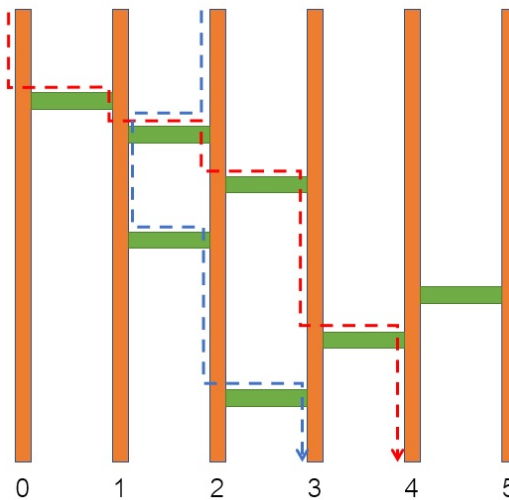




ลิง ลิง

เมื่อหลายปีก่อน ลิงน้อย "ต๋อย" ได้อาศัยอยู่ ณ อุทยานแห่งชาติผาแต้ม และโดนท่านเทพารักษ์ประจำอุทยาน ทดสอบสติปัญญาของลิงต๋อยอยู่บ่อยครั้ง ครั้นเมื่อลิงน้อยได้ยินข่าวเกี่ยวกับนครหรรษาที่มีโอบตามาเกิด จึงตั้งปณิธานว่าจะไม่ตกหลุมพรางกลัวยทิพย์ของท่านเทพารักษ์อีกต่อไป และจะลองไต่ไปลิ้มลองผลไม้จำปาตะ และ ม่วงมุด ณ ที่แห่งนั้นให้ได้ การที่ลิงน้อยจะเดินทางออกจากอุทยานแห่งชาติผาแต้มได้ ก็จำเป็นต้องลงจากเสาวิเศษจนมาถึงพื้น

เสาวิเศษมีลักษณะเป็นเสาที่เรียงต่อกันเป็นแถวหน้ากระดานจำนวน N ต้น เสาเหล่านั้นแต่ละต้นถูกกำกับด้วยหมายเลข 0 ถึง $N - 1$ เรียงจากซ้ายไปขวา เสาสองต้นใดที่เรียงติดกันอาจจะมียกไม้พิเศษเชื่อมระหว่างเสาสองต้นนั้นและอาจจะมียกไม้พิเศษเชื่อมมากกว่า 1 กิ่งก็ได้ แต่อย่างไรก็ตาม กิ่งไม้พิเศษที่ปรากฏจะอยู่ที่ความสูงแตกต่างกันเสมอ การออกจากอุทยานลิงน้อยต้องไต่เสาจากบนลงล่างเท่านั้น ไม่สามารถไต่ขึ้นกลับไปได้ และหากลิงน้อยกำลังไต่ลงเสาแล้วเจอกิ่งไม้พิเศษ กิ่งไม้พิเศษนั้นจะบังคับลิงน้อยให้ไปยังเสาข้างเคียงเสมอ



ตัวอย่างการไต่เสาของลิงน้อย แสดงดังรูปข้างต้น รูปนี้แสดงปริศนาที่มีเสาวิเศษ (สีน้ำตาล) จำนวน 6 ต้น และกิ่งไม้พิเศษ (สีเขียว) จำนวน 7 กิ่ง หากลิงน้อยเลือกที่จะเริ่มไต่เสาจากเสาที่ 1 เส้นทางไต่เสาก็เป็นไปตามเส้นประสีแดง ซึ่งเห็นได้ว่าเมื่อเริ่มไต่จากเสาที่ 0 ลงมา กิ่งไม้พิเศษจะบังคับลิงน้อยไปยังเสาที่ 1, 2, 3 และ 4 ลำดับ ซึ่งโดยวิธีนี้ลิงน้อยจะถึงพื้น ณ เสาที่ 4 แต่ถ้าหากลิงน้อยไต่เสาจากเสาที่ 2 เส้นทางไต่เสาก็เป็นไปตามเส้นประน้ำเงิน เห็นได้ว่าเมื่อเริ่มไต่จากเสาที่ 2 ลงมา กิ่งไม้พิเศษจะบังคับลิงน้อยไปยังเสาที่ 1 ย้อนกลับไปเสาที่ 2 และไปยังเสาที่ 3 ตามลำดับ วิธีที่สองนี้ ลิงน้อยจะถึงพื้น ณ เสาที่ 3

