

ZeroJudge f580 2. 骰子

ZeroJudge f580 2. 骰子

<https://zerojudge.tw/ShowProblem?problemid=f580>

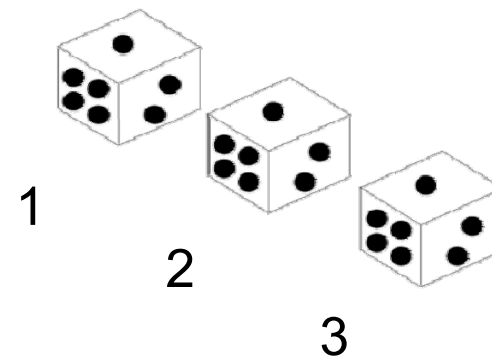
109/07/04 APCS #2 近似題

丁培毅

110/08

問題描述

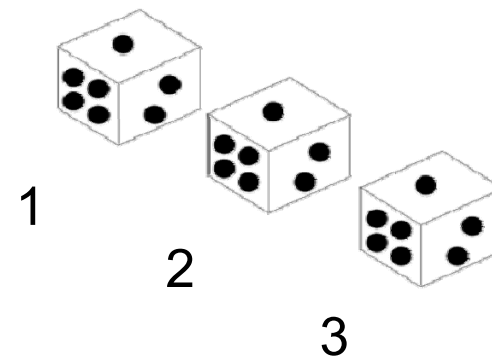
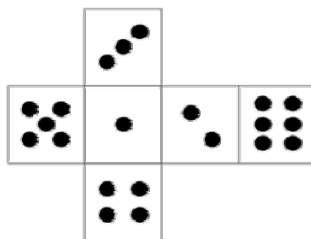
- n 個骰子排成一行，一開始都是
點數 1 朝上、點數 4 朝前、點數 2 朝右



問題描述

- n 個骰子排成一行，一開始都是
點數 1 朝上、點數 4 朝前、點數 2 朝右

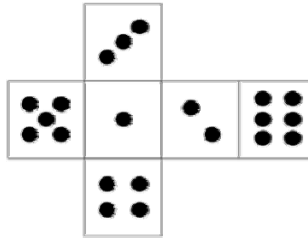
- 骰子的展開圖如右



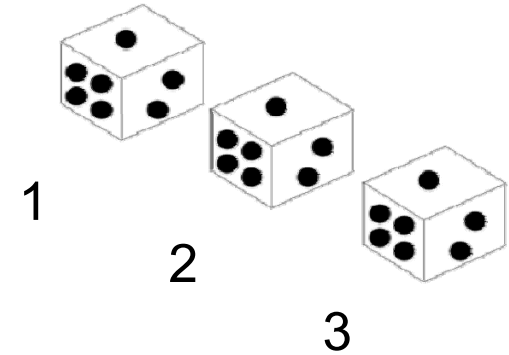
問題描述

- n 個骰子排成一行，一開始都是點數 1 朝上、點數 4 朝前、點數 2 朝右

- 骰子的展開圖如右

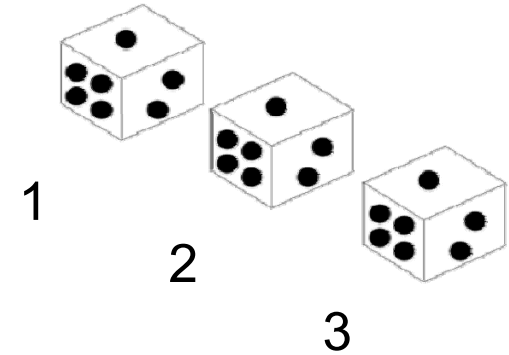


- 對這些骰子進行 m 次的操作，每個操作用兩個整數 a, b 來描述

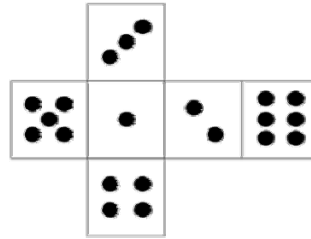


問題描述

- n 個骰子排成一列，一開始都是
點數 1 朝上、點數 4 朝前、點數 2 朝右



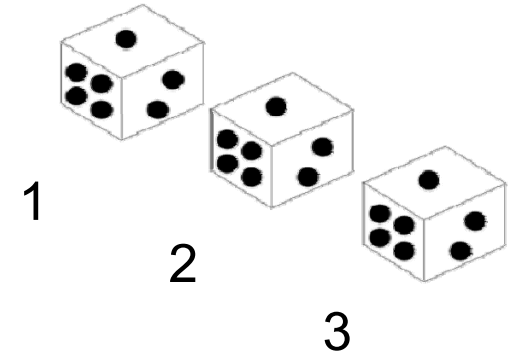
- 骰子的展開圖如右



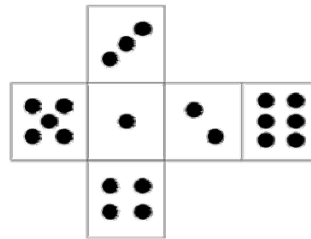
- 對這些骰子進行 m 次的操作，
每個操作用兩個整數 a, b 來描述
 1. 若 $a, b > 0$ ，**交換**位置 a 和位置 b 兩個骰子

問題描述

- n 個骰子排成一行，一開始都是
點數 1 朝上、點數 4 朝前、點數 2 朝右



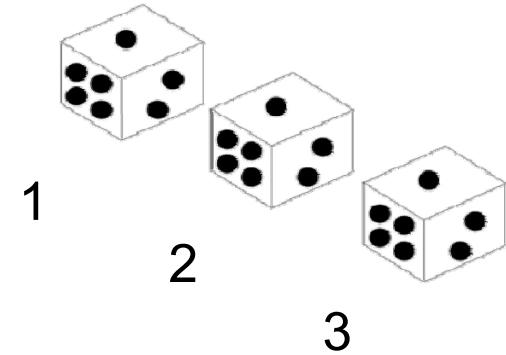
- 骰子的展開圖如右



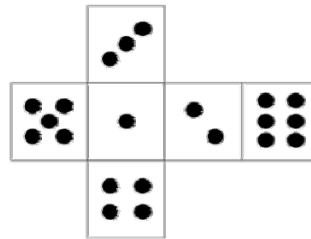
- 對這些骰子進行 m 次的操作，
每個操作用兩個整數 a, b 來描述
 1. 若 $a, b > 0$ ，**交換**位置 a 和位置 b 兩個骰子
 2. 若 b 為 -1 ，將位置 a 的骰子**向前旋轉**

問題描述

- n 個骰子排成一列，一開始都是點數 1 朝上、點數 4 朝前、點數 2 朝右



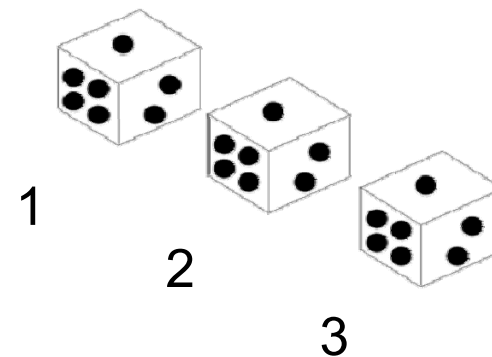
- 骰子的展開圖如右



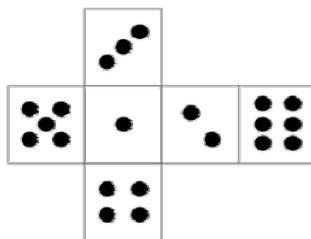
- 對這些骰子進行 m 次的操作，每個操作用兩個整數 a, b 來描述
 1. 若 $a, b > 0$ ，**交換**位置 a 和位置 b 兩個骰子
 2. 若 b 為 -1 ，將位置 a 的骰子**向前旋轉**
 3. 若 b 為 -2 ，將位置 a 的骰子**向右旋轉**

問題描述

- n 個骰子排成一列，一開始都是點數 1 朝上、點數 4 朝前、點數 2 朝右



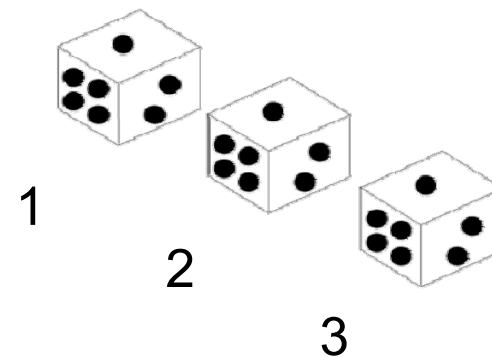
- 骰子的展開圖如右



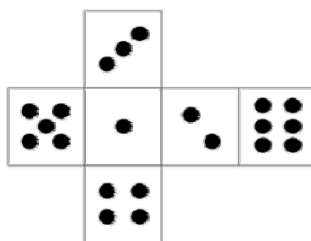
- 對這些骰子進行 m 次的操作，每個操作用兩個整數 a, b 來描述
 1. 若 $a, b > 0$ ，**交換**位置 a 和位置 b 兩個骰子
 2. 若 b 為 -1 ，將位置 a 的骰子**向前旋轉**
 3. 若 b 為 -2 ，將位置 a 的骰子**向右旋轉**
- m 次的操作結束後輸出位置 1 到 n 的骰子朝上的點數

問題描述

- n 個骰子排成一列，一開始都是點數 1 朝上、點數 4 朝前、點數 2 朝右



- 骰子的展開圖如右



- 對這些骰子進行 m 次的操作，
每個操作用兩個整數 a, b 來描述
 1. 若 $a, b > 0$ ，**交換**位置 a 和位置 b 兩個骰子
 2. 若 b 為 -1 ，將位置 a 的骰子**向前旋轉**
 3. 若 b 為 -2 ，將位置 a 的骰子**向右旋轉**
- m 次的操作結束後輸出位置
1 到 n 的骰子朝上的點數

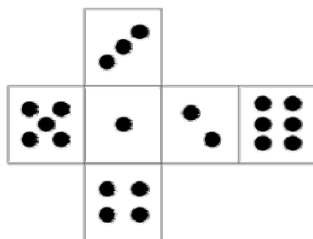
$$1 \leq n \leq 20$$

$$1 \leq m \leq 100$$

問題描述

- n 個骰子排成一列，一開始都是
點數 1 朝上、點數 4 朝前、點數 2 朝右

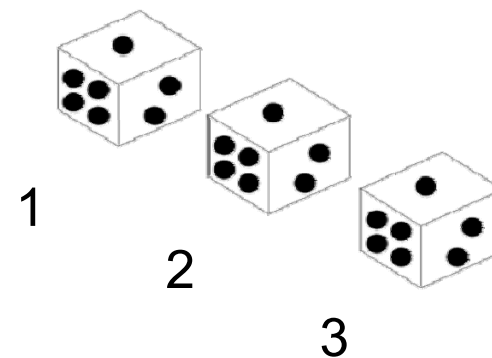
- 骰子的展開圖如右



- 對這些骰子進行 m 次的操作，
每個操作用兩個整數 a, b 來描述
 1. 若 $a, b > 0$ ，**交換**位置 a 和位置 b 兩個骰子
 2. 若 b 為 -1 ，將位置 a 的骰子**向前旋轉**
 3. 若 b 為 -2 ，將位置 a 的骰子**向右旋轉**
- m 次的操作結束後輸出位置
1 到 n 的骰子朝上的點數

$$1 \leq n \leq 20$$

$$1 \leq m \leq 100$$

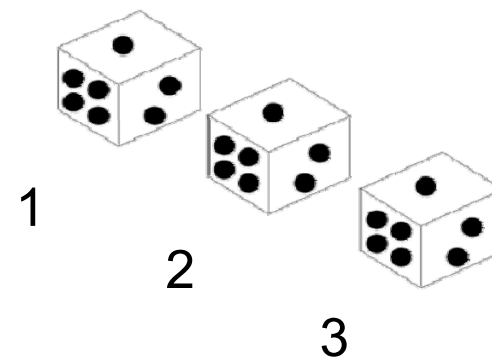


Sample Input

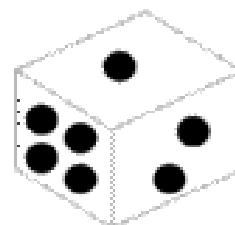
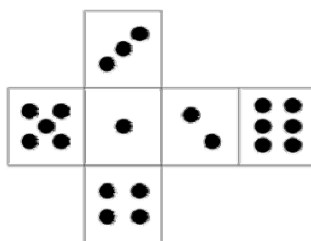
```
1 2    n m
1 -2   右轉
1 -1   前轉
```

