



## ค่าทางด่วน

เมืองต่างๆในญี่ปุ่นถูกเชื่อมต่อกันด้วยเครือข่ายทางด่วน เครือข่ายนี้ประกอบด้วยเมือง  $N$  เมืองและทางด่วน  $M$  เส้น ทางด่วนแต่ละเส้นเชื่อมต่อกันสองเมืองที่เป็นคนละเมืองกัน และไม่มีทางด่วนเส้นใดเชื่อมคู่ของเมืองคู่เดียวกัน เมืองถูกกำกับด้วยหมายเลข  $0$  ถึง  $N - 1$  และทางด่วนถูกกำกับด้วยหมายเลข  $0$  ถึง  $M - 1$  คุณสามารถจะขับรถบนทางด่วนใด ๆ ได้ทั้งสองทิศทาง และคุณสามารถจะเดินทางจากเมืองใด ๆ ไปเมืองอื่นด้วยทางด่วนดังกล่าว

การขับผ่านทางด่วนแต่ละเส้นต้องเสียค่าผ่านทาง ค่าทางด่วนขึ้นอยู่กับสภาพปริมาณรถ ซึ่งอาจจะ น้อย หรือ มาก โดยเมื่อปริมาณรถน้อย ค่าทางด่วนจะมีค่า  $A$  เยน (หน่วยเงินญี่ปุ่น) เมื่อปริมาณรถมาก ค่าทางด่วนจะมีค่า  $B$  เยน ค่า  $A < B$  เสมอ คุณทราบว่าค่า  $A$  และ  $B$  เป็นเท่าไร

คุณมีเครื่องที่จะบอกค่าเดินทางทั้งหมดที่น้อยที่สุด ที่ผู้ขับรถระหว่างเมือง  $S$  และ  $T$  ( $S \neq T$ ) จะต้องจ่าย ภายใต้สภาพปริมาณรถของทางด่วนทุกสายที่คุณต้องกำหนดให้

อย่างไรก็ตาม เครื่องที่คุณมีเป็นแค่เครื่องต้นแบบ โดยค่า  $S$  และ  $T$  เปลี่ยนแปลงไม่ได้ (ฮาร์ดโค้ดในเครื่อง) และคุณไม่รู้ค่านี้มาก่อน คุณอยากทราบว่าค่า  $S$  และ  $T$  นี้คืออะไร คุณจึงวางแผนที่จะบอกสภาพปริมาณรถบนทางด่วนแบบต่าง ๆ ให้กับเครื่องนี้ และใช้ค่าผ่านทางที่เครื่องตอบให้มาอนุมานหาค่า  $S$  และ  $T$  เนื่องจากการใช้เครื่องในการคำนวณค่าผ่านทางในแต่ละครั้งมีราคาสูง คุณจึงไม่อยากใช้เครื่องนี้บ่อยๆ

## รายละเอียดการเขียนโปรแกรม

คุณต้องเขียนฟังก์ชันต่อไปนี้:

```
find_pair(int N, int[] U, int[] V, int A, int B)
```

- $N$ : จำนวนเมือง
- $U$  และ  $V$ : อาร์เรย์ความยาว  $M$  ซึ่ง  $M$  คือจำนวนเส้นทางด่วน โดยที่ทางด่วน  $i$  ( $0 \leq i \leq M - 1$ ) เชื่อมต่อเมือง  $U[i]$  กับเมือง  $V[i]$
- $A$ : ค่าทางด่วนเมื่อปริมาณรถน้อย
- $B$ : ค่าทางด่วนเมื่อปริมาณรถมาก
- ฟังก์ชันนี้จะถูกเรียกเพียงหนึ่งครั้งในแต่ละกรณีทดสอบ
- ให้สังเกตว่าค่าของ  $M$  คือความยาวของอาร์เรย์ และสามารถหาได้ตามที่ระบุในเอกสารข้อสังเกตการเขียนโปรแกรม

ฟังก์ชัน `find_pair` สามารถเรียกฟังก์ชันต่อไปนี้:

```
int64 ask(int[] w)
```

- ความยาวของ  $w$  ต้องเท่ากับ  $M$  หมายความว่า  $w$  บ่งบอกสภาพปริมาณรถบนทางด่วน
- สำหรับแต่ละ  $i$  ( $0 \leq i \leq M - 1$ )  $w[i]$  เป็นค่าปริมาณรถบนทางด่วน  $i$ , ค่า  $w[i]$  ต้องเป็นค่า 0 หรือ 1
  - $w[i] = 0$  หมายความว่าทางด่วน  $i$  มีปริมาณรถน้อย
  - $w[i] = 1$  หมายความว่าทางด่วน  $i$  มีปริมาณรถมาก
- ฟังก์ชันนี้คืนค่าผลรวมค่าทางด่วนที่น้อยที่สุดสำหรับการเดินทางระหว่างเมือง  $S$  และ  $T$  ภายใต้สภาพปริมาณรถที่ได้ระบุใน  $w$
- ฟังก์ชันนี้ถูกเรียกได้ไม่เกิน 100 ครั้ง (ในแต่ละกรณีทดสอบ).

