



วงจร (circuit)

วงจรดิจิทัลรับข้อมูลนำเข้าเป็นตัวแปร N ตัว เรียกเป็น X_1, X_1, \dots, X_N จากนั้นให้ผลลัพธ์ Z

เราจะพิจารณาวงจรดังกล่าวเป็นฟังก์ชัน f คุณได้รับข้อกำหนดของวงจรเป็นการระบุผลลัพธ์ของข้อมูลนำเข้าแต่ละแบบจำนวน M ข้อ โดยข้อกำหนดข้อที่ i อยู่ในรูปแบบต่อไปนี้

$$V_{i,1} \ V_{i,2} \ V_{i,3} \ \dots \ V_{i,N} \ Z_i$$

โดยที่ $V_{i,j} \in \{0, 1\}$ และ $Z_i \in \{0, 1\}$ ซึ่งระบุว่า

$$f(V_{i,1}, V_{i,2}, V_{i,3}, \dots, V_{i,N}) = Z_i,$$

หรือในอีกความหมายหนึ่งคือ ถ้า $X_j = V_{i,j}$ สำหรับทุก ๆ ค่า $1 \leq j \leq N$ วงจรจะต้องให้ผลลัพธ์เท่ากับ Z_i นั้นเอง

เราสามารถสร้างวงจรได้หลายรูปแบบผ่านทางเกทดิจิทัล แต่ในข้อนี้จะสนใจวงจรรูปแบบเฉพาะที่สร้างได้จาก OR gate และ AND gate ที่เขียนเป็นนิพจน์เชิงตรรกะในรูปแบบที่เรียกว่า sum of product นั่นคืออยู่ในรูปของการ OR กันของพจน์ที่เป็นการ AND กันของตัวแปร หรือ นิเสธ (NOT หรือเขียนแทนด้วย \neg) ของตัวแปร ยกตัวอย่างเช่น

$$(X_1 \wedge X_3) \vee (X_2 \wedge X_4 \wedge \neg X_3 \wedge \neg X_6) \vee (\neg X_2 \wedge \neg X_1)$$

เป้าหมายของเราคือได้วงจรที่มีขนาดเล็กที่สุด ในที่นี้ขนาดจะเท่ากับจำนวนเครื่องหมาย \wedge และ \vee ทั้งหมดที่คุณใช้ (ไม่นับ \neg)

พิจารณาตัวอย่างต่อไปนี้ ที่ $N = 2$ สมมติว่ามีข้อกำหนด 3 ข้อดังนี้

ข้อที่	$V_{i,1}$	$V_{i,2}$	Z_i
1	0	0	0
2	1	0	1
3	0	1	1

สังเกตว่าไม่มีการกำหนดว่าผลลัพธ์เมื่อ $X_1 = X_2 = 1$ ดังนั้นวงจรจะให้ผลลัพธ์เป็นอย่างไรก็ได้

วงจรที่ให้ผลลัพธ์สอดคล้องกับข้อกำหนดนี้มีเช่น

$$(X_1 \wedge \neg X_2) \vee (\neg X_1 \wedge X_2)$$

ซึ่งมีขนาดเท่ากับ 3 หน่วย และวงจร

$$(X_1) \vee (X_2)$$

ที่มีขนาดเท่ากับ 1 หน่วย

คุณต้องการหาวงจรที่มีขนาดเล็กที่สุด สำหรับข้อนี้เป็นโจทย์ output only กล่าวคือคุณจะได้รับเพิ่มข้อมูลนำเข้า จากนั้นคุณจะต้องสร้างผลลัพธ์สำหรับแต่ละข้อมูลทดสอบตามรายละเอียดที่จะได้กล่าวถัดไป

ข้อมูลนำเข้า

แฟ้มข้อมูล `xx.in` จะระบุข้อมูลในการทดสอบ $01, \dots, 10$ ในรูปแบบดังนี้

บรรทัดแรกระบุจำนวนเต็ม N และ M จากนั้นอีก M จะระบุข้อกำหนด โดยแต่ละบรรทัดระบุจำนวนเต็มอีก $N + 1$ จำนวนที่มีค่าเป็น 0 หรือเป็น 1 กล่าวคือในบรรทัดที่ i สำหรับ $0 \leq i < M$ จะมีข้อมูลในรูปแบบ

$$V_{i,1} \ V_{i,2} \ V_{i,3} \ \cdots \ V_{i,N} \ Z_i$$

แฟ้มข้อมูล `xx.sol` จะระบุค่า T ซึ่งเป็นจำนวนเกทเป้าหมาย ของข้อมูลทดสอบนั้น ๆ

ข้อมูลส่งออก

บรรทัดแรกระบุจำนวนเต็ม K แทนจำนวนพจน์ที่เกิดจากการ AND กันของตัวแปรหรือนิเสธของตัวแปร จากนั้นอีก K บรรทัดจะระบุข้อมูลของแต่ละพจน์ดังนี้

