

Идеальный город

Леонардо, как и многие другие итальянские ученые и художники его времени, очень интересовался градостроением. Он решил спроектировать идеальный город: комфортабельный, просторный и рациональный в плане использования ресурсов, не такой как тесные и зажатые города средневековья.

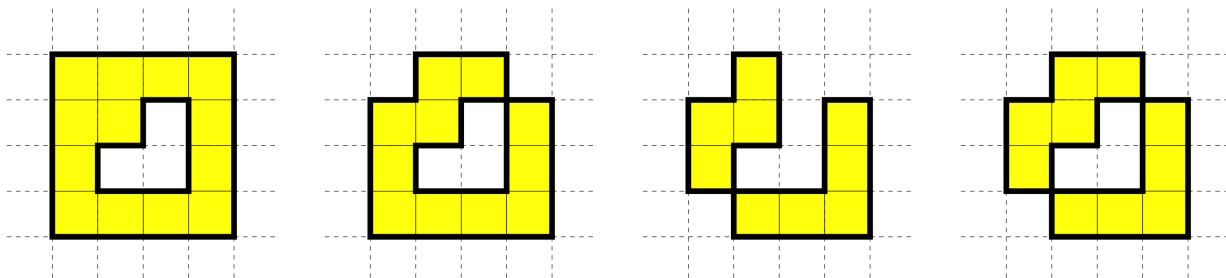
Идеальный город

Город состоит из N кварталов, расположенных на бесконечной квадратной сетке из клеток. Каждая клетка определяется парой координат (строка, столбец). Для клетки (i, j) соседними являются клетки $(i - 1, j)$, $(i + 1, j)$, $(i, j - 1)$ и $(i, j + 1)$. Каждый квартал занимает на сетке в точности одну клетку. Квартал может занимать клетку (i, j) тогда и только тогда, когда $1 \leq i, j \leq 2^{31} - 2$. Для обозначения кварталов мы будем использовать координаты соответствующих клеток. Два квартала называются соседними, если они занимают соседние клетки. В идеальном городе все кварталы соединены так, что нет "дырок" внутри его границы, то есть, клетки должны удовлетворять следующим условиям:

- Для любых двух *пустых* клеток существует не менее одной последовательности соседних между собой *пустых* клеток, соединяющих эти две клетки.
- Для любых двух *непустых* клеток, существует не менее одной последовательности соседних между собой *непустых* клеток, соединяющих эти две клетки.

Пример 1

Ни одна из конфигураций кварталов, приведенных ниже, не является идеальным городом: первые две слева не удовлетворяют первому условию, третья не удовлетворяет второму условию, и четвертая не удовлетворяет обоим условиям.

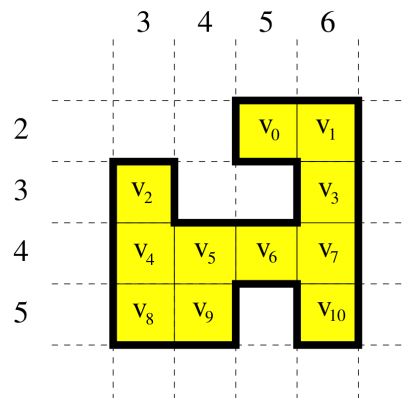


Расстояние

При движении по городу, слово *прыжок* обозначает переход от квартала к соседнему. По пустым клеткам нельзя передвигаться. Пусть v_0, v_1, \dots, v_{N-1} - это координаты кварталов, расположенных на сетке. Для любых двух различных кварталов с координатами v_i и v_j расстояние $d(v_i, v_j)$ - это минимальное количество прыжков, которое необходимо совершить для перехода от одного квартала к другому.

Пример 2

Конфигурация ниже представляет собой идеальный город, состоящий из $N = 11$ кварталов с координатами $v_0 = (2, 5)$, $v_1 = (2, 6)$, $v_2 = (3, 3)$, $v_3 = (3, 6)$, $v_4 = (4, 3)$, $v_5 = (4, 4)$, $v_6 = (4, 5)$, $v_7 = (4, 6)$, $v_8 = (5, 3)$, $v_9 = (5, 4)$ и $v_{10} = (5, 6)$. Например $d(v_1, v_3) = 1$, $d(v_1, v_8) = 6$, $d(v_6, v_{10}) = 2$ и $d(v_9, v_{10}) = 4$.



Постановка задачи

Вы должны написать программу, вычисляющую сумму всех попарных расстояний между кварталами идеального города v_i и v_j для каждого $i < j$. То есть, ваша программа должна вычислять значение следующей суммы:

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

А именно, вы должны реализовать процедуру `DistanceSum(N, X, Y)` которая, по заданным N и двум массивам X и Y , которые описывают город, вычисляет значение по формуле выше. Массивы как X так и Y содержат N элементов; Квартал i имеет координаты $(X[i], Y[i])$ для $0 \leq i \leq N - 1$ и $1 \leq X[i], Y[i] \leq 2^{31} - 2$. Поскольку результат может быть слишком большим, чтобы вместиться в 32 бита, вы должны вывести это число по модулю 1 000 000 000 (один миллиард).

В примере 2, имеется $11 \times 10 / 2 = 55$ пар кварталов. Сумма всех попарных расстояний равна 174.

Подзадача 1 [11 баллов]

В этой подзадаче $N \leq 200$.

Подзадача 2 [21 балл]

В этой подзадаче $N \leq 2\,000$.

Подзадача 3 [23 балла]

В этой подзадаче $N \leq 100\,000$.

Более того, в этой подзадаче есть еще два следующих условия: для любых двух непустых клеток i и j таких, что $X[i] = X[j]$, каждая клетка между ними также является непустой; для любых двух непустых клеток i и j таких, что $Y[i] = Y[j]$, каждая клетка между ними также является непустой.

Подзадача 4 [45 баллов]

В этой подзадаче $N \leq 100\,000$.

Детали реализации

Вы должны отправить на проверку один файл с названием `city.c`, `city.cpp` или `city.pas`. Этот файл должен реализовывать процедуру, описанную выше, используя следующее описание (сигнатуру):

Реализация на C/C++

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

Реализация на Pascal

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

Эти процедуры должны вести себя как описано выше. Конечно, вы можете реализовывать любые другие процедуры для внутреннего использования. Отправляемое вами решение не должно никаким образом взаимодействовать со стандартным потоком ввода/вывода или любым другим файлом.

Пример проверяющего модуля (grader)

Предоставляемый пример проверяющего модуля (grader) использует следующий формат ввода:

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

Ограничения по времени и по памяти

- Ограничение по времени: 1 секунда.
- Ограничение по памяти: 256 мебибайт.