



Broken Line

Azerbaijan nổi tiếng về thảm. Là một nhà thiết kế thảm bậc thầy, bạn muốn tạo ra một thiết kế mới bằng cách vẽ một đường **gấp khúc**. Một đường gấp khúc là một dãy gồm t đoạn trên mặt phẳng hai chiều được xác định bởi một dãy $t + 1$ điểm p_0, \dots, p_t như sau. Với mỗi $0 \leq j \leq t - 1$ có một đoạn thẳng nối điểm p_j với điểm p_{j+1} .

Để thực hiện thiết kế mới, bạn đã đánh dấu n **dấu chấm** trên mặt phẳng hai chiều. Tọa độ của dấu chấm i ($1 \leq i \leq n$) là $(x[i], y[i])$. **Không có hai dấu chấm nào có cùng tọa độ x hoặc cùng tọa độ y.**

Bây giờ bạn muốn tìm một dãy các điểm $(sx[0], sy[0]), (sx[1], sy[1]), \dots, (sx[k], sy[k])$ xác định một đường gấp khúc mà

- bắt đầu từ $(0, 0)$ (nghĩa là $sx[0] = 0$ và $sy[0] = 0$),
- chứa tất cả các dấu chấm (không nhất thiết phải tại các điểm đầu mút của các đoạn) và
- chỉ bao gồm các đoạn ngang hoặc dọc (hai điểm liên tiếp xác định đường gấp khúc có tọa độ x hoặc y bằng nhau).

Đường gấp khúc được phép tự cắt hoặc chồng lại theo bất kỳ cách nào. Cụ thể, mỗi điểm trên mặt phẳng có thể thuộc vào một số đoạn của đường gấp khúc.

Đây là bài toán chỉ cần kết quả đầu ra với cách tính điểm một phần. Bạn được cho 10 tệp đầu vào xác định vị trí của các dấu chấm. Đối với mỗi tệp đầu vào, bạn cần nộp một tệp đầu ra mô tả một đường gấp khúc thỏa mãn các yêu cầu. Đối với mỗi tệp đầu ra mô tả một đường gấp khúc hợp lệ, điểm của bạn phụ thuộc vào **số đoạn** trong đường gấp khúc (xem cách tính điểm ở dưới).

Bạn không cần phải nộp bất kỳ mã nguồn nào cho bài toán này.

Khuôn dạng dữ liệu vào

Mỗi tệp dữ liệu vào theo khuôn dạng sau:

- dòng 1: n
- dòng $1 + i$ (với $1 \leq i \leq n$): $x[i] \ y[i]$

Khuôn dạng kết quả ra

Mỗi tệp kết quả ra phải theo khuôn dạng sau:

- dòng 1: k
- dòng $1 + j$ (với $1 \leq j \leq k$): $sx[j] \ sy[j]$

Chú ý rằng dòng thứ hai phải chứa $sx[1]$ và $sy[1]$ (nghĩa là kết quả ra **không** chứa $sx[0]$ và $sy[0]$). Mỗi $sx[j]$ và $sy[j]$ phải là số nguyên.

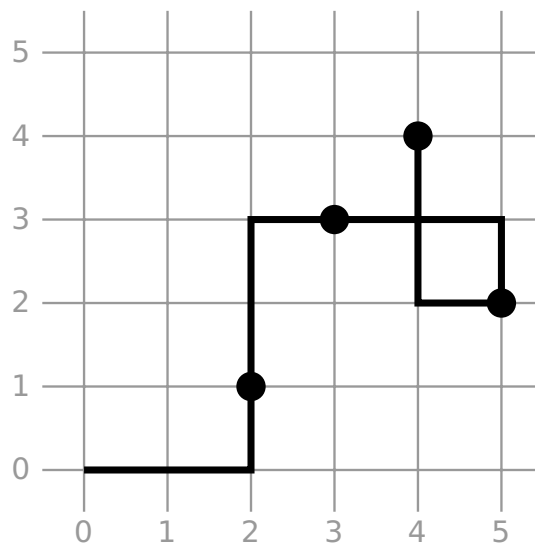
Ví dụ

Với đầu vào:

```
4
2 1
3 3
4 4
5 2
```

một đầu ra hợp lệ có thể là:

```
6
2 0
2 3
5 3
5 2
4 2
4 4
```



Xin lưu ý ví dụ này không có trong dữ liệu đầu vào thực tế của bài toán này.

