



หากันให้เจอ

ลัทธิเซ็ดกำแพง เป็นลัทธิที่เกิดจาก พระ ฑ. แห่งค่ายมีนา โดยหลักการของลัทธินี้มีอยู่ว่า จะต้องจับแพะไปบูชายัญ เพื่อสรรเสริญเทพเจ้าমনาว ผู้ที่มักจะมีกริ้ว อยู่เรื่อย ๆ (พิจารณาภาพประกอบ)



แต่ แพะนั้น ก็หายอยู่เรื่อย ๆ (พิจารณาภาพประกอบ)



สมมติว่า แพะนั้น ซ่อนตัวอยู่ที่ตำแหน่ง n ในขณะที่เทพเจ้ามนาวนั้น ซ่อนตัวอยู่ที่ตำแหน่ง x เทพเจ้าสุริยันแสงจ้า จะสามารถ หลอกล่อแพะและเทพเจ้ามนาวให้เดินออกมา ได้ (พิจารณาภาพประกอบ)



โดยในแต่ละวินาที แพะจะขยับตัวหนึ่งก้าว (กล่าวคือ ไปยังตำแหน่งที่ติดกับ u) และเทพเจ้ามะนาวจะขยับตัวหนึ่งก้าว (ไปยังตำแหน่งที่ติดกับ v) เท่านั้น (ห้ามมีใครคนใดคนหนึ่งหยุดนิ่ง) โดยในแต่ละวินาที เทพเจ้าสุริยันแสงจ้า จะต้องออกคำสั่ง **หลอกล่อ** โดยพูดชื่อตำแหน่ง u' และ v' ที่ติดกับ u และ v ตามลำดับ ไปเรื่อย ๆ จนกระทั่งถึงวินาทีหนึ่งที่แพะกับเทพเจ้ามะนาวมาเจอกัน (หรือว่า มาต่อยกัน... พิจารณาภาพประกอบ...)



หากเทพเจ้าสุริยันแสงจ้าสามารถทำการดังกล่าวได้จนทำให้แพะได้พบ(ต่อ)กับเทพเจ้ามะนาวได้ เราจะเรียกคู่อันดับ (u, v) ว่า **ตำแหน่งเริ่มต้นที่เป็นไปได้**

ให้กราฟไม่ระบุทิศทางไม่ระบุน้ำหนักที่ประกอบด้วย N จุดยอดและ M เส้นเชื่อม (จุดยอด 0 ถึง $N - 1$) โดยรับประกันว่าไม่มีเส้นเชื่อมที่เชื่อมจุดยอดเข้ากับตัวเอง และไม่มีคู่เส้นเชื่อมที่เชื่อมจุดยอดคู่เดียวกัน จงนับจำนวนคู่อันดับ (u, v) ที่เป็น **ตำแหน่งเริ่มต้นที่เป็นไปได้**

ในทางคณิตศาสตร์ สามารถนิยามได้ว่า $(u, v) \in \{0, \dots, N - 1\}^2$ เป็น **ตำแหน่งเริ่มต้นที่เป็นไปได้** เมื่อ เราสามารถหาลำดับจำกัด $(u_0, u_1, u_2, \dots, u_t)$ และ $(v_0, v_1, v_2, \dots, v_t)$ สำหรับบาง t ได้ โดยที่ $u_0, u_1, \dots, u_t, v_0, v_1, \dots, v_t \in \{0, \dots, N - 1\}$ และลำดับนั้นสอดคล้องกับเงื่อนไขดังนี้

- สำหรับทุกจำนวน $0 \leq i < t$ นั้น $u_i \neq u_{i+1}$ และมีเส้นเชื่อมระหว่าง u_i กับ u_{i+1} ในกราฟ
- $u_0 = u$ และ $v_0 = v$
- $u_t = v_t$

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันดังต่อไปนี้

```
long long count_init(int N, int M, vector<int> U, vector<int> V)
```

- ฟังก์ชันนี้จะถูกเรียกเพียงครั้งเดียว
- ตัวแปร N และ M แทนค่า N และ M
- ตัวแปร U และ V เป็นเวกเตอร์ขนาด M โดยแทนเส้นเชื่อมแต่ละเส้น กล่าวคือ ในกราฟจะมีเส้นเชื่อมระหว่างจุดยอด $U[i]$ กับ $V[i]$ สำหรับ i ตั้งแต่ 0 ถึง $M - 1$
- ฟังก์ชันนี้จะต้องคืนค่าจำนวนคู่อันดับ (u, v) ที่เป็นตำแหน่งเริ่มต้นที่เป็นไปได้

ขอบเขต

- $2 \leq N, M \leq 300\,000$
- $U[i]$ และ $V[i]$ เป็นจำนวนเต็มระหว่าง 0 ถึง $N - 1$ สำหรับทุก i ตั้งแต่ 0 ถึง $M - 1$
- $U[i] \neq V[i]$ สำหรับทุก i ตั้งแต่ 0 ถึง $M - 1$
- ไม่มีคู่ $i \neq j$ ที่ $U[i] = U[j]$ และ $V[i] = V[j]$
- ไม่มีคู่ $i \neq j$ ที่ $U[i] = V[j]$ และ $U[j] = V[i]$