เมื่อลิงน้อยจากไป ท่านเทพารักษ์กลัวว่าหากไม่ได้เสกกิ่งไม้พิเศษบ่อย ๆ อาจจะหลงลืมการเสกคาถาได้ ดังนั้นจึงได้สร้างเกมส์ลิงน้อยลงนครขึ้น โดยการเสกให้มีลิงน้อยเป็นจำนวนมากประจำอยู่ ณ เสาทุก ๆ ต้น จากนั้นก็สุ่มกำหนดว่าลิงน้อยที่ไต่ลงมาจากเสาแต่ละต้น จะลงถึงพื้น ณ เสาต้นใด เมื่อได้ข้อมูลแล้ว ท่านเทพารักษ์ก็จะเสกกิ่งไม้พิเศษ เชื่อมระหว่างเสาต้นต่าง ๆ ที่อยู่ติดกัน ณ ความสูงที่แตกต่างกัน หากท่านเทพารักษ์เสกกิ่งไม้พิเศษได้ถูกต้องทุกตำแหน่ง ลิงน้อยที่ไต่ลงมาจากเสาแต่ละต้นก็จะถึงพื้น ณ เสาต้นที่กำหนดไว้ทุกประการ แต่อย่างไรก็ดี อาจจะมีบางครั้งที่ไม่ว่าอย่างไรท่านเทพารักษ์ก็ไม่สามารถเสกกิ่งไม้พิเศษ ให้เป็นไปตามเงื่อนไขที่กำหนดได้

งานของคุณคือ ให้เขียนโปรแกรมที่มีประสิทธิภาพ เพื่อหาว่าท่านเทพารักษ์จะสามารถเสกกิ่งไม้วิเศษเชื่อมต่อระหว่างเสาแล้วทำให้ลิงน้อยจำนวนมากที่ปรากฏอยู่ตามเสาต่าง ๆ สามารถลงสู่พื้นตามเสาที่กำหนดได้หรือไม่ และหากทำได้ ให้หาว่าจะต้องเสกกิ่งไม้วิเศษเป็นจำนวนที่น้อยที่สุดเป็นจำนวนกี่กิ่ง โดยท่านเทพารักษ์จะถามคำถามทั้งหมด Q ครั้ง แต่ละครั้งจะถามว่าหากพิจารณาเฉพาะลิงที่เริ่มต้นตั้งแต่เสาที่ L ถึงเสาที่ R และใส่กิ่งไม้ได้แค่ระหว่างเสาสองต้นที่อยู่ติดกันภายในช่วง $[L, R]$

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้

```
void init_monkeys(vector<int> P, int Q)
```

- ฟังก์ชันนี้จะถูกเรียกเพียงครั้งเดียว
- เวกเตอร์ P บ่งบอกว่า สำหรับจำนวนเต็ม $0 \leq i < N$ ลิงน้อยที่เริ่มไต่ลงมาจากเสาต้นที่ i จะต้องไต่ลงมาถึงพื้น ณ เสาต้นที่ $P[i]$

```
long long minimum_branches(int L, int R)
```

- ฟังก์ชันนี้จะต้องคืนค่าจำนวนกิ่งที่น้อยที่สุดที่จำเป็นต้องเสก เพื่อให้ลิงน้อยทุกตัวลงมาตรงตามเสาที่กำหนดให้ สำหรับลิงที่เริ่มต้นบนสุดตั้งแต่กิ่งที่ L_j ถึงกิ่งที่ R_j เมื่อ L_j และ R_j แทนค่าของตัวแปร L และ R ของคำถามที่ j สำหรับจำนวนเต็ม $0 \leq j < Q$
- หากเป็นไปได้ ให้คืนค่า -1

