รางเหรียญ (Coin)

มีรางเหรียญอยู่ 1 รางที่มีช่องทั้งหมด N ช่องซึ่งจะมีหมายเลขประจำช่องคือ 0 ถึง $N-1$ สำหรับช่องที่ i จะมีความกว้างของช่องคือ S_i เนื่องจากเพื่อนของคุณนั้นเป็นคนติดวัฒนธรรมการละเล่นพื้นบ้านของประเทศข้างเคียงนั่นคือการกลิ้งเหรียญ เขาจึงฝากให้คุณทดสอบการละเล่นดังกล่าว

การละเล่นกลิ้งเหรียญนั้นจะเริ่มโดยการเล่นทั้งหมด Q ครั้งโดยที่แต่ละครั้งจะวางเหรียญก่อนช่อง A_i และจะปล่อยให้กลิ้งมาเรื่อยๆ โดยเริ่มจากช่อง A_i ไปยังช่องที่ N โดยที่เหรียญแต่ละเหรียญนั้นจะมีคุณสมบัติอยู่ 2 ประการคือ

1. จะสามารถเด้งได้ทั้งหมด B_i ครั้งโดยจะเด้งก็ต่อเมื่อมีการกลิ้งผ่านช่อง**ที่มีขนาดใหญ่กว่า**เหรียญและยังเด้งไม่ครบ
2. จะมีขนาดเหรียญ C_i

ในการละเล่นแต่ละครั้งจะจบก็ต่อเมื่อเหรียญกลิ้งไปสุดรางหรือเด้งจนครบแล้วตกช่องใดๆ ซึ่งเพื่อนของคุณเล็งเห็นว่ากติกาดั้งเดิมนั้นอาจจะไม่ท้าทายพอ เขาจึงให้เทปที่แบ่งแยกไม่ได้ขนาด L มาหนึ่งชิ้นสำหรับทุกการละเล่นซึ่งจะสามารถแปะเพื่อปิดช่องในช่วงติดกันที่**ขนาดไม่เกิน** L จำนวน 1 ช่วง พร้อมกับฝากคำถามสุดท้าทายว่าทุกรูปแบบการแปะเทปที่เป็นไปได้จะทำให้เหรียญกลิ้งไปไกลที่สุดที่หมายเลขใด หากกลิ้งไปสุดรางให้ตอบ N

พิจารณาตัวอย่างต่อไปนี้ที่ $N = 7$ และขนาดช่องแต่ละช่องเป็นดังนี้

$$1, 2, 1, 3, 1, 3, 3$$

สมมติว่า $Q = 3, L = 2$ จะมีการละเล่น 3 ครั้ง ดังนี้

- ถ้าการละเล่นครั้งแรกระบุ $A_0 = 1, B_0 = 1, C_0 = 2$ คำตอบคือ 7 โดยเริ่มต้นกลิ้งจากช่องขนาด 2 เด้งที่ช่องหมายเลข 3 และแปะเทปที่ช่อง $5 - 6$ ทำให้กลิ้งไปสุดราง
- ถ้าการละเล่นครั้งที่สองระบุ $A_1 = 1, B_1 = 0, C_1 = 1$ คำตอบคือ 3 โดยเริ่มต้นกลิ้งจากช่องขนาด 2 และแปะเทปที่ช่อง $1 - 2$ ทำให้กลิ้งไปตกที่ช่องหมายเลข 3
- ถ้าการละเล่นครั้งที่สามระบุ $A_2 = 1, B_2 = 1, C_2 = 1$ คำตอบคือ 5 โดยเริ่มต้นกลิ้งจากช่องขนาด 2 เด้งที่ช่องหมายเลข 1 และแปะเทปที่ช่อง $2 - 3$ ทำให้กลิ้งไปตกที่ช่องหมายเลข 5

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันดังต่อไปนี้

```
void initialize(int N, vector<int> Rail, int L)
```

- ฟังก์ชันนี้จะถูกเรียกเพียงครั้งเดียวสำหรับปัญหาย่อย
- $Rail[i]$ จะมีค่าเท่ากับ S_i สำหรับ $0 \leq i \leq N-1$
- ฟังก์ชันนี้ไม่ต้องคืนค่า

```
vector<int> max_dist(vector<vector<int> > Coins)
```

- ฟังก์ชันนี้จะถูกเรียกเพียงครั้งเดียวสำหรับปัญหาย่อย
- $Coins[i][0] = A_i$, $Coins[i][1] = B_i$, $Coins[i][2] = C_i$ สำหรับ $0 \leq i \leq Q-1$
- ฟังก์ชันนี้จะต้องคืนค่า vector ขนาดเท่ากับ Q โดยที่ช่องที่ i จะต้องคืนค่าหมายเลขช่องที่เหรียญกลิ้งไปได้ไกลที่สุดในคำถามที่ i

ขอบเขต

- $2 \leq N, Q \leq 2 \times 10^5$
- $1 \leq S_i, C_i \leq 10^9$
- $0 \leq A_i \leq N-1$
- $0 \leq B_i \leq N$

ปัญหาย่อย

1. (7 คะแนน) $N, Q \leq 3000$
2. (11 คะแนน) C_i เท่ากันหมด, $L \leq 1$
3. (13 คะแนน) C_i เท่ากันหมด
4. (9 คะแนน) S_i แตกต่างกันไม่เกิน 10 ค่า
5. (17 คะแนน) $B_i = 0$
6. (11 คะแนน) $L = 0$
7. (11 คะแนน) $L \leq 10$
8. (21 คะแนน) ไม่มีเงื่อนไขเพิ่มเติม

ตัวอย่าง

ตัวอย่างที่ 1

```
initialize(7, [1, 2, 1, 3, 1, 3, 3], 2)
```

หลังจากนั้นจะมีการเรียก `max_dist` 1 ครั้ง

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

จะคืนค่า `[7, 3, 5]`

ตัวอย่างที่ 2

```
initialize(7, [1, 2, 3, 4, 3, 2, 1], 0)
```

หลังจากนั้นจะมีการเรียก `max_dist` 1 ครั้ง

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

จะคืนค่า `[7, 3]`

เกรดเดอร์ตัวอย่าง

เกรดเดอร์ตัวอย่างจะมีการรับข้อมูลดังนี้

- บรรทัดที่ 1: $N\ Q\ L$
- บรรทัดที่ 2: $S_0\ S_1\ \ldots S_{N-1}$
- บรรทัดที่ 3 ถึง $Q+2$: $A_i\ B_i\ C_i$

ข้อจำกัด

- Time limit: 1.5 seconds
- Memory limit: 64 MB