

# Ideālā pilsēta

Leonardo, tāpat kā daudziem citiem tā laika itāļu zinātniekiem un māksliniekiem, ļoti interesēja pilsētu plānošana. Viņa mērķis bija izplānot ideālo pilsētu: ērtu, plašu un ekonomisku resursu ziņā, pretēji šaurajām, saspiestajām viduslaiku pilsētām.

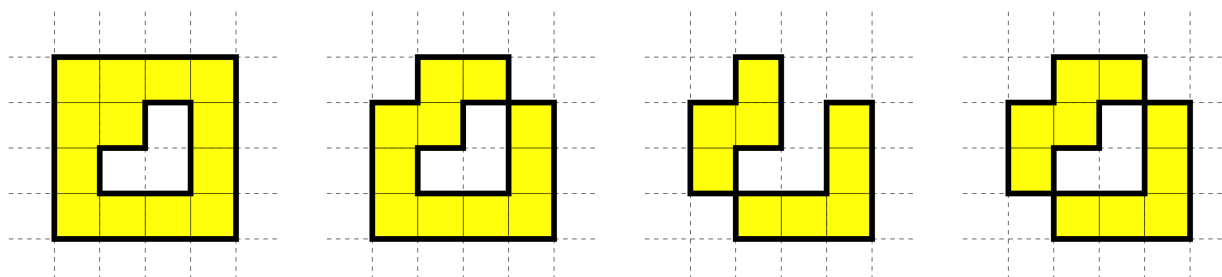
## Ideālā pilsēta

Pilsētā ir  $N$  kvartāli, kas ir novietoti bezgalīgā rūtiņu režģī. Katru režģa rūtiņu viennozīmīgi nosaka tās koordinātas (rinda, kolona). Dotai rūtiņai  $(i, j)$ , blakus esošās rūtiņas ir:  $(i - 1, j)$ ,  $(i + 1, j)$ ,  $(i, j - 1)$  un  $(i, j + 1)$ . Katrs kvartāls režģī aizņem tieši vienu rūtiņu. Kvartāla (rūtiņas)  $(i, j)$  koordinātas var būt robežās  $1 \leq i, j \leq 2^{31} - 2$ . Mēs lietosim rūtiņas koordinātas, lai identificētu kvartālus. Divi kvartāli atrodas blakus, ja tie atrodas blakus rūtiņās. Ideālajā pilsētā kvartāli ir savienoti tādā veidā, ka neveidojas “caurumi”, tas ir šūnām ir jābūt spēkā zemāk esošajiem nosacījumiem.

- Starp jebkurām divām *tukšām* rūtiņām ir vismaz viena tās savienojosa *tukšu* rūtiņu virkne.
- Starp jebkurām divām *netukšām* rūtiņām ir vismaz viena tās savienojosa *netukšu* rūtiņu virkne.

## 1. piemērs

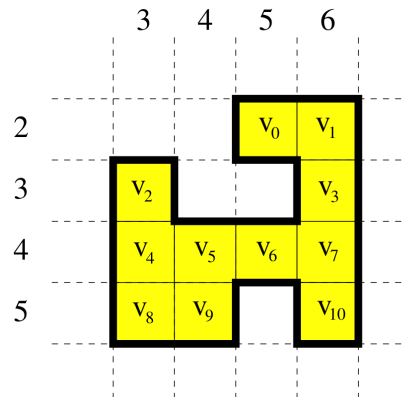
Neviens no zemāk redzamajiem kvartālu izvietojumiem neatbilst ideālai pilsētai: pirmie divi neatbilst pirmajam nosacījumam, trešais neatbilst otrajam nosacījumam un ceturtais neatbilst nevienam no abiem nosacījumiem.



## Attālums

Staigājot pa pilsētu, *solis* nozīmē pāriet no viena kvartāla uz kaimiņu kvartālu. Uz tukšām rūtiņām nevar pāriet. Pilsētas ar  $N$  kvartāliem koordinātas apzīmēsīm ar  $v_0, v_1, \dots, v_{N-1}$ . Jebkuriem diviem dažādiem kvartāliem ar koordinātām  $v_i$  un  $v_j$ , attālums starp tām  $d(v_i, v_j)$  ir mazākais soļu skaits, kas jāveic, lai no viena kvartāla nonāktu otrā.

