

## เชื่อมอภิมหาต้นไม้ (supertrees)

การ์เด้นบายเดอะเบย์ (Gardens by the Bay) เป็นสวนสาธารณะขนาดใหญ่ในสิงคโปร์ ในสวนมีหอคอยที่เรียกว่า อภิมหาต้นไม้ ทั้งหมด  $n$  หอ หอคอยเหล่านี้ถูกกำกับด้วยหมายเลข  $0$  ถึง  $n - 1$  เราต้องการสร้างสะพานจำนวน อย่างน้อยศูนย์เส้น โดยแต่ละเส้นสะพานจะเชื่อมหอคอยสองหอที่แตกต่างกัน และเราสามารถเดิน ไปกลับ บนสะพานได้ ห้ามมีสะพานสองเส้นใด ๆ เชื่อมหอคอยคู่เดียวกัน

เส้นทางจากหอคอย  $x$  ไปหอคอย  $y$  เป็นลำดับของหอคอยหนึ่งหอหรือมากกว่าที่:

- หอคอยแรกในลำดับ คือ  $x$
- หอคอยสุดท้ายในลำดับ คือ  $y$
- ทุก ๆ หอคอยในลำดับต้อง ไม่ซ้ำกัน
- ทุก ๆ คู่ของหอคอยที่อยู่ในลำดับต่อกัน จะต้องถูกเชื่อมด้วยสะพาน

ดังนั้นจากนิยามนี้ จะมีเพียงเส้นทางเดียวจากหอคอยใด ๆ ไปหาตัวมันเอง และจำนวนเส้นทาง (ที่แตกต่างกัน) จากหอคอย  $i$  ไปหอคอย  $j$  จะเท่ากับจำนวนเส้นทางจากหอคอย  $j$  ไปหอคอย  $i$

สถาปนิกผู้ออกแบบสะพานต้องการสร้างสะพานที่ทำให้ทุก ๆ คู่ของหอคอย  $0 \leq i, j \leq n - 1$  มีเส้นทาง (ที่แตกต่างกัน) จากหอคอย  $i$  ไปหอคอย  $j$  ทั้งหมด  $p[i][j]$  เส้นทางพอดี โดยที่  $0 \leq p[i][j] \leq 3$

คุณต้องสร้างสะพานให้ตรงกับความต้องการของสถาปนิก หรือบอกว่าไม่สามารถสร้างสะพานตามได้เงื่อนไขดังกล่าว

## รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้:

```
int construct(int[][] p)
```

- $p$ : อาร์เรย์ขนาด  $n \times n$  ที่แสดงถึงความต้องการของสถาปนิก
- ถ้าการสร้างสะพานสามารถทำได้ ฟังก์ชันนี้จะต้องเรียกฟังก์ชัน `build` (ดูรายละเอียดด้านล่าง) จำนวนหนึ่งครั้งพอดีเพื่อรายงานการสร้างสะพาน และหลังจากนั้นฟังก์ชันต้องคืนค่า `1`
- แต่ถ้าสร้างไม่ได้ ฟังก์ชันนี้จะต้องคืนค่า `0` โดยไม่ทำการเรียกฟังก์ชัน `build` เลย
- ฟังก์ชันนี้จะถูกเรียก 1 ครั้งพอดี

ฟังก์ชัน `build` เป็นดังนี้:

```
void build(int[][] b)
```

- $b$ : อาร์เรย์ขนาด  $n \times n$  โดยที่  $b[i][j] = 1$  เมื่อมีสะพานเชื่อมหอคอย  $i$  และหอคอย  $j$  และให้  $b[i][j] = 0$  ถ้าหากไม่มีสะพาน
- ให้สังเกตว่าอาร์เรย์นี้จะต้องมีคุณสมบัติ  $b[i][j] = b[j][i]$  สำหรับ  $0 \leq i, j \leq n - 1$  และ  $b[i][i] = 0$  สำหรับ  $0 \leq i \leq n - 1$ .

