

# ไกลที่สุด (Longest)

ณ ประเทศแห่งหนึ่ง ประเทศที่เต็มไปด้วยผู้คนที่มีพื้นเพที่หลากหลายด้วยความที่ประเทศนี้มีพื้นที่จำกัดจึงทำให้สามารถรองรับเมืองได้ไม่เกิน  $N$  เมือง ในช่วงแรก (ปีที่ 0) ประเทศแห่งนี้มีเพียงเมืองเดียว (ให้เป็นเมืองที่ 0) ก็เพียงพอต่อการอยู่อาศัยของประชากรทั้งประเทศ แต่เมื่อเวลาผ่านไปจำนวนประชากรก็เริ่มเพิ่มขึ้นเรื่อยๆ จนทำให้เมืองที่มีอยู่นั้นไม่พอ จึงทำให้ในแต่ละปีจะมีการสร้างเมืองใหม่ขึ้น โดยในปีที่  $i$  จะมีการสร้างเมืองที่  $i$  ขึ้น และจะทำการสร้างถนนแบบสองทิศทางระหว่างเมืองที่  $i$  และเมือง  $p[i]$  เดิม (ซึ่งเป็นการการันตีว่า  $i > p[i]$ ) โดยถนนเส้นนี้จะมีความยาว  $w[i]$  สำหรับ  $0 < i < N$

ในแต่ละปีการเดินทางที่ไกลเกินไปนั้นอาจทำให้เกิดความเหนื่อยล้าได้ ผู้นำประเทศจึงต้องการที่จะตกแต่งถนนระหว่างคูเมืองที่ห่างกันมากที่สุดเพื่อให้ผู้ที่เดินทางไกลมากนั้นผ่อนคลายสายตา (มั้ง?) ผู้นำประเทศจึงต้องการว่าจ้างคุณในการหา **คูเมืองที่มีระยะทางห่างกันมากที่สุดในแต่ละปี** (ถ้ามีคูเมืองที่ห่างกันมากที่สุดหลายเมืองสามารถตอบคูเมืองใดก็ได้)

แต่ด้วยข้อจำกัดทางเครื่องมือ ทำให้คุณไม่สามารถวัดระยะทางตรงๆ ระหว่างเมืองได้ คุณมีเพียงตะเกียงวิเศษเท่านั้นที่จะช่วยในการหาเมืองที่ไกลที่สุด โดยเมื่อคุณลูบตะเกียงวิเศษครั้งนึง อลาตินจะลอยออกมาตอบคำถามหนึ่งคำถาม ซึ่งคุณสามารถถามอลาตินได้แค่ว่าระยะทางจากเมือง  $a$  ไปเมือง  $x$  กับระยะทางจากเมือง  $b$  ไปเมือง  $x$  อันไหนมีระยะทางที่น้อยกว่ากัน

เนื่องจากคุณไม่ต้องการให้ตะเกียงวิเศษเป็นรอยมาก จึงต้องการลูบตะเกียงวิเศษ (ถามคำถามอลาติน) ด้วยจำนวนครั้งรวมที่น้อยที่สุดที่เป็นไปได้ แต่ยังสามารถหาคูเมืองที่ห่างกันมากที่สุดในแต่ละปีทั้ง  $N$  ปีได้ (ปีที่ 0 ถึง  $N - 1$ ) ให้คุณเขียนโปรแกรมเพื่อทำให้ตะเกียงเป็นรอยน้อยที่สุด (ถามคำถามอลาตินน้อยครั้งที่สุด)

## รายละเอียดการเขียนโปรแกรม

```
vector<pair<int,int>> longest(int N)
```

- ฟังก์ชันนี้จะถูกเรียกใช้เพียงครั้งเดียว
- ฟังก์ชันจะต้องคืนค่าเป็น vector ที่ประกอบไปด้วย pair จำนวน  $N$  คู่ โดยที่ pair คู่ที่  $i$  จะเป็นเมือง 2 เมืองที่ห่างกันมากที่สุด (เมืองใดขึ้นก่อนก็ได้) ในปีที่  $i$  สำหรับทุก  $0 \leq i < N$  ซึ่งถ้าในปีนั้นประกอบด้วยคูเมืองที่สอดคล้องหลายคู่ สามารถตอบคูใดก็ได้

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

- การคืนค่าของฟังก์ชัน find
  - หาก ระยะทางจาก  $a$  ไป  $x$  น้อยกว่า ระยะทางจาก  $b$  ไป  $x$  จะ return true
  - หาก ระยะทางจาก  $a$  ไป  $x$  มากกว่าหรือเท่ากับ ระยะทางจาก  $b$  ไป  $x$  จะ return false

## ขอบเขต

- $2 \leq N \leq 5\,000$
- $0 \leq p[i] < i$  สำหรับทุก  $0 < i < N$
- $1 \leq w[i] \leq 1\,000\,000\,000$  สำหรับทุก  $0 \leq i < N$

