



## โปรแกรมเห็นแจ้ง

คุณกำลังทำการสร้างโปรแกรมการมองเห็นสำหรับหุ่นยนต์ ทุกครั้งที่หุ่นยนต์ถ่ายภาพ ภาพดังกล่าวจะถูกเก็บเป็นภาพขาวดำในหน่วยความจำของหุ่นยนต์ รูปภาพแต่ละรูปจะเก็บเป็นตารางกริดของจุด (pixel) ขนาด  $H \times W$  ที่แถวจะมีหมายเลข 0 จนถึง  $H - 1$  และคอลัมน์มีหมายเลข 0 จนถึง  $W - 1$  ในรูปดังกล่าวจะมีจุดสีดำ สองจุดพอดี เสมอ จุดที่เหลือทั้งหมดจะเป็นสีขาว

หุ่นยนต์สามารถประมวลผลรูปแต่ละรูปด้วยโปรแกรมที่ประกอบด้วยคำสั่งแบบง่าย ๆ คุณจะได้รับค่า  $H$ ,  $W$ , และจำนวนเต็มบวก  $K$  เป้าหมายของคุณคือเขียนฟังก์ชันที่สร้างโปรแกรมสำหรับหุ่นยนต์ที่ เมื่อรับรูปภาพใด ๆ จะตรวจสอบว่า ระยะทาง ระหว่างจุดสีดำสองจุดนั้นเท่ากับ  $K$  พอดีหรือไม่ ในที่นี้ ระยะทางระหว่างจุดที่แถว  $r_1$  และคอลัมน์  $c_1$  กับจุดที่แถว  $r_2$  และคอลัมน์  $c_2$  จะเท่ากับ  $|r_1 - r_2| + |c_1 - c_2|$  ในสูตรนี้  $|x|$  จะแทนค่าสัมบูรณ์ของ  $x$  ที่มีค่าเท่ากับ  $x$  ถ้า  $x \geq 0$  และเท่ากับ  $-x$  ถ้า  $x < 0$

เราจะอธิบายการทำงานของหุ่นยนต์ดังนี้

หน่วยความจำของหุ่นยนต์คืออาร์เรย์ที่มีขนาดใหญ่มากพอของช่อง ที่มีดัชนีเริ่มต้นที่ 0 แต่ละช่องจะสามารถเก็บค่า 0 หรือ 1 และค่าที่เก็บนี้เมื่อมีการกำหนดไปแล้ว จะไม่สามารถเปลี่ยนแปลงได้ รูปจะถูกเก็บไล่ไปทีละแถวในช่องที่มีดัชนี 0 ไปจนถึงช่องที่มีดัชนี  $H \cdot W - 1$  แถวแรกของรูปจะถูกเก็บในช่องดัชนี 0 ไปจนถึง  $W - 1$  และแถวสุดท้ายจะถูกเก็บในช่อง  $(H - 1) \cdot W$  ไปจนถึง  $H \cdot W - 1$  นั่นคือ ถ้าจุดที่แถว  $i$  และคอลัมน์  $j$  เป็นสีดำ ค่าของช่อง  $i \cdot W + j$  จะเป็น 1 ไม่เช่นนั้นจะเป็น 0

โปรแกรมของหุ่นยนต์จะเป็นลำดับของ คำสั่ง ที่มีการระบุหมายเลขต่อเนื่องกันไปโดยเริ่มจาก 0 เมื่อโปรแกรมเริ่มทำงาน, คำสั่งเหล่านี้จะถูกนำไปประมวลไล่ไปทีละคำสั่ง แต่ละคำสั่งจะอ่านค่าจากช่องหน่วยความจำหนึ่งช่องหรือมากกว่านั้น (เราเรียกค่าเหล่านี้ว่าเป็น ข้อมูลนำเข้า ของคำสั่ง) และสร้างค่าผลลัพธ์หนึ่งค่าที่มีค่าเป็น 0 หรือ 1 (เราเรียกค่านี้ว่าเป็น ข้อมูลส่งออก ของคำสั่ง) ผลลัพธ์ของคำสั่งที่  $i$  จะถูกเก็บที่ช่อง  $H \cdot W + i$  ข้อมูลนำเข้าของคำสั่งที่  $i$  จะเป็นได้แค่ค่าที่มาจากช่องที่เก็บข้อมูลจุดของรูปภาพ หรือข้อมูลส่งออกของคำสั่งก่อนหน้า นั่นคือ เป็นช่อง 0 ถึง  $H \cdot W + i - 1$

