



## บริษัทขนส่งสินค้า (Company)

ประเทศแห่งหนึ่งมีบริษัท 100 000 บริษัท มีหมายเลข 1 ถึง 100 000 บางบริษัทเป็นบริษัทเล็ก ๆ ยังไม่ได้ไปตั้งที่เมืองไหน ประเทศนี้มีเมือง  $N$  เมือง แต่ละเมืองจะอนุญาตให้มีหนึ่งบริษัทมาตั้ง เมืองที่  $i$  จะมีบริษัทหมายเลข  $C_i$  ตั้งอยู่ เมื่อเริ่มต้นประเทศดังกล่าวมีถนนที่เชื่อมระหว่างสองเมืองทั้งหมด  $M$  เส้น

แน่นอนว่าสังคมปัจจุบัน การแข่งขันในทางธุรกิจเป็นไปในทางที่ดุเดือดอย่างมากจึงทำให้บริษัทต่าง ๆ มีการตั้งกลุ่มเพื่อขนส่งวัสดุ โดยจะเรียกว่า **กลุ่มการค้า** จำนวน  $P$  กลุ่ม กลุ่มที่  $j$  จะมีสมาชิกทั้งหมด  $P_j$  บริษัท โดยจะระบุเป็นหมายเลขบริษัท รับประกันว่าในแต่ละกลุ่มการค้า จะมีเฉพาะบริษัทที่ได้ไปตั้งอยู่ที่บางเมืองแล้วเท่านั้น บางบริษัทอาจจะอยู่ในหลายกลุ่มการค้าได้

สำหรับกลุ่มการค้าใด ๆ จะประสบความสำเร็จจะต้องสามารถ **ค้าขายกันได้** โดยจะต้องเข้าเงื่อนไขทั้งหมดดังนี้

1. สามารถตั้งศูนย์รวมสินค้าไว้ที่หนึ่งในเมืองที่สมาชิกได้ไปตั้งอยู่ และสำหรับทุกบริษัทที่เป็นสมาชิกต้องสามารถเดินทางไปหาเมืองนั้นได้
2. หากบริษัทที่เป็นสมาชิกกลุ่มการค้านั้นมีอยู่หลายเมือง ทุกเมืองดังกล่าวต้องสามารถเดินทางไปศูนย์รวมสินค้าได้

สำหรับถนนในแต่ละเส้นจะถูกปรับปรุงตามลำดับการรับมา กล่าวคือถนนเส้นที่  $k$  จะเดินทางไม่ได้หลังจบวันที่  $k$  ไปแล้ว เมื่อ  $0 \leq k \leq M - 1$  ถ้าหากสำหรับแต่ละกลุ่มการค้าจะสามารถค้าขายกันได้ถึงวันที่เท่าไร หากสามารถค้าขายได้แม้ว่าจะไม่มีถนนให้ตอบ  $M$

## รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันดังต่อไปนี้:

```
void init(int N, int M, vector<int> C, vector<vector<int>> Road)
```

- ฟังก์ชันนี้จะถูกเรียกเพียงครั้งเดียวโดยตัวแปร  $N$  และ  $M$  แทนค่า  $N$  และ  $M$  ตามลำดับ
- เวกเตอร์ `Road` จะมีขนาด  $M$  และประกอบด้วยเวกเตอร์ที่มีขนาด 2 โดยที่ถนนเส้นที่  $i$  จะเชื่อมเมือง `Road[i][0]` และ `Road[i][1]` เมื่อ  $0 \leq i \leq M - 1$

```
vector<int> dissolve(vector<vector<int>> Group)
```

- ฟังก์ชันนี้จะถูกเรียกเพียงครั้งเดียว
- เวกเตอร์ `Group` จะมีขนาด  $P$  ในช่องที่  $j$  จะประกอบด้วยเวกเตอร์ที่มีขนาด  $P_j$  แทนบริษัทที่อยู่ในกลุ่มการค้าที่  $j$
- ฟังก์ชันนี้ต้องคืนค่าเวกเตอร์ขนาด  $P$  แทนวันสุดท้ายที่แต่ละกลุ่มการค้าจะสามารถค้าขายกันได้

## ขอบเขต

- $1 \leq N, P \leq 100\,000$
- $N - 1 \leq M \leq 200\,000$
- $1 \leq C_i \leq 100\,000$
- $1 \leq P_i \leq N$
- $1 \leq \sum P_i \leq 200\,000$
- รับประกันว่าเมืองทุกเมืองสามารถเดินทางไปหากันได้

## ปัญหาย่อย

1. (5 คะแนน)  $P = 1, N \leq 1\,000, M \leq 2\,000$
2. (8 คะแนน)  $\sum P_i \leq 10\,000, N \leq 1\,000, M \leq 2\,000$
3. (17 คะแนน)  $P = 1$
4. (9 คะแนน)  $P_j = 1, M = N - 1$  และค่า  $C_i$  ต่างกันไม่เกิน 100 ค่า
5. (14 คะแนน)  $P_j = 1, M = N - 1$
6. (7 คะแนน)  $M = N - 1$
7. (23 คะแนน)  $P_j = 1$
8. (17 คะแนน) ไม่มีเงื่อนไขเพิ่มเติม

## ตัวอย่าง

```
init(8, 8, [5, 1, 2, 2, 1, 3, 3, 4],  
[[1, 2], [1, 5], [7, 4], [5, 7], [2, 3], [7, 6], [4, 5], [0, 1]])
```

ต่อมาจะเรียกฟังก์ชัน `dissolve` 1 ครั้ง

```
dissolve([[1], [2], [3], [4], [1, 4], [1, 3, 4], [1, 2, 3, 4]])
```

จะคืนค่า `[1, 4, 3, 8, 1, 1, 0]`

## เกรตเตอร์ตัวอย่าง

เกรตเตอร์ตัวอย่างจะอ่านข้อมูลนำเข้าดังนี้

- บรรทัดที่ 1:  $N \ M \ P$
- บรรทัดที่ 2:  $C_0 \ C_1 \ C_2 \ \dots \ C_{N-1}$
- บรรทัดที่ 3 ถึง  $M + 2$ :  $Road[i][0] \ Road[i][1]$
- บรรทัดที่  $M + 3$  ถึง  $M + P + 2$ :  $P_j \ Group[k][0] \ Group[k][1] \ \dots \ Group[k][P_j - 1]$

เกรตเตอร์ตัวอย่างจะส่งออกมาที่ได้รับจากฟังก์ชัน `dissolve` 1 ครั้ง

## ข้อจำกัด

- Time limit: 1.25 seconds
- Memory limit: 512 MB