ขอบเขต

- $2 \leq N \leq 200\,000$
- $1 \leq Q \leq 10\,000$
- $0 \leq P[i] < N$ สำหรับทุกจำนวนเต็ม $0 \leq i < N$
- $0 \leq L_i \leq R_i < N$ สำหรับทุกจำนวนเต็ม $0 \leq j < Q$

ปัญหาย่อย

1. (4 คะแนน) $Q = 1, L_0 = 0, R_0 = N - 1$ และคำตอบคือ -1 หรือ 0 อย่างไม่อย่างหนึ่งเท่านั้น
2. (13 คะแนน) $Q = 1, L_0 = 0, R_0 = N - 1$ และ $N \leq 5$
3. (5 คะแนน) $Q = 1, L_0 = 0, R_0 = N - 1$ และ $P[i] = (i + 1) \bmod N$ สำหรับทุกจำนวนเต็ม $0 \leq i < N$
4. (9 คะแนน) $Q = 1, L_0 = 0, R_0 = N - 1$ และ $P[i] = N - i - 1$ สำหรับทุกจำนวนเต็ม $0 \leq i < N$
5. (25 คะแนน) $Q = 1, L_0 = 0, R_0 = N - 1$ และ $N \leq 1\,000$
6. (21 คะแนน) $Q = 1, L_0 = 0$ และ $R_0 = N - 1$
7. (5 คะแนน) คำตอบคือ -1 หรือ 0 อย่างไม่อย่างหนึ่งเท่านั้น
8. (18 คะแนน) ไม่มีเงื่อนไขเพิ่มเติม

ตัวอย่าง

ตัวอย่างที่ 1

```
init_monkeys([2, 2, 1], 2)
```

ต่อมามีการเรียก

```
minimum_branches(0, 2)
```

จะต้องคืนค่า -1

```
minimum_branches(1, 2)
```

จะต้องคืนค่า 1

ตัวอย่างที่ 2

```
init_monkeys([1, 0, 2], 1)
```

ต่อมามีการเรียก

```
minimum_branches(0, 2)
```

จะต้องคืนค่า 1

ตัวอย่างที่ 3

```
init_monkeys([1, 2, 0], 2)
```

ต่อมามีการเรียก

```
minimum_branches(0, 2)
```

จะต้องคืนค่า 2

```
minimum_branches(0, 1)
```

จะต้องคืนค่า -1

ตัวอย่างที่ 4

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

ต่อมามีการเรียก

```
minimum_branches(0, 3)
```

จะต้องคืนค่า 0

```
minimum_branches(1, 1)
```

จะต้องคืนค่า 0

```
minimum_branches(2, 3)
```

จะต้องคืนค่า 0

ตัวอย่างที่ 5

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

ต่อมามีการเรียก

```
minimum_branches(0, 3)
```

จะต้องคืนค่า 6

เกรตเตอร์ตัวอย่าง

เกรตเตอร์ตัวอย่างอ่านข้อมูลนำเข้าดังรูปแบบต่อไปนี้:

- บรรทัดที่ 1: N
- บรรทัดที่ 2: $P[0] \ P[1] \ P[2] \ \dots \ P[N-1]$
- บรรทัดที่ 3: Q
- บรรทัดที่ $4 + i$: $L_i \ R_i$

เกรตเตอร์ตัวอย่างจะส่งออกข้อมูลที่คืนค่ามาจากฟังก์ชัน `minimum_branches` ทั้งหมดทั้ง Q รอบ โดยส่งออกทีละบรรทัด

ข้อจำกัด

- Time limit: 7.5 seconds
- Memory limit: 512 MB

