

## 理想の都市 (Ideal city)

レオナルド (Leonardo) は、イタリアの同世代の他の科学者や芸術家と同様に、都市の計画や設計にとっても興味を持っていた。彼の目的は理想の都市 — 快適で、見かけがよく、資源を合理的に使用し、狭く窮屈な中世の都市からかけ離れた都市 — を作ることであった。

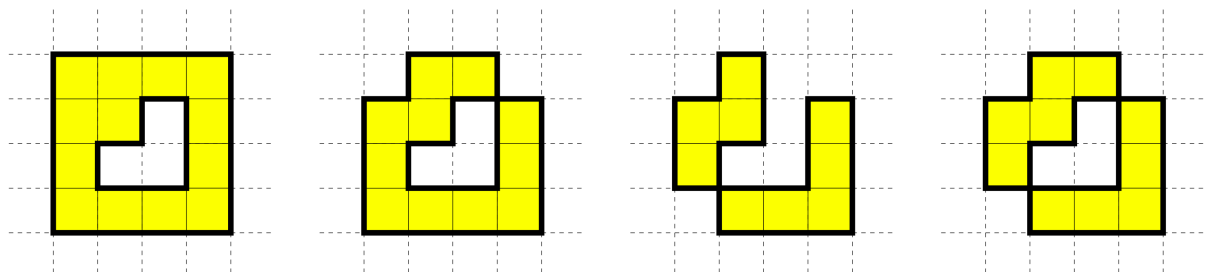
### 理想の都市 (The ideal city)

都市は、無限に広がるマス目に置かれた  $N$  個のブロックからなる。各マスはその座標 (行, 列) で表される。マス  $(i, j)$  に対し、それと隣接するマスは  $(i - 1, j)$ ,  $(i + 1, j)$ ,  $(i, j - 1)$ ,  $(i, j + 1)$  である。マス目に置かれた各ブロックはちょうど 1 つのマス进行を覆う。ブロックは  $1 \leq i, j \leq 2^{31} - 2$  を満たすマス  $(i, j)$  にのみ置かれうる。以下、マスの座標をそのマスに置かれたブロックを指すのにも使う。2 つのブロックが隣接しているとは、それらが置かれているマスが隣接していることをいう。理想の都市において、すべてのブロックはその境界の内部に「穴」がないようにしてつながっている：すなわち、すべてのマスは以下の 2 条件を満たす。

- どの 2 つの空のマスについても、その一方のマスからはじめて、隣接する空のマスに移動する操作を何回か行うことでもう一方のマスに到達することができる。
- どの 2 つの空でないマスについても、その一方のマスからはじめて、隣接する空でないマスに移動する操作を何回か行うことでもう一方のマスに到達することができる。

### 例 1 (Example 1)

下図に示されたブロックの配置はいずれも理想の都市でない：左から 1 つ目と 2 つ目の配置は 1 つ目の条件を満たさない。左から 3 つ目の配置は 2 つ目の条件を満たさない。左から 4 つ目の配置はいずれの条件も満たさない。



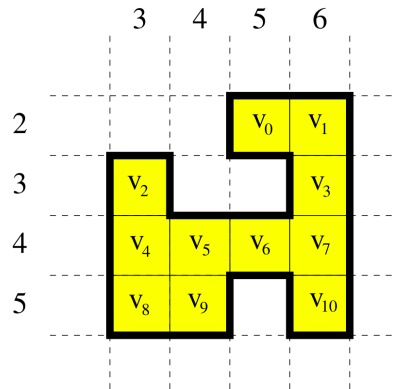
### 距離 (Distance)

都市の中を移動するために、あるブロックからそれと隣接したブロックに移動することをホップ (hop) と呼ぶ。空のマスを通ることは出来ない。  $v_0, v_1, \dots, v_{N-1}$  をマス目に置かれた  $N$  個のブロックの座標とする。相異なる座標  $v_i, v_j$  にある 2 つのブロックに対し、それらの距離  $d(v_i, v_j)$  とはその一方のブロックからもう一方のブロックへ移動するために必要なホップの回

数の最小値のこととする。

## 例 2 (Example 2)

下図の配置は  $N = 11$  個のブロックからなる理想の都市を表す。ブロックの座標は  $v_0 = (2, 5)$ ,  $v_1 = (2, 6)$ ,  $v_2 = (3, 3)$ ,  $v_3 = (3, 6)$ ,  $v_4 = (4, 3)$ ,  $v_5 = (4, 4)$ ,  $v_6 = (4, 5)$ ,  $v_7 = (4, 6)$ ,  $v_8 = (5, 3)$ ,  $v_9 = (5, 4)$ ,  $v_{10} = (5, 6)$  である。このとき、例えば  $d(v_1, v_3) = 1$ ,  $d(v_1, v_8) = 6$ ,  $d(v_6, v_{10}) = 2$ ,  $d(v_9, v_{10}) = 4$  である。



## 問題文 (Statement)

あなたは、理想の都市が与えられたとき、ブロック  $v_i$  と  $v_j$  の距離をすべての組  $v_i, v_j$  ( $i < j$ ) について総和をとったものを計算するプログラムを書かなければならない。つまり、あなたのプログラムは次の式の値を計算しなければならない：

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

具体的には、あなたは  $N$  と都市の情報を示す 2 つの配列  $X, Y$  が与えられたとき上の式を計算するルーチン  $\text{DistanceSum}(N, X, Y)$  を実装しなければならない。配列  $X, Y$  のサイズはともに  $N$  である。各  $i$  ( $0 \leq i \leq N - 1$ ) について  $1 \leq X[i], Y[i] \leq 2^{31} - 2$  を満たし、ブロック  $i$  の座標が  $(X[i], Y[i])$  であることを表す。結果は 32 ビット整数に収まらないほど大きくなる可能性があるため、あなたは結果を 1 000 000 000 (10 億) で割った余りを答えなければならない。

例 2 において、ブロックのペアは  $11 \times 10 / 2 = 55$  個ある。すべての組についての距離の総和は 174 である。

## 小課題 1 [11 点]

$N \leq 200$  と仮定してよい。

## 小課題 2 [21 点]

$N \leq 2\,000$  と仮定してよい。

## 小課題 3 [23 点]

$N \leq 100\,000$  と仮定してよい.

さらに, 次の 2 つの条件が成り立つ:

- $X[i] = X[j]$  を満たす空でない 2 つのマス  $i, j$  に対し, それらの間にあるマスはすべて空でない.
- $Y[i] = Y[j]$  を満たす空でない 2 つのマス  $i, j$  に対し, それらの間にあるマスはすべて空でない.

## 小課題 4 [45 点]

$N \leq 100\,000$  と仮定してよい.

### 実装の詳細 (Implementation details)

あなたは `city.c`, `city.cpp` または `city.pas` という名前のファイルをちょうど 1 つ提出しなければならない. そのファイルは上で説明されたサブプログラムを以下のシグネチャを用いて実装しなければならない.

#### C/C++ プログラム

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

#### Pascal プログラム

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

このサブプログラムは上記の通りに動作しなければならない. もちろん, 内部での使用のために他のサブプログラムを実装することは自由である. あなたの提出は標準入力・標準出力, あるいは他のファイルといかなる方法でもやりとりしてはならない.

#### 採点プログラムのサンプル (Sample grader)

課題環境に与えられる採点プログラムのサンプルは以下の書式の入力を読み込む:

- 1 行目:  $N$ .
- 2 行目から  $N + 1$  行目まで:  $X[i], Y[i]$ .

### 時間とメモリの制限 (Time and Memory limits)

- 時間制限: 1 秒.
- メモリ制限: 256 MiB.