



Brekende lijn

Azerbaidjan is beroemd om haar tapijten. Als meester tapijontwerper wil je een nieuw ontwerp maken door een **brekende lijn** te tekenen. Een brekende lijn is een rij van t lijnsegmenten op een twee-dimensionale vlakke, die door een rij van $t + 1$ punten p_0, \dots, p_t als volgt gedefinieerd wordt: voor elke $0 \leq j \leq t - 1$ is er een segment dat punt p_j verbindt met punt p_{j+1} .

Om een nieuw ontwerp te maken heb je al n **stippen** gemarkeerd op een twee-dimensionale vlakke. De coördinaten van stip i ($1 \leq i \leq n$) zijn $(x[i], y[i])$. **Geen twee stippen hebben dezelfde x-coördinaat, of dezelfde y-coördinaat.**

Je wilt nu een rij punten $(sx[0], sy[0]), (sx[1], sy[1]), \dots, (sx[k], sy[k])$ vinden die een brekende lijn definiëren die:

- Begint op $(0, 0)$ (dus $sx[0] = 0$ en $sy[0] = 0$).
- Alle stippen bevat (niet per se als eindpunten van de segmenten).
- Alleen uit horizontale of verticale segmenten bestaat (twee opeenvolgende punten van de brekende lijn hebben dezelfde x- of y-coördinaat).

De brekende lijn mag zichzelf op elke manier doorkruisen of overlappen. Anders gezegd: ieder punt op de vlakke kan bij een willekeurig aantal segmenten van de brekende lijn horen.

Dit is een output-only taak met deelscore. Je krijgt 10 inputbestanden met de locaties van de stippen. Voor elk inputbestand moet je een outputbestand inzenden dat een brekende lijn omschrijft met de nodige eigenschappen. Voor elk outputbestand met een geldige brekende lijn is je score afhankelijk van het **aantal segmenten** in de brekende lijn. (zie de 'Score'-sectie verderop)

Het is niet de bedoeling om code in te zenden voor deze taak.

Invoerformaat

Elk inputbestand is in het volgende formaat:

- regel 1: n
- regel $1 + i$ (voor $1 \leq i \leq n$): $x[i] \ y[i]$

Uitvoerformaat

Elk outputbestand moet het volgende formaat hebben:

- regel 1: k
- regel $1 + j$ (voor $1 \leq j \leq k$): $sx[j] \ sy[j]$

Let op: de tweede regel moet $sx[1]$ en $sy[1]$ bevatten, dus de output **moet niet** $sx[0]$ en $sy[0]$ bevatten. Elke $sx[j]$ en $sy[j]$ moeten gehele getallen zijn.

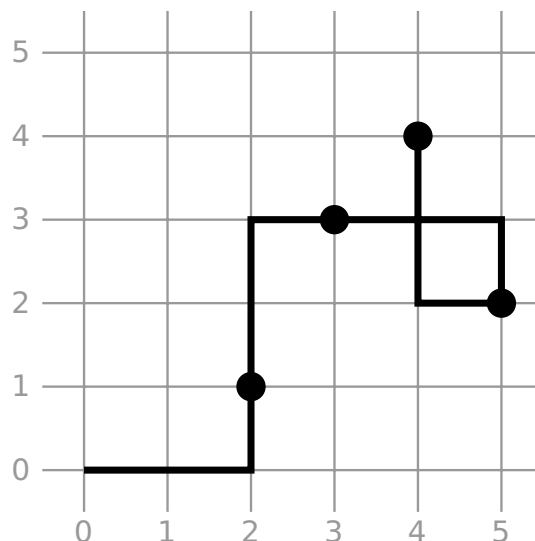
Voorbeeld

Voor de voorbeeldinput:

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

is een mogelijke geldige output:

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



Dit voorbeeld zit niet in de werkelijke inputs van deze taak.

Randvoorwaarden

- $1 \leq n \leq 100\,000$
- $1 \leq x[i], y[i] \leq 10^9$
- Alle waarden van $x[i]$ en $y[i]$ zijn gehele getallen.
- Geen twee stippen hebben dezelfde x of dezelfde y coördinaten, dus $x[i_1] \neq x[i_2]$ en $y[i_1] \neq y[i_2]$ voor $i_1 \neq i_2$.
- $-2 \cdot 10^9 \leq sx[j], sy[j] \leq 2 \cdot 10^9$
- De grootte van elk ingezonden bestand (output of zip) mag niet meer dan 15MB zijn.

Score

Voor elk testgeval kan je maximaal 10 punten krijgen. De output voor een testgeval krijg 0 punten als deze niet een geldige brekende lijn definieert. Anders wordt je score bepaald met een aflopende rij c_1, \dots, c_{10} die per testcase verschilt.

Neem aan dat je oplossing een geldige brekende lijn is die uit k segmenten bestaat. Je krijgt dan:

- i punten als $k = c_i$ (voor $1 \leq i \leq 10$).
- $i + \frac{c_i - k}{c_i - c_{i+1}}$ punten als $c_{i+1} < k < c_i$ (voor $1 \leq i \leq 9$).
- 0 punten als $k > c_1$.
- 10 punten als $k < c_{10}$.

De rij c_1, \dots, c_{10} voor elke testcase staat hieronder.

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

Visualizer

In de bijlage van deze taak zit een script dat het mogelijk maakt om input- en outputbestanden te visualiseren.

Om een inputbestand te visualiseren gebruik je het volgende commando:

```
python vis.py [inputbestand]
```

Je kan je oplossing ook visualiseren door het volgende commando te gebruiken. Om technische redenen laat de visualizer alleen **de eerste 1000 segmenten** zien van het outputbestand.

```
python vis.py [inputbestand] --solution [outputbestand]
```

Voorbeeld:

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