問題描述

- n 個骰子排成一列，一開始都是
點數 1 朝上、點數 4 朝前、點數 2 朝右



- 骰子的展開圖如右



Sample Input		
1 2	n	m
1 -2	右轉	
1 -1	前轉	

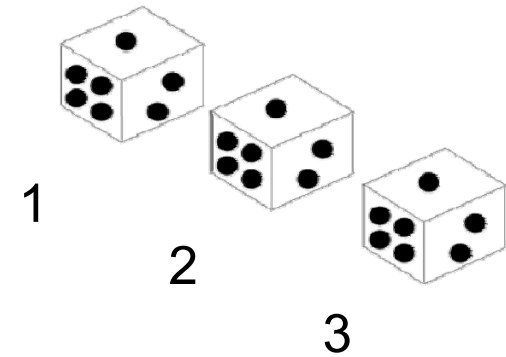
- 對這些骰子進行 m 次的操作，
每個操作用兩個整數 a, b 來描述
 - 若 $a, b > 0$ ，**交換**位置 a 和位置 b 兩個骰子
 - 若 b 為 -1 ，將位置 a 的骰子**向前旋轉**
 - 若 b 為 -2 ，將位置 a 的骰子**向右旋轉**
- m 次的操作結束後輸出位置
1 到 n 的骰子朝上的點數

$$1 \leq n \leq 20$$

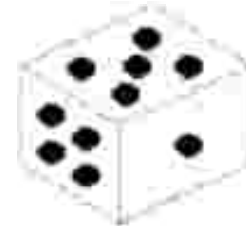
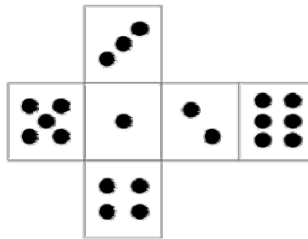
$$1 \leq m \leq 100$$

問題描述

- n 個骰子排成一列，一開始都是
點數 1 朝上、點數 4 朝前、點數 2 朝右



- 骰子的展開圖如右



Sample Input		
1 2	n	m
1 -2	右轉	
1 -1	前轉	

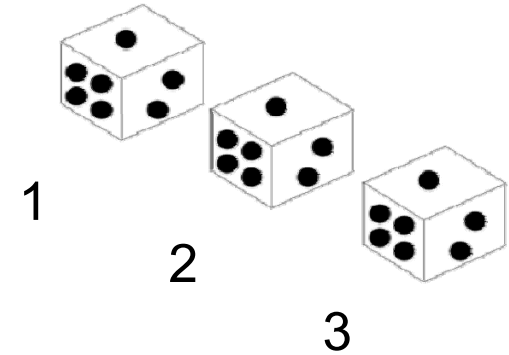
- 對這些骰子進行 m 次的操作，
每個操作用兩個整數 a, b 來描述
 - 若 $a, b > 0$ ，**交換**位置 a 和位置 b 兩個骰子
 - 若 b 為 -1 ，將位置 a 的骰子**向前旋轉**
 - 若 b 為 -2 ，將位置 a 的骰子**向右旋轉**
- m 次的操作結束後輸出位置
1 到 n 的骰子朝上的點數

$$1 \leq n \leq 20$$

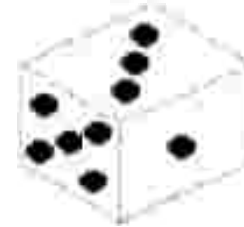
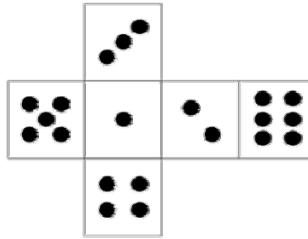
$$1 \leq m \leq 100$$

問題描述

- n 個骰子排成一列，一開始都是點數 1 朝上、點數 4 朝前、點數 2 朝右



- 骰子的展開圖如右



Sample Input		
1 2	n	m
1 -2	右轉	
1 -1	前轉	

- 對這些骰子進行 m 次的操作，每個操作用兩個整數 a, b 來描述
 - 若 $a, b > 0$ ，**交換**位置 a 和位置 b 兩個骰子
 - 若 b 為 -1 ，將位置 a 的骰子**向前旋轉**
 - 若 b 為 -2 ，將位置 a 的骰子**向右旋轉**
- m 次的操作結束後輸出位置 1 到 n 的骰子朝上的點數

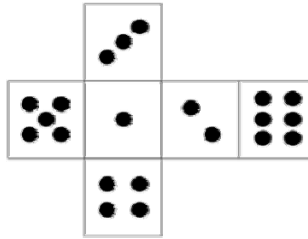
$$1 \leq n \leq 20$$

$$1 \leq m \leq 100$$

問題描述

- n 個骰子排成一列，一開始都是點數 1 朝上、點數 4 朝前、點數 2 朝右

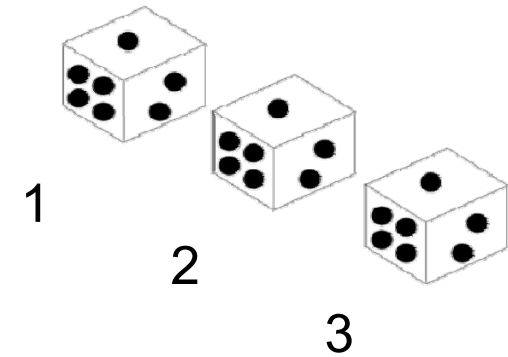
- 骰子的展開圖如右



- 對這些骰子進行 m 次的操作，每個操作用兩個整數 a, b 來描述
 - 若 $a, b > 0$ ，**交換**位置 a 和位置 b 兩個骰子
 - 若 b 為 -1 ，將位置 a 的骰子**向前旋轉**
 - 若 b 為 -2 ，將位置 a 的骰子**向右旋轉**
- m 次的操作結束後輸出位置 1 到 n 的骰子朝上的點數

$$1 \leq n \leq 20$$

$$1 \leq m \leq 100$$



Sample Input

1 2 n m

1 -2 右轉

1 -1 前轉

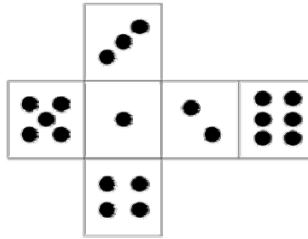
Sample Output

3

問題描述

- n 個骰子排成一列，一開始都是點數 1 朝上、點數 4 朝前、點數 2 朝右

- 骰子的展開圖如右



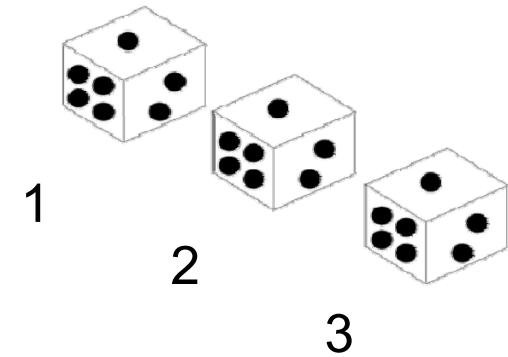
- 對這些骰子進行 m 次的操作，每個操作用兩個整數 a, b 來描述

- 若 $a, b > 0$ ，**交換**位置 a 和位置 b 兩個骰子
- 若 b 為 -1 ，將位置 a 的骰子**向前旋轉**
- 若 b 為 -2 ，將位置 a 的骰子**向右旋轉**

- m 次的操作結束後輸出位置 1 到 n 的骰子朝上的點數

$$1 \leq n \leq 20$$

$$1 \leq m \leq 100$$



Sample Input

```
1 2    n m
1 -2   右轉
1 -1   前轉
```

Sample Output

3

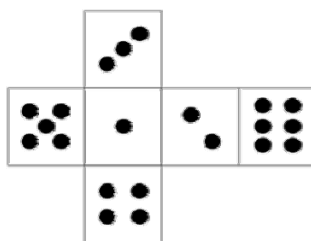
Sample Input

```
3 3
2 -1
3 -2
3 1
```

問題描述

- n 個骰子排成一列，一開始都是點數 1 朝上、點數 4 朝前、點數 2 朝右

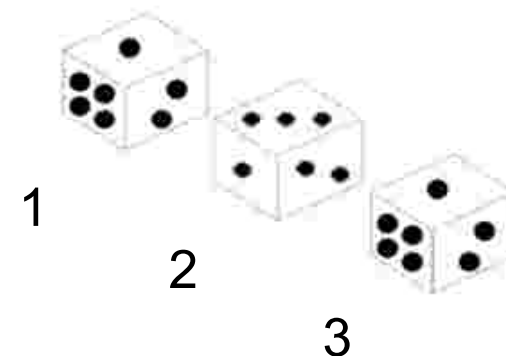
- 骰子的展開圖如右



- 對這些骰子進行 m 次的操作，每個操作用兩個整數 a, b 來描述
 - 若 $a, b > 0$ ，**交換**位置 a 和位置 b 兩個骰子
 - 若 b 為 -1 ，將位置 a 的骰子**向前旋轉**
 - 若 b 為 -2 ，將位置 a 的骰子**向右旋轉**
- m 次的操作結束後輸出位置 1 到 n 的骰子朝上的點數

$$1 \leq n \leq 20$$

$$1 \leq m \leq 100$$



Sample Input

```
1 2    n m
1 -2   右轉
1 -1   前轉
```

Sample Output

3

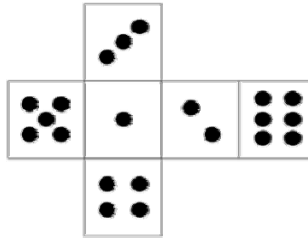
Sample Input

```
3 3
2 -1
3 -2
3 1
```


問題描述

- n 個骰子排成一列，一開始都是點數 1 朝上、點數 4 朝前、點數 2 朝右

- 骰子的展開圖如右



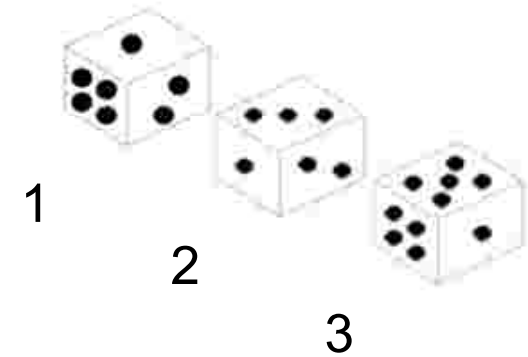
- 對這些骰子進行 m 次的操作，每個操作用兩個整數 a, b 來描述

- 若 $a, b > 0$ ，**交換**位置 a 和位置 b 兩個骰子
- 若 b 為 -1 ，將位置 a 的骰子**向前旋轉**
- 若 b 為 -2 ，將位置 a 的骰子**向右旋轉**

