

## Kota ideal

Leonardo, seperti banyak ilmuwan dan artis lain pada zamannya, sangat tertarik akan tata kota dan desain perkotaan. Dia ingin membuat model kota yang ideal: nyaman, luas, dan menggunakan sumber daya secara rasional, berbeda dari kota-kota pada abad pertengahan yang sempit dan sesak.

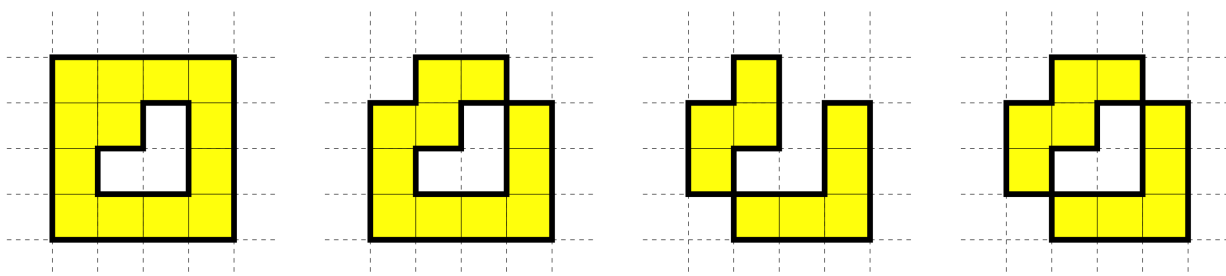
### Kota ideal tersebut

Kota tersebut terbentuk dari  $N$  balok yang diletakkan pada sebuah grid tak terbatas yang terdiri dari sel-sel. Setiap sel diidentifikasi dengan sepasang koordinat (baris, kolom). Untuk sebuah sel  $(i, j)$ , sel-sel yang bersebelahan antara lain:  $(i - 1, j)$ ,  $(i + 1, j)$ ,  $(i, j - 1)$ , dan  $(i, j + 1)$ . Setiap balok, ketika diletakkan pada grid, menutupi tepat salah satu sel. Sebuah balok dapat diletakkan pada sel  $(i, j)$  jika dan hanya jika  $1 \leq i, j \leq 2^{31} - 2$ . Kita akan menggunakan koordinat dari sel-sel juga untuk menunjukkan balok-balok yang ada di atas mereka. Dua buah balok dikatakan bersebelahan jika mereka diletakkan pada sel-sel bersebelahan. Pada sebuah kota ideal, semua balok-baloknya terhubung sedemikian sehingga tidak ada "lubang" di dalam batasnya, yaitu, sel-sel itu harus memenuhi kedua kondisi di bawah ini.

- Untuk setiap dua sel "kosong", ada setidaknya sebuah sekuens (rangkaian) sel-sel "kosong" bersebelahan yang menghubungkan mereka.
- Untuk setiap dua sel "tidak kosong", ada setidaknya sebuah sekuens sel-sel "tidak kosong" bersebelahan yang menghubungkan mereka.

### Contoh 1

Tidak ada di antara konfigurasi balok-balok di bawah ini yang merepresentasikan sebuah kota ideal: dua terkiri pertama tidak memenuhi kondisi pertama, yang ketiga tidak memenuhi kondisi kedua, dan yang keempat tidak memenuhi kondisi manapun.



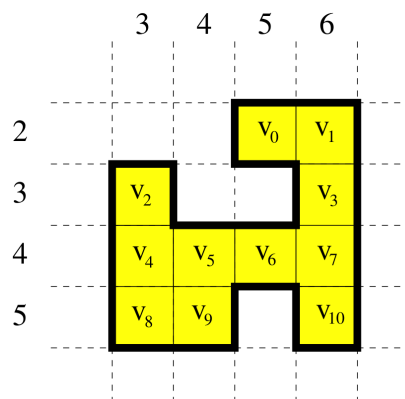
### Jarak

Ketika menjelajahi sebuah kota, sebuah *lompatan* berarti pergi dari satu blok ke blok lain yang

bersebelahan. Sel-sel kosong tidak dapat dijelajahi. Anggap  $v_0, v_1, \dots, v_{N-1}$  adalah koordinat dari  $N$  buah balok yang diletakkan pada grid. Untuk setiap dua balok berbeda pada koordinat  $v_i$  dan  $v_j$ , jarak mereka  $d(v_i, v_j)$  adalah banyak lompatan tersedikit untuk pergi dari salah satu dari kedua balok ini ke balok yang lainnya.

