

## 2559\_2\_Recursive\_L3\_Double\_Hanoi

มีเสาอยู่ 3 ต้น มีจานกลม (มีรูตรงกลาง) อยู่ 3 ใบขนาดไม่เท่ากัน เสียบในเสาด้านซ้าย วางเรียงจานใบเล็กวางทับอยู่บนใบใหญ่กว่า จากบนลงล่าง (รูปที่ 1 บนสุด) ปัญหาที่อยากรู้คือ จะย้ายจานทั้ง 3 ใบอย่างไรจากต้นซ้ายมายังต้นขวา (รูปที่ 1 ล่างสุด) โดยมีเงื่อนไขคือ ย้ายได้ทีละใบ และใบใหญ่ห้ามทับใบเล็ก คำตอบคือให้ย้ายดังลำดับภาพที่แสดงในรูปที่ 1 จากบนลงล่าง

กำหนดให้จานแต่ละใบมีหมายเลขตามขนาดจาน ใบเล็กหมายเลข 1 ฟังก์ชัน `hanoi(n)` ข้างล่างนี้แสดงลำดับการย้ายจานให้ได้ตามต้องการ (โดยสามารถใช้ได้กับกรณีจาน  $n$  ใบใดๆ  $n \geq 0$ )

```
def hanoi(n, left, mid, right):
    if n == 0: return
    hanoi(n-1, left, right, mid)
    print(n, ': ', left, '->', right)
    hanoi(n-1, mid, left, right)
```

`hanoi(3, 'L', 'M', 'R')` # ได้ผลแสดงทางขวานี้

```
1 : L -> R
2 : L -> M
1 : R -> M
3 : L -> R
1 : M -> L
2 : M -> R
1 : L -> R
```

โปรแกรมข้างบนนี้มีหลักการทำงานคือ ให้ `hanoi(n, left, mid, right)` หาวิธีย้ายจานหมายเลข 1 ถึง  $n$  จากเสา `left` ไปเสา `right` โดยใช้เสา `mid` เป็นเสาชั่วคราวระหว่างการย้าย จะได้ว่า

`hanoi(n, left, mid, right)` หาวิธีการย้ายโดย

1. ทำ `hanoi(n-1, left, right, mid)` คือย้ายจานหมายเลข 1 ถึง  $n-1$  จาก `left` ไป `mid` ให้ได้ โดยใบใหญ่สุดหมายเลข  $n$  อยู่ที่เสา `left` ตลอดเวลา
2. จากนั้นก็ย้ายจานหมายเลข  $n$  จาก `left` ไป `right`
3. ตามด้วยการทำ `hanoi(n-1, mid, left, right)` คือย้ายจานหมายเลข 1 ถึง  $n-1$  ไปจาก `mid` (ที่เป็นผลจากขั้นตอนที่ 1) ไป `right` (ซึ่งก็จะทับใบ  $n$  ที่เราย้ายในขั้นตอนที่ 2)

ปัญหาข้างบนนี้มีชื่อว่า **Tower of Hanoi** แต่ปัญหาที่จะให้เขียนในโจทย์นี้เรียกว่า **Double Hanoi** เหมือน **Tower of Hanoi** บวกเงื่อนไขเพิ่มเติมคือ มีจาน  $2n$  ใบ จานแต่ละหมายเลขจะมี 2 ใบ (แต่ละขนาดมี 2 ใบ) แต่มีสีต่างกัน ใบหนึ่งดำ อีกใบหนึ่งขาว (จึงมีจานหมายเลข  $1W, 1B, 2W, 2B, \dots, nW, nB$ )

- ตอนเริ่มต้น จานวางทับกันจากบนลงล่าง ใบบนไม่ใหญ่กว่าใบล่าง และวางสลับสีกัน
- ตอนย้ายจาน ก็เหมือนเดิม ใบใหญ่ห้ามทับใบเล็ก
- เป้าหมายคือ ย้ายจานทั้ง  $2n$  ใบ จาก เสาซ้าย ไปไว้ที่เสาขวา ที่วางในลักษณะเดียวกัน

ขอเสนอวิธีการย้ายแบบง่าย ๆ (ที่ไม่ได้จำนวนครั้งการย้ายที่น้อยสุดนะ) ให้ทำตามรูปที่ 2 ทางขวานี้

