

## Circulation with Lower Bounds (circlower)

กำหนดกราฟแบบมีทิศทาง  $G = (V, E)$  ซึ่งมีคุณสมบัติต่อไปนี้

- มีฟังก์ชัน  $n$  จาก  $E$  ไปยังเซตของจำนวนเต็มที่ไม่เป็นลบ เรียกว่า *capacity function*
- มีฟังก์ชัน  $l$  จาก  $E$  ไปยังเซตของจำนวนเต็มที่ไม่เป็นลบ เรียกว่า *lower bound function*

เราเรียกฟังก์ชัน  $f$  จาก  $E$  ไปยังเซตของจำนวนเต็มที่ไม่เป็นลบว่าเป็น circulation with lower bound ถ้าหากมันสอดคล้องกับสมบัติต่อไปนี้

- Flow Conservation: สำหรับ vertex  $v$  ทุกๆ vertex

$$\sum_{(v,w) \in E} f(v,w) - \sum_{(w,v) \in E} f(w,v) = 0$$

- Capacity Constraint:  $l(v,w) \leq f(v,w) \leq u(v,w)$  สำหรับ edge  $(v,w)$  ทุกๆ edge

จะเขียนฟังก์ชัน

```
bool circlower(int n, int m, Edge *edges, int *f)
```

โดยที่

- $n$  คือจำนวน vertex โดยที่  $3 \leq n \leq 1,000$
- $m$  คือจำนวน edge โดยที่  $2 \leq m \leq 50,000$
- Edge เป็น struct ที่มีนิยามดังนี้

```
struct Edge {  
    int v;  
    int w;  
    int u;  
    int l;  
}
```

โดยที่

- $v$  และ  $w$  เป็นหมายเลข vertex ซึ่งมีค่าตั้งแต่ 0 ถึง  $n-1$
- $u$  เป็น capacity ของ edge ซึ่งมีค่าตั้งแต่ 0 ถึง  $1,000,000$
- $l$  เป็น capacity ของ edge ซึ่งมีค่าตั้งแต่ 0 ถึง  $1,000,000$  และมีค่าไม่เกิน  $u$
- มีความหมายว่ามี direct edge จาก  $v$  ไปยัง  $w$  ซึ่งมี capacity เท่ากับ  $u$  และมี lower bound เท่ากับ  $l$

- edges เป็นอะเรย์ของ edge ซึ่งมีขนาด m ของพอร์ต
- f เป็นอะเรย์ขนาด m ของ ซึ่งฟังก์ชันของคุณจะต้องเขียนคำตอบให้ f โดยที่
  - $f[i]$  มีค่าเท่ากับ flow ที่อยู่ใน  $edge[i]$
  - f ต้องสอดคล้องกับ capacity constraint และ flow conservation
- ฟังก์ชันของจะคืนค่า true ถ้าหากว่ามี circulation ที่สอดคล้องกับเงื่อนไขทั้งหมด และคืนค่า false ถ้าหากไม่มี circulation ดังกล่าว หากฟังก์ชันของคุณไม่จำเป็นต้องเติมค่าในอะเรย์ f แต่อย่างใด อันนี้
- ให้เขียนโปรแกรมคำวายภาษา C++ เท่านั้น
- โปรแกรมของคุณต้องมีข้อความ `#include "circlower.h"` อยู่ในส่วนต้นของโปรแกรม
- โปรแกรมของคุณจะต้องทำงานเสร็จในเวลา 2 วินาที และใช้หน่วยความจำไม่เกิน 32 MB หลังจาก คอมไพล์รวมกับโค้ดตรวจสอบกรรมการด้วยแล้ว