

## แผนที่ (Mapping)

อาณาจักรแห่งหนึ่ง มีเมืองจำนวน  $N$  เมือง (เรียกเป็นเมือง 1 ถึงเมือง  $N$ ) เมืองดังกล่าวเชื่อมต่อกันเป็นกราฟต้นไม้ที่มีน้ำหนัก (weighted tree) แต่ละเมืองจะเชื่อมกับเมืองอื่นไม่เกิน 3 เมืองและรับประกันว่าระยะทางระหว่างคูเมืองใด ๆ จะแตกต่างกันเสมอ

อาณาจักรนี้มีจุดศูนย์กลาง (เมืองหลวง) อยู่ที่เมือง  $S$  แต่ละเมืองมีระดับความสำคัญที่แตกต่างกันตั้งแต่ 1 ถึง  $N$  ตามระยะทางจากเมืองหลวง กล่าวคือเมืองหลวงหรือ  $S$  จะมีความสำคัญระดับ 1, เมืองที่อยู่ใกล้  $S$  ที่สุดจะมีความสำคัญระดับ 2, เมืองที่ไกลรองลงมาจะมีความสำคัญระดับ 3 ไปเรื่อยๆจนถึงเมืองที่ไกลที่สุดจะมีความสำคัญระดับ  $N$

แต่ละเมืองจะมีอุปกรณ์ส่งสัญญาณ โดยอุปกรณ์นี้จะมีหลักการทำงานคือ เราต้องเลือกเมืองปลายทาง 2 เมือง สมมติให้เป็น  $A$  และ  $B$  และทำการเลือกสถานีต้นทาง สมมติให้เป็น  $X$  เมื่อเปิดเครื่อง อุปกรณ์จะส่งสัญญาณจากเมืองต้นทาง  $X$  ไปเรื่อยๆ จนเจอกับเมืองปลายทาง หลังจากนั้นเครื่องจะบอกว่าจากเมืองต้นทางไปเมืองปลายทางเมืองไหนใกล้กว่ากัน (บอกว่า  $dis[X][A]$  หรือ  $dis[X][B]$  อันไหนน้อยกว่ากัน) เป็นไปได้ที่  $A$  หรือ  $B$  จะเท่ากับ  $X$  ซึ่งในกรณีนี้ระยะจาก  $X$  ไปยังเมืองตัวเองจะเป็น 0 และเมืองดังกล่าวจะใกล้กว่าอีกเมืองเสมอ

แต่!!! ในบางกรณี อุปกรณ์ที่อยู่ในทุก ๆ เมืองนั้น **อาจจะ** เป็นของปลอมทำให้ความแม่นยำไม่เที่ยงตรง กล่าวคือ ในกรณีที่เมืองปลายทาง  $A$  และ  $B$  มีระดับความสำคัญต่างกัน 1 เครื่องนี้จะทำงานตรงกันข้าม นั่นคือ บอกว่า  $dis[X][A]$  หรือ  $dis[X][B]$  อันไหนมากกว่ากัน

**หมายเหตุ** ในการแก้ปัญหาข้อนี้ คุณจะทราบแน่ชัดว่าชุดอุปกรณ์ที่ติดตั้งสำหรับทุกเมืองนั้นเป็นของจริงหรือของปลอม ซึ่งถ้าเป็นของจริงจะเป็นของจริงทุกเมือง และถ้าเป็นของปลอมจะเป็นของปลอมทุกเมือง

อยู่มาวันหนึ่ง ผู้ปกครองอาณาจักรต้องการที่จะทำแผนที่ (ที่ระบุแค่ข้อมูลว่าแต่ละเมืองเชื่อมกันอย่างไร โดยไม่ต้องทราบถึงระยะทางระหว่างเมือง) ผู้ปกครองจึงต้องการให้คุณช่วยสร้างแผนที่ขึ้นมาจากชุดอุปกรณ์นี้ให้ โดยคุณจะใช้อุปกรณ์ได้ไม่เกิน  $M$  ครั้ง ( $M$  จะถูกระบุในแต่ละปัญหาย่อย)

## รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้

```
vector<pair<int,int>> mapping(int N, int S)
```

- ฟังก์ชันนี้จะถูกเรียกใช้เพียงครั้งเดียว
- $N$  แทนจำนวนเมืองในอาณาจักรนี้ และเนื่องจากในกรณีที่ชุดอุปกรณ์เป็นของจริง เมืองหลวงและระดับความสำคัญไม่มีผลในการคิดคำนวณ จึงให้  $S = -1$  แทนกรณีที่อุปกรณ์เป็นของจริง แต่ถ้าอุปกรณ์เป็นของปลอม  $S$  จะแทนเมืองที่เป็นเมืองหลวงดังที่กล่าวไว้ในโจทย์ข้างต้น
- ฟังก์ชันจะต้องคืนค่าเป็น vector ที่ประกอบไปด้วย pair จำนวน  $N - 1$  คู่ โดยแต่ละคู่จะแทนคู่ของ

