



Game

เจียนเจียเป็นเด็กหนุ่มที่ชอบเล่นเกม เมื่อมีใครถามคำถามกับเขา แทนที่จะตอบตรง ๆ เขาเลือกที่จะเล่นเกมกับคนที่ถามแทน เจียนเจียได้ไปพบเพื่อนของเขาที่ชื่อเหมยยู และได้บอกเธอเกี่ยวกับเครือข่ายเที่ยวบินของไต้หวัน ในไต้หวันมีเมืองอยู่ n เมือง (กำกับด้วยหมายเลข $0, \dots, n-1$) เมืองเหล่านี้บางเมืองจะเชื่อมต่อกันด้วยเที่ยวบินตรง เที่ยวบินแต่ละเที่ยวจะเชื่อมเมืองสองเมืองเข้าด้วยกัน และสามารถใช้เดินทางได้ทั้งสองทิศทาง

เหมยยูถามเจียนเจียว่า มันเป็นไปได้หรือไม่ที่จะเดินทางระหว่างคูเมืองทุกคูเมืองโดยเครื่องบิน (ไม่ว่าจะด้วยเที่ยวบินตรงหรือบินอ้อม) เจียนเจียไม่ต้องการที่จะบอกคำตอบ แต่เลือกที่จะเล่นเกมแทน โดยเหมยยูจะสามารถถามเจียนเจียด้วยคำถามในรูปแบบ "มี เที่ยวบินตรงที่เชื่อมระหว่างเมือง x กับเมือง y หรือไม่" และ เจียนเจียจะตอบคำถามดังกล่าวทันทีที่เหมยยูจะถามคำถามดังกล่าวสำหรับทุก ๆ คูเมือง คูเมืองละหนึ่งครั้งพอดี รวมจำนวนคำถามทั้งหมด $r = n(n-1)/2$ คำถาม เราจะถือว่าเหมยยูชนะเกมนี้ถ้าหากว่า หลังจากที่เหมยยูได้คำตอบของคำถามจำนวน i คำถามแรก สำหรับค่า $i < r$ แล้วเหมยยูสามารถสรุปได้ว่าเครือข่ายเที่ยวบินนี้เชื่อมต่อกันหรือไม่ กล่าวคือ เธอสามารถสรุปได้ว่ามันเป็นไปได้ หรือ เป็นไปไม่ได้ ที่จะเดินทางระหว่างทุก ๆ คูเมืองโดยเครื่องบิน (ไม่ว่าจะด้วยเที่ยวบินตรงหรือบินอ้อม) แต่ถ้าหากไม่เป็นเช่นนั้นแล้ว กล่าวคือ ถ้าเหมยยู จำเป็นที่จะต้องทราบคำตอบของคำถามทั้งหมด r คำถาม เราจะถือว่าเจียนเจียเป็นผู้ชนะเกมนี้

เพื่อให้เกมนี้น่าสนุกยิ่งขึ้นสำหรับเจียนเจีย พวกเขาได้ตกลงกันว่าเจียนเจียไม่ต้องใช้เครือข่ายเที่ยวบินที่เป็นอยู่จริง ๆ ในไต้หวัน และสามารถสมมติเครือข่ายขึ้นมาเองขณะที่กำลังเล่นเกมอยู่ได้ และตอบคำตอบตามคำถามก่อนหน้าของเหมยยู งานของคุณคือช่วยให้เจียนเจียเอาชนะเกมนี้ โดยพิจารณาว่าเขาควรจะตอบคำถามอย่างไรดี

ตัวอย่าง

เราอธิบายกฎของเกมนี้ด้วยตัวอย่าง 3 ตัวอย่าง แต่ละตัวอย่างมี $n = 4$ เมือง และ $r = 6$ รอบของคำถามและคำตอบ

ในตัวอย่างแรก (ดังตารางถัดไป) เจียนเจีย แพ้เนื่องจากว่าหลังจากการถามตอบรอบที่ 4 เหมยยูรู้ได้อย่างชัดเจนว่าเราสามารถเดินทางระหว่างคูเมืองใด ๆ ก็ได้โดยเครื่องบิน โดยไม่ขึ้นอยู่กับว่าเจียนเจียตอบอย่างไรในคำถามที่ 5 และ 6

| รอบ | คำถาม | คำตอบ |
|-------|-------|-------|
| 1 | 0, 1 | yes |
| 2 | 3, 0 | yes |
| 3 | 1, 2 | no |
| 4 | 0, 2 | yes |
| ----- | ----- | ----- |
| 5 | 3, 1 | no |
| 6 | 2, 3 | no |