มีคำสั่งทั้งหมดสี่รูปแบบ

- NOT: มีข้อมูลนำเข้าแค่หนึ่งตัว ข้อมูลส่งออกจะเท่ากับ 1 ถ้าข้อมูลนำเข้าเท่ากับ 0 ไม่เช่นนั้นข้อมูลส่งออกจะมีค่าเป็น 0
- AND: มีข้อมูลนำเข้าหนึ่งตัวหรือมากกว่านั้น ข้อมูลส่งออกจะเป็น 1 ก็ต่อเมื่อข้อมูลนำเข้า ทั้งหมด มีค่าเป็น 1
- OR: มีข้อมูลนำเข้าหนึ่งตัวหรือมากกว่านั้น ข้อมูลส่งออกจะเป็น 1 ก็ต่อเมื่อมีข้อมูลนำเข้า อย่างน้อยหนึ่งตัว ที่มีค่า 1
- XOR: มีข้อมูลนำเข้าหนึ่งตัวหรือมากกว่านั้น ข้อมูลส่งออกจะเป็น 1 ก็ต่อเมื่อจำนวนของข้อมูลนำเข้า ที่มีค่าเป็น 1 เป็นเลขคี่

ข้อมูลส่งออกของคำสั่งสุดท้ายจะต้องเป็น 1 ถ้าระยะทางระหว่างจุดดำสองจุดมีค่าเท่ากับ  $K$  และเป็น 0 ใน

กรณีอื่น ๆ

## รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้:

```
void construct_network(int H, int W, int K)
```

- $H, W$ : ขนาดของรูปที่หุ่นยนต์ถ่าย
- $K$ : จำนวนเต็มบวก
- ฟังก์ชันนี้จะต้องสร้างโปรแกรมสำหรับหุ่นยนต์ สำหรับรูปภาพใด ๆ ที่หุ่นยนต์ได้ถ่าย โปรแกรมจะต้องตรวจสอบว่าระยะทางระหว่างจุดดำสองจุดเท่ากับ  $K$  พอดีหรือไม่

ฟังก์ชันจะต้องเรียกฟังก์ชันด้านล่างต่อไปนี้เพื่อที่จะเพิ่มคำสั่งเข้าไปในโปรแกรมของหุ่นยนต์ (ที่เริ่มต้นด้วยโปรแกรมว่าง ๆ)

```
int add_not(int N)
int add_and(int[] Ns)
int add_or(int[] Ns)
int add_xor(int[] Ns)
```

- เพิ่มคำสั่ง NOT, AND, OR, หรือ XOR ตามลำดับ
- $N$  (สำหรับ `add_not`): ดัชนีของช่องที่ใช้เป็นข้อมูลนำเข้าของคำสั่ง NOT ที่เพิ่มเข้าไป
- $Ns$  (สำหรับ `add_and`, `add_or`, `add_xor`): อาร์เรย์ที่ระบุรายการของดัชนีของช่องที่ใช้เป็นข้อมูลนำเข้าของคำสั่งที่เพิ่มเข้าไป
- แต่ละคำสั่งจะคืนค่าดัชนีของช่องในหน่วยความจำที่จะเก็บผลลัพธ์ของคำสั่งดังกล่าว การเรียกฟังก์ชันเหล่านี้ต่อเนื่องกันจะคืนค่าเป็นจำนวนเต็มต่อเนื่องกันโดยเริ่มจาก  $H \cdot W$