- m 次的操作結束後輸出位置 1 到 n 的骰子朝上的點數

$$1 \leq n \leq 20$$

$$1 \leq m \leq 100$$



Sample Input

```
1 2    n m
1 -2   右轉
1 -1   前轉
```

Sample Output

3

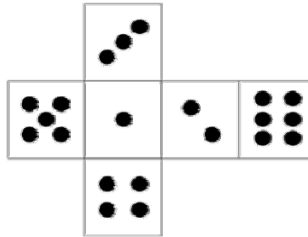
Sample Input

```
3 3
2 -1
3 -2
3 1
```

問題描述

- n 個骰子排成一列，一開始都是點數 1 朝上、點數 4 朝前、點數 2 朝右

- 骰子的展開圖如右



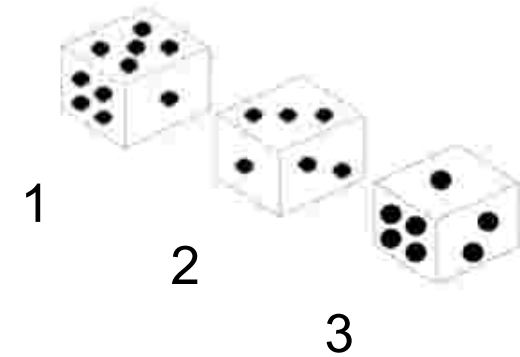
- 對這些骰子進行 m 次的操作，每個操作用兩個整數 a, b 來描述

- 若 $a, b > 0$ ，**交換**位置 a 和位置 b 兩個骰子
- 若 b 為 -1 ，將位置 a 的骰子**向前旋轉**
- 若 b 為 -2 ，將位置 a 的骰子**向右旋轉**

- m 次的操作結束後輸出位置 1 到 n 的骰子朝上的點數

$$1 \leq n \leq 20$$

$$1 \leq m \leq 100$$



Sample Input
1 2 n m
1 -2 右轉
1 -1 前轉

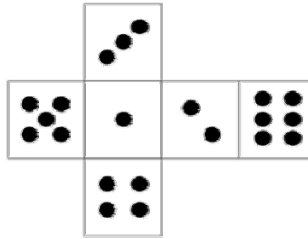
Sample Output
3

Sample Input
3 3
2 -1
3 -2
3 1

問題描述

- n 個骰子排成一列，一開始都是點數 1 朝上、點數 4 朝前、點數 2 朝右

- 骰子的展開圖如右

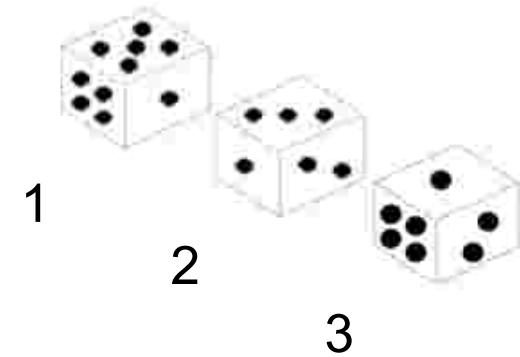


- 對這些骰子進行 m 次的操作，每個操作用兩個整數 a, b 來描述

- 若 $a, b > 0$ ，**交換**位置 a 和位置 b 兩個骰子
- 若 b 為 -1 ，將位置 a 的骰子**向前旋轉**
- 若 b 為 -2 ，將位置 a 的骰子**向右旋轉**

- m 次的操作結束後輸出位置 1 到 n 的骰子朝上的點數

$$1 \leq n \leq 20$$
$$1 \leq m \leq 100$$



Sample Input
1 2 n m
1 -2 右轉
1 -1 前轉

Sample Output
3

Sample Input
3 3
2 -1
3 -2
3 1

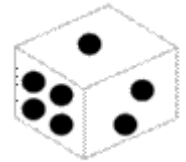
Sample Output
5 3 1

問題分析

- 在這個題目裡，每一個骰子的狀態會改變多次，所以第一件事情是**如何**在程式裡**表示**一個**骰子的狀態**

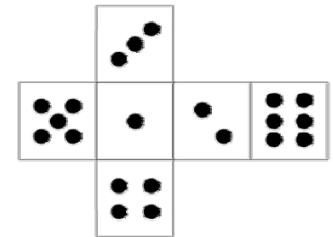
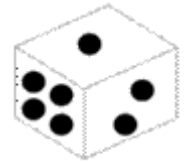
問題分析

- 在這個題目裡，每一個骰子的狀態會改變多次，所以第一件事情是**如何**在程式裡**表示**一個**骰子的狀態**
- 每個骰子都有六個面：**上前右下後左**，如果依序用**六個整數**代表這六面的點數，就可以精確地描述一個骰子目前的狀態，例如右邊這個骰子我們可以用 **142635** 來描述



問題分析

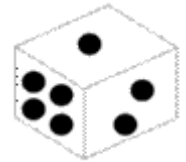
- 在這個題目裡，每一個骰子的狀態會改變多次，所以第一件事情是**如何**在程式裡**表示**一個**骰子的狀態**
- 每個骰子都有六個面：**上前右下後左**，如果依序用**六個整數**代表這六面的點數，就可以精確地描述一個骰子目前的狀態，例如右邊這個骰子我們可以用 **142635** 來描述
- 其中**下後左**這三面雖然從立體透視圖中沒辦法看到，但是可以在展開圖上看到 **5** 點的對面是 **2** 點，**1** 點的對面是 **6** 點，**4** 點的對面是 **3** 點，這個規律不論骰子**怎麼轉都不會改變**，也就是**相對兩面的點數和是 7**



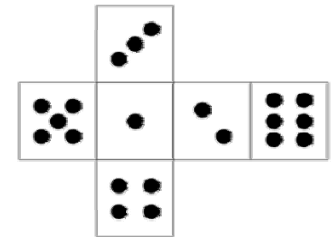
問題分析

- 在這個題目裡，每一個骰子的狀態會改變多次，所以第一件事情是**如何**在程式裡**表示**一個**骰子的狀態**

- 每個骰子都有六個面：**上前右下後左**，如果依序用**六個整數**代表這六面的點數，就可以精確地描述一個骰子目前的狀態，例如右邊這個骰子我們可以用 **142635** 來描述



- 其中**下後左**這三面雖然從立體透視圖中沒辦法看到，但是可以在展開圖上看到 **5** 點的對面是 **2** 點，**1** 點的對面是 **6** 點，**4** 點的對面是 **3** 點，這個規律不論骰子**怎麼轉都不會改變**，也就是**相對兩面的點數和是 7**

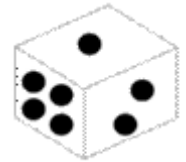


- 所以在描述一個骰子的狀態時，其實只需要描述**上前右**三面的點數就夠了，完全不需要描述另外三面，所以題目說一開始有 n 個骰子，只描述看得到的三個面的點數而已：上 **1**、前 **4**、右 **2**

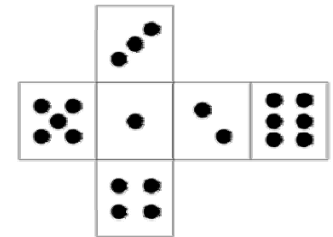
問題分析

- 在這個題目裡，每一個骰子的狀態會改變多次，所以第一件事情是**如何**在程式裡**表示**一個**骰子的狀態**

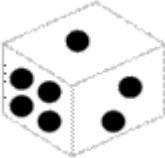
- 每個骰子都有六個面：**上前右下後左**，如果依序用**六個整數**代表這六面的點數，就可以精確地描述一個骰子目前的狀態，例如右邊這個骰子我們可以用 **142635** 來描述



- 其中**下後左**這三面雖然從立體透視圖中沒辦法看到，但是可以在展開圖上看到 **5** 點的對面是 **2** 點，**1** 點的對面是 **6** 點，**4** 點的對面是 **3** 點，這個規律不論骰子**怎麼轉都不會改變**，也就是**相對兩面的點數和是 7**



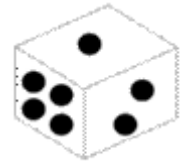
- 所以在描述一個骰子的狀態時，其實只需要描述**上前右**三面的點數就夠了，完全不需要描述另外三面，所以題目說一開始有 n 個骰子，只描述看得到的三個面的點數而已：上 **1**、前 **4**、右 **2**

- 骰子的狀態 

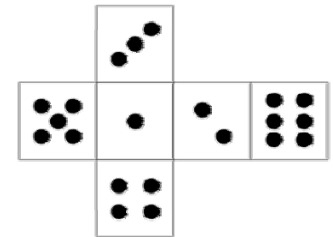
問題分析

- 在這個題目裡，每一個骰子的狀態會改變多次，所以第一件事情是**如何**在程式裡**表示**一個**骰子的狀態**

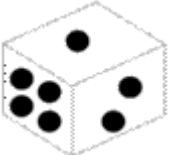
- 每個骰子都有六個面：**上前右下後左**，如果依序用**六個整數**代表這六面的點數，就可以精確地描述一個骰子目前的狀態，例如右邊這個骰子我們可以用 **142635** 來描述

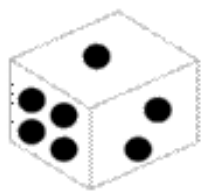


- 其中**下後左**這三面雖然從立體透視圖中沒辦法看到，但是可以在展開圖上看到 **5** 點的對面是 **2** 點，**1** 點的對面是 **6** 點，**4** 點的對面是 **3** 點，這個規律不論骰子**怎麼轉都不會改變**，也就是**相對兩面的點數和是 7**



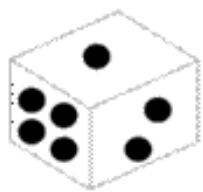
- 所以在描述一個骰子的狀態時，其實只需要描述**上前右**三面的點數就夠了，完全不需要描述另外三面，所以題目說一開始有 n 個骰子，只描述看得到的三個面的點數而已：上 **1**、前 **4**、右 **2**

- 骰子的狀態  **上前右:142**



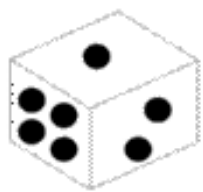
問題分析 (續)

- 骰子的狀態有多少種？



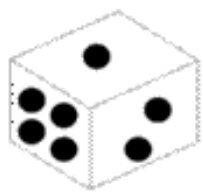
問題分析 (續)

- 骰子的狀態有多少種？ 是 $6*5*4=120$ 嗎？



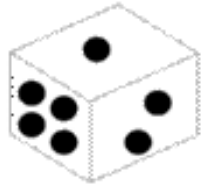
問題分析 (續)

- 骰子的狀態有多少種？ 是 $6*5*4=120$ 嗎？ 不是



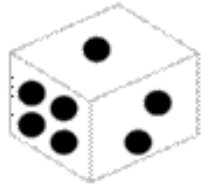
問題分析 (續)

- 骰子的狀態有多少種？ 是 $6*5*4=120$ 嗎？ 不是
骰子旋轉時每一面的四個相鄰面點數都是固定的，並不是
六面拆開來隨便組合的



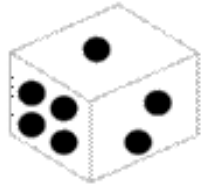
問題分析 (續)

- 骰子的狀態有多少種？ 是 $6*5*4=120$ 嗎？ 不是
骰子旋轉時每一面的四個相鄰面點數都是固定的，並不是六面拆開來隨便組合的
隨機挑選一個點數作為前面的點數，然後向右旋轉四次得到四種不同的狀態，所以是 $6*4=24$ 種不同的狀態



問題分析 (續)

- 骰子的狀態有多少種？ 是 $6*5*4=120$ 嗎？ 不是
骰子旋轉時每一面的四個相鄰面點數都是固定的，並不是六面拆開來隨便組合的
隨機挑選一個點數作為前面的點數，然後向右旋轉四次得到四種不同的狀態，所以是 $6*4=24$ 種不同的狀態
- 模擬：骰子旋轉時狀態如何改變？



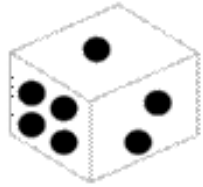
問題分析 (續)

- 骰子的狀態有多少種？ 是 $6*5*4=120$ 嗎？ 不是
骰子旋轉時每一面的四個相鄰面點數都是固定的，並不是六面拆開來隨便組合的
隨機挑選一個點數作為前面的點數，然後向右旋轉四次得到四種不同的狀態，所以是 $6*4=24$ 種不同的狀態
- 模擬：骰子旋轉時狀態如何改變？

- 前旋



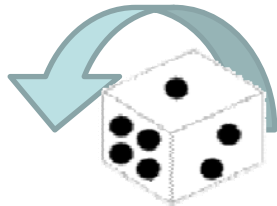
上前右: 142



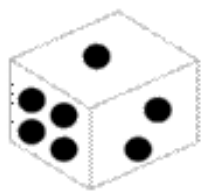
問題分析 (續)