## Contoh 2

Konfigurasi di bawah ini merepresentasikan sebuah kota ideal yang terdiri dari  $N = 11$  balok pada koordinat-koordinat  $v_0 = (2, 5)$ ,  $v_1 = (2, 6)$ ,  $v_2 = (3, 3)$ ,  $v_3 = (3, 6)$ ,  $v_4 = (4, 3)$ ,  $v_5 = (4, 4)$ ,  $v_6 = (4, 5)$ ,  $v_7 = (4, 6)$ ,  $v_8 = (5, 3)$ ,  $v_9 = (5, 4)$ , dan  $v_{10} = (5, 6)$ . Sebagai contoh,  $d(v_1, v_3) = 1$ ,  $d(v_1, v_8) = 6$ ,  $d(v_6, v_{10}) = 2$ , dan  $d(v_9, v_{10}) = 4$ .



## Statement

Diberikan sebuah kota ideal, tugas Anda adalah menulis sebuah program untuk menghitung jumlah dari semua pasang jarak antara balok  $v_i$  dan  $v_j$  yang mana  $i < j$ . Secara formal, program Anda harus menghitung nilai dari jumlah di bawah ini:

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

Khususnya, Anda diharapkan untuk mengimplementasi sebuah routine `DistanceSum(N, X, Y)` yang menghitung rumus di atas setelah diberikan  $N$  dan dua buah array  $X$  dan  $Y$  yang mendeskripsikan kota tersebut. Kedua array  $X$  dan  $Y$  berukuran  $N$ ; balok  $i$  berada pada koordinat  $(X[i], Y[i])$  untuk  $0 \leq i \leq N - 1$ , dan  $1 \leq X[i], Y[i] \leq 2^{31} - 2$ . Karena hasilnya bisa terlalu besar ketika direpresentasikan menggunakan 32 bit, Anda harus memberikan hasil ini setelah dimodulo dengan 1 000 000 000 (satu milyar).

Pada Contoh 2, ada  $11 \times 10 / 2 = 55$  pasangan balok. Jumlah dari jarak semua pasang adalah 174.

## Subtask 1 [11 poin]

Anda boleh berasumsi bahwa  $N \leq 200$ .

## Subtask 2 [21 poin]

Anda boleh berasumsi bahwa  $N \leq 2\,000$ .

## Subtask 3 [23 poin]

Anda boleh berasumsi bahwa  $N \leq 100\,000$ .

Sebagai tambahan, kedua kondisi berikut ini terpenuhi: untuk dua sel tidak kosong  $i$  dan  $j$  manapun yang mana  $X[i] = X[j]$ , setiap sel di antara mereka pasti juga tidak kosong; untuk dua sel tidak kosong  $i$  dan  $j$  manapun yang mana  $Y[i] = Y[j]$ , setiap sel di antara mereka juga tidak kosong.

## Subtask 4 [45 poin]

Anda boleh berasumsi bahwa  $N \leq 100\,000$ .

## Detail implementasi

Anda harus mengumpulkan tepat sebuah file, yang diberi nama `city.c`, `city.cpp` atau `city.pas`. File ini harus berisi implementasi dari subprogram yang dideskripsikan di atas dengan signature berikut ini.

### Program C/C++

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

### Program Pascal

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

Subprogram ini harus berperilaku seperti yang dijelaskan di atas. Tentu saja, Anda boleh mengimplementasikan subprogram lain untuk kepentingan internalnya. Program yang Anda kumpulkan tidak boleh berinteraksi dengan cara apapun melalui standard input/output, maupun melalui file lain.

### Contoh grader

Contoh grader yang disediakan mengharapkan masukan dengan format berikut ini:

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

## Batas Waktu dan Memori

- Batas waktu: 1 detik.

- Batas memori: 256 MiB.