## ปัญหาย่อย

หมายเหตุ  $degree[i]$  แทนจำนวนเมืองที่ติดกับเมืองที่  $i$  ในปีที่  $N - 1$

1. (5 คะแนน)  $degree[0] = 1$  และ  $degree[i] \leq 2$  สำหรับทุก  $1 \leq i < N$
2. (20 คะแนน)  $N \leq 300$
3. (75 คะแนน)  $N \leq 5000$

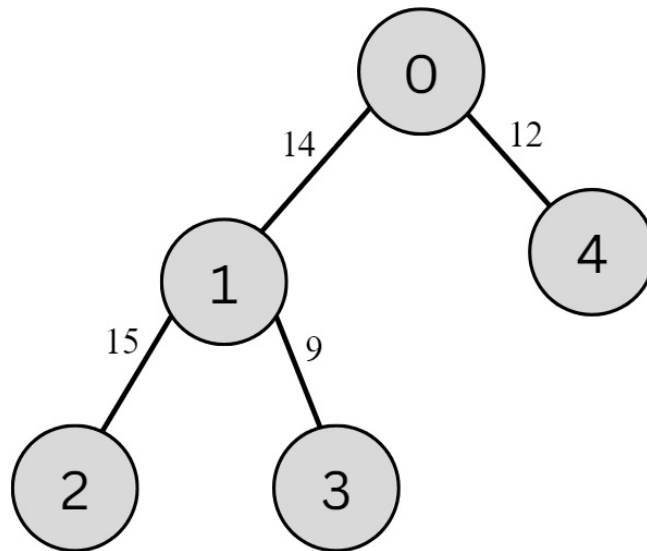
ในปัญหาย่อยที่ 3 จะมีการให้คะแนนบางส่วน กำหนดค่า  $M$  แทนจำนวนครั้งของการเรียกใช้ฟังก์ชัน `find` โดยคะแนนที่ได้จะถูกประมาณเป็นทศนิยมสองตำแหน่งดังนี้

เงื่อนไข	คะแนนที่ได้
$M > 2\,500\,000$	0
$1\,250\,000 < M \leq 2\,500\,000$	$18 - \frac{M - 1\,250\,000}{70\,000}$
$100\,000 < M \leq 1\,250\,000$	$36 - \frac{M - 100\,000}{64\,000}$
$10\,000 < M \leq 100\,000$	$60 - \frac{M - 10\,000}{3\,750}$
$9\,995 < M \leq 10\,000$	$20\,060 - 2M$
$M \leq 9\,995$	75

## ตัวอย่าง

```
longest(5)
```

สมมติให้ประเทศในปีที่ 4 (ปีสุดท้ายที่สร้างเมืองเพิ่ม) เป็นดังนี้



ในส่วนของการเรียกฟังก์ชัน find

- $\text{find}(0, 1, 3)$  จะ **return false** เนื่องจากระยะทางจากเมือง 0 ไปเมือง 3 มากกว่าระยะทางจากเมือง 1 ไปเมือง 3
- $\text{find}(0, 2, 4)$  จะ **return true** เนื่องจากระยะทางจากเมือง 0 ไปเมือง 4 น้อยกว่าระยะทางจากเมือง 2 ไปเมือง 4
- $\text{find}(3, 1, 4)$  จะ **return false** เนื่องจากระยะทางจากเมือง 3 ไปเมือง 4 มากกว่าระยะทางจากเมือง 1 ไปเมือง 4

ในส่วนของการตอบคำถาม

- ในปีที่ 0 ซึ่งมีเพียงเมืองเดียวคือเมือง 0 ดังนั้นคู่มืองที่ห่างกันมากที่สุดจึงเป็นเมือง 0 กับเมือง 0
- ในปีที่ 1 ประเทศจะมีเมือง 0, 1 คู่มืองที่ห่างกันมากที่สุดคือเมือง 0 และเมือง 1 โดยมีระยะทางเป็น 14
- ในปีที่ 2 ประเทศจะมีเมือง 0, 1, 2 คู่มืองที่ห่างกันมากที่สุดคือเมือง 0 และเมือง 2 โดยมีระยะทางเป็น 29
- ในปีที่ 3 ประเทศจะมีเมือง 0, 1, 2, 3 คู่มืองที่ห่างกันมากที่สุดคือเมือง 0 และเมือง 2 โดยมีระยะทางเป็น 29
- ในปีที่ 4 ประเทศจะมีเมือง 0, 1, 2, 3, 4 คู่มืองที่ห่างกันมากที่สุดคือเมือง 2 และเมือง 4 โดยมีระยะทางเป็น 41

ดังนั้นฟังก์ชันนี้ต้องคืนค่าเป็น  $[[0, 0], [0, 1], [0, 2], [2, 0], [2, 4]]$

## เกรตเตอร์ตัวอย่าง

เกรตเตอร์ตัวอย่างจะรับข้อมูลดังนี้

- บรรทัดที่ 1:  $N$
- บรรทัดที่ 2 ถึง  $N$ :  $p[i] \ w[i]$

ข้อมูลส่งออกสำหรับเกรตเตอร์ตัวอย่างจะแสดงความถูกต้องของคำตอบ  
เรียกใช้ฟังก์ชัน find

โดยถ้าคำตอบถูกจะแสดงจำนวนการ

## ข้อจำกัด

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