

## ติดต่อส่งต่อ (secret deal)

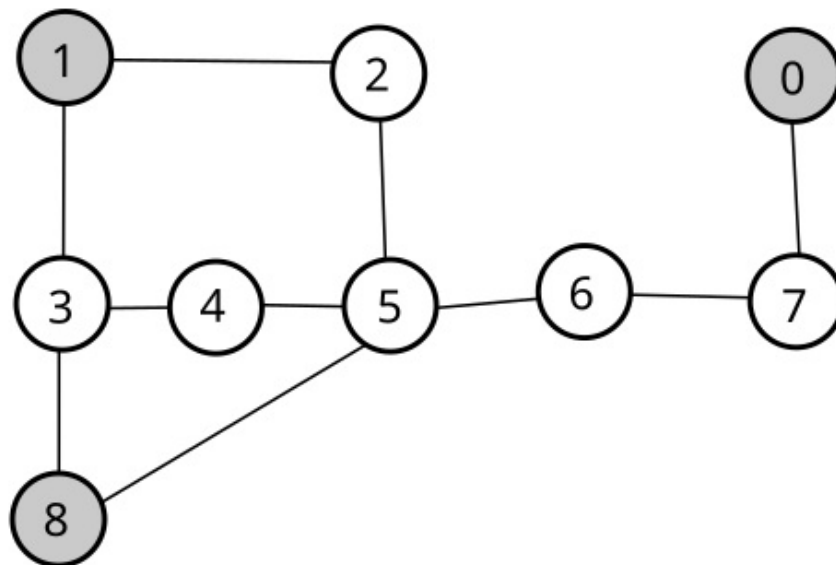
มีคน  $N$  คน ในคนกลุ่มนี้ มีคู่ของคนจำนวน  $M$  คู่ที่รู้จักกัน คู่ของคนที่รู้จักกันสามารถติดต่อส่งรับข้อมูลให้กันได้ (เป็นการสื่อสารสองทาง) รับประกันว่าระหว่างคู่ของคนใด ๆ จะมีวิธีการในการส่งรับข้อมูลระหว่างกันผ่านทาง การส่งต่อกันระหว่างคู่ของคนที่รู้จักกันเสมอ ในบรรดาคน  $N$  คนนี้ มีสมาชิกลับที่ต้องการจะดำเนินการเรื่อง ลึกลับบางอย่าง สมาชิกของสมาคมนี้มี  $K$  คน ( $2 \leq K \leq N$ )

การที่คนของสมาคมนี้จะติดต่อสื่อสารกันได้ จะทำผ่านทาง การส่งข้อมูลผ่านทางคู่ของคนที่รู้จักกัน (ที่มีระบุไว้  $M$  คู่ตอนต้น) เท่านั้น เนื่องจากข้อมูลเป็นข้อมูลลับ จึงไม่ค่อยอยากให้คนนอกสมาคมรู้เรื่องเกี่ยวกับการสื่อสาร นอกจากนี้อีก การจะไปขอดำเนินการสื่อสารผ่านคนนอกสมาคมนั้นก็ เป็นเรื่องที่ต้องเสียค่าใช้จ่ายเพื่อจ้างบุคคล เหล่านั้นอีกด้วย

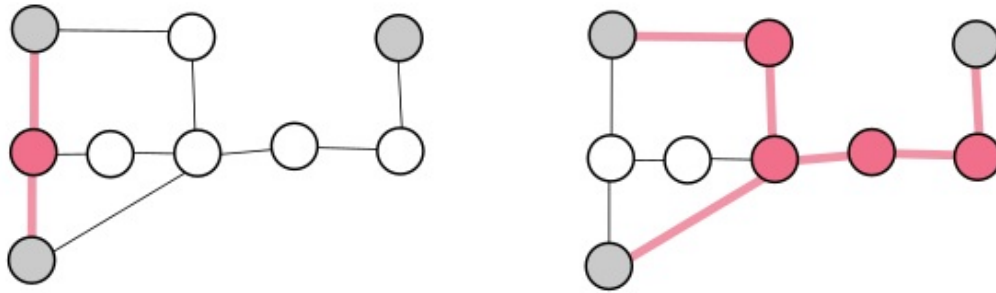
ในการจะดำเนินการเรื่องลึกลับเรื่องหนึ่ง มีความจำเป็นจะต้องให้สมาชิกของสมาคมลับจำนวน  $L$  คน สามารถติดต่อสื่อสารกันได้ ( $2 \leq L \leq 5$ )

คุณจะต้องเขียนโปรแกรมเพื่อคำนวณหาวิธีทำให้คนจำนวน  $L$  คนในสมาคมลับ ( $L$  คนใดก็ได้) สื่อสารกันได้ โดยการส่งข้อความผ่านคนนอกสมาคม ด้วยค่าใช้จ่ายน้อยที่สุด

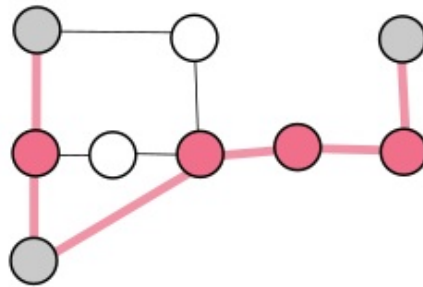
พิจารณาตัวอย่างต่อไปนี้ ที่  $N = 9$ ,  $M = 10$  และคนทุกคนนอกสมาคมเก็บค่าจ้างคนละ 1 บาท สมาชิกสมาคมลับแสดงเป็นวงกลมสีเทา เส้นแสดงคู่ที่ส่งรับข้อมูลกันได้