## ตัวอย่าง

### ตัวอย่างที่ 1

พิจารณาการเรียกต่อไปนี้:

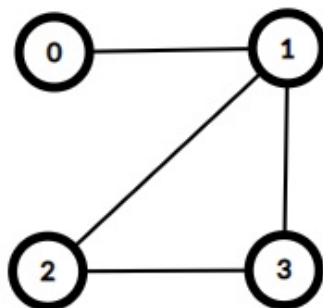
```
construct([[1, 1, 2, 2], [1, 1, 2, 2], [2, 2, 1, 2], [2, 2, 2, 1]])
```

ซึ่งหมายความว่า ต้องมีหนึ่งเส้นทางระหว่างหอคอย 0 และหอคอย 1 สำหรับคู่ของหอคอยอื่น ๆ  $(x, y)$  ที่  $0 \leq x < y \leq 3$  ต้องมีทั้งหมด 2 เส้นทางระหว่างหอคอย  $x$  และหอคอย  $y$

ตัวอย่างนี้สามารถแก้ได้โดยการสร้างสะพาน 4 เส้น ที่เชื่อมระหว่างหอคอย (0, 1), (1, 2), (1, 3) และ (2, 3)

ฟังก์ชัน `construct` ต้องรายงานการสร้างสะพานโดยเรียก

- `build([[0, 1, 0, 0], [1, 0, 1, 1], [0, 1, 0, 1], [0, 1, 1, 0]])`



และฟังก์ชันนี้ต้องคืนค่า 1

ในกรณีนี้ มีวิธีสร้างสะพานหลายแบบที่ตรงกับความต้องการ ซึ่งทุกแบบถือว่าเป็นคำตอบที่ถูกต้อง

### ตัวอย่างที่ 2

พิจารณาการเรียกต่อไปนี้:

```
construct([[1, 0], [0, 1]])
```

ซึ่งหมายความว่าสองหอคอยนี้ต้องไม่สามารถเดินทางถึงกันได้ ทำได้โดยการไม่สร้างสะพานใด ๆ

ดังนั้นฟังก์ชัน `construct` ต้องเรียก

- `build([[0, 0], [0, 0]])`

หลังจากนั้นให้คืนค่า 1

### ตัวอย่างที่ 3

พิจารณาการเรียกต่อไปนี้:

```
construct([[1, 3], [3, 1]])
```

ซึ่งหมายความว่า ต้องมีทั้งหมด 3 เส้นทางระหว่างหอคอย 0 และหอคอย 1 ทำให้ไม่มีวิธีสร้างสะพานที่ตรงความต้องการนี้ได้ ดังนั้นฟังก์ชัน `construct` ต้องคืนค่า 0 และต้องไม่เรียกฟังก์ชัน `build`

### ข้อจำกัด

- $1 \leq n \leq 1000$
- $p[i][i] = 1$  (สำหรับ  $0 \leq i \leq n - 1$ )
- $p[i][j] = p[j][i]$  (สำหรับ  $0 \leq i, j \leq n - 1$ )
- $0 \leq p[i][j] \leq 3$  (สำหรับ  $0 \leq i, j \leq n - 1$ )

### ปัญหาย่อย

1. (11 คะแนน)  $p[i][j] = 1$  (สำหรับ  $0 \leq i, j \leq n - 1$ )
2. (10 คะแนน)  $p[i][j] = 0$  หรือ 1 (สำหรับ  $0 \leq i, j \leq n - 1$ )
3. (19 คะแนน)  $p[i][j] = 0$  หรือ 2 (สำหรับ  $i \neq j, 0 \leq i, j \leq n - 1$ )
4. (35 คะแนน)  $0 \leq p[i][j] \leq 2$  (สำหรับ  $0 \leq i, j \leq n - 1$ ) และมีการสร้างสะพานอย่างน้อยหนึ่งแบบที่ตรงกับความต้องการ
5. (21 คะแนน)  $0 \leq p[i][j] \leq 2$  (สำหรับ  $0 \leq i, j \leq n - 1$ )
6. (4 คะแนน) ไม่มีข้อจำกัดเพิ่มเติม

### เกรตเดอร์ตัวอย่าง

เกรตเดอร์ตัวอย่างอ่านข้อมูลนำเข้าในรูปแบบต่อไปนี้

- บรรทัด 1:  $n$
- บรรทัด  $2 + i$  ( $0 \leq i \leq n - 1$ ):  $p[i][0] \ p[i][1] \ \dots \ p[i][n - 1]$

ข้อมูลส่งออกของเกรตเดอร์จะเป็นดังต่อไปนี้

- บรรทัด 1: ค่าที่คืนจากฟังก์ชัน `construct`

ถ้าค่าที่คืนจาก `construct` เป็น 1 เกรตเดอร์ตัวอย่างจะพิมพ์ข้อมูลต่อไปนี้เพิ่มเติม

- บรรทัด  $2 + i$  ( $0 \leq i \leq n - 1$ ):  $b[i][0] \ b[i][1] \ \dots \ b[i][n - 1]$