- 骰子的狀態有多少種？ 是 $6*5*4=120$ 嗎？ 不是
骰子旋轉時每一面的四個相鄰面點數都是固定的，並不是六面拆開來隨便組合的
隨機挑選一個點數作為前面的點數，然後向右旋轉四次得到四種不同的狀態，所以是 $6*4=24$ 種不同的狀態
- 模擬：骰子旋轉時狀態如何改變？

- 前旋



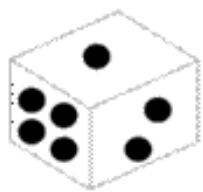
上前右: 142



問題分析 (續)

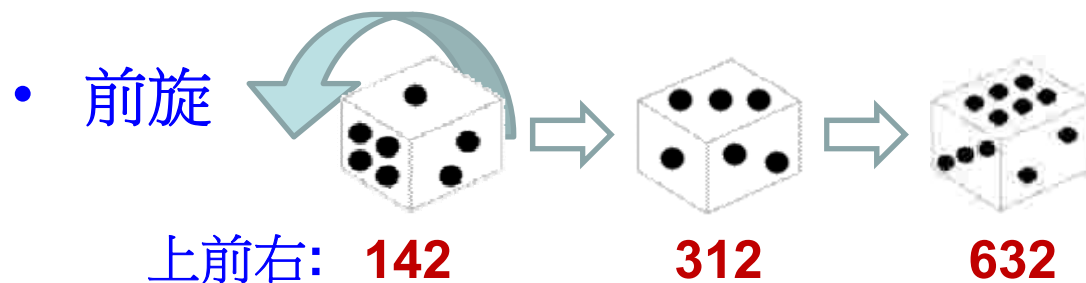
- 骰子的狀態有多少種？ 是 $6*5*4=120$ 嗎？ 不是
骰子旋轉時每一面的四個相鄰面點數都是固定的，並不是六面拆開來隨便組合的
隨機挑選一個點數作為前面的點數，然後向右旋轉四次得到四種不同的狀態，所以是 $6*4=24$ 種不同的狀態
- 模擬：骰子旋轉時狀態如何改變？

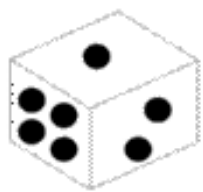




問題分析 (續)

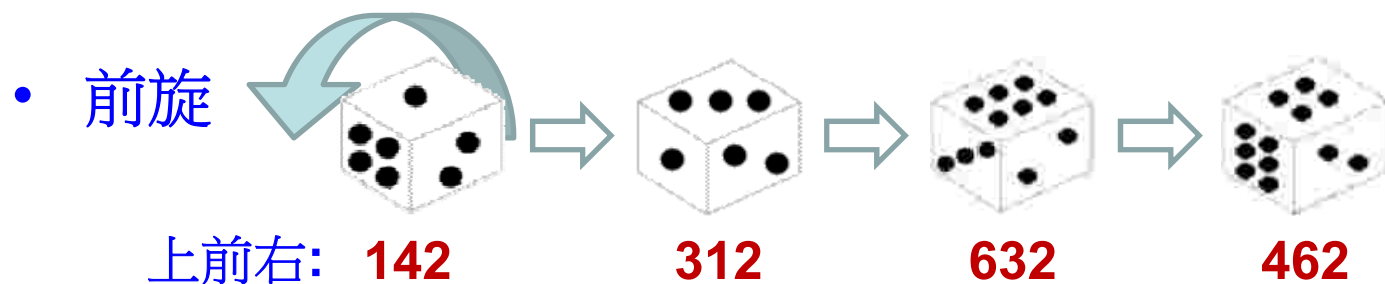
- 骰子的狀態有多少種？ 是 $6*5*4=120$ 嗎？ 不是
骰子旋轉時每一面的四個相鄰面點數都是固定的，並不是六面拆開來隨便組合的
隨機挑選一個點數作為前面的點數，然後向右旋轉四次得到四種不同的狀態，所以是 $6*4=24$ 種不同的狀態
- 模擬：骰子旋轉時狀態如何改變？

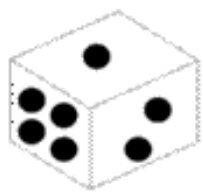




問題分析 (續)

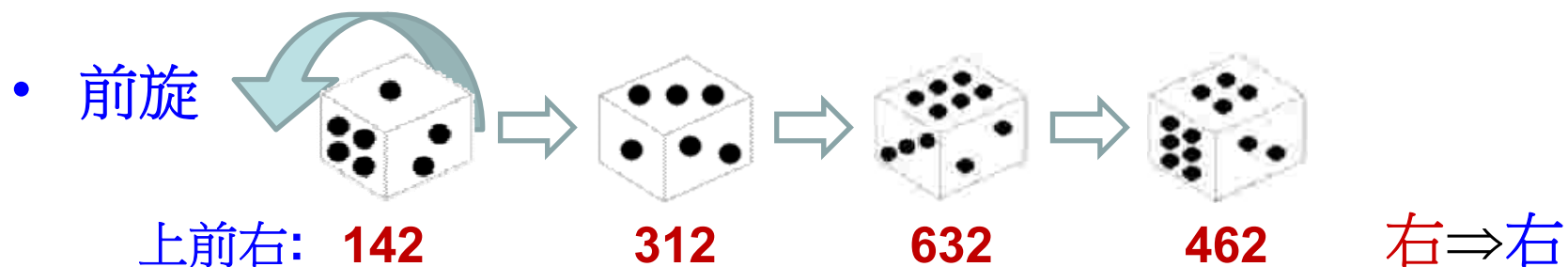
- 骰子的狀態有多少種？ 是 $6*5*4=120$ 嗎？ 不是
骰子旋轉時每一面的四個相鄰面點數都是固定的，並不是六面拆開來隨便組合的
隨機挑選一個點數作為前面的點數，然後向右旋轉四次得到四種不同的狀態，所以是 $6*4=24$ 種不同的狀態
- 模擬：骰子旋轉時狀態如何改變？

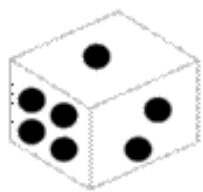




問題分析 (續)

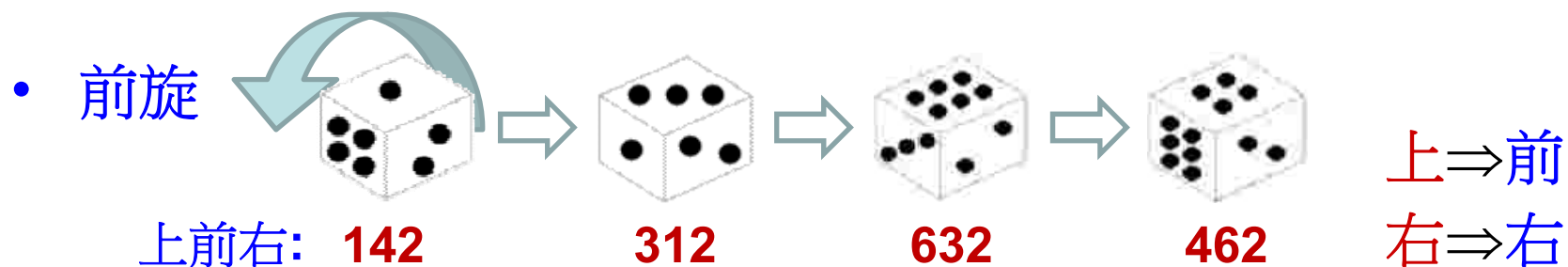
- 骰子的狀態有多少種？ 是 $6*5*4=120$ 嗎？ 不是
骰子旋轉時每一面的四個相鄰面點數都是固定的，並不是六面拆開來隨便組合的
隨機挑選一個點數作為前面的點數，然後向右旋轉四次得到四種不同的狀態，所以是 $6*4=24$ 種不同的狀態
- 模擬：骰子旋轉時狀態如何改變？

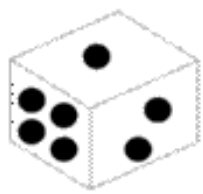




問題分析 (續)

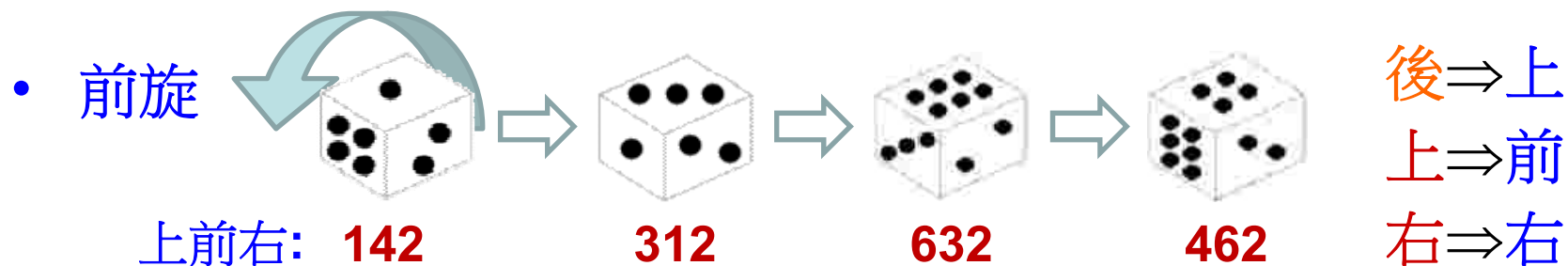
- 骰子的狀態有多少種？ 是 $6*5*4=120$ 嗎？ 不是
骰子旋轉時每一面的四個相鄰面點數都是固定的，並不是六面拆開來隨便組合的
隨機挑選一個點數作為前面的點數，然後向右旋轉四次得到四種不同的狀態，所以是 $6*4=24$ 種不同的狀態
- 模擬：骰子旋轉時狀態如何改變？

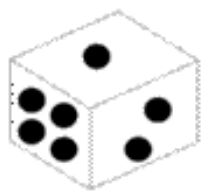




問題分析 (續)

- 骰子的狀態有多少種？ 是 $6*5*4=120$ 嗎？ 不是
骰子旋轉時每一面的四個相鄰面點數都是固定的，並不是六面拆開來隨便組合的
隨機挑選一個點數作為前面的點數，然後向右旋轉四次得到四種不同的狀態，所以是 $6*4=24$ 種不同的狀態
- 模擬：骰子旋轉時狀態如何改變？

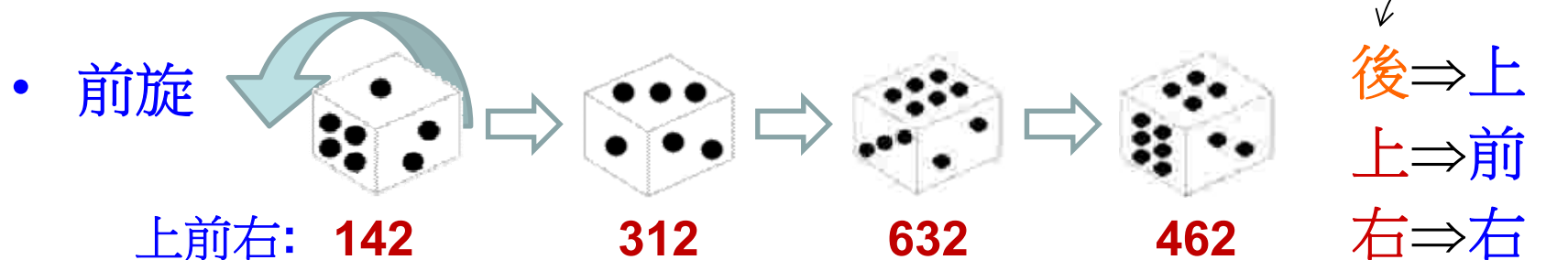


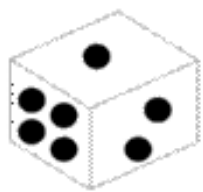


問題分析 (續)