วิธีการส่งที่มีค่าใช้จ่ายน้อยที่สุดกรณีนี้ที่  $L = 2$  และ  $L = 3$  แสดงดังรูปด้านล่าง โดยเมื่อ  $L = 2$  ค่าใช้จ่ายคือ 1 บาท และเมื่อ  $L = 3$  ค่าใช้จ่ายคือ 4 บาท



พิจารณากรณีที่คนนอกสมาคมมีการเก็บค่าจ้างเปลี่ยนไป สมมติว่าคนที่ 2 เก็บค่าใช้จ่าย 10 บาท และคนที่ 6 เก็บค่าใช้จ่าย 3 บาท วิธีการส่งที่มีค่าใช้จ่ายน้อยที่สุดกรณีนี้  $L = 3$  แสดงดังรูปด้านล่าง และมีค่าใช้จ่ายเท่ากับ 6 บาท



## รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้ที่จะถูกเรียกใช้แค่หนึ่งครั้ง

```
int make_deal(int N, int M, int K, int L,
             vector<int> A, vector<int> B, vector<int> W)
```

- $N$  แทนจำนวนคน และ  $M$  แทนจำนวนคู่ของคนที่อยู่จักกัน
- เวกเตอร์  $A$  และ  $B$  มีขนาด  $M$  เป็นข้อมูลความสัมพันธ์ กล่าวคือ สำหรับ  $0 \leq i < M$ , คนที่  $A[i]$  รู้จักกับ  $B[i]$  สามารถสื่อสารกันได้
- เวกเตอร์  $W$  มีขนาด  $N$  โดย  $W[j]$  แทนค่าใช้จ่ายที่ต้องจ่ายถ้าจะสื่อสารผ่านคนที่  $j$  สำหรับสมาชิกในสมาคมกลับ ค่านี้จะเป็น 0 กล่าวคือ ถ้า  $j$  เป็นสมาชิกของสมาคมกลับ  $W[j] = 0$  จะมีค่าในเวกเตอร์  $W$  ที่เป็นศูนย์จำนวน  $K$  ค่าพอดี
- ฟังก์ชันจะต้องคืนค่าของค่าใช้จ่ายที่น้อยที่สุดที่จำเป็นในการจ้างคนนอกสมาคม เพื่อให้คนในสมาคม  $L$  คนสามารถติดต่อสื่อสารกันได้

## เงื่อนไข

- $2 \leq N \leq 100\,000$
- $1 \leq M \leq 250\,000$
- $0 \leq A[i] < N$  และ  $0 \leq B[i] < N$ , สำหรับทุก ๆ จำนวนเต็ม  $0 \leq i < M$
- $0 \leq W[j] \leq 10\,000$ , สำหรับทุก ๆ จำนวนเต็ม  $0 \leq j < N$  รับประกันว่าจะมีค่าที่เป็น 0 จำนวน  $K$  ค่าพอดี
- $2 \leq L \leq 5$

สำหรับข้อนี้ จะมีเงื่อนไขสำหรับ  $N$  และ  $M$  เป็นกรณีพิเศษ ขึ้นอยู่กับค่าของ  $L$  ดังตารางต่อไปนี้

$L$	เงื่อนไขที่รับประกัน
2	$N + M \leq 250\,000$
3	$N + M \leq 100\,000$
4	$N + M \leq 30\,000$
5	$N + M \leq 4\,000$

## ปัญหาย่อย

1. (5 คะแนน)  $L \leq 3; N \leq 1\,000; M \leq 2\,000$ .
2. (13 คะแนน)  $L = 2$ .
3. (8 คะแนน)  $L = 3; M = N - 1$
4. (10 คะแนน)  $L = 3$ .
5. (15 คะแนน)  $L = 4; N \leq 1\,000; M \leq 2\,000$ .
6. (10 คะแนน)  $L = 4$ .
7. (19 คะแนน)  $L = K$ .
8. (20 คะแนน) ไม่มีเงื่อนไขเพิ่มเติม

## ตัวอย่าง

พิจารณาตัวอย่างด้านบนที่  $L = 2$  และคนนอกสมาคมเก็บค่าใช้จ่ายคนละ 1 บาท เกรดเดอร์จะเรียกฟังก์ชันดังนี้

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

ฟังก์ชันจะต้องคืนค่า 1

ในตัวอย่างเดียวกัน ถ้า  $L = 3$  เกรดเดอร์จะเรียกฟังก์ชันดังนี้

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

ฟังก์ชันจะต้องคืนค่า 4

และในตัวอย่างสุดท้าย ที่คนที่ 2 เก็บค่าใช้จ่าย 10 บาทและคนที่ 6 เก็บค่าใช้จ่าย 3 บาท เมื่อ  $L = 3$  เกรดเดอร์จะเรียกฟังก์ชันดังนี้

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

ฟังก์ชันจะต้องคืนค่า 6

## เกรตเตอร์ตัวอย่าง

เกรตเตอร์ตัวอย่างจะอ่านข้อมูลดังนี้:

- บรรทัดที่ 1:  $N M K L$
- บรรทัดที่  $2 + i$ , สำหรับ  $0 \leq i < M$ :  $A[i] B[i]$
- บรรทัดที่  $2 + M$ :  $W[0] W[1] \dots W[N - 1]$

เกรตเตอร์ตัวอย่างจะพิมพ์ค่าที่คืนจากฟังก์ชัน `make_deal`

## ขีดจำกัด

- Time limit: 10 seconds
- Memory limit: 512 MB