

Pathway In KVIS

หลังจากที่คุณกลับมาจากการปิดเทอมครั้งยิ่งใหญ่ คุณก็ได้พบว่าโรงเรียนของเราได้เปลี่ยนแปลงไปอย่างสิ้นเชิงด้วย **การทางอาคาร** ของเจ้าปโยศเวตกุมาร ทำให้โรงเรียนของเราได้ถูกแบ่งแยกออกเป็น N อาคาร และมีถนนแบบไม่ระบุทิศทางระหว่างอาคารอยู่ทั้งหมด M เส้น เชื่อม ในถนนที่ i จะประกอบด้วยอาคารเริ่มต้นคืออาคารหมายเลข $U[i]$ อาคารสิ้นสุดคืออาคารหมายเลข $V[i]$ และความยาวของถนนเส้นนั้นมีค่า $W[i]$ เมตร โดยคุณได้เข้ามาในอาคารของโรงเรียนทางอาคารที่ A และเพื่อนของคุณได้เข้ามาทางอาคารที่ B คุณและเพื่อนของคุณต้องการที่จะหาจุดนัดพบที่สามารถเจอกันได้อย่างเร็วที่สุด แต่ด้วยความที่ว่าคุณและเพื่อนของคุณอาจมีความเร็วในการเดินไม่เท่ากัน เราจึงกำหนดให้คุณใช้เวลาในการเดิน X วินาทีต่อเมตร และเพื่อนของคุณใช้เวลา Y วินาทีต่อเมตร เราต้องการหาว่าคุณควรจะพบกับเพื่อนของคุณในอาคารหมายเลขใดจึงจะประหยัดเวลาของพวกคุณทั้งสองคนได้มากที่สุดก่อนที่คุณจะถูกเจ้าปโยศเวตกุมารพบ (ถ้าหากมีหลายอาคารที่ใช้เวลาน้อยสุดเท่ากัน ให้เลือกตออาคารที่มีหมายเลขน้อยที่สุด)

พิจารณาตัวอย่างด้านล่าง

จากตัวอย่างจะเห็นได้ว่าคุณอยู่ที่อาคารหมายเลข 0 และเพื่อนของคุณอยู่ที่อาคารหมายเลข 7 โดยถ้าให้คุณใช้เวลาในการเดิน 3 วินาทีต่อเมตร และเพื่อนของคุณใช้เวลาในการเดิน 1 วินาทีต่อเมตร คุณสามารถเดินจากอาคาร 0 ไปสิ้นสุดที่อาคารหมายเลข 2 โดยใช้เวลากี่ทั้งหมด $(3 \cdot 5) = 15$ วินาที และเพื่อนของคุณจะเดินจากอาคารหมายเลข 7 ไปยังอาคารหมายเลข 5 และสิ้นสุดที่อาคารหมายเลข 2 ซึ่งใช้เวลากี่ทั้งหมด $(1 \cdot 14) + (1 \cdot 9) = 14 + 9 = 23$ วินาที ซึ่งการเลือกพบกันที่อาคารหมายเลข 2 โดยใช้เวลาในการเดินเพื่อมาพบกันทั้งหมด 23 วินาที ถือว่าเป็นเลือกวิธีการเดินที่ดีที่สุดแล้ว

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้

```
int find_building(int N, int A, int B, int X, int Y, vector U, vector V, vector W)
```

- N คือจำนวนอาคารทั้งหมดในอาณาเขตของโรงเรียน
- A คือหมายเลขของอาคารที่คุณเข้ามาในอาณาเขตของโรงเรียน
- B คือหมายเลขของอาคารที่เพื่อนของคุณเข้ามาในอาณาเขตของโรงเรียน
- X คือจำนวนวินาทีที่คุณใช้ในการเดินในแต่ละเมตร
- Y คือจำนวนวินาทีที่เพื่อนของคุณใช้ในการเดินในแต่ละเมตร
- $U[i]$ คือหมายเลขอาคารเริ่มต้นของถนนเส้นที่ i สำหรับทุก i ตั้งแต่ 0 ถึง $M - 1$
- $V[i]$ คือหมายเลขอาคารสิ้นสุดของถนนเส้นที่ i สำหรับทุก i ตั้งแต่ 0 ถึง $M - 1$
- $W[i]$ คือความยาวของถนนเส้นที่ i สำหรับทุก i ตั้งแต่ 0 ถึง $M - 1$