ปัญหาย่อย

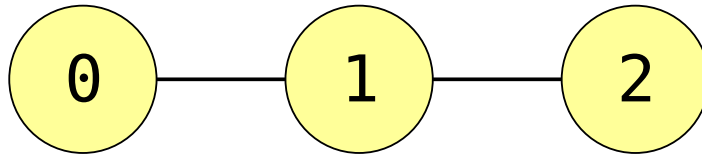
1. (11 คะแนน) รับประกันว่า $M = N - 1$ และ $U[i] = i$ และ $V[i] = i + 1$ สำหรับทุก i ตั้งแต่ 0 ถึง $N - 2$
2. (14 คะแนน) รับประกันว่า $M = N - 1$ และกราฟนั้นเชื่อมต่อกัน กล่าวคือสำหรับทุกจุดยอด u, v จะมีทางเดิน p_0, p_1, \dots, p_t สำหรับบาง t ที่ $p_0 = u$ และ $p_t = v$
3. (5 คะแนน) รับประกันว่ากราฟจะอยู่ในรูปแบบที่สร้างขึ้นมาจากสามเหลี่ยม (กล่าวคือ มีจุดยอด u, v, w ที่ u เชื่อมอยู่กับ v, v เชื่อมอยู่กับ w และ w เชื่อมอยู่กับ u) โดยมีวิธีการสร้างดังนี้: เริ่มแรกมีเพียงจุดยอด u, v กับ w และ เส้นเชื่อม uv, vw และ wu แต่ว่า ต่อมา จะมีจุดยอดใหม่เพิ่มมาที่ละหนึ่งจุด แล้วนำจุดยอดนั้นไปเชื่อมกับจุดยอดบางจุดยอดที่มีอยู่ ทำซ้ำไปเรื่อย ๆ จนจุดยอดครบทั้ง N จุด ต่อมา จะมีเส้นเชื่อมใหม่เพิ่มขึ้นมาที่ละเส้น ทำซ้ำไปเรื่อย ๆ จนเส้นเชื่อมครบทั้ง M เส้น
4. (9 คะแนน) รับประกันว่ากราฟเชื่อมต่อกัน
5. (17 คะแนน) $N \leq 100$
6. (21 คะแนน) $N \leq 1\,000$
7. (23 คะแนน) ไม่มีเงื่อนไขเพิ่มเติม

ตัวอย่างที่ 1

พิจารณาตัวอย่างต่อไปนี้

```
count_init(3, 2, [0, 1], [1, 2])
```

ตัวอย่างแสดงกราฟต่อไปนี้



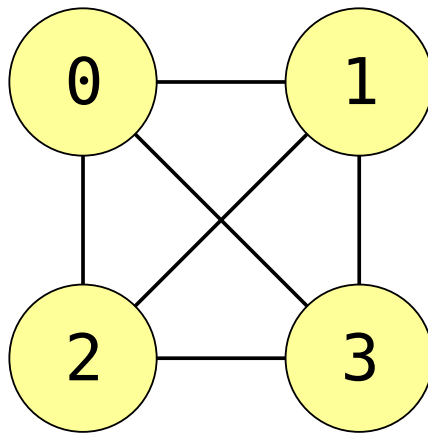
สังเกตได้ว่า หากแพะอยู่ที่ 0 และเทพเจ้ามะนาวอยู่ที่ 1 จะ **ไม่มีวิธีที่จะเดินไปหากันได้** เพราะทุกครั้งที่แพะขยับจากตำแหน่ง 0 ไปที่ 1 เทพเจ้ามะนาวก็ต้องขยับจาก 1 ไป 0 หรือ 2 ทำให้ไม่มีช่วงเวลาใดเลยที่แพะและเทพเจ้ามะนาวอยู่ที่ตำแหน่งเดียวกัน

ในกรณีนี้ คำตอบคือ 5 เกิดจากตำแหน่งเริ่มต้นที่เป็นไปได้ดังต่อไปนี้ (0, 0), (1, 1), (2, 2), (0, 2), (2, 0)

ฟังก์ชันจึงต้องคืนค่า 5

ตัวอย่างที่ 2

```
count_init(4, 6, [0, 1, 2, 3, 0, 1], [1, 2, 3, 0, 2, 3])
```



ฟังก์ชันจะต้องคืนค่า 16

เกรตเตอร์ตัวอย่าง

เกรตเตอร์ตัวอย่างจะรับข้อมูลนำเข้าดังต่อไปนี้:

- บรรทัดที่ 1: $N M$
- บรรทัดที่ $2+i$ ถึง $2 + M - 1$: $U[i] V[i]$

เกรตเตอร์ตัวอย่างจะส่งออกค่าที่คืนมาจากฟังก์ชัน `count_init`

ขอบเขต

- Time limit: 2 seconds
- Memory limit: 256 MB