Các ràng buộc

- $1 \leq n \leq 100\,000$
- $1 \leq x[i], y[i] \leq 10^9$
- Tất cả các giá trị của $x[i]$ và $y[i]$ đều nguyên.
- Không có hai dấu chấm nào có cùng tọa độ x hoặc cùng tọa độ y , nghĩa là $x[i_1] \neq x[i_2]$ và $y[i_1] \neq y[i_2]$ với $i_1 \neq i_2$.
- $-2 \cdot 10^9 \leq sx[j], sy[j] \leq 2 \cdot 10^9$
- Kích thước của mỗi tệp được nộp (có thể là kết quả đầu ra hoặc tệp nén) không được vượt quá 15MB.

Cách tính điểm

Đối với mỗi trường hợp kiểm thử, bạn có thể nhận tới 10 điểm. Bạn nhận 0 điểm nếu kết quả đầu ra của bạn không xác định một đường gấp khúc thỏa mãn các yêu cầu. Ngược lại, điểm số sẽ được xác định bằng cách sử dụng dãy giảm c_1, \dots, c_{10} , thay đổi theo trường hợp kiểm thử.

Giả sử lời giải của bạn là một đường gấp khúc hợp lệ bao gồm k đoạn. Khi đó, bạn sẽ nhận được

- i điểm, nếu $k = c_i$ (với $1 \leq i \leq 10$),
- $i + \frac{c_i - k}{c_i - c_{i+1}}$ điểm, nếu $c_{i+1} < k < c_i$ (với $1 \leq i \leq 9$),
- 0 điểm, nếu $k > c_1$,
- 10 điểm, nếu $k < c_{10}$.

Dãy c_1, \dots, c_{10} với mỗi trường hợp kiểm thử được cho dưới đây.

Testcases	01	02	03	04	05	06	07-10
n	20	600	5 000	50 000	72 018	91 891	100 000
c_1	50	1 200	10 000	100 000	144 036	183 782	200 000
c_2	45	937	7 607	75 336	108 430	138 292	150 475
c_3	40	674	5 213	50 671	72 824	92 801	100 949
c_4	37	651	5 125	50 359	72 446	92 371	100 500
c_5	35	640	5 081	50 203	72 257	92 156	100 275
c_6	33	628	5 037	50 047	72 067	91 941	100 050
c_7	28	616	5 020	50 025	72 044	91 918	100 027
c_8	26	610	5 012	50 014	72 033	91 906	100 015
c_9	25	607	5 008	50 009	72 027	91 900	100 009
c_{10}	23	603	5 003	50 003	72 021	91 894	100 003

Trình hiển thị

Trong các tệp đính kèm của bài toán này, có một kịch bản cho phép bạn trực quan hóa các tệp đầu vào và đầu ra.

Để trực quan hóa một tập tin đầu vào, hãy sử dụng lệnh sau:

```
python vis.py [input file]
```

Bạn cũng có thể trực quan lời giải của mình cho một dữ liệu đầu vào nào đó, sử dụng câu lệnh sau. Do các giới hạn kỹ thuật, trình hiển thị được cung cấp chỉ hiển thị 1000 **đoạn đầu tiên** của tệp kết quả đầu ra.

```
python vis.py [input file] --solution [output file]
```

Ví dụ:

```
python vis.py examples/00.in --solution examples/00.out
```