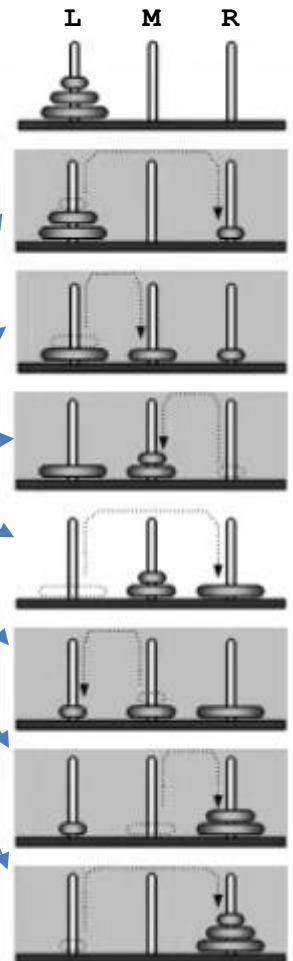
1. เริ่มด้วยทำ **Double Hanoi** เพื่อย้ายจาน  $1B, 1W$  ถึง  $(n-1)B, (n-1)W$  จากเสาซ้าย ไป เสาขวา
2. ย้ายจาน หมายเลข  $nW$  จากเสาซ้ายไปกลาง ตามด้วยย้ายจาน หมายเลข  $nB$  จาก เสาซ้ายไปกลาง
3. ทำ **Double Hanoi** เพื่อย้ายจาน  $1B, 1W$  ถึง  $(n-1)B, (n-1)W$  จากเสาขวา กลับมา เสาซ้าย
4. และ 5 น่าจะเดาได้จากรูป ว่าต้องทำอะไร

จงปรับฟังก์ชันข้างล่างนี้ให้ทำตามขั้นตอนที่กำหนดไว้

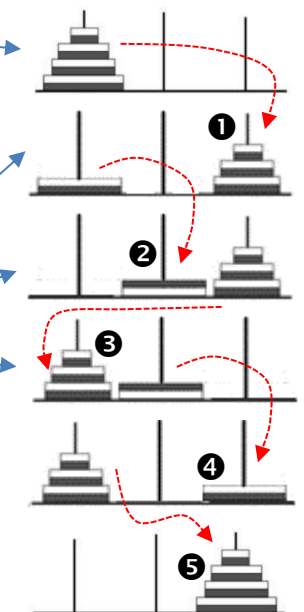
```
def dhanoi(n, left, mid, right):
    if n == 0: return
    dhanoi(n-1, _____)
    print(str(n)+'W', ': ', left, '->', _____)
    print(str(n)+'B', ': ', left, '->', _____)
    dhanoi(n-1, _____)
    ???
```

```
# ①
# ②
# ②
# ③
# ④
# ④
# ⑤
```

`exec(input().strip())` # don't remove this line



รูปที่ 1 Tower of Hanoi



รูปที่ 2 Double Hanoi

ข้อมูลนำเข้า

คำสั่งในการทดสอบฟังก์ชันที่เขียน

ข้อมูลส่งออก

ผลที่ได้จากคำสั่งที่ป้อนเป็นข้อมูลนำเข้า

ตัวอย่าง

input	Output
<code>dhanoi (2, 'L', 'M', 'R')</code>	1W : L -> M 1B : L -> M 1B : M -> R 1W : M -> R 2W : L -> M 2B : L -> M 1W : R -> M 1B : R -> M 1B : M -> L 1W : M -> L 2B : M -> R 2W : M -> R 1W : L -> M 1B : L -> M 1B : M -> R 1W : M -> R
<code>dhanoi (3, 'L', 'M', 'R')</code>	1W : L -> M 1B : L -> M 1B : M -> R 1W : M -> R 2W : L -> M 2B : L -> M 1W : R -> M 1B : R -> M 1B : M -> L 1W : M -> L 2B : M -> R 2W : M -> R 1W : L -> M 1B : L -> M 1B : M -> R 1W : M -> R 3W : L -> M 3B : L -> M 1W : R -> M 1B : R -> M 1B : M -> L 1W : M -> L 2W : R -> M 2B : R -> M 1W : L -> M 1B : L -> M 1B : M -> R 1W : M -> R 2B : M -> L 2W : M -> L 1W : R -> M 1B : R -> M 1B : M -> L 1W : M -> L 3B : M -> R 3W : M -> R 1W : L -> M 1B : L -> M 1B : M -> R 1W : M -> R 2W : L -> M 2B : L -> M 1W : R -> M 1B : R -> M 1B : M -> L 1W : M -> L 2B : M -> R 2W : M -> R 1W : L -> M 1B : L -> M 1B : M -> R 1W : M -> R