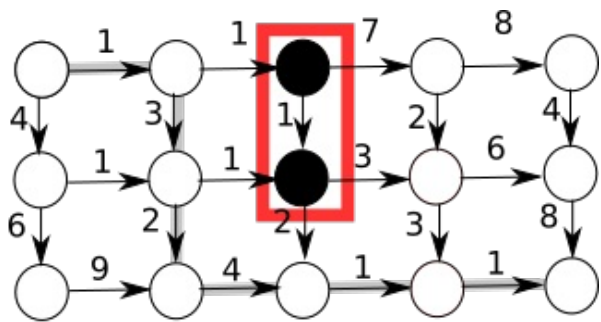
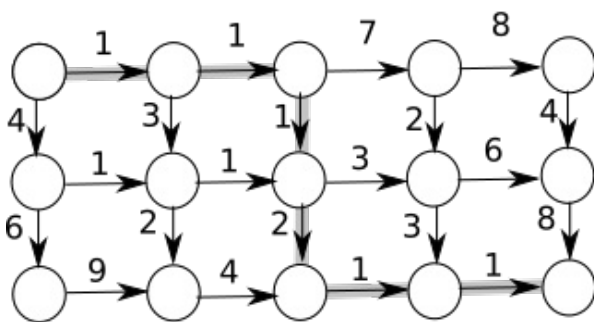




กีดขวาง

เมืองแห่งหนึ่งมีโครงสร้างเป็นตารางกริดขนาด $N \times M$ แยก ($4 \leq N \cdot M \leq 200\,000$) กล่าวคือ มีถนนเดินทางเดียวจำนวน N เส้น รุ่งจากแนวซ้ายไปขวา เรียกเป็นถนนหมายเลข 0 ถึง $N - 1$ และมีซอยที่เดินทางได้ทางเดียวจำนวน M ซอย รุ่งจากบนลงล่าง เรียกเป็นซอย 0 ถึง $M - 1$ ตัวอย่างของเมืองที่ $N = 3$ และ $M = 5$ แสดงดังรูปด้านล่าง (ซ้าย)

เราจะเรียกจุดที่ถนนและซอยตัดกันว่าเป็นแยก โดยแยกที่เกิดจากถนนที่ i พบกับซอยที่ j จะเรียกว่าแยก (i, j)



การเดินทางระหว่างถนนและซอยที่เดินทางเดียวที่เชื่อมระหว่างสองแยกจะใช้เวลาอาจจะแตกต่างกัน บ้านของคุณอยู่ที่แยก $(0, 0)$ คุณต้องการไปโรงเรียนที่แยก $(N - 1, M - 1)$ โดยใช้เวลาน้อยที่สุด เวลาในการเดินทางจากบ้านไปยังโรงเรียนจะเท่ากับเวลารวมของถนนและซอยที่คุณต้องเดินผ่าน ในรูปด้านบน ซ้าย เส้นทางที่ใช้เวลาน้อยที่สุดใช้เวลา 7 หน่วย (แสดงเป็นสีเทาด้านหลัง)

ในแต่ละวัน เมืองดังกล่าวจะทำการปรับปรุงแยกบางแยก ทำให้ไม่สามารถเดินทางผ่านได้ ในตัวอย่างด้านขวาแสดงแยกที่มีการปรับปรุง การปรับปรุงจะเท่ากับเซตของแยกที่ระบุด้วยค่า r และ c ที่จะหมายความว่ามีการปิดแยก (r', c) สำหรับทุก ๆ $0 \leq r' \leq r$ สังเกตว่าการปิดแยกนี้จะไปตามแนวซอยจะปิดจากขอบด้านบนของเมืองเสมอ

แม้ว่าแยกจะปิดไปอย่างไรก็ตาม คุณก็ยังต้องการเดินทางไปยังโรงเรียนให้ใช้เวลาน้อยที่สุดอยู่ดี ในตัวอย่างข้างต้นคุณสามารถปรับทางเดินไปโรงเรียนให้ใช้เวลา 12 หน่วย (รูปขวา)

ในโจทย์ข้อนี้จะมีคำถามทั้งสิ้น Q คำถาม คำถามแต่ละคำถามเป็นอิสระต่อกัน

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้

```
void initialize(int N, int M, int Q,  
               vector<vector<int>>& rowlen,  
               vector<vector<int>>& collen)
```

