สวนน้ำพุ

ในสวนสาธารณะใกล้ๆ มี **น้ำพุ** n อ่าง แต่ละน้ำพุกำกับด้วยหมายเลข 0 ถึง $n-1$ เราจะพิจารณาให้น้ำพุอยู่บนจุดในระนาบสองมิติ กล่าวคือน้ำพุ i ($0 \leq i \leq n-1$) เป็นจุด $(x[i], y[i])$ ซึ่ง $x[i]$ และ $y[i]$ เป็น **เลขจำนวนเต็มคู่** ที่ตั้งของน้ำพุทั้งหมดจะแตกต่างกัน

ทิโมธีเป็นสถาปนิกที่ถูกจ้างให้สร้าง **ถนน** และวางตำแหน่ง **ม้านั่ง** ในแต่ละถนน ถนนเป็นส่วนของเส้นตรงความยาว 2 ใน **แนวนอน** หรือ **แนวตั้ง** ที่จุดปลายทางอยู่ที่น้ำพุสองอ่างที่ต่างกัน ถนนควรถูกสร้างโดยที่คุณสามารถที่จะเดินทางไปมาจากสองน้ำพุใดๆ ผ่านทางถนนได้ ในตอนเริ่มต้นไม่มีถนนอยู่ในสวนสาธารณะ

ในแต่ละถนนจะมีม้านั่งหนึ่งตัว **เท่านั้น** และม้านั่งจะถูกเลือกให้อยู่กับถนนนั้น (เช่นหันหน้าเข้าถนน) ม้านั่งแต่ละตัวจะถูกวางในตำแหน่ง (a, b) โดยที่ a และ b เป็น **เลขจำนวนเต็มคี่** ตำแหน่งของม้านั่งแต่ละตัวต้อง **แตกต่าง** กัน ม้านั่งที่ตำแหน่ง (a, b) สามารถถูกเลือกให้อยู่กับถนนหนึ่งได้ก็ต่อเมื่อ จุดปลายทาง **ทั้งสอง** ของถนนนั้น เป็น $(a-1, b-1)$, $(a-1, b+1)$, $(a+1, b-1)$ หรือ $(a+1, b+1)$ เท่านั้น ตัวอย่างเช่น ม้านั่งที่ตำแหน่ง $(3, 3)$ สามารถถูกเลือกให้อยู่กับถนน ซึ่งต้องเป็นส่วนของเส้นตรงหนึ่งในสี่แบบดังต่อไปนี้ เท่านั้น $(2, 2)$ – $(2, 4)$, $(2, 4)$ – $(4, 4)$, $(4, 4)$ – $(4, 2)$, $(4, 2)$ – $(2, 2)$

จงช่วยทิโมธีหาว่าเป็นไปได้หรือไม่ที่จะสร้างถนน วางตำแหน่งและเลือกม้านั่งให้อยู่กับถนน โดยเป็นไปตามข้อกำหนดข้างต้นทั้งหมด ถ้าหากเป็นไปได้ให้ระบุคำตอบ ในกรณีที่มีมากกว่าหนึ่งคำตอบที่เป็นไปได้คุณสามารถรายงานคำตอบที่ถูกต้องอันใดก็ได้

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้:

```
int construct_roads(int[] x, int[] y)
```

- x, y: อาร์เรย์สองอาร์เรย์ที่มีขนาด n สำหรับแต่ละ i ($0 \leq i \leq n-1$) น้ำพุ i จะอยู่ที่จุด $(x[i], y[i])$ โดยที่ $x[i]$ และ $y[i]$ เป็นจำนวนเต็มคู่
- ถ้าการก่อสร้างเป็นไปได้ ฟังก์ชันดังกล่าวจะต้องเรียกฟังก์ชัน `build` (ดูรายละเอียดด้านล่าง) หนึ่งครั้ง เพื่อที่จะรายงานคำตอบ จากนั้นให้ฟังก์ชันจบการทำงานและคืนค่า 1
- ถ้าเป็นไปไม่ได้ ฟังก์ชันจะต้องคืนค่า 0 โดยไม่มีการเรียกฟังก์ชัน `build`
- ฟังก์ชันนี้จะถูกเรียกใช้หนึ่งครั้งเท่านั้น

ฟังก์ชันที่คุณเขียนสามารถเรียกฟังก์ชันด้านล่างเพื่อรายงานวิธีการก่อสร้างถนนและการวางตำแหน่งม้านั่ง

```
void build(int[] u, int[] v, int[] a, int[] b)
```

