

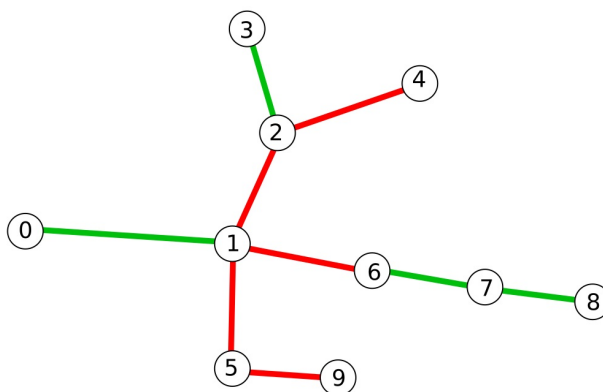


## ค่ารถไฟ (expensive)

เมือง  $N$  เมืองเชื่อมถึงกันทั้งหมดด้วยรถไฟ  $N - 1$  สาย รถไฟแต่ละสายจะเชื่อมเมืองสองเมือง เนื่องจากแนวคิดพื้นฐานในการเชื่อมต่อระบบรถไฟนี้จึงรับประกันว่าเมืองที่เชื่อมกับรถไฟสายเดียวเท่านั้นจะมีไม่เกิน  $L$  เมือง

ระบบรถไฟนี้มีบริษัทเดินรถสองบริษัทคือบริษัท A และบริษัท B รถไฟแต่ละสายจะเดินรถด้วยบริษัทบริษัทเดียวเท่านั้น ในช่วงเวลาหนึ่ง ๆ แต่ละบริษัทจะคิดค่าโดยสารสายรถไฟที่ตนเองเดินทุกสายราคาเท่ากันทั้งหมด เนื่องจากการเดินรถโดยเอกชนที่จะตั้งราคาเดินรถอย่างไรก็ได้และคนก็แทบไม่มีทางเลือกอื่น ชีวิตก็ดูจะอัดอัดพอทน อย่างไรก็ตาม เพื่อคุ้มครองผู้บริโภค หน่วยงานจึงอยากทราบว่าด้วยค่าโดยสารของที่ทั้งสองบริษัทระบุมาคนที่ต้องเดินทางและเสียค่าใช้จ่ายสูงสุด จะต้องเสียเงินกี่บาท

พิจารณาตัวอย่างต่อไปนี้ ที่  $N = 10$  สายรถไฟของบริษัท A แสดงเป็นสีแดง สายรถไฟของบริษัท B แสดงเป็นสีเขียว



ถ้าทั้งบริษัท A และ B ตั้งราคาค่าโดยสาร 1 บาท ค่าโดยสารแพงสุดคือการเดินทางระหว่างเมือง 3 ไปยังเมือง 8 คือ 5 บาท

อย่างไรก็ตามถ้า บริษัท A ปรับราคาเป็น 5 บาท ส่วนบริษัท B ใช้ราคา 1 บาทเท่าเดิม ค่าโดยสารที่แพงที่สุดจะเป็นการเดินทางระหว่างเมือง 4 ไปยังเมือง 9 คือ 20 บาท

คุณจะต้องตอบคำถามลักษณะนี้จำนวน  $Q$  ครั้ง

## รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันสองฟังก์ชันต่อไปนี้

```
void initialize(int N, int Q, vector<pair<int,int>> R, vector<int> C)
```

- ฟังก์ชันนี้จะถูกเรียกใช้ 1 ครั้ง

- อาร์เรย์  $R$  ระบุข้อมูลการเชื่อมต่อสายรถไฟ กล่าวคือสำหรับ  $i$  ที่  $0 \leq i < N - 1$  สายรถไฟที่  $i$  เชื่อมระหว่างเมือง  $R[i].first$  กับ  $R[i].second$
- อาร์เรย์  $C$  ระบุบริษัท กล่าวคือสำหรับ  $i$  ที่  $0 \leq i < N - 1$  สายรถไฟที่  $i$  จะเป็นของบริษัท A ถ้า  $C[i] = 1$  และจะเป็นของบริษัท B ถ้า  $C[i] = 2$

เกรดเดอร์จะเรียกฟังก์ชัน `most_expensive` อีก  $Q$  ครั้ง

```
long long most_expensive(int x, int y)
```

- ฟังก์ชันนี้จะถูกเรียกทั้งสิ้น  $Q$  ครั้ง
- คุณจะต้องตอบค่ารถไฟที่แพงที่สุดที่ผู้โดยสารจะต้องจ่าย ถ้าบริษัท A คิดค่าโดยสาร  $x$  บาทต่อสาย และบริษัท B คิดค่าโดยสาร  $y$  บาทต่อสาย

## เงื่อนไข

- $3 \leq N \leq 1\,000\,000$
- $1 \leq Q \leq 2\,000\,000$
- $2 \leq L \leq \min\{2\,000, N - 1\}$
- สำหรับทุกจำนวนเต็ม  $0 \leq i < N - 1$ 
  - $0 \leq R[i].first \leq N - 1$
  - $0 \leq R[i].second \leq N - 1$
  - $C[i] \in \{1, 2\}$
- $0 \leq x, y \leq 1\,000\,000\,000$

## ปัญหาย่อย

1. (4 คะแนน) สำหรับทุกจำนวนเต็ม  $0 \leq i < N - 1, C[i] = 1$
2. (5 คะแนน)  $L = 3$  และมีสถานีหนึ่งสถานีที่เชื่อมกับรถไฟสามสาย
3. (8 คะแนน)  $Q, N \leq 80$
4. (32 คะแนน)  $N \leq 2\,000$
5. (5 คะแนน)  $Q \leq 3$
6. (21 คะแนน)  $Q \leq 1\,000$
7. (25 คะแนน) ไม่มีเงื่อนไขเพิ่มเติม

## ตัวอย่าง

จากตัวอย่างข้างต้น เกรดเดอร์ตัวอย่างจะเรียกฟังก์ชัน `initialize` ดังนี้

```
initialize(10, 2, [[0, 1], [1, 2], [2, 4], [2, 3], [1, 6],
                  [6, 7], [7, 8], [5, 1], [5, 9]],
          [2, 1, 1, 2, 1, 2, 2, 1, 1])
```

หลังจากนั้นจะเรียกฟังก์ชัน `most_expensive` จำนวน 2 ครั้งดังนี้ ครั้งแรกจะเรียก

```
most_expensive(1, 1)
```

ซึ่งจะต้องตอบ 5 ส่วนครั้งที่สองจะเรียก

```
most_expensive(5, 1)
```

ซึ่งจะต้องตอบ 20

## เกรตเตอร์ตัวอย่าง

เกรตเตอร์ตัวอย่างอ่านข้อมูลนำเข้าในรูปแบบดังนี้

- บรรทัดที่ 1:  $N \ Q$
- บรรทัดที่  $2 + i$  ถึง  $2 + N - 2$ :  $R[i].first \ R[i].second \ C[i]$
- บรรทัดที่  $1 + N + j$  ถึง  $1 + N + Q - 1$ :  $x \ y$

เกรตเตอร์ตัวอย่างจะพิมพ์ค่าที่คืนจากฟังก์ชัน `most_expensive`

## ขีดจำกัด

- Time limit: 5 seconds
- Memory limit: 512 MB