



## Начупена линия

Азербайджан е известен с килимите си. Като майстор на килими вие искате да направите нов дизайн чрез начертаване на **начупена линия**. Начупената линия е редица от  $t$  праволинейни отсечки в двумерната равнина, която се дефинира чрез редица от  $t + 1$  точки  $p_0, \dots, p_t$  по следния начин:

За всяко  $0 \leq j \leq t - 1$  има отсечка, свързваща точките  $p_j$  и  $p_{j+1}$ .

За новия дизайн, вие вече имате маркирани  $n$  **точки** в двумерната равнина. Координатите на точките  $i$  ( $1 \leq i \leq n$ ) са  $(x[i], y[i])$ . **Няма две точки с еднакви  $x$  или с еднакви  $y$  координати.**

Сега искате да намерите редица от точки  $(sx[0], sy[0]), (sx[1], sy[1]), \dots, (sx[k], sy[k])$ , която дефинира начупена линия, такава че:

- започва от  $(0, 0)$  (т.е.,  $sx[0] = 0$  и  $sy[0] = 0$ ),
- съдържа всички точки (незадължително като крайни точки на отсечките от линията) и
- съдържа само хоризонтални или вертикални отсечки (две последователни точки, които дефинират начупената линия трябва да имат равна  $x$  или  $y$  координата).

За начупената линия е позволено да пресича или застъпва себе си, т.е. всяка точка от равнината може да принадлежи на произволен брой отсечки от начупената линия.

Тази задача е output-only. Дадени са 10 входни файла, задаващи разположение на точки. За всеки входен файл вие трябва да събмитнете изходен файл, описващ търсена начупена линия със зададените по-горе свойства. За всеки изходен файл, описващ валидна начупена линия, вашите точки зависят от **броя на отсечките**, които образуват начупената линия, което подробно е описано в раздела Оценяване. За тази задачи вие НЕ събмитвате сорс.

## Вход

Всеки входен файл има следния формат:

- ред 1:  $n$
- ред  $1 + i$  (за  $1 \leq i \leq n$ ):  $x[i] \ y[i]$

## Изход

Всеки изходен файл трябва да бъде в следния формат:

- ред 1:  $k$
- ред  $1 + i$  (за  $1 \leq i \leq k$ ):  $sx[i] \ sy[i]$

Забележете, че вторият ред трябва да съдържа  $sx[1]$  и  $sy[1]$  (т.е изходът **НЕ** съдържа  $sx[0]$  и  $sy[0]$ ). Всичките  $sx[i]$  и  $sy[i]$  трябва да са цели числа.

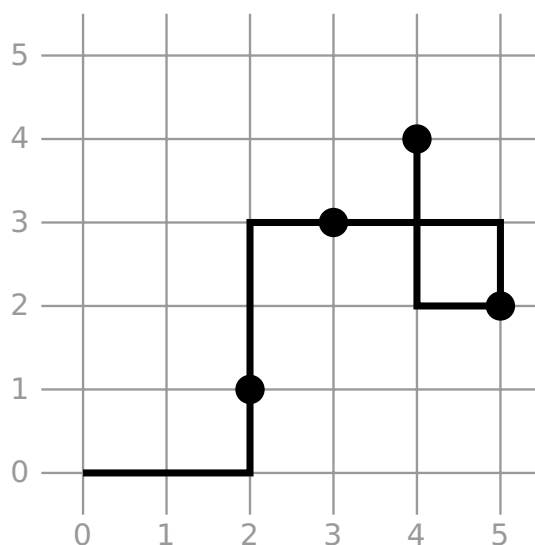
## Пример

За примерния вход:

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

един възможен валиден изход е:

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



Забележка: този пример НЕ е измежду входовете, с които ще се тества задачата.

## Ограничения

- $1 \leq n \leq 100\,000$
- $1 \leq x[i], y[i] \leq 10^9$
- Всички стойности на  $x[i]$  и  $y[i]$  са цели числа.
- Никои две точки нямат еднакви  $x$  или еднакви  $y$  координати, т.е.  $x[i_1] \neq x[i_2]$  и  $y[i_1] \neq y[i_2]$  за  $i_1 \neq i_2$ .
- $-2 \cdot 10^9 \leq sx[j], sy[j] \leq 2 \cdot 10^9$
- Размерът на всеки събмитнат от вас файл (даже и да е zip) не трябва да надминава 15MB.

## Оценяване

За всеки тест вие получавате максимално по 10 точки. Вашият изход за тест получава 0 точки, ако той не удовлетворява изискванията за начупена линия. В противен случай точките ви се определят чрез намаляваща редица  $c_1, \dots, c_{10}$ , която е различна за различните тестове.

Ако приемем, че вашият изход е валидна начупена линия, състояща се от  $k$  отсечки. Тогава вие получавате

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

По-долу е дадена редицата  $c_1, \dots, c_{10}$  за всеки от тестовете:

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

## Визуализатор

В атачмента за задачата има скрипт, чрез който може да визуализирате входа и изхода.

За да визуализирате входа, използвайте следната команда:

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

Вие може да визуализирате вашия изход за някои от входовете, чрез командата (по технически причини, предоставеният визуализатор може да показва само **първите 1000 отсечки** от вашия изход):

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

Пример:

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