



## แฟ่งตัวในรถไฟ

รถไฟฟหุรหามีตู้หังลัน  $N$  ตู้ แต่ละตู้มีสองชั้น ปกติทุกคนจะอยู่หือชั้นล่าง ชั้นบนจะมีทางเชื่อมระหว่างตู้แต่จะเปิดก็ต่อเมื่อมีงานเลี้ยงเท่านั้น คุณอยู่ปลายขบวน คือตู้ที่ 0 ส่วนตู้ที่  $N - 1$  คือหัวขบวนและเครื่องจักรอยู่หือตู้หั้น คุณต้องการเดินทางไปตู้ที่  $N - 1$  แต่หนทางเดียวที่จะเดินไปมาระหว่างตู้ได้คือรอให้มีงานเลี้ยงจัดชั้นเท่านั้น

ตลอดการเดินทางนาน  $M$  วันทุกวันจะมีงานเลี้ยงจัดชั้นหนึ่งงาน โดยแต่ละงานจะจัดหือตู้รถไฟที่ติดกัน กล่าวคือ งานในวันที่  $i$  เมื่อ  $0 \leq i \leq M - 1$  จะจัดในรถไฟตู้ที่  $s[i]$  ถึงตู้ที่  $t[i]$  (รวม  $s[i]$  และ  $t[i]$  ด้วย) คนที่เข้าร่วมงานจะเดินไปมาระหว่างตู้ดังกล่าวได้ แต่จะออกไปนอกขอบเขตหั้นไม่ได้ ถ้าในวันหั้นคุณอยู่หือชั้นล่างของตู้รถไฟที่อยู่ในงานเลี้ยง (หมายเลขตั้งแต่  $s[i]$  ถึง  $t[i]$ ) คุณจะสามารรถเดินขึ้นไปชั้นบนและร่วมงาน จากนั้นเดินไปมาระหว่างตู้ที่อยู่ในงานหั้น ๆ และแอบแฟ่งตัวลงมาหือชั้นล่างของตู้ใดก็ได้ในงานหั้นเมื่อลันวัน

งานเลี้ยงแต่ละงานมีค่าบัตรผ่านเข้างาน กล่าวคือ งานในวันที่  $i$  มีค่าบัตร  $c[i]$  บาท คุณต้องการเดินทางจากตู้ที่ 1 ไปถึงตู้ที่  $N$  โดยจ่ายเงินน้อยที่สุด โดยสนใจแค่ว่าจะต้องเดินทางไปถึงตู้ที่  $N$  จะเป็นวันใดก็ได้ คุณไม่จำเป็นต้องเข้าร่วมทุกงานเลี้ยง แม้ว่างานเลี้ยงจะจัดหือชั้นสองของตู้หือคุณอยู่ในวันดังกล่าวก็ตาม

พิจารณาตัวอย่างต่อไปนี้ หือ  $N = 7$  และ  $M = 7$  และตารางด้านล่างแสดงข้อมูลงานเลี้ยง

วันที่ ( $i$ )	ตู้เริ่มต้น $s[i]$	ตู้สุดท้ายของงาน $t[i]$	ค่าบัตร $c[i]$
0	0	2	10
1	1	3	50
2	2	6	200
3	2	4	30
4	4	6	200
5	3	6	10
6	0	6	1000

คุณจะไปยังตู้ที่ 6 โดยใช้เงินน้อยที่สุดโดยเข้าร่วมงานเลี้ยงในวันที่ 0 (จ่ายเงิน 10 บาท) จากนั้นเดินทางงานเลี้ยงแล้วลงไปยังตู้ที่ 2 รอจนถึงวันที่ 3 และเข้าร่วมงานเลี้ยงในวันดังกล่าว (จ่ายเงิน 30 บาท) จากนั้นเดินในตู้งานเลี้ยงแล้วลงไปยังตู้ที่ 3 หรือ 4 ก็ได้ เพื่อรอเข้างานวันที่ 5 (จ่ายเงิน 10 บาท) และเดินทางผ่านตู้ในงานเลี้ยงเพื่อไปลงหือตู้ที่ 6 ตามต้องการ รวมเสียเงินค่าเข้างาน  $10 + 30 + 10 = 50$  บาท

## รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้

```
int find_min_cost(int N, int M,  
                  vector<int>& s, vector<int>& t,  
                  vector<int>& c)
```

- ฟังก์ชันนี้จะถูกเรียกหนึ่งครั้ง
- จะระบุจำนวนตู้รถไฟ ( $N$ ) และจำนวนวันที่มีการจัดงานเลี้ยง ( $M$ )
- จะระบุข้อมูลของงานในวันที่  $i$  เมื่อ  $0 \leq i \leq M - 1$  โดย  $s[i]$  เป็นหมายเลขเริ่มต้นของตู้รถไฟของงาน และ  $t[i]$  เป็นหมายเลขสุดท้ายของตู้รถไฟในงาน ค่าบัตรเข้างานคือ  $c[i]$  บาท
- ฟังก์ชันจะต้องคืนค่าค่าใช้จ่ายน้อยที่สุดที่เป็นไปได้ในการไปให้ถึงตู้ที่  $N - 1$  ถ้าไม่สามารถทำได้ให้ตอบ  $-1$

## ขอบเขต

- $1 \leq N \leq 200\,000$ ,  $1 \leq M \leq 200\,000$
- $0 \leq s[i] \leq t[i] \leq N - 1$ , สำหรับทุก ๆ  $0 \leq i \leq M - 1$
- $1 \leq c[i] \leq 3\,000$  สำหรับทุก ๆ  $0 \leq i \leq M - 1$

## ปัญหาย่อย

1. (10 คะแนน)  $N \leq 1\,000$ ,  $M \leq 1\,000$
2. (15 คะแนน)  $M \leq 20\,000$
3. (18 คะแนน)  $s[i] \leq s[i + 1]$ ,  $t[i] \leq t[i + 1]$  สำหรับ  $0 \leq i < M - 1$
4. (18 คะแนน)  $c[i] = 1$
5. (39 คะแนน) ไม่มีเงื่อนไขเพิ่มเติม

## ตัวอย่าง

จากตัวอย่างข้างต้น การเรียกฟังก์ชัน `find_min_cost` จะเป็นดังนี้

```
find_min_cost(7, 7,  
              [0, 1, 2, 2, 4, 3, 0],  
              [2, 3, 6, 4, 6, 6, 6],  
              [10, 50, 200, 30, 200, 10, 1000])
```

ฟังก์ชันที่ทำงานถูกต้องจะคืนค่า 50

## เกรตเตอร์ตัวอย่าง

เกรตเตอร์ตัวอย่างอ่านข้อมูลดังนี้

- บรรทัดที่ 1:  $N, M$
- บรรทัดที่  $2, \dots, M + 1$ :  $s[i], t[i]$  และ  $c[i]$

เกรดเดอร์จะพิมพ์ค่าที่คืนจากฟังก์ชัน `find_min_cost`

## ข้อจำกัด

- Time limit: 1 second
- Memory limit: 512 MB