ฟังก์ชัน `find_pair` สามารถจะเรียกฟังก์ชันต่อไปนี้เพื่อรายงานคำตอบ

```
answer(int s, int t)
```

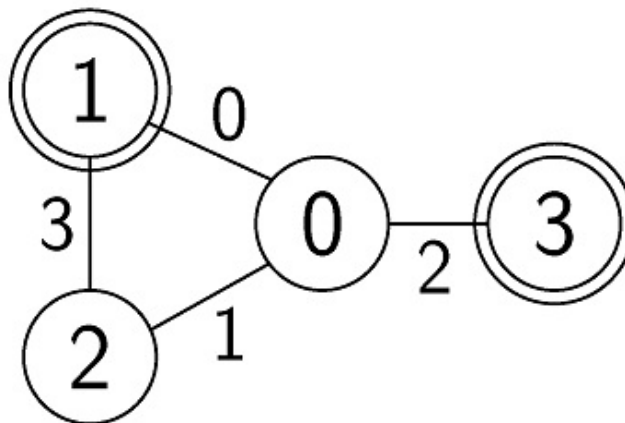
- $s$  และ  $t$  ต้องเป็นคูเมือง  $S$  และ  $T$  (ลำดับไม่สำคัญ).
- ฟังก์ชันนี้ต้องถูกเรียกเพียงหนึ่งครั้งเท่านั้น

ถ้าข้อบังคับข้างบนข้อใดข้อหนึ่งข้างบน โปรแกรมของคุณจะได้รับผลการตรวจเป็น **Wrong Answer**. นอกเหนือจากนั้น โปรแกรมของคุณจะได้รับผลการตรวจเป็น **Accepted** และคะแนนของคุณจะถูกคำนวณจากจำนวนการเรียกคำสั่ง `ask` (กรุณาดูปัญหาย่อย)

## ตัวอย่าง

ให้  $N = 4$ ,  $M = 4$ ,  $U = [0, 0, 0, 1]$ ,  $V = [1, 2, 3, 2]$ ,  $A = 1$ ,  $B = 3$ ,  $S = 1$ , และ  $T = 3$ .

เกรดเดอร์เรียก `find_pair(4, [0, 0, 0, 1], [1, 2, 3, 2], 1, 3)`



ในรูปด้านบน, เส้นเชื่อมที่กำกับด้วยหมายเลข  $i$  แสดงถึงทางด่วน  $i$

การเรียกฟังก์ชัน `ask` ที่เป็นไปได้บางส่วนเป็นต่อไปนี้:

การเรียกฟังก์ชัน	ค่าที่คืน
ask([0, 0, 0, 0])	2
ask([0, 1, 1, 0])	4
ask([1, 0, 1, 0])	5
ask([1, 1, 1, 1])	6

สำหรับการเรียก ask([0, 0, 0, 0]) นั้น ทางด่วนทุกเส้นมีปริมาณรถน้อย และ ค่าทางด่วนแต่ละเส้นจะเป็น 1 เส้นทางที่ถูกที่สุดจาก  $S = 1$  ถึง  $T = 3$  คือ  $1 \rightarrow 0 \rightarrow 3$  ค่าทางด่วนทั้งหมดสำหรับเส้นทางนี้คือ 2 ดังนั้นฟังก์ชันนี้จึงคืนค่า 2

สำหรับคำตอบที่ถูกต้อง, ฟังก์ชัน find\_pair ต้องเรียก answer(1, 3) หรือ answer(3, 1)

แฟ้ม sample-01-in.txt ในชุดแฟ้มแนบที่ถูกบีบอัดไว้ตรงกับตัวอย่างนี้ มีตัวอย่างอื่นถูกเก็บไว้ในแฟ้มแนบด้วย

