กุญแจ

สถาปนิกทิโมธีได้ออกแบบเกมหนีออกจากห้องเกมขึ้นมาใหม่ เกมนี้มีห้อง n ห้อง มีหมายเลขห้องกำกับตั้งแต่ 0 ถึง $n-1$ ในขณะเริ่มต้นเกม แต่ละห้องจะมีกุญแจอยู่ในห้อง ห้องละหนึ่งดอกเท่านั้น กุญแจแต่ละดอกมีชนิดกำกับอยู่ โดยชนิดระบุด้วยจำนวนเต็ม ซึ่งมีค่าระหว่าง 0 ถึง $n-1$ รวมหัวท้าย กุญแจที่อยู่ในห้องที่ i (โดยที่ $0 \leq i \leq n-1$) เป็นกุญแจชนิด $r[i]$ ทั้งนี้ ห้องหลายห้องอาจมีกุญแจชนิดเดียวกันได้ นั่นก็คือ ค่าของ $r[i]$ ไม่จำเป็นต้องแตกต่างกันทั้งหมด

นอกจากนี้ ยังมีทางเชื่อม**แบบสองทิศทาง**ทั้งสิ้น m ทางเชื่อม มีหมายเลขทางเชื่อมกำกับตั้งแต่ 0 ถึง $m-1$ ทางเชื่อม j (โดยที่ $0 \leq j \leq m-1$) เชื่อมระหว่างห้อง $u[j]$ กับ $v[j]$ ซึ่งแตกต่างกัน ห้องคู่หนึ่ง ๆ อาจถูกเชื่อมด้วยทางเชื่อมหลายทางเชื่อมได้

เกมนี้เล่นทีละคน โดยผู้เล่นจะเก็บกุญแจและเดินทางระหว่างห้องโดยใช้ทางเชื่อม เราจะกล่าวว่า ผู้เล่น**เดินทาง**ด้วยทางเชื่อม j เมื่อเขาใช้ทางเชื่อมนี้เพื่อเคลื่อนที่จากห้อง $u[j]$ ไปยังห้อง $v[j]$ หรือในทางกลับกันก็ได้ ผู้เล่นจะสามารถเดินทางด้วยทางเชื่อม j ได้ หากเขาได้เก็บกุญแจชนิด $c[j]$ ก่อนหน้านี้แล้วเท่านั้น

ในทุกขณะที่เล่นเกม ผู้เล่นจะอยู่ในห้องใดห้องหนึ่ง หากผู้เล่นอยู่ในห้อง x เขาจะสามารถกระทำการได้สองแบบ

- เก็บกุญแจที่อยู่ในห้อง x ซึ่งเป็นชนิด $r[x]$ (เว้นเสียแต่ว่าผู้เล่นจะได้เก็บกุญแจดอกนั้นแล้ว)
- เดินทางด้วยทางเชื่อม j เมื่อ $u[j] = x$ หรือ $v[j] = x$ โดยมีข้อแม้ว่าผู้เล่นจะต้องเคยเก็บกุญแจชนิด $c[j]$ มาก่อนหน้านี้แล้ว ทั้งนี้ ผู้เล่นจะ**ไม่**ทิ้งกุญแจใด ๆ ที่เขาเคยเก็บไป

ผู้เล่น**เริ่ม**เล่นเกมในห้อง s โดยไม่มีกุญแจใด ๆ เราจะกล่าวว่า ห้อง t **สามารถไปถึงได้**จากห้อง s ถ้าผู้เล่นที่เริ่มเกมในห้อง s สามารถใช้ลำดับของการกระทำการตามที่ได้กล่าวไปข้างต้น เพื่อไปให้ถึงห้อง t ได้

สำหรับแต่ละห้อง i (โดยที่ $0 \leq i \leq n-1$) ให้ $p[i]$ แทนจำนวนห้องที่สามารถไปถึงได้จากห้อง i ทิโมธีต้องการทราบเซตของหมายเลข i ที่จะให้ค่า $p[i]$ น้อยที่สุด จากบรรดาค่า i ในช่วง $0 \leq i \leq n-1$

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้:

```
int[] find_reachable(int[] r, int[] u, int[] v, int[] c)
```

- r: อาร์เรย์ความยาว n สำหรับแต่ละ i (โดยที่ $0 \leq i \leq n-1$) กุญแจในห้อง i เป็นกุญแจชนิด $r[i]$
- u, v: อาร์เรย์ความยาว m สองอาร์เรย์ สำหรับแต่ละ j (โดยที่ $0 \leq j \leq m-1$) ทางเชื่อม j เชื่อมระหว่างห้อง $u[j]$ กับห้อง $v[j]$
- c: อาร์เรย์ความยาว m สำหรับแต่ละ j (โดยที่ $0 \leq j \leq m-1$) ชนิดของกุญแจที่ต้องใช้เดินทางด้วยทางเชื่อม j คือชนิด $c[j]$

- ฟังก์ชันนี้ต้องคืนค่าเป็นอาร์เรย์ a ความยาว n สำหรับแต่ละ $0 \leq i \leq n-1$ ค่าของ $a[i]$ ต้องเป็น 1 ถ้า $p[i] \leq p[j]$ สำหรับทุกค่า j ในช่วง $0 \leq j \leq n-1$ มิฉะนั้น ค่าของ $a[i]$ ต้องเป็น 0

ตัวอย่าง

ตัวอย่างที่ 1

พิจารณาการเรียกฟังก์ชันต่อไปนี้:

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

ถ้าผู้เล่นเริ่มเกมที่ห้อง 0 เขาสามารถใช้ลำดับของการกระทำการต่อไปนี้:

ห้องปัจจุบัน	การกระทำการ
0	เก็บกุญแจชนิด 0
0	เคลื่อนที่ด้วยทางเชื่อม 0 ไปยังห้อง 1
1	เก็บกุญแจชนิด 1
1	เคลื่อนที่ด้วยทางเชื่อม 2 ไปยังห้อง 2
2	เคลื่อนที่ด้วยทางเชื่อม 2 ไปยังห้อง 1
1	เคลื่อนที่ด้วยทางเชื่อม 3 ไปยังห้อง 3

ดังนั้น ห้อง 3 สามารถไปถึงได้จากห้อง 0 ในทำนองเดียวกัน เราสามารถสร้างลำดับของการกระทำเพื่อแสดงว่าทุกห้องสามารถไปถึงได้จากห้อง 0 จึงทำให้ $p[0] = 4$ ตารางต่อไปนี้แสดงห้องที่สามารถไปถึงได้จากห้องเริ่มต้นทุกห้อง:

ห้องเริ่มต้น i	ห้องที่สามารถไปถึงได้	$p[i]$
0	$[0, 1, 2, 3]$	4
1	$[1, 2]$	2
2	$[1, 2]$	2
3	$[1, 2, 3]$	3

ค่าที่ต่ำที่สุดของ $p[i]$ ในบรรดาห้องทั้งหมดคือ 2 ซึ่งค่านี้เกิดขึ้นสำหรับ $i = 1$ หรือ $i = 2$ ดังนั้น ฟังก์ชันนี้ต้องคืนค่า $[0, 1, 1, 0]$

ตัวอย่างที่ 2

```
find_reachable([0, 1, 1, 2, 2, 1, 2],
               [0, 0, 1, 1, 2, 3, 3, 4, 4, 5],
               [1, 2, 2, 3, 3, 4, 5, 5, 6, 6],
               [0, 0, 1, 0, 0, 1, 2, 0, 2, 1])
```

ตารางต่อไปนี้แสดงห้องที่สามารถไปถึงได้:

ห้องเริ่มต้น i	ห้องที่สามารถไปถึงได้	$p[i]$
0	$[0, 1, 2, 3, 4, 5, 6]$	7
1	$[1, 2]$	2
2	$[1, 2]$	2
3	$[3, 4, 5, 6]$	4
4	$[4, 6]$	2
5	$[3, 4, 5, 6]$	4
6	$[4, 6]$	2

ค่าที่ต่ำที่สุดของ $p[i]$ ในบรรดาห้องทั้งหมดคือ 2 ซึ่งค่านี้เกิดขึ้นสำหรับ $i \in \{1, 2, 4, 6\}$ ดังนั้น ฟังก์ชันนี้ต้องคืนค่า $[0, 1, 1, 0, 1, 0, 1]$

ตัวอย่างที่ 3

```
find_reachable([0, 0, 0], [0], [1], [0])
```

ตารางต่อไปนี้แสดงห้องที่สามารถไปถึงได้:

ห้องเริ่มต้น i	ห้องที่สามารถไปถึงได้	$p[i]$
0	$[0, 1]$	2
1	$[0, 1]$	2
2	$[2]$	1

ค่าที่ต่ำที่สุดของ $p[i]$ ในบรรดาห้องทั้งหมดคือ 1 ซึ่งค่านี้เกิดขึ้นสำหรับ $i = 2$ ดังนั้น ฟังก์ชันนี้ต้องคืนค่า $[0, 0, 1]$

ข้อจำกัด

- $2 \leq n \leq 300\,000$
- $1 \leq m \leq 300\,000$
- $0 \leq r[i] \leq n - 1$ สำหรับทุก $0 \leq i \leq n - 1$
- $0 \leq u[j], v[j] \leq n - 1$ และ $u[j] \neq v[j]$ สำหรับทุก $0 \leq j \leq m - 1$

- $0 \leq c[j] \leq n-1$ สำหรับทุก $0 \leq j \leq m-1$

ปัญหาย่อย

1. (9 คะแนน) $c[j] = 0$ สำหรับทุก $0 \leq j \leq m-1$ และ $n, m \leq 200$
2. (11 คะแนน) $n, m \leq 200$
3. (17 คะแนน) $n, m \leq 2000$
4. (30 คะแนน) $c[j] \leq 29$ (สำหรับทุก $0 \leq j \leq m-1$) และ $r[i] \leq 29$ (สำหรับทุก $0 \leq i \leq n-1$)
5. (33 คะแนน) ไม่มีข้อจำกัดเพิ่มเติม

เกรดเดอร์ตัวอย่าง

เกรดเดอร์ตัวอย่างอ่านข้อมูลนำเข้าในรูปแบบต่อไปนี้:

- บรรทัดที่ 1: n m
- บรรทัดที่ 2: $r[0]$ $r[1]$ $\ldots$ $r[n-1]$
- บรรทัดที่ $3+j$ (สำหรับทุก $0 \leq j \leq m-1$): $u[j]$ $v[j]$ $c[j]$

เกรดเดอร์ตัวอย่างจะพิมพ์สิ่งที่ `find_reachable` คืนค่ามาในรูปแบบต่อไปนี้:

- บรรทัดที่ 1: $a[0]$ $a[1]$ $\ldots$ $a[n-1]$