- ฟังก์ชันนี้จะถูกเรียกหนึ่งครั้ง
- ฟังก์ชันนี้จะระบุจำนวนถนน (N) และจำนวนซอย (M)
- เวลาในการเดินทางบนถนนแต่ละเส้นเมื่อเดินทางระหว่างซอยที่ติดกันจะระบุใน `rowlen` กล่าวคือ `rowlen` จะเป็นเวกเตอร์ขนาด N ของเวกเตอร์ขนาด $M - 1$ ที่ `rowlen[i][j]` ระบุเวลาที่ใช้ในการเดินทางระหว่างแยก (i, j) ไปยัง $(i, j + 1)$ เมื่อ $0 \leq i \leq N - 1$ และ $0 \leq j \leq M - 2$
- เวลาในการเดินทางบนซอยแต่ละซอยเมื่อเดินทางระหว่างถนนที่ติดกันจะระบุใน `collen` กล่าวคือ `collen` จะเป็นเวกเตอร์ขนาด $N - 1$ ของเวกเตอร์ขนาด M ที่ `collen[i][j]` ระบุระยะเวลาในการเดินทางระหว่างแยก (i, j) ไปยัง $(i + 1, j)$ เมื่อ $0 \leq i \leq N - 2$ และ $0 \leq j \leq M - 1$

คุณต้องเขียนฟังก์ชันต่อไปนี้เพื่อตอบคำถาม

```
int ask(int r, int c)
```

- ฟังก์ชันนี้จะถูกเรียกทั้งสิ้น Q ครั้งเพื่อเป็นการระบุว่าคุณ ๆ แยก (r', c) ที่ $0 \leq r' \leq r$ จะถูกปิดไม่ให้เดินทางผ่าน
- ฟังก์ชันจะต้องคืนจำนวนเต็มแทนเวลาที่น้อยที่สุดในการเดินทางจาก $(0, 0)$ ไปยัง $(N - 1, M - 1)$ ถ้าไม่สามารถไปได้ให้ตอบ -1

ขอบเขต

- $2 \leq N \leq 100\,000$, $2 \leq M \leq 100\,000$, $4 \leq N \cdot M \leq 200\,000$.
- $Q \leq 100\,000$
- $1 \leq \text{rowlen}[i][j] \leq 10\,000$ (for all $0 \leq i \leq N - 1, 0 \leq j < M - 1$)
- $1 \leq \text{collen}[j][i] \leq 10\,000$ (for all $0 \leq j \leq M - 1, 0 \leq i < N - 1$)

ปัญหาย่อย

1. (11 คะแนน) $Q = 1$
2. (12 คะแนน) $N \cdot M \leq 1\,000$, $Q \leq 1\,000$
3. (17 คะแนน) $r = 0$
4. (17 คะแนน) $r = N - 2$
5. (43 คะแนน) ไม่มีเงื่อนไขเพิ่มเติม

ตัวอย่าง

จากตัวอย่างข้างต้น การเรียกฟังก์ชัน `initialize` เริ่มต้นจะเป็นดังนี้

```
initialize(3, 5, 2,
          [[1, 1, 7, 8],
           [1, 1, 3, 6],
           [9, 4, 1, 1]],
          [[4, 3, 1, 2, 4],
           [6, 2, 2, 3, 8]])
```

มีการระบุว่าจะมีคำถามทั้งสิ้น 2 คำถาม

- เมื่อเกรตเดอร์ถาม `ask(0, 4)` ฟังก์ชันจะต้องคืนค่า 7
- เมื่อเกรตเดอร์ถาม `ask(1, 2)` ฟังก์ชันจะต้องคืนค่า 12

เกรตเดอร์ตัวอย่าง

เกรตเดอร์ตัวอย่างอ่านข้อมูลดังนี้

- บรรทัดที่ 1: N, M, Q
- บรรทัดที่ $2, \dots, N + 1$: `rowlen[i][0] rowlen[i][1] ... rowlen[i][M - 2]`
- บรรทัดที่ $N + 2, \dots, 2N$: `collen[i][0] collen[i][1] ... collen[i][M - 1]`
- บรรทัดที่ $2N + 1, \dots, 2N + Q$: $r\ c$ ของแต่ละคำถาม

เกรตเดอร์จะพิมพ์ค่าที่คืนจากฟังก์ชัน `ask`

ข้อจำกัด

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