## ข้อจำกัด

- $2 \leq N \leq 90\,000$
- $1 \leq M \leq 130\,000$
- $1 \leq A < B \leq 1\,000\,000\,000$
- ในแต่ละ  $0 \leq i \leq M - 1$ 
  - $0 \leq U[i] \leq N - 1$
  - $0 \leq V[i] \leq N - 1$
  - $U[i] \neq V[i]$
- $(U[i], V[i]) \neq (U[j], V[j])$  และ  $(U[i], V[i]) \neq (V[j], U[j])$ ,  $(0 \leq i < j \leq M - 1)$
- คุณสามารถเดินทางทางเมืองใดๆไปเมืองอื่นโดยใช้ทางด่วน
- $0 \leq S \leq N - 1$
- $0 \leq T \leq N - 1$
- $S \neq T$

ในปัญหานี้ เกรดเดอร์จะไม่พลิกแพลง กล่าวคือ  $S$  และ  $T$  จะถูกกำหนดเป็นค่าคงที่ไว้ตั้งแต่ที่เกรดเดอร์เริ่มต้นทำงาน และค่าของมันจะไม่เปลี่ยนไปในขณะที่โปรแกรมของคุณทำงาน

## ปัญหาย่อย

1. (5 คะแนน) หนึ่งใน  $S$  หรือ  $T$  เป็น 0,  $N \leq 100$ ,  $M = N - 1$
2. (7 คะแนน) หนึ่งใน  $S$  หรือ  $T$  เป็น 0,  $M = N - 1$
3. (6 คะแนน)  $M = N - 1$ ,  $U[i] = i$ ,  $V[i] = i + 1$ ,  $(0 \leq i \leq M - 1)$
4. (33 คะแนน)  $M = N - 1$
5. (18 คะแนน)  $A = 1$ ,  $B = 2$
6. (31 คะแนน) ไม่มีข้อกำหนดเพิ่มเติม

ถ้าโปรแกรมของคุณได้รับผลการตรวจเป็น **Accepted** และเรียก ask จำนวน  $X$  ครั้ง คะแนน  $P$  ในแต่ละกรณีทดสอบจะขึ้นอยู่กับปัญหาย่อยซึ่งถูกคำนวณดังต่อไปนี้

- ปัญหาย่อย 1.  $P = 5$
- ปัญหาย่อย 2. ถ้า  $X \leq 60$  แล้ว  $P = 7$  นอกจากนั้น  $P = 0$
- ปัญหาย่อย 3. ถ้า  $X \leq 60$  แล้ว  $P = 6$  นอกจากนั้น  $P = 0$
- ปัญหาย่อย 4. ถ้า  $X \leq 60$  แล้ว  $P = 33$  นอกจากนั้น  $P = 0$
- ปัญหาย่อย 5. ถ้า  $X \leq 52$  แล้ว  $P = 18$  นอกจากนั้น  $P = 0$
- ปัญหาย่อย 6.
  - ถ้า  $X \leq 50$ ,  $P = 31$
  - ถ้า  $51 \leq X \leq 52$ ,  $P = 21$
  - ถ้า  $53 \leq X$ ,  $P = 0$

โปรดสังเกตว่าคะแนนในแต่ละปัญหาย่อยเป็นค่าคะแนนที่น้อยที่สุดในแต่ละกรณีทดสอบในปัญหาย่อยนั้น

## เกรดเดอร์ตัวอย่าง

เกรดเดอร์ตัวอย่างอ่านข้อมูลนำเข้าในรูปแบบต่อไปนี้

- บรรทัดที่ 1:  $N M A B S T$
- บรรทัดที่  $2 + i$  ( $0 \leq i \leq M - 1$ ):  $U[i] V[i]$

ถ้าโปรแกรมของคุณได้รับผลการตรวจเป็น **Accepted** เกรดเดอร์ตัวอย่างจะพิมพ์ Accepted:  $q$  โดยค่า  $q$  คือจำนวนครั้งที่ ask ถูกเรียก

ถ้าโปรแกรมของคุณได้รับผลการตรวจเป็น **Wrong Answer**, เกรดเดอร์ตัวอย่างจะพิมพ์ Wrong Answer: MSG โดย MSG อาจเป็นหนึ่งในข้อความต่อไปนี้:

- answered not exactly once: ฟังก์ชัน answer ถูกเรียกไม่เท่ากับหนึ่งครั้ง
- w is invalid: ค่าความยาว  $w$  ที่กำหนดให้ ask ไม่เท่ากับ  $M$  หรือ  $w[i]$  ไม่เป็น 0 หรือ 1 สำหรับค่า  $i$  ( $0 \leq i \leq M - 1$ ) บางค่า
- more than 100 calls to ask: ฟังก์ชัน ask ถูกเรียกมากกว่า 100 ครั้ง
- {s, t} is wrong: ฟังก์ชัน answer ถูกเรียกด้วยคู่  $s$  และ  $t$  ที่ผิด