เมืองที่มีทางเชื่อมต่อกัน โดยลำดับการเรียงไม่มีผลในการตรวจ

ฟังก์ชัน mapping จะสามารถเรียกใช้ฟังก์ชัน send\_signal ที่มีรายละเอียดดังด้านล่าง

```
bool send_signal(int x, int a, int b)
```

- การคืนค่าของฟังก์ชัน send\_signal เป็นดังนี้
  - หาก ระยะทางจาก  $a$  ไป  $x$  น้อยกว่า ระยะทางจาก  $b$  ไป  $x$  จะ return true
  - หาก ระยะทางจาก  $a$  ไป  $x$  มากกว่า ระยะทางจาก  $b$  ไป  $x$  จะ return false

หมายเหตุ ถ้าอุปกรณ์เป็นของปลอม และระดับความสำคัญของเมือง  $a$  และเมือง  $b$  ต่างกัน 1 send\_signal จะคืนค่าที่ตรงกันข้ามแทน (ดูเพิ่มเติมจากตัวอย่าง)

- จะสามารถเรียกใช้ฟังก์ชันได้ไม่เกิน  $M$  ครั้ง (จะถูกระบุในแต่ละปัญหาย่อย)

## ขอบเขต

- $2 \leq N \leq 1000$
- $1 \leq S \leq N$  หรือ  $S = -1$

## ปัญหาย่อย

หมายเหตุ  $M$  คือจำนวนครั้งที่มากที่สุดที่สามารถเรียก send\_signal ได้

1. (5 คะแนน) แต่ละเมืองเชื่อมกับเมืองอื่นไม่เกิน 2 เมือง และ  $S = -1$  และ  $M = 2\,000\,000$
2. (13 คะแนน) แต่ละเมืองเชื่อมกับเมืองอื่นไม่เกิน 2 เมือง และ  $S = -1$  และ  $M = 30\,000$
3. (57 คะแนน)  $S = -1$  และ  $M = 2\,000\,000$
4. (7 คะแนน)  $M = 2\,000\,000$
5. (12 คะแนน)  $M = 26\,000$
6. (6 คะแนน)  $M = 23\,000$

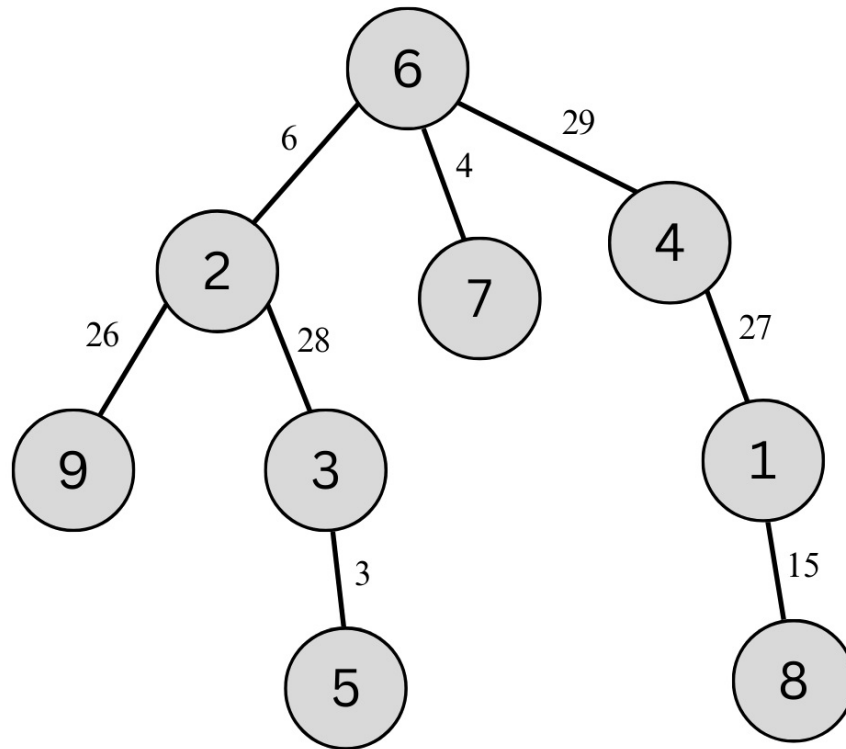
ในปัญหาย่อยที่ 3 จะมีการให้คะแนนบางส่วน กำหนดค่า  $X$  แทนจำนวนการเรียกใช้ send\_signal โดยคะแนนที่ได้จะถูกประมาณเป็นทศนิยมสองตำแหน่งดังนี้

เงื่อนไข	คะแนนที่ได้
$X > 2\,000\,000$	0
$50\,000 < X \leq 1\,000\,000$	15
$23\,000 < X \leq 50\,000$	$e^{\frac{613}{\sqrt{X}}}$
$X \leq 23\,000$	57