Zemāk redzamajā zīmējumā redzama ideāla pilsēta ar  $N = 11$  kvartāliem koordinātās  $v_0 = (2, 5)$ ,  $v_1 = (2, 6)$ ,  $v_2 = (3, 3)$ ,  $v_3 = (3, 6)$ ,  $v_4 = (4, 3)$ ,  $v_5 = (4, 4)$ ,  $v_6 = (4, 5)$ ,  $v_7 = (4, 6)$ ,  $v_8 = (5, 3)$ ,  $v_9 = (5, 4)$ , and  $v_{10} = (5, 6)$ . Piemēram,  $d(v_1, v_3) = 1$ ,  $d(v_1, v_8) = 6$ ,  $d(v_6, v_{10}) = 2$ , un  $d(v_9, v_{10}) = 4$ .



## Uzdevums

Jūsu uzdevums ir uzrakstīt programmu, kas dotai ideālai pilsētai aprēķina attālumu summu starp visiem kvartāliem  $v_i$  un  $v_j$  kuriem  $i < j$ . Formāli, jūsu programmai ir jāaprēķina summa:

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

jums jāuzraksta funkcija `DistanceSum(N, X, Y)`, kas dotam  $N$  un diviem masīviem  $X$  un  $Y$ , kas apraksta pilsētu, aprēķina augstāk esošās formulas vērtību. Gan  $X$ , gan  $Y$  garums ir  $N$ ;  $i$ -tā kvartāla ( $0 \leq i \leq N - 1$ ) koordinātas ir  $(X[i], Y[i])$ , un  $1 \leq X[i], Y[i] \leq 2^{31} - 2$ . Tā kā rezultāts var būt lielāks kā skaitlis ko var attēlot 32 bitu mainīgajos, atbilde jāizvada pēc moduļa 1 000 000 000 (viens miljards).

Otrajā piemērā ir  $11 \times 10 / 2 = 55$  kvartālu pāru. Savstarpējo attālumu summa ir 174.

### 1. apakšuzdevums [11 punkti]

Jūs varat pieņemt, ka  $N \leq 200$ .

### 2. apakšuzdevums [21 punkti]

Jūs varat pieņemt, ka  $N \leq 2\,000$ .

### 3. apakšuzdevums [23 punkti]

Jūs varat pieņemt, ka  $N \leq 100\,000$ .

Papildus ir spēkā divi nosacījumi: jebkurām divām netukšām rūtīnām  $i$  un  $j$  tādām, ka  $X[i] = X[j]$ , visas rūtīņas, kas atrodas starp tām ir netukšas; jebkurām divām netukšām rūtīnām  $i$  un  $j$  tādām, ka  $Y[i] = Y[j]$ , visas rūtīņas, kas atrodas starp tām ir netukšas.

## 4. apakšuzdevums [45 punkti]

Jūs varat pieņemt, ka  $N \leq 100\,000$ .

### Implementācijas detaļas

Jums ir jāiesūta viens fails ar nosaukumu `city.c`, `city.cpp` vai `city.pas`. Šajā failā jābūt realizētai augstāk aprakstītajai apakšprogrammai ar šādu signatūru.

#### C/C++ programmas

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

#### Pascal programmas

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

Apakšprogrammai jādarbojas iepriekš aprakstītajā veidā. Protams, varat implementēt arī citas apakšprogrammas. Iesūtītās programmas nedrīkst neko rakstīt / lasīt no standarta izvada / ievada vai kāda cita faila.

#### Paraugvērtētājs

Piedāvātā paraugvērtētāja ievaddatiem jābūt šādā formā:

- 1. rinda:  $N$ ;
- Katrā no nākamajām  $N$  rindām:  $X[i]$ ,  $Y[i]$ .

### Izpildes laika un atmiņas ierobežojumi

- Izpildes laika ierobežojums: 1 sekunde.
- Atmiņas ierobežojums: 256 MiB.