จำนวนเต็มตัวแรก L ระบุจำนวนตัวแปรหรือนิเสธของตัวแปร จากนั้นถ้าจะมีจำนวนเต็มอีก L ตัวระบุตัวแปรหรือนิเสธของตัวแปรในพจน์นั้น ถ้าจำนวนเต็มดังกล่าวเป็นจำนวนเต็มบวก i หมายความว่าในพจน์นี้มีตัวแปร X_i ถ้าเป็นจำนวนเต็มลบ $-i$ แสดงว่าในพจน์นี้มีนิเสธของตัวแปรคือ $\neg X_i$

เงื่อนไข

- $2 \leq N \leq 20$
- $1 \leq M \leq 10\,000$

ข้อมูลทดสอบ

จะมีข้อมูลทดสอบ 10 ชุดที่มีรายละเอียดดังนี้

ชุดที่	N	T
1	2	1
2	4	5
3	8	93
4	10	1238
5	12	1012
6	12	1789
7	15	2008
8	15	4649
9	20	835
10	20	3871

การให้คะแนน

จากตารางข้างต้นจะมีค่า T ของแต่ละชุดทดสอบอยู่ ซึ่งเป็นจำนวนเกทเป้าหมาย หากผู้เข้าแข่งขันใช้จำนวน AND และ OR gate รวมกัน S ครั้ง จะได้คะแนนดังนี้

เงื่อนไข	อัตราส่วนคะแนนที่ได้ต่อคะแนนเต็ม
$S \leq T$	1.0
$S > T$	$1.75 \left(1 - \left(\frac{S}{T}\right)^{0.75}\right)$

ตัวอย่าง

จากตัวอย่างข้างต้น เพิ่มข้อมูลนำเข้าจะเป็นดังนี้

```
2 3
0 0 0
1 0 1
0 1 1
```

ข้อมูลส่งออกที่สอดคล้องกับนิพจน์ $(X_1 \wedge \neg X_2) \vee (\neg X_1 \wedge X_2)$ คือ

```
2
2 1 -2
2 -1 2
```

วงจรนี้จะมีขนาดเท่ากับ 3 หน่วย และข้อมูลส่งออกที่สอดคล้องกับนิพจน์ $(X_1) \vee (X_2)$ คือ

```
2
1 1
1 2
```

ซึ่งจะเป็นวงจรขนาด 1

เกรตเตอร์ตัวอย่าง

สำหรับข้อนี้ เกรตเตอร์ตัวอย่างจะมีชื่อว่า `local_checker.cpp`

เกรตเตอร์ตัวอย่างจะรับข้อมูลชื่อแฟ้มข้อมูลทาง command line argument แล้วอ่านแฟ้มข้อมูลสองแฟ้มคือแฟ้มข้อมูลนำเข้า และแฟ้มข้อมูลส่งออกในรูปแบบตามด้านบน ถ้าวงจรทำงานได้ตามข้อกำหนดจะพิมพ์ขนาดของวงจร ถ้าไม่ได้จะพิมพ์ข้อผิดพลาด

วิธีการใช้งานเกรตเตอร์ตัวอย่าง:

1. ทำการคอมไพล์ไฟล์ `local_checker.cpp` ด้วยคำสั่ง `g++ -std=c++11 -O2 local_checker.cpp -o local_checker`
2. หากต้องการตรวจสอบข้อมูลนำเข้า `aaa.bbb` และข้อมูลส่งออก `ccc.ddd` สามารถใช้คำสั่ง `./local_checker aaa.bbb ccc.ddd` ได้

หากผู้เข้าแข่งขันมีปัญหาในการใช้เกรตเตอร์ตัวอย่าง กรุณาแจ้งผู้คุมสอบ