

理想城市

像许多同龄的意大利科学家和艺术家一样，Leonardo 对城市规划和城区设计很感兴趣。

他致力于构建一个理想城：舒适，宽敞以及合理利用资源，以此远离以往狭窄局促的中世纪城市。

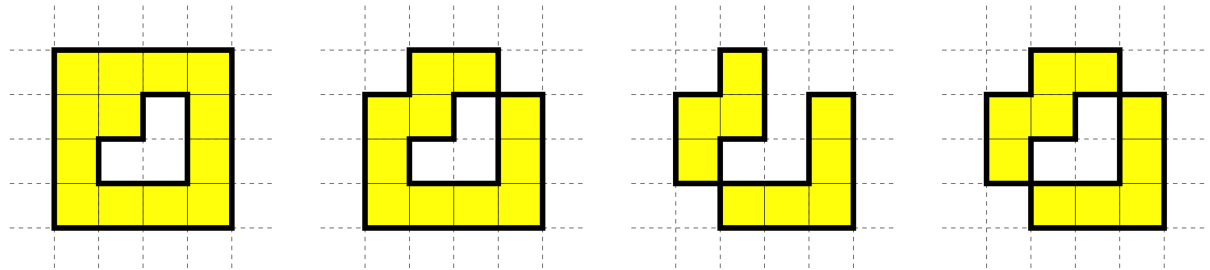
理想城

理想城由放置在一个无限的网格单元 (grid of cells) 中的 N 个方块 (blocks) 组成。每个单元格由一对坐标来标识 (行, 列)。单元格 $(0, 0)$ 位于网格的左上角。给定一个单元格 (i, j) ，与之相邻的单元格 (如果存在的话) 分别为： $(i-1, j)$ ， $(i+1, j)$ ， $(i, j-1)$ ，和 $(i, j+1)$ 。每个方块 (当放置在网格上时) 刚好覆盖一个单元格。一个方块能够被放置在单元格 (i, j) 上，当且仅当 $1 \leq i, j \leq 2^{31}-2$ 。我们将使用单元格的坐标同时来代表单元格上面的方块。两个方块是相邻的如果它们被放在相邻的单元格中。理想城中的所有方块均按照下述方式来连接：在边界内部没有“洞”，换句话说，所有单元格必须满足下述两个条件：

- 对于任意两个空白单元格，至少存在一个相邻的空白单元格的序列使它们连通。
- 对于任意两个非空的单元格，至少存在一个相邻的非空单元格的序列使它们连通。

例 1

以下 4 个图中的方块放置均不满足理想城的条件。前两个图不满足第一个条件。第 3 个图不满足第二个条件，第 4 个图两个条件均不满足。

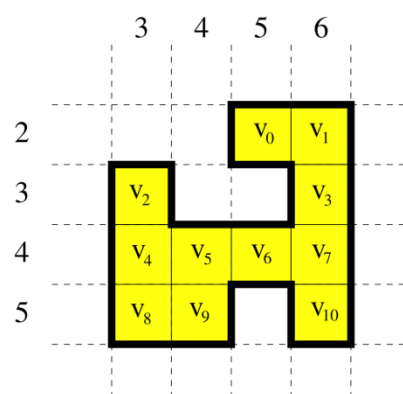


距离

当遍历理想城时，一个跳步 (*hop*) 代表从一个方块走到一个相邻的方块。空白单元格不能被遍历。令 v_0, v_1, \dots, v_{N-1} 是放置在单元格上的 N 个方块的坐标。对于任意两个不同的方块 (坐标为 v_i 和 v_j)，它们的距离 $d(v_i, v_j)$ 是从二个方块之一走到另一个方块所需的最小跳步 (*hops*) 数目。

例 2

下图的方块放置表示了一个 $N = 11$ 的理想城。其坐标分别为 $v_0 = (2, 5)$, $v_1 = (2, 6)$, $v_2 = (3, 3)$, $v_3 = (3, 6)$, $v_4 = (4, 3)$, $v_5 = (4, 4)$, $v_6 = (4, 5)$, $v_7 = (4, 6)$, $v_8 = (5, 3)$, $v_9 = (5, 4)$, $v_{10} = (5, 6)$ 。其中, $d(v_1, v_3) = 1$, $d(v_1, v_8) = 6$, $d(v_6, v_{10}) = 2$, and $d(v_9, v_{10}) = 4$ 。



说明

你的任务是，给定一个理想城，编写一个程序，对于任意 $i < j$ ，计算方块 v_i 和 v_j 之间的所有两两之间 (pairwise) 的距离和。亦可形式化的表示为，你的程序应当求出下列和式的值。

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

特别地，你需要实现一个子例程 $\text{DistanceSum}(N, X, Y)$ ，对于给定的 N 和描述理想城的数组 X 和 Y ，计算上述公式。 X 和 Y 的大小均为 N ；方块 i 位于坐标 $(X[i], Y[i])$ ，对于 $0 \leq i \leq N-1$ ，和 $1 \leq X[i], Y[i] \leq 2^{31}-2$ 。由于使用 32 位表示结果可能太大，你应当将结果按 1 000 000 000 (10 亿) 取模输出。

在例 2 中，共有 $11 \times 10 / 2 = 55$ 对方块。所有 pairwise 距离的和是 174.

子任务 1 [11 分]

你可以假设 $N \leq 200$.

子任务 2 [21 分]

你可以假设 $N \leq 2\,000$.

子任务 3 [23 分]

你可以假设 $N \leq 100\,000$.

此外，下列两个条件成立：对于任意两个非空的单元格 i 和 j ，使得 $X[i]=X[j]$ ，它们二者之间的每一个单元格也是非空的；对于任意两个非空的单元格 i 和 j ，使得 $Y[i]=Y[j]$ ，它们二者之间的每一个单元格也是非空的。

子任务 4 [45 分]

你可以假设 $N \leq 100\,000$.

实现细节

你必须只提交一个名为 `city.c`, `city.cpp` 或 `city.pas` 的文件。该文件必须实现上面描述的子程序并使用下列内容。

C/C++ programs

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

Pascal programs

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

这些子程序必须具有上面描述的功能。当然你可以根据需要实现其他的子程序供内部使用。你的提交程序不能以任意方式与标准输入/输出交互，也不能与其他文件交互。

Sample grader

The sample grader provided with the task environment will expect input in the following format:

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

Time and Memory limits

- Time limit: 1 second.
- Memory limit: 256 MiB.