

ช่วยเหลือ (rescue)

มีคนต้องการความช่วยเหลือ N คน คนที่ i อยู่ที่พิกัด $(X[i], Y[i])$ ระหว่างการทำงาน อาจจะมีคนที่ต้องการความช่วยเหลือเปลี่ยนใจได้

คุณจะได้รับคำสั่งสองแบบ รวมจำนวน Q คำสั่ง

แบบที่ 1: เป็นคำถามในรูปแบบ: (A, B, C, D) (โดยที่ $A \leq C$ และ $B \leq D$) ที่ระบุจุดมุมของรูปสี่เหลี่ยมมุมฉาก ที่จุดมุมล่างซ้ายคือ (A, B) และมุมบนขวาคือ (C, D) คุณจะตอบคำถามดังนี้

- ถ้ามีคนที่ต้องการความช่วยเหลืออยู่ในขอบเขตสี่เหลี่ยมมุมฉาก (นั่นคือ มีบาง i ที่ $A \leq X[i] \leq C$ และ $B \leq Y[i] \leq D$) ให้คืนค่าจำนวนคนที่อยู่ในขอบเขตสี่เหลี่ยมมุมฉากนั้น เพื่อให้ค่าที่คืนแตกต่างจากในอีกกรณีให้คืนค่าเป็นจำนวนเต็มลบ
- ถ้าไม่มีใครที่ต้องการความช่วยเหลือในขอบเขตนั้นเลย ให้ตอบระยะทางสั้นที่สุดที่คนที่ต้องการความช่วยเหลือสักคนจะเดินทางมายังขอบเขตสี่เหลี่ยมมุมฉากนี้ โดยให้พิจารณาว่าคนจะสามารถเดินทางได้ตามแนวแกน x หรือแกน y เท่านั้น ถ้าไม่มีใครสามารถมาได้เลยให้ตอบ 0

แบบที่ 2: เป็นการระบุว่าคนที่ R ไม่ต้องการความช่วยเหลือแล้ว (หลังจากคำสั่งนี้ ให้คิดเสมือนว่าคนที่ R ไม่มีอยู่เลยได้) รับประกันว่าก่อนจะได้รับคำสั่งนี้ คนที่ R จะยังต้องการความช่วยเหลืออยู่

พิจารณาตัวอย่างต่อไปนี้ สมมติ $N = 5, Q = 8$ โดยที่พิกัดของแต่ละคนเป็นดังนี้

i	$X[i]$	$Y[i]$
0	1	0
1	10	10
2	30	2
3	5	8
4	7	12

จากนั้นจะได้คำสั่ง Q คำสั่ง

- ถ้าได้รับคำสั่งแบบที่ 1 เป็นคำถามที่ถาม $(0, 0, 10, 10)$ ที่ระบุขอบเขตสี่เหลี่ยมที่มีจุดมุมล่างซ้ายอยู่ที่ $(0, 0)$ จุดมุมบนขวาที่ $(10, 10)$ พบว่ามี 3 คนอยู่ในขอบเขตดังกล่าว คำตอบคือ -3
- ถ้าได้รับคำสั่งแบบที่ 1 เป็นคำถามที่ถาม $(4, 1, 9, 8)$ ที่ระบุขอบเขตสี่เหลี่ยมที่มีจุดมุมล่างซ้ายอยู่ที่ $(4, 1)$ จุดมุมบนขวาที่ $(9, 8)$ พบว่ามี 1 คนอยู่ในขอบเขตดังกล่าว คำตอบคือ -1
- ถ้าได้รับคำสั่งแบบที่ 1 เป็นคำถามที่ถาม $(4, 1, 9, 2)$ ที่ระบุขอบเขตสี่เหลี่ยมที่มีจุดมุมล่างซ้ายอยู่ที่ $(4, 1)$ จุดมุมบนขวาที่ $(9, 2)$ พบว่าไม่มีใครอยู่ในขอบเขตเลย แต่คนที่ 2, 3 และ 4 สามารถเดินทางมายังขอบเขตได้ด้วยการเดินทางตามแนวแกน x หรือ y สังเกตว่าคนที่ 3 ที่พิกัด $(5, 8)$ สามารถเดินทางมายังขอบเขตได้ ด้วย

