Thành phố lý tưởng

Giống như nhiều nhà khoa học và nghệ sĩ người Ý cùng lứa tuổi, Leonardo rất quan tâm đến quy hoạch và thiết kế thành phố. Ông tập trung xây dựng thành phố lý tưởng: thuận tiện, rộng rãi và hợp lý trong việc sử dụng các nguồn tài nguyên, tránh xa sự chật hẹp và tù túng của các thành phố thời Trung Cổ.

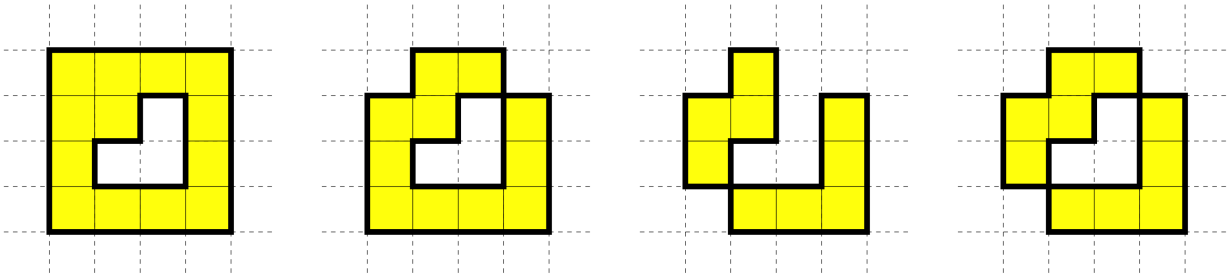
Thành phố lý tưởng

Thành phố được xây dựng bởi N khối được đặt trên một lưới ô vuông vô hạn. Mỗi ô vuông được xác định bởi cặp tọa độ (dòng, cột). Đối với ô (i, j) cho trước, các ô liền kề là: (i - 1, j), (i + 1, j), (i, j - 1) và (i, j + 1). Mỗi khối khi được đặt trên lưới sẽ phủ kín đúng một ô. Một khối có thể được đặt vào ô (i,j) khi và chỉ khi $1 \leq i, j \leq 2^{31} - 2$. Chúng ta cũng sử dụng các tọa độ của các ô để chỉ các khối nằm trên các ô đó. Hai khối là liền kề nếu chúng được đặt tại hai ô liền kề. Trong thành phố lý tưởng tất cả các khối được liên kết với nhau sao cho không có "lỗ hổng" trong đường biên, nghĩa là các khối phải thỏa mãn hai điều kiện dưới đây.

- Với hai ô *trống* bất kỳ, tồn tại ít nhất một dãy các ô *trống* liền kề kết nối chúng.
- Với hai ô *không trống* bất kỳ, tồn tại ít nhất một dãy các ô *không trống* liền kề kết nối chúng.

Ví dụ 1

Không có cấu hình khối nào dưới đây là biểu diễn của thành phố lý tưởng: Hai cấu hình đầu bên trái không thỏa mãn điều kiện thứ nhất, cấu hình thứ ba không thỏa mãn điều kiện thứ hai, còn cấu hình thứ tư không thỏa mãn cả hai điều kiện.

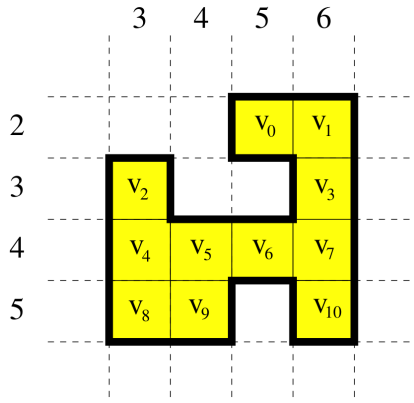


Khoảng cách

Khi di chuyển trong thành phố, một *bước nhảy* chỉ việc di chuyển từ một khối sang khối liền kề. Ô trống không thể di chuyển qua. Gọi $v_0, v_1, \ldots, v_{N-1}$ là tọa độ của N khối đặt trên lưới. Với hai khối khác nhau có tọa độ v_i và v_j, khoảng cách giữa chúng $d(v_i, v_j)$ là số bước nhảy ít nhất cần thực hiện để di chuyển từ một trong hai khối đến khối kia.