- 骰子的狀態有多少種？ 是 $6*5*4=120$ 嗎？ 不是
骰子旋轉時每一面的四個相鄰面點數都是固定的，並不是六面拆開來隨便組合的
隨機挑選一個點數作為前面的點數，然後向右旋轉四次得到四種不同的狀態，所以是 $6*4=24$ 種不同的狀態

- 模擬：骰子旋轉時狀態如何改變？

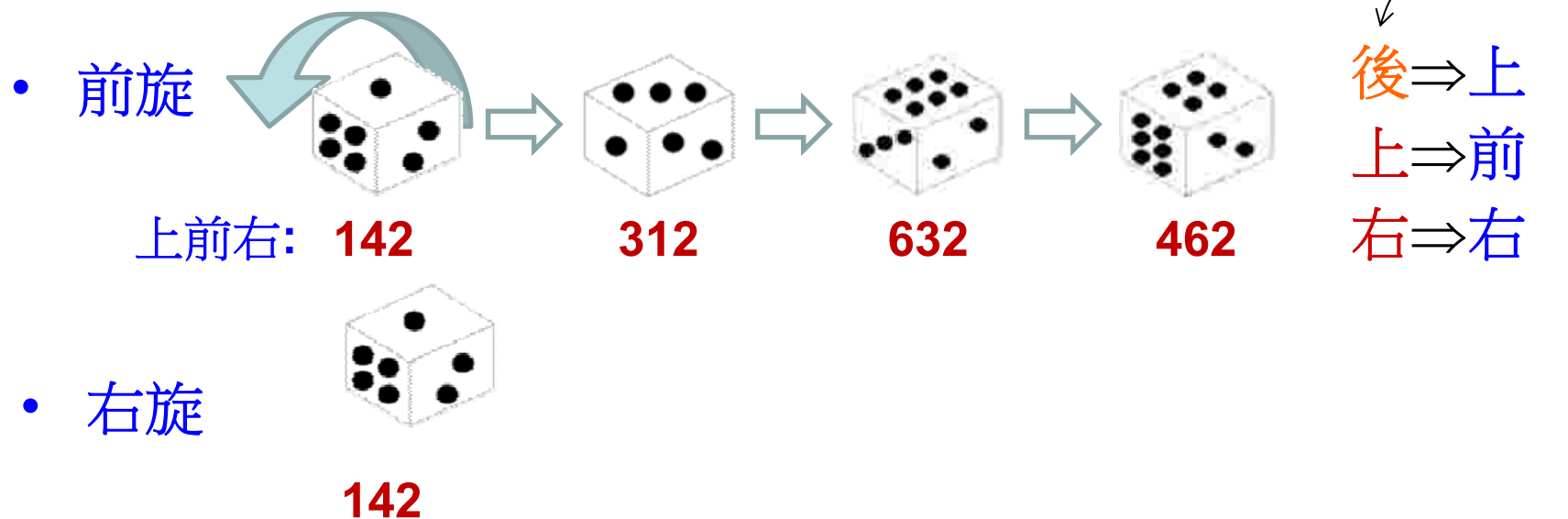


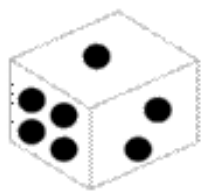


問題分析 (續)

- 骰子的狀態有多少種？ 是 $6*5*4=120$ 嗎？ 不是
骰子旋轉時每一面的四個相鄰面點數都是固定的，並不是六面拆開來隨便組合的
隨機挑選一個點數作為前面的點數，然後向右旋轉四次得到四種不同的狀態，所以是 $6*4=24$ 種不同的狀態

- 模擬：骰子旋轉時狀態如何改變？

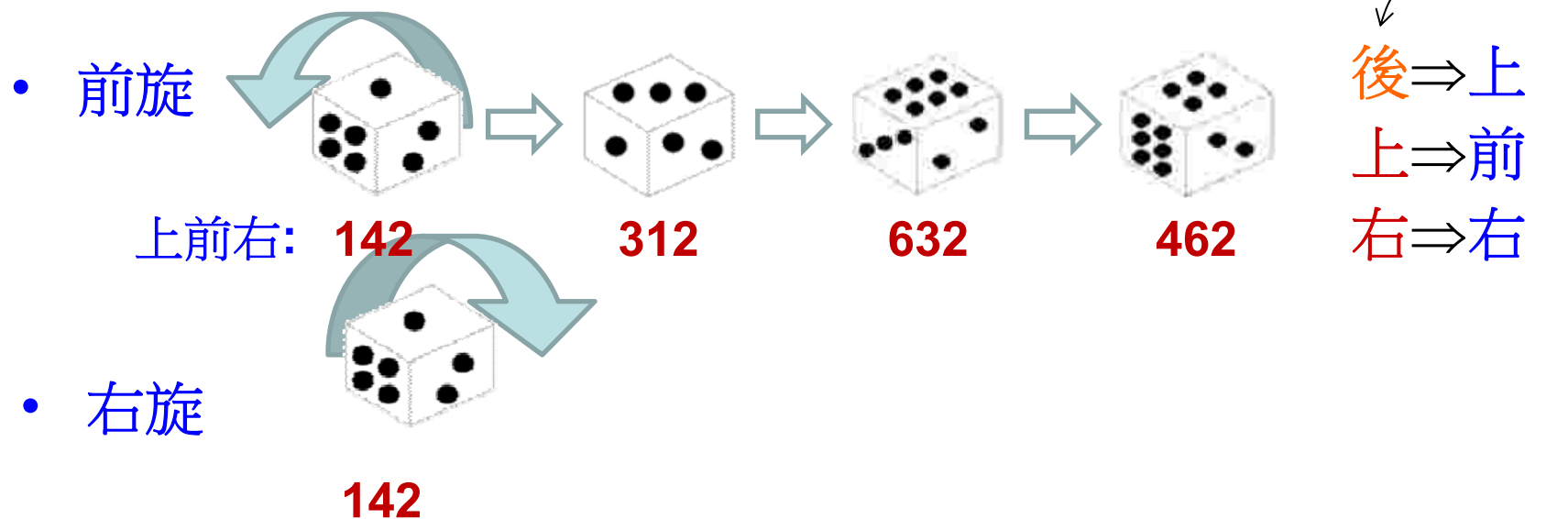


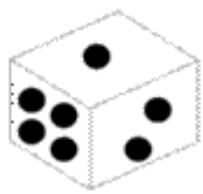


問題分析 (續)

- 骰子的狀態有多少種？ 是 $6*5*4=120$ 嗎？ 不是
骰子旋轉時每一面的四個相鄰面點數都是固定的，並不是六面拆開來隨便組合的
隨機挑選一個點數作為前面的點數，然後向右旋轉四次得到四種不同的狀態，所以是 $6*4=24$ 種不同的狀態

- 模擬：骰子旋轉時狀態如何改變？

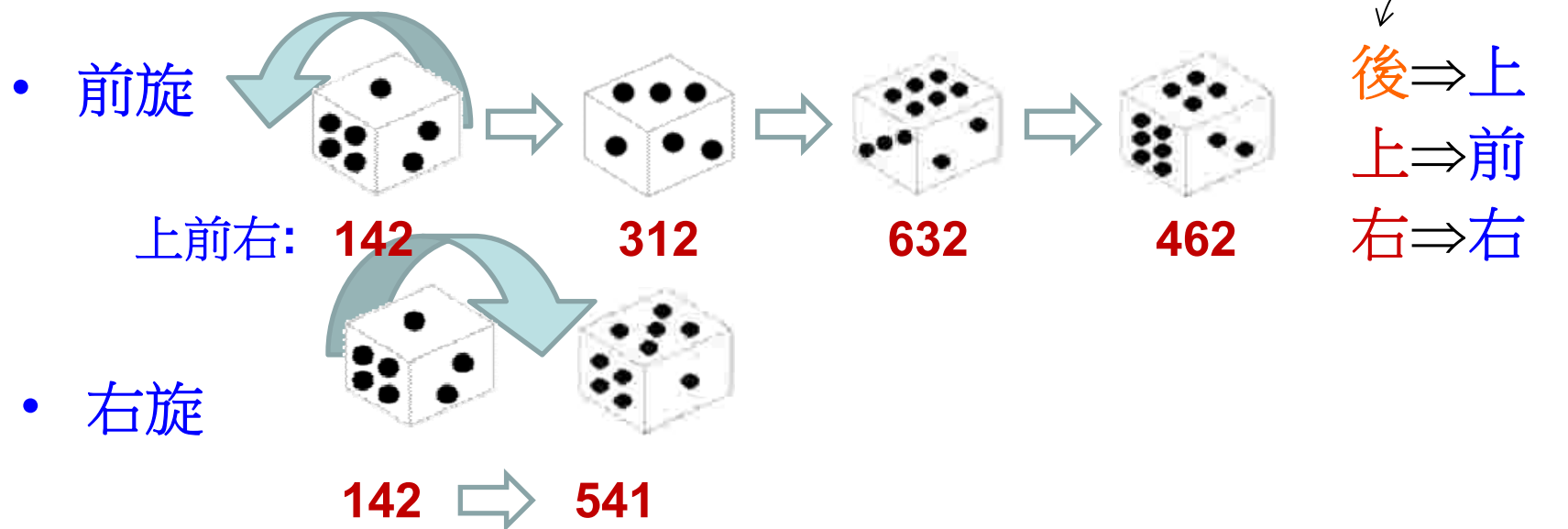


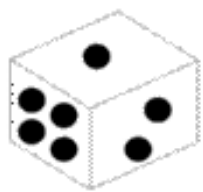


問題分析 (續)

- 骰子的狀態有多少種？ 是 $6*5*4=120$ 嗎？ 不是
骰子旋轉時每一面的四個相鄰面點數都是固定的，並不是六面拆開來隨便組合的
隨機挑選一個點數作為前面的點數，然後向右旋轉四次得到四種不同的狀態，所以是 $6*4=24$ 種不同的狀態

- 模擬：骰子旋轉時狀態如何改變？

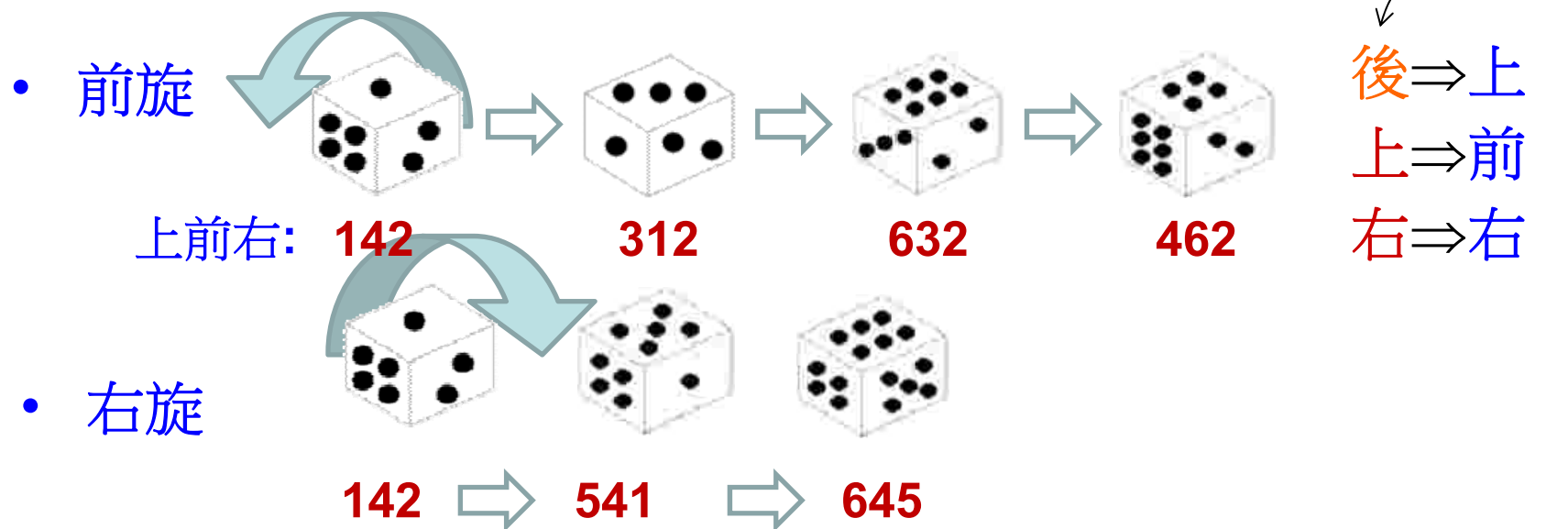


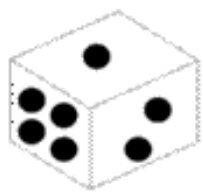


問題分析 (續)