- ระยะทาง 6 หน่วยซึ่งเป็นระยะสั้นที่สุด คำตอบจึงเป็น 6
- ถ้าได้รับคำสั่งแบบที่ 1 เป็นคำถามที่ถาม (5, 0, 6, 3) ที่ระบุขอบเขตสี่เหลี่ยมที่มีจุดมุมล่างซ้ายอยู่ที่ (5, 0) จุดมุมบนขวาที่ (6, 3) พบว่าไม่มีใครอยู่ในขอบเขตเลย แต่คนที่ 0, 2 และ 3 สามารถเดินทางมายังขอบเขตได้ด้วยการเดินตามแนวแกน x หรือ y สังเกตว่าคนที่ 0 ที่พิกัด (1, 0) สามารถเดินทางมายังขอบเขตได้ ด้วยระยะทาง 4 หน่วยซึ่งเป็นระยะสั้นที่สุด คำตอบจึงเป็น 4
- ถ้าได้รับคำสั่งแบบที่ 1 เป็นคำถามที่ถาม (2, 2, 2, 2) ที่ระบุขอบเขตสี่เหลี่ยมที่มีจุดมุมล่างซ้ายอยู่ที่ (2, 2) จุดมุมบนขวาที่ (2, 2) พบว่าไม่มีใครอยู่ในขอบเขตเลย แต่คนที่ 2 สามารถเดินทางมายังขอบเขตได้ด้วยการเดินตามแนวขนานแกน x ด้วยระยะทาง 28 หน่วย คำตอบจึงเป็น 28
- ถ้าได้รับคำสั่งแบบที่ 1 เป็นคำถามที่ถาม (3, 5, 4, 7) ที่ระบุขอบเขตสี่เหลี่ยมที่มีจุดมุมล่างซ้ายอยู่ที่ (3, 5) จุดมุมบนขวาที่ (4, 7) พบว่าไม่มีใครอยู่ในขอบเขตเลย และไม่มีใครสามารถเดินทางมายังขอบเขตได้โดยการเดินตามแนวขนานแกน x หรือแกน y ดังนั้นคำตอบจึงเป็น 0
- หลังจากนั้นสมมติว่าได้คำสั่งแบบที่ 2 ที่ระบุ $R = 0$ เมื่อได้รับคำสั่งดังกล่าว เราจะพิจารณาว่าคนที่ 0 ที่อยู่ที่พิกัด (1, 0) ไม่ต้องการความช่วยเหลืออีกต่อไป
- จากนั้นถ้าได้รับคำสั่งแบบที่ 1 เป็นคำถามที่ถาม (5, 0, 6, 3) ที่ระบุขอบเขตสี่เหลี่ยมที่มีจุดมุมล่างซ้ายอยู่ที่ (5, 0) จุดมุมบนขวาที่ (6, 3) พบว่าไม่มีใครอยู่ในขอบเขตเลย แต่เหลือแค่คนที่ 2 ที่ยังต้องการความช่วยเหลือ และสามารถเดินทางมายังขอบเขตได้ด้วยการเดินตามแนวแกน x หรือ y สังเกตว่าคนที่ 2 และ 3 สามารถเดินทางมาได้ และคนที่ 3 ที่พิกัด (5, 8) สามารถเดินทางมายังขอบเขตได้ ด้วยระยะทาง 5 หน่วยซึ่งเป็นระยะสั้นที่สุด คำตอบจึงเป็น 5

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้

```
void init_rescue(int N, int Q, vector<int> X, vector<int> Y)
```

- ฟังก์ชันดังกล่าวจะถูกเรียกหนึ่งครั้ง

```
int answer_query(int A, int B, int C, int D)
```

- แสดงว่ามีคำสั่งแบบที่ 1 ซึ่งคุณจะต้องตอบคำถาม (A, B, C, D) นั่นคือให้พิจารณาขอบเขตสี่เหลี่ยมที่มีมุมล่างซ้ายคือ (A, B) และมุมบนขวาคือ (C, D)

```
void remove_request(int R)
```

- แสดงว่ามีคำสั่งแบบที่ 2 ที่ระบุว่าคนที่ R ไม่ต้องการความช่วยเหลือแล้ว รับประกันว่าจะไม่มีค่า R ได้โดยเรียกมากกว่าหนึ่งครั้ง

เงื่อนไข

- $1 \leq N \leq 100\,000$
- $1 \leq Q \leq 100\,000$
- $0 \leq X[i] \leq 200\,000$ สำหรับทุก ๆ ค่า $0 \leq i < N$

