



## ฮานอยคู่ (doublehanoi)

หอคอยฮานอยเป็นเกมที่ทุกคนรู้จัก และบางคนก็อาจจะเคยเล่น ในเกมปกติเรามีจานอยู่  $N$  ใบ แต่ในเกมนี้จะมีจาน  $2N$  ใบ แต่หมายเลขจานแค่  $N$  แบบ นั่นคือสำหรับจานแต่ละหมายเลขเราจะมีจานสองใบ สำหรับหมายเลข  $0$  ถึง  $N - 1$  จานหมายเลข  $i$  มีขนาด  $i$  โดยที่ขนาด  $0$  คือขนาดเล็กที่สุดและขนาด  $N - 1$  มีขนาดใหญ่ที่สุด

จาน  $2N$  ใบเหล่านี้เสียบอยู่บนเสา 3 เสา หมายเลข  $0, 1$ , และ  $2$  เราจะมีเงื่อนไขว่าจานใบใหญ่จะซ้อนทับจานใบเล็กไม่ได้

ในโจทย์ทั่วไป เราต้องการย้ายจานทั้งหมดที่ตอนแรกเสียบอยู่ในเสาที่  $0$  ไปยังเสาที่  $3$  ในข้อนี้คุณอาจจะได้รับรูปแบบเริ่มต้นและสิ้นสุดที่แตกต่างจากนั้น

พิจารณาตัวอย่างต่อไปนี้ สมมติ  $N = 2$  มีจาน 4 ใบดังนี้ ถ้าเราต้องการย้ายจากสถานะด้านซ้ายไปสถานะด้านขวา เราจะเริ่มโดยการหยิบจานหมายเลข  $0$  ที่เสา  $1$  ไปที่เสา  $2$  จากนั้นย้ายจานหมายเลข  $1$  จากเสา  $0$  มาเสา  $1$  ทำไปเรื่อย ๆ เราจะย้ายจานไปเป็นสถานะด้านขวาได้โดยย้ายจาน 7 ครั้ง



## รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้

```
long long hanoi(int N,
                vector<int> S0, vector<int> S1, vector<int> S2,
                vector<int> T0, vector<int> T1, vector<int> T2)
```

- ฟังก์ชันนี้จะถูกเรียก 1 ครั้ง
- อาร์เรย์  $S0, S1, S2$  แทนสถานะเริ่มต้นในรูปแบบดังนี้  $S0$  จะเป็นรายการของจานในเสาที่  $0$ ,  $S1$  จะเป็นรายการของจานในเสาที่  $1$ , และ  $S2$  จะเป็นรายการของจานในเสาที่  $2$  โดยจะเป็นหมายเลขจานจาก  $0$  จนถึง  $N - 1$  เรียงจากบนลงล่าง (นั่นคือจะเป็นตัวเลขเรียงจากน้อยไปหามาก) รับประกันว่าจำนวนเต็ม  $i$  ระหว่าง  $0$  ถึง  $N - 1$  จะปรากฏแค่ในสองเวกเตอร์เท่านั้น
- อาร์เรย์  $T0, T1, T2$  แทนสถานะเป้าหมายในรูปแบบเดียวกัน
- ฟังก์ชันต้องคืนค่าจำนวนการย้ายที่น้อยครั้งที่ที่สุด จากรูปสถานะ  $S$  ไปยังรูปแบบ  $T$  รับประกันว่าคำตอบจะอยู่ในขอบเขตของ `long long`

## เงื่อนไข

- $1 \leq N \leq 60$
- $0 \leq S0[i] \leq N - 1, 0 \leq S1[i] \leq N - 1, 0 \leq S2[i] \leq N - 1$
- $0 \leq T0[i] \leq N - 1, 0 \leq T1[i] \leq N - 1, 0 \leq T2[i] \leq N - 1$

## ปัญหาย่อย

1. (6 points)  $N = 2$
2. (13 points)  $N \leq 5$
3. (10 points) ที่สถานะเริ่มต้นจานทั้งหมดอยู่เสา 0 ที่สถานะปลายทางจานทั้งหมดอยู่เสา 2
4. (36 points) ที่สถานะปลายทางจานทั้งหมดอยู่เสา 2
5. (35 points) ไม่มีเงื่อนไขอื่นๆ

## ตัวอย่าง

จากตัวอย่าง เกรดเดอร์จะเรียกฟังก์ชัน `hanoi` ดังนี้

```
hanoi(2, [1], [0], [0,1],  
      [], [0,1,1], [0])
```

ฟังก์ชันจะต้องคืนค่า 7

## เกรดเดอร์ตัวอย่าง

เกรดเดอร์ตัวอย่างจะอ่านข้อมูลนำเข้าดังนี้

- Line 1:  $N$
- Line 2:  $S0.size$   $S0[0]$   $S0[1]$   $S0[2]$  ...
- Line 3:  $S1.size$   $S1[0]$   $S1[1]$   $S1[2]$  ...
- Line 4:  $S2.size$   $S2[0]$   $S2[1]$   $S2[2]$  ...
- Line 5:  $T0.size$   $T0[0]$   $T0[1]$   $T0[2]$  ...
- Line 6:  $T1.size$   $T1[0]$   $T1[1]$   $T1[2]$  ...
- Line 7:  $T2.size$   $T2[0]$   $T2[1]$   $T2[2]$  ...

เกรดเดอร์ตัวอย่างจะพิมพ์ค่าที่คืนจากฟังก์ชัน `hanoi`

## ขีดจำกัด

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