โปรแกรมของหุ่นยนต์จะมีคำสั่งได้ไม่เกิน 10 000 คำสั่ง คำสั่งทั้งหมดจะสามารถอ่านค่าจากช่องหน่วยความจำได้ไม่เกิน 1 000 000 คำรวมกัน กล่าวคือ ความยาวรวมของอาร์เรย์  $Ns$  ในการเรียก `add_and`, `add_or` และ `add_xor` เมื่อรวมกับจำนวนครั้งที่เรียก `add_not` จะต้องไม่เกิน 1 000 000

หลังจากเพิ่มคำสั่งสุดท้ายแล้ว ฟังก์ชัน `construct_network` จะต้องจบการทำงาน โปรแกรมหุ่นยนต์จะถูกตรวจสอบกับรูปจำนวนหนึ่ง โค้ดคำตอบของคุณจะผ่านข้อมูลทดสอบถ้าสำหรับรูปทดสอบเหล่านี้ ข้อมูลส่งออกของคำสั่งสุดท้ายมีค่าเป็น 1 ก็ต่อเมื่อระยะทางระหว่างจุดดำสองจุดในรูปมีค่าเท่ากับ  $K$

การตรวจโค้ดคำตอบของคุณอาจจะได้ผลลัพธ์เป็นข้อความแสดงความผิดพลาดอันใดอันหนึ่งดังต่อไปนี้

- Instruction with no inputs: มีการส่งอาร์เรย์ว่างให้กับฟังก์ชัน `add_and`, `add_or`, หรือ `add_xor`
- Invalid index: มีการส่งดัชนีของช่องผิดพลาด (อาจจะเป็นลบ) ให้กับฟังก์ชัน `add_and`, `add_or`, `add_xor`, หรือ `add_not`
- Too many instructions: ฟังก์ชันของคุณพยายามที่จะเพิ่มคำสั่งมากกว่า 10 000 คำสั่ง
- Too many inputs: คำสั่งของคุณเมื่อรวมทั้งหมดแล้วมีการอ่านค่าข้อมูลในช่องหน่วยความจำ

มากกว่า 1 000 000 ค่า

## ตัวอย่าง

สมมติว่า  $H = 2$ ,  $W = 3$ ,  $K = 3$  มีแค่สองกรณีเท่านั้นที่มีระยะระหว่างจุดดำสองจุดเท่ากับ 3

0	1	2
3	4	5

0	1	2
3	4	5

- กรณีที่ 1: จุดดำอยู่ที่ 0 และ 5
- กรณีที่ 2: จุดดำอยู่ที่ 2 และ 3

คำตอบที่เป็นไปได้แบบหนึ่งในการสร้างโปรแกรมของหุ่นยนต์จะมีการเรียกฟังก์ชันดังนี้

1. `add_and([0, 5])` เพิ่มคำสั่งที่มีข้อมูลส่งออกเป็น 1 ก็ต่อเมื่อรูปภาพอยู่ในกรณีที่ 1 ข้อมูลส่งออกจะเก็บที่ช่องที่มีดัชนีเท่ากับ 6
2. `add_and([2, 3])` เพิ่มคำสั่งที่มีข้อมูลส่งออกเป็น 1 ก็ต่อเมื่อรูปภาพอยู่ในกรณีที่ 2 ข้อมูลส่งออกจะเก็บที่ช่องที่มีดัชนีเท่ากับ 7
3. `add_or([6, 7])` เพิ่มคำสั่งที่มีข้อมูลส่งออกเป็น 1 ก็ต่อเมื่อรูปภาพอยู่ในกรณีใดกรณีหนึ่งจากกรณีด้านบน