- $0 \leq Y[i] \leq 2000$ สำหรับทุก ๆ ค่า $0 \leq i < N$
- ไม่มีคนสองคนอยู่ที่พิกัดเดียวกัน
- $0 \leq A \leq C \leq 200\,000, 0 \leq B \leq D \leq 2000$
- ในกรณีที่มี $Y[i] \neq 0, X[i] \leq 2000$ และ $C \leq 2000$
- $0 \leq R < N$

ปัญหาย่อย

1. (5 คะแนน) $N \leq 1000, Q \leq 1000$ ทุกพิกัดมีค่าไม่เกิน 2000
2. (5 คะแนน) ไม่มีคำสั่งแบบที่ 2, $Y[i] = 0, B = D = 0$, ในทุกคำถามจะมีคนอยู่ในขอบเขตเสมอ (คำตอบจะมีค่าเป็นลบในทุกคำถาม)
3. (5 คะแนน) ไม่มีคำสั่งแบบที่ 2, $Y[i] = 0, B = D = 0$
4. (5 คะแนน) ไม่มีคำสั่งแบบที่ 2, $X[i] \leq 2000, C \leq 2000$, ในทุกคำถามจะมีคนอยู่ในขอบเขตเสมอ (คำตอบจะมีค่าเป็นลบในทุกคำถาม)
5. (5 คะแนน) ไม่มีคำสั่งแบบที่ 2, $X[i] \leq 2000, C \leq 2000$
6. (10 คะแนน) $Y[i] = 0, B = D = 0$, ในทุกคำถามจะมีคนอยู่ในขอบเขตเสมอ (คำตอบจะมีค่าเป็นลบในทุกคำถาม)
7. (10 คะแนน) $Y[i] = 0, B = D = 0, A = C$
8. (15 คะแนน) $Y[i] = 0, B = D = 0$
9. (20 คะแนน) $X[i] \leq 2000, C \leq 2000$, ในทุกคำถามจะมีคนอยู่ในขอบเขตเสมอ (คำตอบจะมีค่าเป็นลบในทุกคำถาม)
10. (20 คะแนน) $X[i] \leq 2000, C \leq 2000$, ไม่มีเงื่อนไขเพิ่มเติม

สังเกตว่า ปัญหาย่อยที่มี $Y[i] \neq 0$ จะไม่ครอบคลุมปัญหาย่อยที่มี $Y[i] = 0$

ตัวอย่าง

จากตัวอย่างด้านบน เกรตเตอร์จะเรียกฟังก์ชัน

```
init_rescue(5, 6, [1, 10, 30, 5, 7], [0,10, 2, 8, 12])
```

หลังจากนั้นจะมีการเรียก `answer_query` จำนวน 6 ครั้งดังด้านล่าง:

```
answer_query(0, 0, 10, 10)
```

ซึ่งจะต้องคืนค่า -3

```
answer_query(4, 1, 9, 8)
```

ซึ่งจะต้องคืนค่า -1

```
answer_query(4, 1, 9, 2)
```

ซึ่งจะต้องคืนค่า 6

```
answer_query(5, 0, 6, 3)
```

ซึ่งจะต้องคืนค่า 4

```
answer_query(2, 2, 2, 2)
```

ซึ่งจะต้องคืนค่า 28

```
answer_query(3, 5, 4, 7)
```

ซึ่งจะต้องคืนค่า 0

หลังจากนั้นจะมีการเรียก

```
remove_request(0)
```

เพื่อระบุว่าคนที่ 0 ไม่ต้องการความช่วยเหลือแล้ว

สุดท้ายจะมีการเรียก

```
answer_query(5, 0, 6, 3)
```

ซึ่งจะต้องคืนค่า 5

เกรดเดอร์ตัวอย่าง

เกรดเดอร์ตัวอย่างจะอ่านข้อมูลในรูปแบบดังนี้:

- Line 1: N Q
- Line 2 to $1 + N$: $X[i]$ $Y[i]$
- Line $2 + N$ to $1 + N + Q$: แต่ละบรรทัดจะขึ้นด้วยจำนวนเต็ม K
 - ถ้า $K = 1$ จะเป็นคำสั่งแบบที่ 1 และตามด้วยจำนวนเต็ม A B C D
 - ถ้า $K = 2$ จะเป็นคำสั่งแบบที่ 2 และตามด้วยจำนวนเต็ม R

เกรดเดอร์ตัวอย่างจะพิมพ์ค่าที่คืนจาก `answer_query`

ขีดจำกัด

- Time limit: 2 seconds
- Memory limit: 512 MB