- 骰子的**狀態**有多少種？ 是 $6*5*4=120$ 嗎？ 不是
骰子旋轉時每一面的四個相鄰面點數都是固定的，並不是六面拆開來隨便組合的
隨機挑選一個點數作為**前**面的點數，然後向右旋轉四次得到四種不同的狀態，所以是 $6*4=24$ 種不同的狀態

- **模擬**：骰子**旋轉**時**狀態**如何改變？

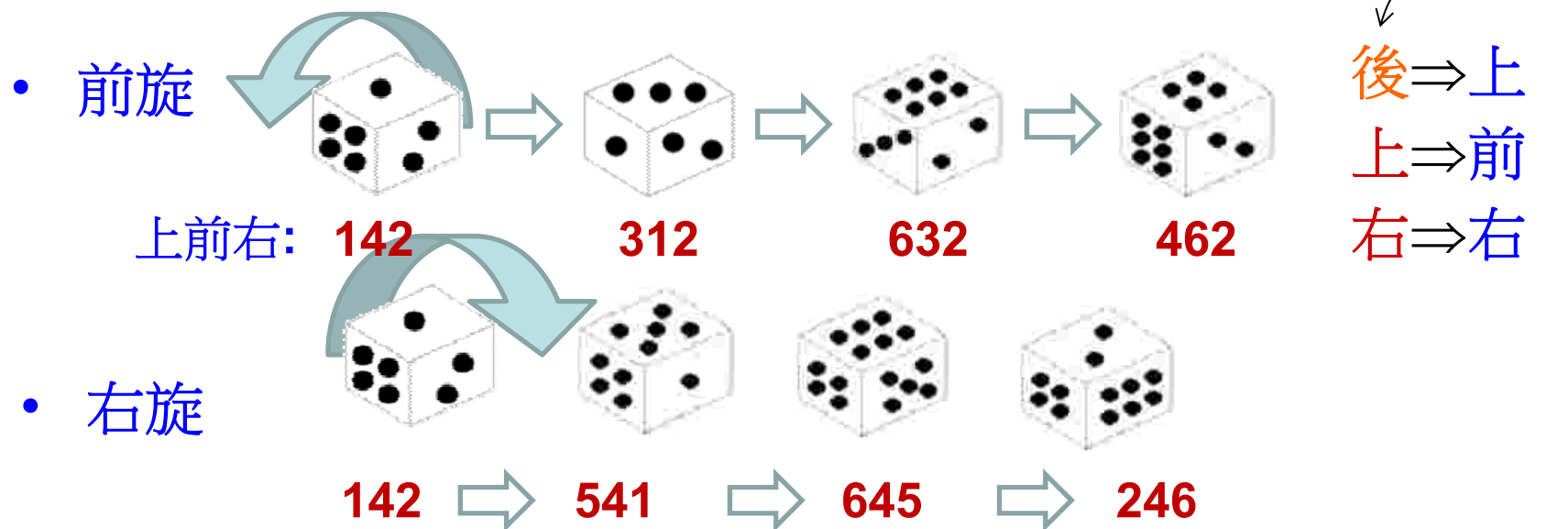


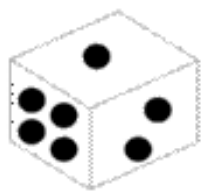


問題分析 (續)

- 骰子的狀態有多少種？ 是 $6*5*4=120$ 嗎？ 不是
骰子旋轉時每一面的四個相鄰面點數都是固定的，並不是六面拆開來隨便組合的
隨機挑選一個點數作為前面的點數，然後向右旋轉四次得到四種不同的狀態，所以是 $6*4=24$ 種不同的狀態

- 模擬：骰子旋轉時狀態如何改變？

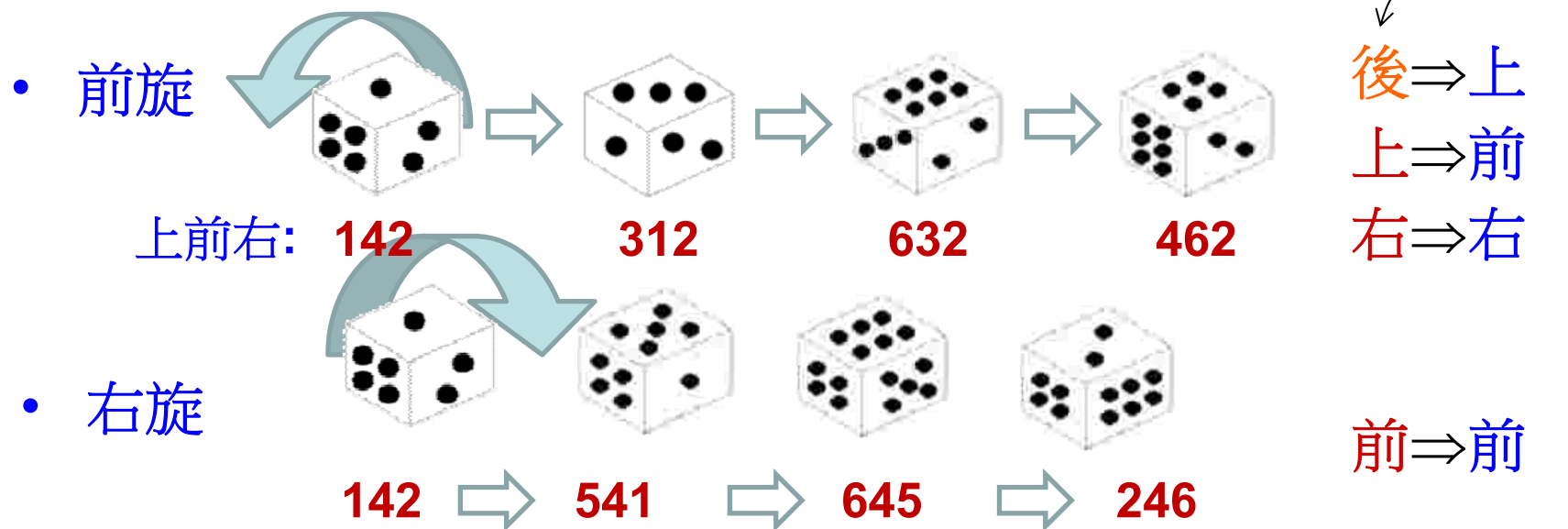


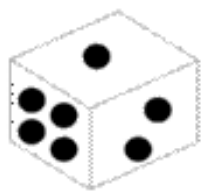


問題分析 (續)

- 骰子的狀態有多少種？ 是 $6*5*4=120$ 嗎？ 不是
骰子旋轉時每一面的四個相鄰面點數都是固定的，並不是六面拆開來隨便組合的
隨機挑選一個點數作為前面的點數，然後向右旋轉四次得到四種不同的狀態，所以是 $6*4=24$ 種不同的狀態

- 模擬：骰子旋轉時狀態如何改變？

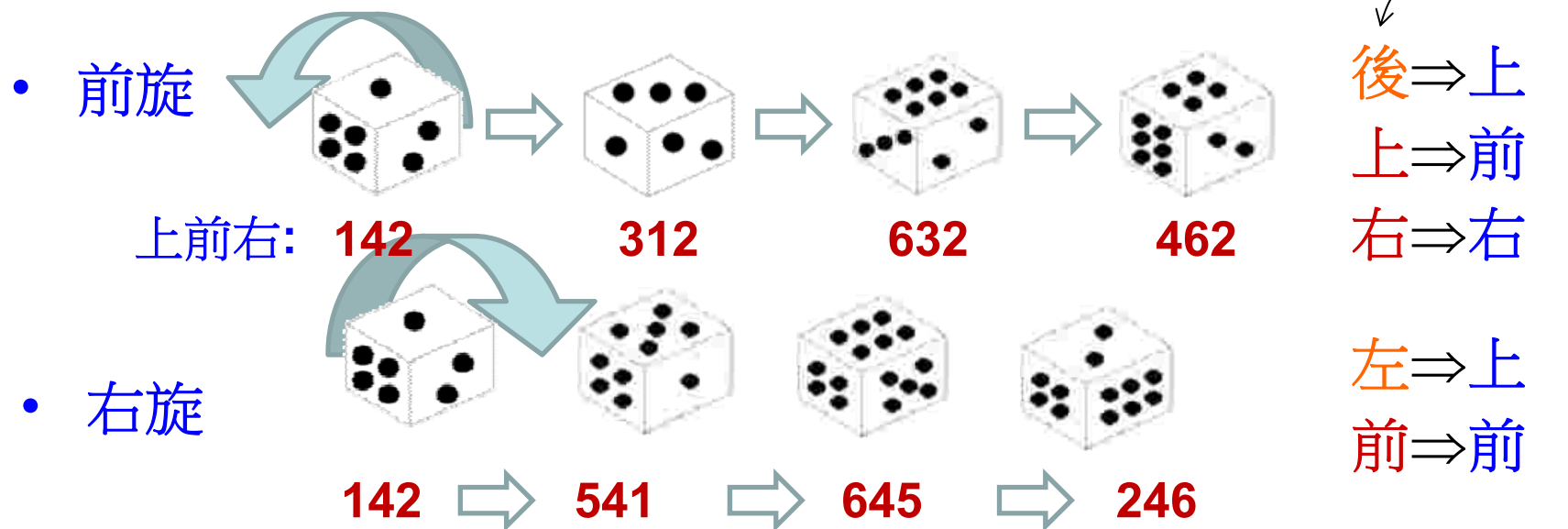


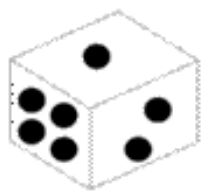


問題分析 (續)

- 骰子的狀態有多少種？ 是 $6*5*4=120$ 嗎？ 不是
骰子旋轉時每一面的四個相鄰面點數都是固定的，並不是六面拆開來隨便組合的
隨機挑選一個點數作為前面的點數，然後向右旋轉四次得到四種不同的狀態，所以是 $6*4=24$ 種不同的狀態

- 模擬：骰子旋轉時狀態如何改變？

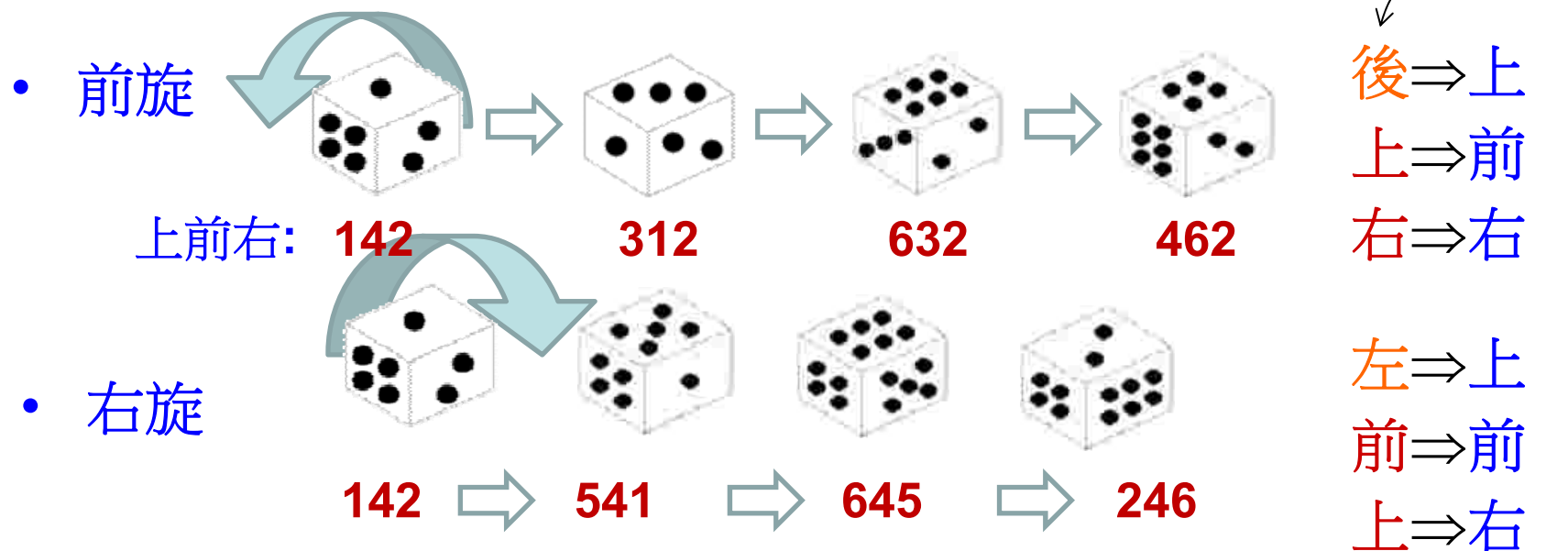


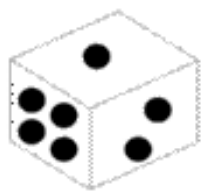


問題分析 (續)