ฟังก์ชันนี้จะถูกเรียกเพียงครั้งเดียวเมื่อโปรแกรมเริ่มทำงาน

ขอบเขต

- $2 \leq N \leq 100\,000$
- $N - 1 \leq M \leq 300\,000$
- $0 \leq A, B < N$ และ $A \neq B$

- $1 \leq X, Y \leq 1000$
- $0 \leq U[i], V[i] < N$ สำหรับทุก i ตั้งแต่ 0 ถึง $M - 1$
- $1 \leq W[i] \leq 10\,000$ สำหรับทุก i ตั้งแต่ 0 ถึง $M - 1$

ปัญหาย่อย

ปัญหาย่อย	คะแนน	เงื่อนไขเพิ่มเติม
1	3	จะมีถนนโดยตรงจาก A ไปยัง B เสมอ
2	8	รับประกันว่า $M = N - 1$, $U[i] = i$, $V[i] = i + 1$ และ $W[i] = 1$ สำหรับทุก i ตั้งแต่ 0 ถึง $N - 2$
3	12	รับประกันว่า $M = N - 1$, $U[i] = i$, และ $V[i] = i + 1$ สำหรับทุก i ตั้งแต่ 0 ถึง $N - 2$
4	17	$W[i] = 1$ สำหรับทุก i ตั้งแต่ 0 ถึง $M - 1$
5	15	$N \leq 100$
6	10	$X = 1$ และ $Y = 1$
7	15	$N \leq 2\,000$
8	20	ไม่มีเงื่อนไขเพิ่มเติม

ตัวอย่าง

สมมติว่าคุณเข้ามาในโรงเรียนที่อาคารหมายเลข 0 เพื่อนของคุณเข้ามาในโรงเรียนที่อาคารหมายเลข 7 คุณใช้เวลาในการเดิน 3 วนาทีต่อเมตร และเพื่อนของคุณใช้เวลาในการเดิน 1 วนาทีต่อเมตร

เกรดเดอร์จะเรียกฟังก์ชัน `find_building` ดังนี้

```
find_building(8, 9, 0, 7, 3, 1,
[0, 2, 4, 1, 2, 2, 3, 4, 5],
[2, 4, 6, 3, 3, 5, 5, 5, 7],
[5, 3, 11, 8, 12, 9, 7, 6, 14])
```

ซึ่งจะคืนค่า 2

เกรดเดอร์ตัวอย่าง

รูปแบบข้อมูลนำเข้า:

บรรทัดที่ 1: $N M A B$

บรรทัดที่ 2: $X Y$

บรรทัดที่ 3 ถึง $M + 2$: $U[i];V[i];W[i];$ สำหรับทุก i ตั้งแต่ 0 ถึง $M-1$

เกรดเดอร์ตัวอย่างจะแสดงผลที่ `return` มาจากฟังก์ชัน `find_building`

ตัวอย่างการทำงานจากข้อมูลนำเข้า



```
8 9 0 7
3 1
0 2 5
2 4 3
4 6 11
1 3 8
2 5 12
3 5 7
4 5 6
5 7 14
```

ทรอดเตอร์ตัวอย่างจะเรียก `find_building(8, 9, 0, 7, 3, 1, [0, 2, 4, 1, 2, 2, 3, 4, 5], [2, 4, 6, 3, 3, 5, 5, 5, 7], [5, 3, 11, 8, 12, 9, 7, 6, 14])` และจะหากฟังก์ชันถูกต้องจะส่งออก

2

ข้อจำกัด

- Time limit: 1.5 seconds
- Memory limit: 512 MB