $(i, j + 1)$ . Kiekvienas lentelėje patalpintas blokas pilnai uždengia vieną kvadratėlį. Bloką galima patalpinti ant kvadratėlio  $(i, j)$  tada ir tik tada, jei  $1 \leq i, j \leq 2^{31} - 2$ . Kvadratėlių koordinatės naudosime ir ant jų esančių blokų apibūdinimui. Du blokai yra gretimi, jei jie patalpinti ant gretimų kvadratėlių. Tobulame mieste visi blokai yra sujungti taip, kad miesto viduje nėra skylių. T.y. kvadratėliai turi tenkinti tokias dvi sąlygas:

- bet kuriuos du *tuščius* kvadratėlius jungia bent viena gretimų *tuščių* kvadratėlių seka;
- bet kuriuos du *netuščius* kvadratėlius jungia bent viena gretimų *netuščių* kvadratėlių seka;

### 1-as pavyzdys

Nė vienas iš žemiau pateiktų blokų išdėstymų nėra tobulo miesto planas: kairiausi du išdėstymai neatitinka pirmosios sąlygos, trečiasis neatitinka antrosios sąlygos, o ketvirtasis neatitinka abiejų sąlygų.

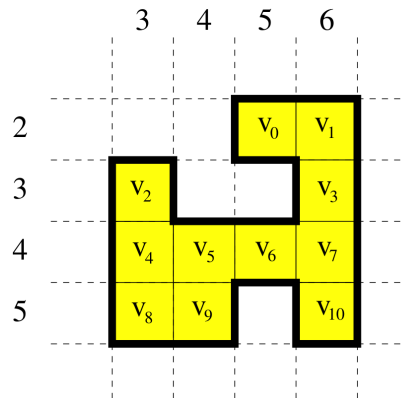


### Atstumas

Judant mieste, vienas *žingsnis* reiškia perėjimą iš vieno bloko į gretimą. Tuščiais kvadratėlius judėti negalima. Tarkime, kad  $v_0, v_1, \dots, v_{N-1}$  yra lentelėje patalpintų  $N$  blokų koordinatės. Bet kuriems dviems blokams  $v_i$  and  $v_j$ , jų atstumas  $d(v_i, v_j)$  lygus mažiausiam žingsnių, atliekamų pereinant iš vieno šių blokų į kitą, skaičiui.

## 2-as pavyzdys

Žemiau pateiktas blokų išdėstymas atitinka tobulą miestą sudarytą iš  $N = 11$  blokų, kurių koordinatės yra  $v_0 = (2, 5)$ ,  $v_1 = (2, 6)$ ,  $v_2 = (3, 3)$ ,  $v_3 = (3, 6)$ ,  $v_4 = (4, 3)$ ,  $v_5 = (4, 4)$ ,  $v_6 = (4, 5)$ ,  $v_7 = (4, 6)$ ,  $v_8 = (5, 3)$ ,  $v_9 = (5, 4)$  ir  $v_{10} = (5, 6)$ . Pavyzdžiui,  $d(v_1, v_3) = 1$ ,  $d(v_1, v_8) = 6$ ,  $d(v_6, v_{10}) = 2$  ir  $d(v_9, v_{10}) = 4$ .



## Užduotis

Parašykite programą, kuri, gavusi tobulo miesto išplanavimą, apskaičiuotų atstumų tarp visų blokų porų  $v_i$  ir  $v_j$  (kur  $i < j$ ) sumą. Konkrečiau, jūsų programa turi apskaičiuoti šios sumos vertę:

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

Realizuokite paprogramę `DistanceSum(N, X, Y)`, kuri apskaičiuotų aukščiau pateiktos formulės reikšmę.  $X$  ir  $Y$  yra masyvai iš  $N$  elementų;  $i$ -ojo bloko koordinatės yra  $(X[i], Y[i])$ , kur  $0 \leq i \leq N - 1$  ir  $1 \leq X[i], Y[i] \leq 2^{31} - 2$ . Kadangi rezultatas gali būti labai didelis ir netilpti į 32 bitų skaičių, jums reikia apskaičiuoti jo dalybos iš 1 000 000 000 (vieno milijardo) liekaną.

Antrame pavyzdyje yra  $11 \times 10 / 2 = 55$  blokų poros. Visų porų atstumų suma yra 174.

### 1 užduotis [11 taškų]

$N \leq 200$ .

### 2 užduotis [21 taškas]

$N \leq 2\,000$ .

### 3 užduotis [23 taškai]

$N \leq 100\,000$ .

Taip pat galioja ir šios dvi sąlygos: bet kuriems dviems blokams  $i$  ir  $j$ , kur  $X[i] = X[j]$ , visi tarp jų esantys kvadratai nėra tušti; bet kuriems dviems blokams  $i$  ir  $j$ , kur  $Y[i] = Y[j]$ , visi tarp jų esantys kvadratai taip pat nėra tušti.

## 4 užduotis [45 taškai]

$N \leq 100\,000$ .

## Realizacija

Turite pateikti vieną failą, pavadintą `city.c`, `city.cpp` arba `city.pas`. Šiame faile turi būti įrašytos paprogramės su tokiomis antraštėmis.

### Programuojantiems C/C++

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

### Programuojantiems Paskaliu

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

Ši paprogramė turi veikti taip, kaip aprašyta aukščiau. Be abejo, galite sukurti daugiau paprogramių vidiniam vartojimui. Jūsų pateiktas sprendimas neturi dirbti (skaityti ar rašyti) nei su standartiniu įvedimo/išvedimo įrenginiu, nei su jokia kitu failu.

### Pavyzdinis vertintojas

Pavyzdinis vertintojas duomenis perskaito tokiu formatu:

- 1-a eilutė:  $N$ ;
- 2-a, ...,  $(N + 1)$ -a eilutės:  $X[i] \ Y[i]$ .

## Laiko ir atminties ribojimai

- Laiko ribojimas: 1 sekundė.
- Atminties ribojimas: 256 MiB.