- ให้ m เป็นจำนวณถนนทั้งหมดในการก่อสร้าง

- u, v: อาร์เรย์สองอาร์เรย์ที่มีขนาด m ที่ระบุถนนที่จะต้องสร้าง ถนนเหล่านี้กำกับด้วยหมายเลขจาก 0 ถึง $m-1$ สำหรับแต่ละ j ($0 \leq j \leq m-1$) ถนน j เชื่อมระหว่างน้ำพุ $u[j]$ และ $v[j]$ ถนนแต่ละเส้นจะต้องเป็นส่วนของเส้นตรงในแนวตั้งหรือแนวนอนความยาว 2 หน่วย ถนนที่แตกต่างกันจะสามารถมีจุดร่วมกันได้แค่จุดเดียวที่ตำแหน่งน้ำพุ ถนนต่าง ๆ ที่สร้างจะต้องรับประกันว่าจะสามารถเดินทางจากน้ำพุใด ๆ ไปยังน้ำพุอื่น ๆ โดยผ่านทางถนนเหล่านี้ได้ทั้งหมด
- a, b: อาร์เรย์สองอาร์เรย์ขนาด m ระบุม้านั่ง สำหรับแต่ละ j ($0 \leq j \leq m-1$) ม้านั่งจะถูกวางที่ตำแหน่ง $(a[j], b[j])$ และจะถูกเลือกให้กับถนน j ห้ามม้านั่งสองอันอยู่ที่ตำแหน่งเดียวกัน

ตัวอย่าง

ตัวอย่างที่ 1

จากการเรียกฟังก์ชัน:

```
construct_roads([4, 4, 6, 4, 2], [4, 6, 4, 2, 4])
```

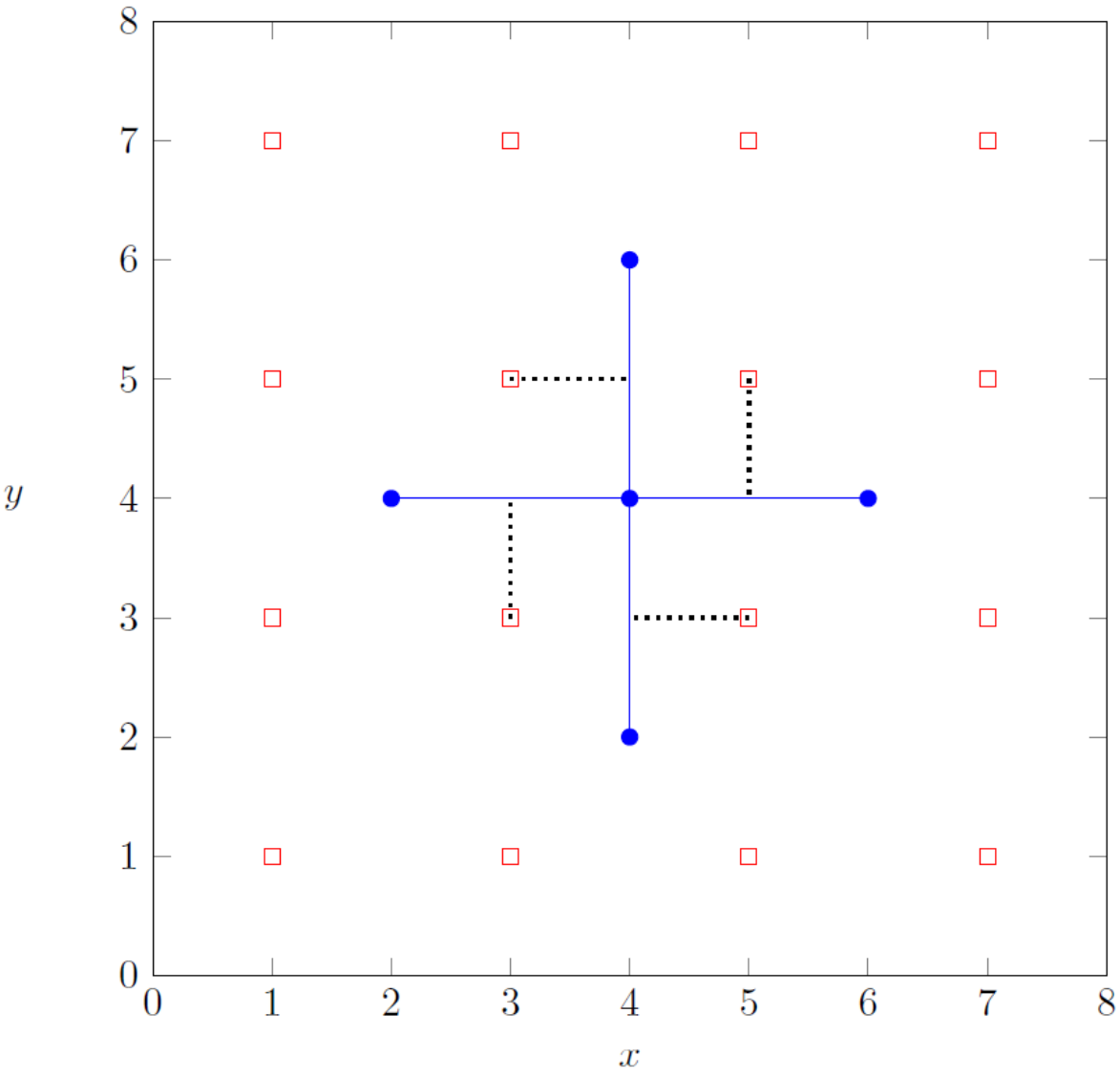
ซึ่งหมายความว่ามีน้ำพุ 5 อ่าง:

- น้ำพุ 0 อยู่ที่ $(4, 4)$,
- น้ำพุ 1 อยู่ที่ $(4, 6)$,
- น้ำพุ 2 อยู่ที่ $(6, 4)$,
- น้ำพุ 3 อยู่ที่ $(4, 2)$,
- น้ำพุ 4 อยู่ที่ $(2, 4)$

คุณสามารถสร้างถนน 4 ถนนได้โดยที่ถนนแต่ละถนนเชื่อมระหว่างน้ำพุสองอ่าง และวางม้านั่งดังต่อไปนี้

ชื่อถนน	ชื่อของน้ำพุที่ถนนเชื่อม	ตำแหน่งของม้านั่งที่ถูกเลือก
0	$0, 2$	$(5, 5)$
1	$0, 1$	$(3, 5)$
2	$3, 0$	$(5, 3)$
3	$4, 0$	$(3, 3)$

โดยคำตอบเป็นไปตามแผนภาพต่อไปนี้:



เมื่อจะส่งคำตอบนี้, ฟังก์ชัน `construct_roads` ควรจะเรียกฟังก์ชันดังต่อไปนี้

- `build([0, 0, 3, 4], [2, 1, 0, 0], [5, 3, 5, 3], [5, 5, 3, 3])`

โดยที่ฟังก์ชันจะคืนค่า 1

สังเกตว่าในกรณีนี้มีคำตอบที่เป็นไปได้มากกว่าหนึ่งคำตอบที่ตรงตามข้อกำหนด ซึ่งถือว่าเป็นคำตอบที่ถูกต้อง ตัวอย่างเช่น สามารถที่จะเรียกฟังก์ชัน `build([1, 2, 3, 4], [0, 0, 0, 0], [5, 5, 3, 3], [5, 3, 3, 5])` แล้วฟังก์ชันคืนค่าเป็น 1

ตัวอย่างที่ 2

จากการเรียกฟังก์ชัน:

```
construct_roads([2, 4], [2, 6])
```

น้ำพุ 0 จะอยู่ที่ตำแหน่ง $(2, 2)$ และน้ำพุ 1 จะอยู่ที่ตำแหน่ง $(4, 6)$ เนื่องจากเป็นไปที่ไม่ได้ที่จะสร้างถนนและตรงไปตามข้อกำหนด `construct_roads` ต้องคืนค่า 0 โดยที่ไม่เรียกฟังก์ชัน `build`

ข้อจำกัด

- $1 \leq n \leq 200\,000$
- $2 \leq x[i], y[i] \leq 200\,000$ (สำหรับทุก $0 \leq i \leq n-1$)
- $x[i]$ และ $y[i]$ เป็นเลขจำนวณเต็มคู่ (สำหรับทุก $0 \leq i \leq n-1$)
- ไม่มีน้ำพุสองอันอยู่ตำแหน่งเดียวกัน

ปัญหาย่อย

1. (5 คะแนน) $x[i] = 2$ (สำหรับทุก $0 \leq i \leq n-1$)
2. (10 คะแนน) $2 \leq x[i] \leq 4$ (สำหรับทุก $0 \leq i \leq n-1$)
3. (15 คะแนน) $2 \leq x[i] \leq 6$ (สำหรับทุก $0 \leq i \leq n-1$)
4. (20 คะแนน) มีอย่างมากหนึ่งคำตอบที่ทำให้สามารถสร้างถนนโดยที่จะสามารถเดินทางระหว่างน้ำพุสองอ่างใดๆบนถนน
5. (20 คะแนน) ไม่มีน้ำพุสี่อ่างที่รวมเป็นสี่เหลียมจัตุรัส 2×2
6. (30 คะแนน) ไม่มีข้อจำกัดอื่นๆอันใด

เกรดเดอร์ตัวอย่าง

เกรดเดอร์ตัวอย่างจะอ่านข้อมูลนำเข้าในรูปแบบต่อไปนี้

- บรรทัดที่ 1 : n
- บรรทัดที่ $2+i$ ($0 \leq i \leq n-1$): $x[i]$ $y[i]$

เกรดเดอร์ตัวอย่างเขียนข้อมูลส่งออกในรูปแบบต่อไปนี้

- บรรทัดที่ 1: คืนค่าของ `construct_roads`

ถ้าผลลัพท์จาก `construct_roads` เป็น 1 และ `build(u, v, a, b)` ถูกเรียก เกรดเดอร์จะพิมพ์ค่าเพิ่มเติม

- บรรทัดที่ 2: m
- บรรทัดที่ $3+j$ ($0 \leq j \leq m-1$): $u[j]$ $v[j]$ $a[j]$ $b[j]$