Ví dụ 2

Cấu hình dưới đây là một thành phố lý tưởng được tạo bởi N=11 khối đặt tại các tọa độ $v_0 = (2, 5)$, $v_1 = (2, 6)$, $v_2 = (3, 3)$, $v_3 = (3, 6)$, $v_4 = (4, 3)$, $v_5 = (4, 4)$, $v_6 = (4, 5)$, $v_7 = (4, 6)$, $v_8 = (5, 3)$, $v_9 = (5, 4)$, và $v_{10} = (5, 6)$. Chẳng hạn, $d(v_1, v_3) = 1$, $d(v_1, v_8) = 6$, $d(v_6, v_{10}) = 2$,và $d(v_9, v_{10}) = 4$.



Phát biểu bài toán

Nhiệm vụ của bạn là, cho trước thành phố lý tưởng, viết chương trình tính tổng các khoảng cách giữa tất cả các cặp khối v_i và v_j với $i < j$. Một cách hình thức, chương trình cần tính giá trị của tổng sau:

$$\sum d(v_i, v_j), \text{ trong đó } 0 \le i < j \le N - 1$$

Cụ thể, bạn phải cài đặt chương trình con `DistanceSum`(N, X, Y) tính công thức trên đối với N và hai mảng X, Y cho trước để mô tả thành phố. Cả hai mảng X và Y đều có kích thước N; khối thứ i có tọa độ (X[i], Y[i]) với $0 \le i \le N - 1$ và $1 \le X[i], Y[i] \le 2^{31} - 2$. Do kết quả có thể quá lớn đối với biểu diễn sử dụng 32 bit, bạn phải đưa ra giá trị của nó theo mô đun 1 000 000 000 (một tỉ).

Trong ví dụ 2, có $11 \times 10 / 2 = 55$ cặp khối. Tổng khoảng cách tất cả các cặp khối là 174.

Subtask 1 [11 điểm]

Bạn được giả thiết rằng $N \le 200$.

Subtask 2 [21 điểm]

Bạn được giả thiết rằng $N \le 2\,000$.

Subtask 3 [23 điểm]

Bạn được giả thiết rằng $N \le 100\,000$.

Thêm vào đó, giả thiết rằng dữ liệu thỏa mãn hai điều kiện sau: với hai ô không trống i và j mà X[i] = X[j] thì tất cả các ô giữa chúng đều không trống; với hai ô không trống i và j mà Y[i] = Y[j] thì tất cả các ô giữa chúng cũng đều không trống.

Subtask 4 [45 điểm]

Bạn được giả thiết rằng $N \leq 100\,000$.

Chi tiết cài đặt

Bạn phải nộp duy nhất một file có tên là `city.c`, `city.cpp` hoặc `city.pas`. File này phải cài đặt chương trình con mô tả ở trên sử dụng các mẫu sau đây.

C/C++ programs

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

Pascal programs

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

Chương trình con này phải hoạt động như mô tả ở trên. Tất nhiên bạn có thể cài đặt các chương trình con khác để sử dụng nội bộ. Các lần giao nộp không được giao tiếp dưới bất kỳ hình thức nào với input/output chuẩn hoặc với bất kỳ file nào khác.

Sample grader

Chương trình grader được cung cấp cùng với môi trường tác nghiệp sẽ nhận đầu vào theo khuôn dạng sau:

- dòng 1: N;
- dòng 2, …, N + 1: X[i], Y[i].

Hạn chế thời gian và bộ nhớ

- Hạn chế thời gian: 1 giây.
- Hạn chế bộ nhớ: 256 MiB.