## ข้อจำกัด

- $1 \leq H \leq 200$
- $1 \leq W \leq 200$
- $2 \leq H \cdot W$
- $1 \leq K \leq H + W - 2$

## ปัญหาย่อย

1. (10 คะแนน)  $\max(H, W) \leq 3$
2. (11 คะแนน)  $\max(H, W) \leq 10$
3. (11 คะแนน)  $\max(H, W) \leq 30$
4. (15 คะแนน)  $\max(H, W) \leq 100$
5. (12 คะแนน)  $\min(H, W) = 1$
6. (8 คะแนน) จุดที่อยู่ในแถวที่ 0 และคอลัมน์ที่ 0 จะเป็นสีดำเสมอ
7. (14 คะแนน)  $K = 1$
8. (19 คะแนน) ไม่มีเงื่อนไขเพิ่มเติมอื่น ๆ

## เกรดเดอรัตัวอย่าง

เกรดเดอร์ตัวอย่างอ่านข้อมูลนำเข้าในรูปแบบต่อไปนี้:

- บรรทัด 1:  $H \ W \ K$
- บรรทัด  $2 + i$  ( $i \geq 0$ ):  $r_1[i] \ c_1[i] \ r_2[i] \ c_2[i]$
- บรรทัดสุดท้าย:  $-1$

แต่ละบรรทัดยกเว้นบรรทัดสุดท้ายและบรรทัดแรกจะระบุข้อมูลของรูป โดยระบุตำแหน่งของจุดดำสองจุด เราจะเรียกรูปที่ระบุในบรรทัด  $2 + i$  ว่ารูปที่  $i$  จุดดำจุดหนึ่งจะอยู่ที่แถว  $r_1[i]$  และคอลัมน์  $c_1[i]$  อีกจุดจะอยู่ที่แถว  $r_2[i]$  และคอลัมน์  $c_2[i]$

เกรดเดอร์ตัวอย่างจะเริ่มโดยเรียก `construct_network(H, W, K)` ถ้า `construct_network` ทำงานผิดพลาดที่ระบุไว้ในโจทย์ เกรดเดอร์ตัวอย่างจะพิมพ์ข้อความแสดงการผิดพลาดที่ระบุไว้ตอนท้ายของส่วนรายละเอียดการเขียนโปรแกรมและจบการทำงาน

ไม่เช่นนั้น เกรดเดอร์ตัวอย่างจะสร้างข้อมูลส่งออกสองชุด

สำหรับข้อมูลส่งออกชุดแรก เกรดเดอร์ตัวอย่างจะพิมพ์ข้อมูลส่งออกของโปรแกรมหุ่นยนต์ในรูปแบบต่อไปนี้:

- บรรทัด  $1 + i$  ( $0 \leq i$ ): ข้อมูลส่งออกของคำสั่งสุดท้ายของโปรแกรมหุ่นยนต์เมื่อทำงานกับรูปที่  $i$  (มีค่าเป็น 1 หรือ 0)

สำหรับข้อมูลส่งออกชุดที่สอง เกรดเดอร์ตัวอย่างจะเขียนแฟ้ม `log.txt` ในไดเรกทอรีปัจจุบัน ที่มีรูปแบบดังนี้:

- บรรทัด  $1 + i$  ( $0 \leq i$ ):  $m[i][0] \ m[i][1] \ \dots \ m[i][c - 1]$

ลำดับในแถว  $1 + i$  ระบุค่าที่อยู่ในหน่วยความจำของหุ่นยนต์ช่องต่าง ๆ หลังจากที่โปรแกรมทำงานเสร็จเมื่อได้รับรูปภาพที่  $i$  เป็นข้อมูลนำเข้า กล่าวคือ  $m[i][j]$  ระบุค่าของช่อง  $j$  สังเกตว่าค่าของ  $c$  (ความยาวของลำดับ) จะเท่ากับ  $H \cdot W$  บวกกับจำนวนของคำสั่งของโปรแกรมของหุ่นยนต์