

Идеален град

Леонардо, како и многу други италијански научници и уметници од неговото време, бил екстремно заинтересиран за планирање на градови и урбанистички дизајн. Тој си поставил како цел да моделира еден идеален град: комфорен, простран и рационален во употребата на неговите ресурси, далеку од тесните, клаустрофобични градови на Средниот Век.

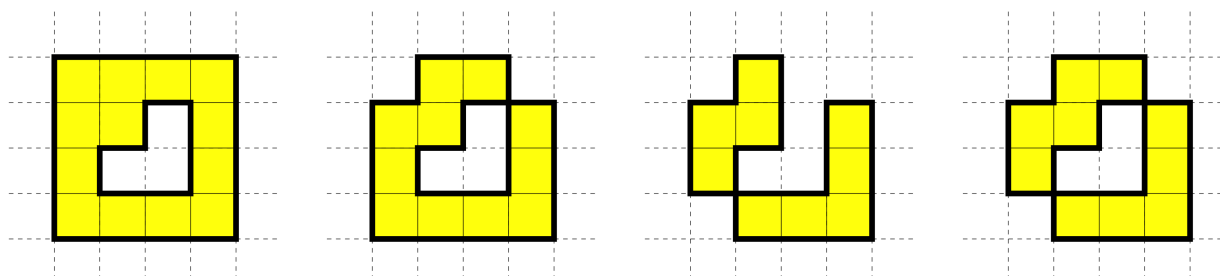
Идеалниот град

Градот е составен од N блокови поставени на бесконечна мрежа од клетки. Секоја клетка се идентификува со пар од координати (редица, колона). За дадена клетка (i, j) , нејзини соседни клетки се: $(i - 1, j)$, $(i + 1, j)$, $(i, j - 1)$ и $(i, j + 1)$. Секој блок, кога ќе се постави на мрежата, покрива точно една од клетките. Даден блок може да се постави врз клетката (i, j) ако и само ако $1 \leq i, j \leq 2^{31} - 2$. Координатите на клетките исто така ќе ги користиме и за идентификација на блоковите што се наоѓаат врз нив. Два блока се соседни ако се поставени врз соседни клетки. Во идеален град, сите блокови се поврзани на таков начин што не постојат “дупки” во рамки на неговите граници, т.е. клетките мора да ги задоволуваат двата услова наведени подолу.

- За кои било две *празни* клетки, постои барем една секвенца од соседни *празни* клетки кои што ги поврзуваат нив.
- За кои било две *непразни* клетки, постои барем една секвенца од соседни *непразни* клетки кои што ги поврзуваат нив.

Пример 1

Ниту една од конфигурациите на блокови дадени подолу не претставува идеален град: првите две (од лево) не го задоволуваат првиот услов, третата не го задоволува вториот услов, додека пак четвртата конфигурација не задоволува ниту еден од двата услова.

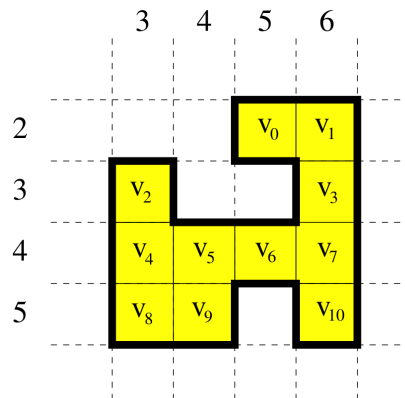


Растојание

При изминување на градот, *скок* означува одење (премин) од еден блок на некој негов соседен блок. Празните клетки не можат да се изминуваат. Нека v_0, v_1, \dots, v_{N-1} се координатите на N -те блокови поставени на мрежата. За кои било два различни блока со координати v_i и v_j , растојанието $d(v_i, v_j)$ помеѓу нив е најмалиот број на скокови кои што се потребни за да се премине од едниот на другиот.

Пример 2

Конфигурацијата дадена подолу претставува идеален град составен од $N = 11$ блокови со координати $v_0 = (2, 5)$, $v_1 = (2, 6)$, $v_2 = (3, 3)$, $v_3 = (3, 6)$, $v_4 = (4, 3)$, $v_5 = (4, 4)$, $v_6 = (4, 5)$, $v_7 = (4, 6)$, $v_8 = (5, 3)$, $v_9 = (5, 4)$ и $v_{10} = (5, 6)$. На пример, $d(v_1, v_3) = 1$, $d(v_1, v_8) = 6$, $d(v_6, v_{10}) = 2$ и $d(v_9, v_{10}) = 4$.



Задача

Ваша задача е, за даден идеален град, да напишете програма која ќе го пресметува збирот на сите растојанија по парови т.е. сите растојанија помеѓу оние блокови v_i и v_j за кои што $i < j$. Формално, вашата програма треба да ја пресметува вредноста на следниот збир:

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

Конкретно, треба да имплементирате рутина (функција) $\text{DistanceSum}(N, X, Y)$ која што, за дадено N и за дадени две низи X и Y кои што го опишуваат градот, ја пресметува бараната вредност според погорната формула. Двете низи X и Y се со големина N ; блокот i има координати $(X[i], Y[i])$ за $0 \leq i \leq N - 1$, и $1 \leq X[i], Y[i] \leq 2^{31} - 2$. Бидејќи резултатот може да биде преголем за да се претстави со користење на 32 бита, истиот треба да се врати по модул 1 000 000 000 (еден билион).

Во Пример 2, постојат $11 \times 10 / 2 = 55$ парови од блокови. Збирот на сите растојанија по парови е 174.

Подзадача 1 [11 поени]

Може да претпоставите дека $N \leq 200$.

Подзадача 2 [21 поен]

Може да претпоставите дека $N \leq 2\,000$.

Подзадача 3 [23 поени]

Може да претпоставите дека $N \leq 100\,000$.

Дополнително, важат и следниве два услова: за кои било две непразни клетки i и j такви што $X[i] = X[j]$, важи дека секоја клетка помеѓу нив е исто така непразна; за кои било две непразни клетки i и j такви што $Y[i] = Y[j]$, важи дека секоја клетка помеѓу нив е исто така непразна.

Подзадача 4 [45 поени]

Може да претпоставите дека $N \leq 100\,000$.

Имплементациски детали

Треба да предадете точно еден документ, со име `city.c`, `city.cpp` или `city.pas`. Документот треба да содржи имплементација на потпрограмата опишана погоре користејќи ги следниве потписи.

C/C++ програми

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

Pascal програми

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

Оваа потпрограма треба да се однесува како што беше опишано погоре. Се разбира, дозволено е да имплементирате и други потпрограми за нејзина внатрешна употреба. Вашите предадени задачи не смеат да имаат никаков вид на интеракција ниту со стандардниот влез/излез, ниту пак со кој било друг документ.

Пример-оценувач

Пример-оценувачот кој што е обезбеден за задачата ќе очекува влез во следниов формат:

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

Временско и Мемориско ограничување

- Временско ограничување: 1 секунда.
- Мемориско ограничување: 256 MiB.