ในตัวอย่างถัดมาเหมยอยู่สามารถพิสูจน์ได้ว่าหลังจากการถามตอบรอบที่ 3 ไม่ว่าเจียนเจียจะตอบคำถามที่ 4,5 หรือ 6 อย่างไม่รู้ก็ตาม เรา *ไม่สามารถ*เดินทางระหว่างเมือง 0 และ 1 โดยเครื่องบิน ดังนั้นเจียนเจียจึงแพ้เช่นกัน

| รอบ | คำถาม | คำตอบ |
|-------|-------|-------|
| 1 | 0, 3 | no |
| 2 | 2, 0 | no |
| 3 | 0, 1 | no |
| ----- | ----- | ----- |
| 4 | 1, 2 | yes |
| 5 | 1, 3 | yes |
| 6 | 2, 3 | yes |

ในตัวอย่างสุดท้ายเหมยอยู่ไม่สามารถสรุปได้ว่าเราสามารถเดินทางระหว่างคูเมืองใด ๆ ก็ได้โดยเครื่องบิน จนกระทั่งคำถามทั้ง 6 คำถามได้รับการตอบจนหมด ดังนั้นเจียนเจียจึงเป็นฝ่าย *ชนะ* เกมนี้ กล่าวโดยละเอียดคือ เพราะว่าเจียนเจียตอบว่า "มี" ในคำถามสุดท้าย (ดังตารางต่อไปนี้) มันจะเป็นไปได้ที่จะเดินทางระหว่างคูเมืองทุกคูเมือง แต่ถ้าเจียนเจียตอบว่า "ไม่มี" ในคำถามสุดท้าย จะทำให้การเดินทางระหว่างคูเมืองทุกคูเมืองเป็นไปได้

| รอบ | คำถาม | คำตอบ |
|-----|-------|-------|
| 1 | 0, 3 | no |
| 2 | 1, 0 | yes |
| 3 | 0, 2 | no |
| 4 | 3, 1 | yes |
| 5 | 1, 2 | no |
| 6 | 2, 3 | yes |

งานของคุณ

กรุณาเขียนโปรแกรมที่ช่วยให้เจียนเจียเอาชนะเกมนี้ ให้สังเกตว่า ทั้งเหมยอยู่และเจียนเจีย นั้นไม่ทราบถึงยุทธวิธีของอีกฝ่าย เหมยอยู่สามารถถามคำถามเกี่ยวกับคูเมืองใด ๆ ในลำดับใดก็ได้ และเจียนเจีย จะต้องตอบคำตอบทันทีโดยที่ยังไม่ทราบถึงคำถามในอนาคต คุณจะต้องเขียนฟังก์ชันสองฟังก์ชันต่อไปนี้

- initialize(n) -- เราจะเรียก initialize ของคุณก่อน. พารามิเตอร์ n คือจำนวนของเมือง
- hasEdge(u, v) -- หลังจากนั้นเราจะเรียก hasEdge เป็นจำนวน $r = n(n - 1)/2$ ครั้ง การเรียกเหล่านี้ระบุถึงการถามคำถามของเหมยอยู่ตามลำดับที่เธอถาม คุณจะต้องตอบว่ามันมีเที่ยวบินบินตรงระหว่างเมือง u และ v หรือไม่ กล่าวโดยละเอียดคือ ฟังก์ชันจะต้องคืนค่า 1 ถ้ามีเที่ยวบินบินตรง แต่จะคืนค่า 0 ถ้าไม่มีเที่ยวบินตรง

งานย่อย

แต่ละงานย่อยประกอบด้วยเกมหลาย ๆ เกม คุณจะได้คะแนนในงานย่อยก็ต่อเมื่อโปรแกรมของคุณทำให้เจียนเจียชนะในทุก ๆ เกม

| งานย่อย | คะแนน | n |
|---------|-------|-----|
|---------|-------|-----|

| งานย่อย | คะแนน | n |
|---------|-------|----------------------|
| 1 | 15 | $n = 4$ |
| 2 | 27 | $4 \leq n \leq 80$ |
| 3 | 58 | $4 \leq n \leq 1500$ |

รายละเอียดการเขียนโปรแกรม

คุณจะต้องส่งแฟ้มหนึ่งแฟ้ม เรียกว่า `game.c`, `game.cpp` หรือ `game.pas` โดยที่แฟ้มนี้จะต้องเขียนโปรแกรมย่อยดังที่กล่าวไว้ข้างบน โดยใช้รูปแบบดังต่อไปนี้

โปรแกรมภาษา C/C++

```
void initialize(int n);
int hasEdge(int u, int v);
```

โปรแกรมภาษา Pascal

```
procedure initialize(n: longint);
function hasEdge(u, v: longint): longint;
```

เกรตเตอร์ตัวอย่าง

เกรตเตอร์ตัวอย่างจะอ่านข้อมูลนำเข้าดังรูปแบบต่อไปนี้

- บรรทัดที่ 1: n
- หลังจากนั้นอีก r บรรทัด: แต่ละบรรทัดมีจำนวนเต็มสองจำนวน u และ v ซึ่งระบุถึงคำถามเกี่ยวกับเมือง u และ v