- 骰子的狀態有多少種？ 是 $6*5*4=120$ 嗎？ 不是
骰子旋轉時每一面的四個相鄰面點數都是固定的，並不是六面拆開來隨便組合的
隨機挑選一個點數作為前面的點數，然後向右旋轉四次得到四種不同的狀態，所以是 $6*4=24$ 種不同的狀態

- 模擬：骰子旋轉時狀態如何改變？

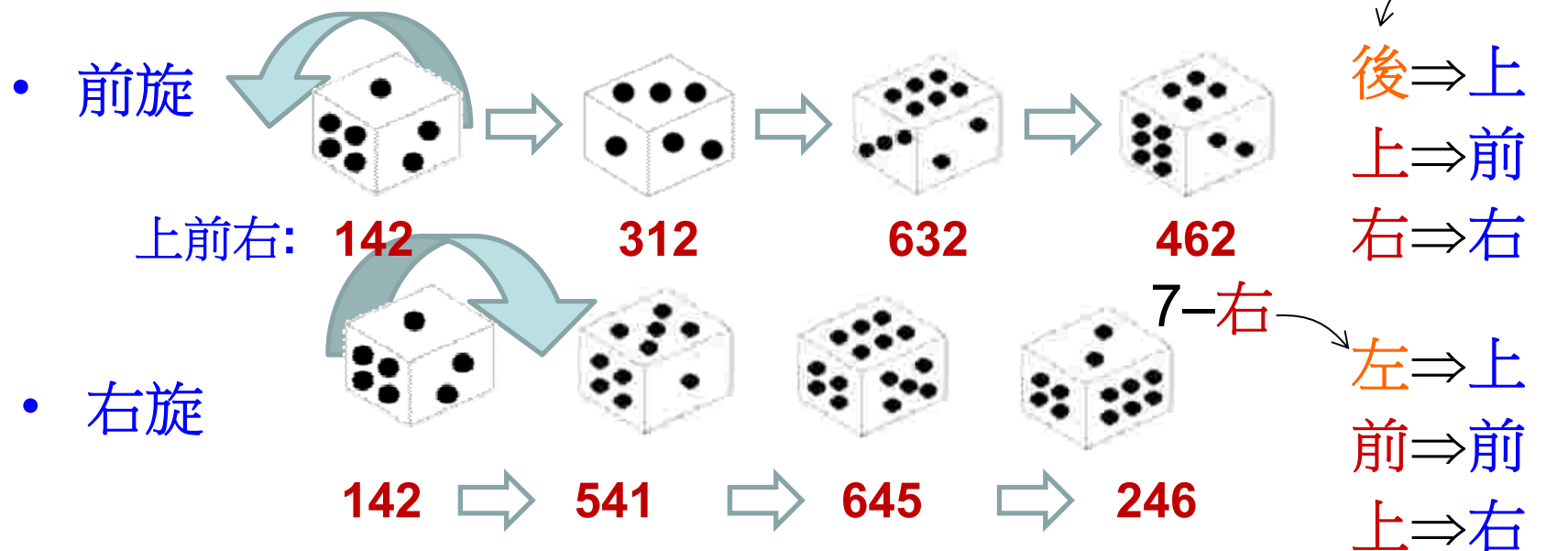




問題分析 (續)

- 骰子的狀態有多少種？ 是 $6*5*4=120$ 嗎？ 不是
骰子旋轉時每一面的四個相鄰面點數都是固定的，並不是六面拆開來隨便組合的
隨機挑選一個點數作為前面的點數，然後向右旋轉四次得到四種不同的狀態，所以是 $6*4=24$ 種不同的狀態

- 模擬：骰子旋轉時狀態如何改變？



實作

$1 \leq n \leq 20$

$1 \leq m \leq 100$

Sample Input

1 2 n m

1 -2 右轉

1 -1 前轉

Sample Output

3

實作

位置 1~20
上前右點數

```
01 int i, j, n, m, dice[21][3], a, b, t;  
02 while (2==scanf("%d%d", &n, &m)) {  
03     for (i=1; i<=n; i++)  
04         dice[i][0]=1, dice[i][1]=4, dice[i][2]=2;
```

$1 \leq n \leq 20$
 $1 \leq m \leq 100$



Sample Input

1 2 n m

1 -2 右轉

1 -1 前轉

Sample Output

3

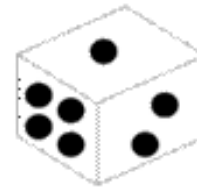
17 }

實作

位置 1~20
上前右點數

```
01 int i, j, n, m, dice[21][3], a, b, t;  
02 while (2==scanf("%d%d", &n, &m)) {  
03     for (i=1; i<=n; i++)  
04         dice[i][0]=1, dice[i][1]=4, dice[i][2]=2;  
05     for (i=0; i<m; i++) {
```

$1 \leq n \leq 20$
 $1 \leq m \leq 100$



Sample Input

1 2 n m

1 -2 右轉

1 -1 前轉

Sample Output

3

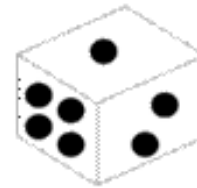
```
14     }  
15     for (i=1; i<=n; i++)  
16         printf("%d%c", dice[i][0], i==n?"\n ':' ');  
17 }
```

實作

位置 1~20 上前右點數

```
01 int i, j, n, m, dice[21][3], a, b, t;  
02 while (2==scanf("%d%d", &n, &m)) {  
03     for (i=1; i<=n; i++)  
04         dice[i][0]=1, dice[i][1]=4, dice[i][2]=2;  
05     for (i=0; i<m; i++) {  
06         scanf("%d%d", &a, &b);
```

$1 \leq n \leq 20$
 $1 \leq m \leq 100$



Sample Input

1 2 n m

1 -2 右轉

1 -1 前轉

Sample Output

3

```
14     }  
15     for (i=1; i<=n; i++)  
16         printf("%d%c", dice[i][0], i==n?"\n":' ');  
17 }
```

實作

位置 1~20
上前右點數

```
01 int i, j, n, m, dice[21][3], a, b, t;
02 while (2==scanf("%d%d", &n, &m)) {
03     for (i=1; i<=n; i++)
04         dice[i][0]=1, dice[i][1]=4, dice[i][2]=2;
05     for (i=0; i<m; i++) {
06         scanf("%d%d", &a, &b);
07         if (b>0)
08             for (j=0; j<3; j++)
09                 t=dice[a][j],dice[a][j]=dice[b][j],dice[b][j]=t;
```

$1 \leq n \leq 20$

$1 \leq m \leq 100$



骰子交換

```
14     }
15     for (i=1; i<=n; i++)
16         printf("%d%c", dice[i][0], i==n?"\n ':' ');
17 }
```

Sample Input

1 2 n m

1 -2 右轉

1 -1 前轉

Sample Output

3

實作

位置 1~20
上前右點數

```

01 int i, j, n, m, dice[21][3], a, b, t;
02 while (2==scanf("%d%d", &n, &m)) {
03     for (i=1; i<=n; i++)
04         dice[i][0]=1, dice[i][1]=4, dice[i][2]=2;
05     for (i=0; i<m; i++) {
06         scanf("%d%d", &a, &b);
07         if (b>0)
08             for (j=0; j<3; j++)
09                 t=dice[a][j], dice[a][j]=dice[b][j], dice[b][j]=t;
10         else if (b==-1)
11             t=7-dice[a][1], dice[a][1]=dice[a][0], dice[a][0]=t;

14     }
15     for (i=1; i<=n; i++)
16         printf("%d%c", dice[i][0], i==n?"\n":' ');
17 }

```

$1 \leq n \leq 20$
 $1 \leq m \leq 100$



Sample Input

```

1 2    n m
1 -2   右轉
1 -1   前轉

```

Sample Output

3

骰子交換

前旋 { $7 - \text{前} \Rightarrow \text{上}$
 $\text{上} \Rightarrow \text{前}$

位置 1~20 上前右點數 實作

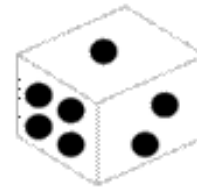
```

01 int i, j, n, m, dice[21][3], a, b, t;
02 while (2==scanf("%d%d", &n, &m)) {
03     for (i=1; i<=n; i++)
04         dice[i][0]=1, dice[i][1]=4, dice[i][2]=2;
05     for (i=0; i<m; i++) {
06         scanf("%d%d", &a, &b);
07         if (b>0)
08             for (j=0; j<3; j++)
09                 t=dice[a][j], dice[a][j]=dice[b][j], dice[b][j]=t;
10         else if (b== -1)
11             t=7-dice[a][1], dice[a][1]=dice[a][0], dice[a][0]=t;
12         else if (b== -2)
13             t=7-dice[a][2], dice[a][2]=dice[a][0], dice[a][0]=t;
14     }
15     for (i=1; i<=n; i++)
16         printf("%d%c", dice[i][0], i==n?"\n":' ');
17 }

```

$1 \leq n \leq 20$

$1 \leq m \leq 100$



Sample Input

1 2 n m

1 -2 右轉

1 -1 前轉

Sample Output

3

骰子交換

前旋 { 7-前 ⇒ 上
 上 ⇒ 前

右旋 { 7-右 ⇒ 上
 上 ⇒ 右

延伸問題

- 這個題目給你一開始骰子的狀態，告訴你對這組骰子的一連串操作，詢問你最後骰子的狀態，有一個很直接的延伸如下：給你一連串的操作，然後告訴你最後骰子的狀態，請問一開始骰子的狀態？

延伸問題

- 這個題目給你一開始骰子的狀態，告訴你對這組骰子的一連串操作，詢問你最後骰子的狀態，有一個很直接的延伸如下：給你一連串的操作，然後告訴你最後骰子的狀態，請問一開始骰子的狀態？
 - 這樣子形式的反向模擬也可以參考
ZeroJudge b965: APCS 105/03/05 #2 矩陣轉換
<https://zerojudge.tw/ShowProblem?problemid=b965>

延伸問題

- 這個題目給你一開始骰子的狀態，告訴你對這組骰子的一連串操作，詢問你最後骰子的狀態，有一個很直接的延伸如下：給你一連串的操作，然後告訴你最後骰子的狀態，請問一開始骰子的狀態？
 - 這樣子形式的反向模擬也可以參考
ZeroJudge b965: APCS 105/03/05 #2 矩陣轉換
<https://zerojudge.tw/ShowProblem?problemid=b965>
- 如果題目給你下面展開圖的五個心算骰子，再給你五個骰子起始的狀態，你要怎樣紀錄狀態並且模擬所有的動作？