หมายเหตุ บนเกรดเดอร์สำหรับทุกปัญหาย่อยยกเว้นปัญหาย่อยที่ 3 ถ้ามีการเรียก send\_signal เกินที่ได้ระบุไว้ในแต่ละปัญหาย่อย เกรดเดอร์จะแสดงผลว่าคำตอบผิด

## ตัวอย่าง

```
mapping(9, -1)
```



พิจารณา กราฟตัวอย่างในกรณีที่  $N = 9$  และเนื่องจาก  $S = -1$  นั้นหมายความว่าอุปกรณ์ชุดนี้เป็นของจริง ตัวอย่างการเรียกใช้ `send_signal` เป็นดังนี้

```
send_signal(2, 3, 7)
```

returns false

```
send_signal(1, 6, 8)
```

returns false

```
send_signal(4, 2, 7)
```

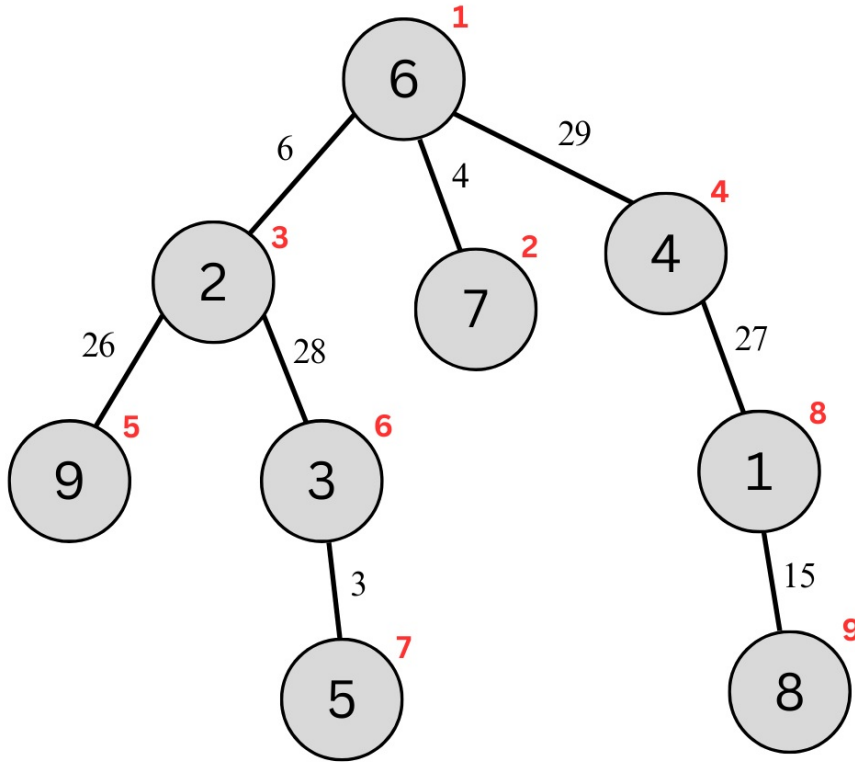
returns false

```
send_signal(9, 9, 3)
```

returns true

จากตัวอย่างข้างต้น ฟังก์ชันจะต้องคืนค่า ((6, 2), (6, 7), (6, 4), (2, 9), (2, 3), (3, 5), (4, 1), (1, 8)) โดยเรียงลำดับอย่างไรก็ได้

```
mapping(9, 6)
```



พิจารณา กราฟตัวอย่างในกรณีที่  $N = 9$  และเนื่องจาก  $S = 6$  นั้นหมายความว่าอุปกรณ์ชุดนี้เป็นของปลอม โดยลำดับความสำคัญของแต่ละเมืองจะเป็นตัวอักษรสีแดงในภาพ ตัวอย่างการเรียกใช้ `send_signal` เป็นดังนี้

```
send_signal(2, 3, 7)
```

returns false

```
send_signal(1, 6, 8)
```

returns false

```
send_signal(4, 2, 7)
```

returns true (ระยะทางจาก 4 ไป 2 มากกว่า 4 ไป 7 แต่เนื่องจากลำดับความสำคัญของ 2 และ 7 ต่างกัน 1 และ อุปกรณ์เป็นของปลอม)

```
send_signal(9, 9, 3)
```

returns true

## เกรตเตอร์ตัวอย่าง

เกรตเตอร์ตัวอย่างจะรับข้อมูลดังนี้

- บรรทัดที่ 1:  $N$   $S$
- บรรทัดที่ 2 ถึง  $N$ :  $a[i]$   $b[i]$   $d[i]$  เพื่อระบุว่าเมือง  $a[i]$  เชื่อมกับเมือง  $b[i]$  ด้วยเส้นเชื่อมที่มีความยาว  $d[i]$

ข้อมูลส่งออกสำหรับเกรตเตอร์ตัวอย่างจะแสดงความถูกต้องของคำตอบ โดยถ้าคำตอบถูกจะแสดงจำนวนการเรียกใช้ฟังก์ชัน `send_signal`

## ข้อจำกัด

- Time limit: 1 seconds
- Memory limit: 512 MB