延伸問題

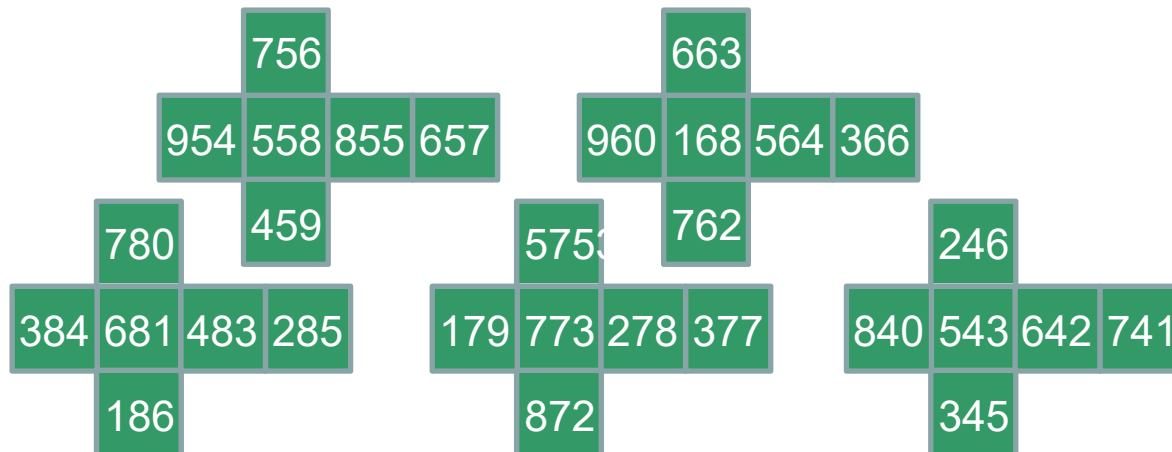
- 這個題目給你一開始骰子的狀態，告訴你對這組骰子的一連串操作，詢問你最後骰子的狀態，有一個很直接的延伸如下：給你一連串的操作，然後告訴你最後骰子的狀態，請問一開始骰子的狀態？

- 這樣子形式的反向模擬也可以參考

ZeroJudge b965: APCS 105/03/05 #2 矩陣轉換

<https://zerojudge.tw/ShowProblem?problemid=b965>

- 如果題目給你下面展開圖的五個心算骰子，再給你五個骰子起始的狀態，你要怎樣紀錄狀態並且模擬所有的動作？



延伸問題

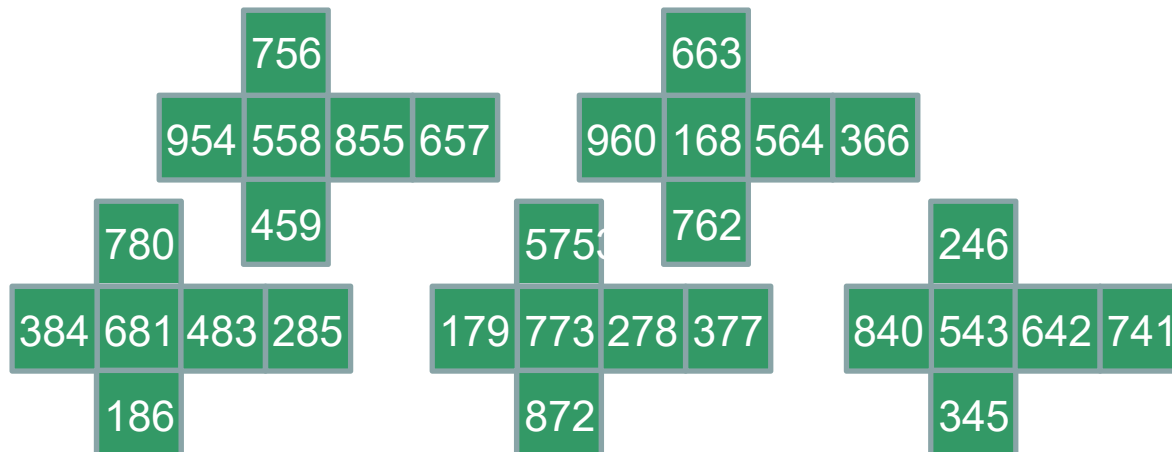
- 這個題目給你一開始骰子的狀態，告訴你對這組骰子的一連串操作，詢問你最後骰子的狀態，有一個很直接的延伸如下：給你一連串的操作，然後告訴你最後骰子的狀態，請問一開始骰子的狀態？

– 這樣子形式的反向模擬也可以參考

ZeroJudge b965: APCS 105/03/05 #2 矩陣轉換

<https://zerojudge.tw/ShowProblem?problemid=b965>

- 如果題目給你下面展開圖的五個心算骰子，再給你五個骰子起始的狀態，你要怎樣紀錄狀態並且模擬所有的動作？



延伸問題 (續)

- ZeroJudge f633: 魔術方塊 2X2

延伸問題 (續)

- ZeroJudge f633: 魔術方塊 2X2

如右圖一開始；

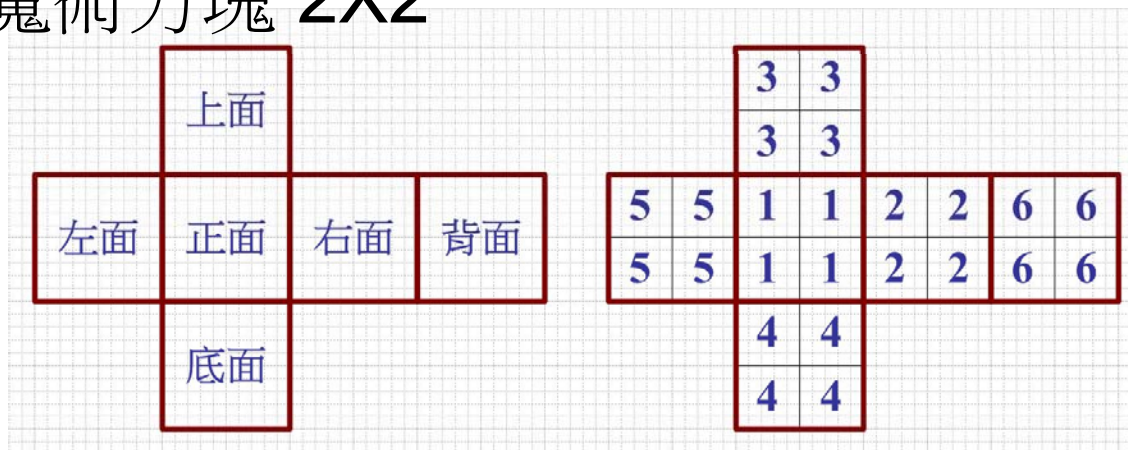
正面全部數字為 1，

背面全部數字為 6，

右面全部數字為 2，

左面全部數字為 5，

上面全部數字為 3，底面全部數字為 4。



延伸問題 (續)

- ZeroJudge f633: 魔術方塊 2X2

如右圖一開始；

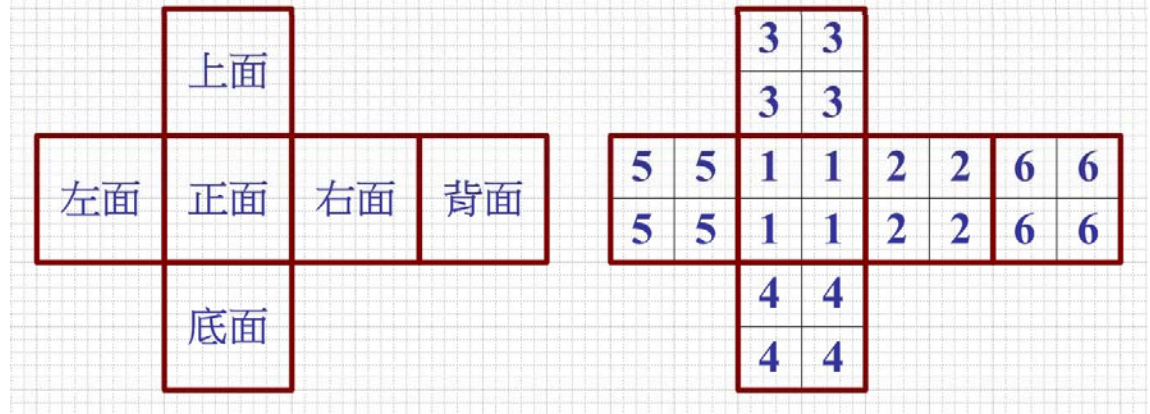
正面全部數字為 1，

背面全部數字為 6，

右面全部數字為 2，

左面全部數字為 5，

上面全部數字為 3，底面全部數字為 4。



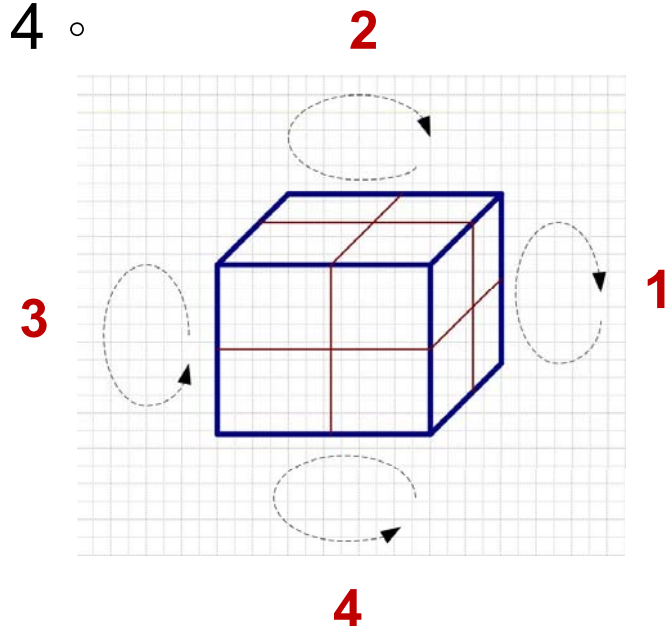
指令為

CMD=1 (右面順時針旋轉90度)

CMD=2 (上面順時針旋轉90度)

CMD=3 (左面順時針旋轉90度)

CMD=4 (底面順時針旋轉90度)



延伸問題 (續)

- ZeroJudge f633: 魔術方塊 2X2

如右圖一開始；

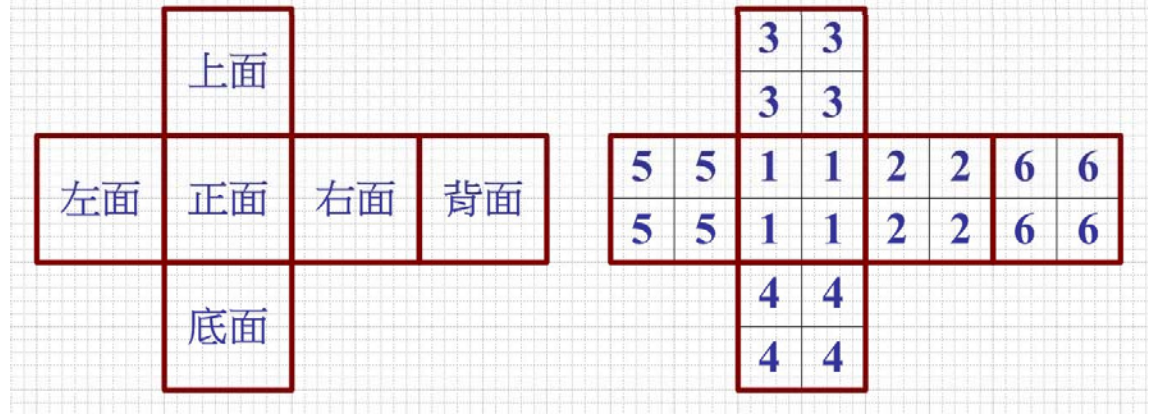
正面全部數字為 1，

背面全部數字為 6，

右面全部數字為 2，

左面全部數字為 5，

上面全部數字為 3，底面全部數字為 4。



指令為

CMD=1 (右面順時針旋轉90度)

CMD=2 (上面順時針旋轉90度)

CMD=3 (左面順時針旋轉90度)

CMD=4 (底面順時針旋轉90度)

一連串指令後請問最後正面的 4 個數字。

