

ZeroJudge f581 3. 圓環出口

ZeroJudge f581 3. 圓環出口

<https://zerojudge.tw/ShowProblem?problemid=f581>

109/07/04 APCS #3 近似題

丁培毅

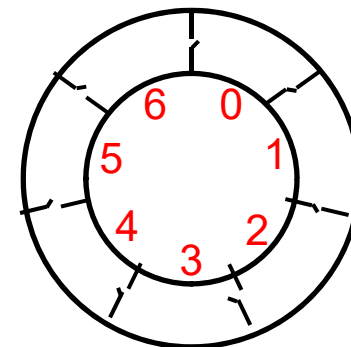
110/08

問題描述

- n 個房間排成環狀，編號由 0 到 $n-1$

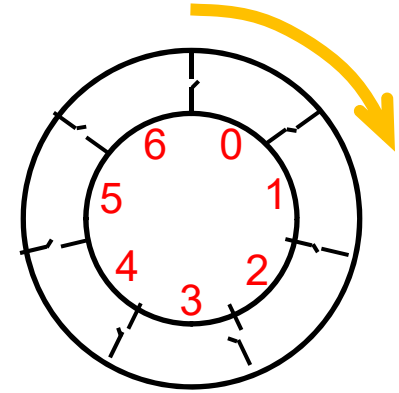
問題描述

- n 個房間排成環狀，編號由 0 到 $n-1$



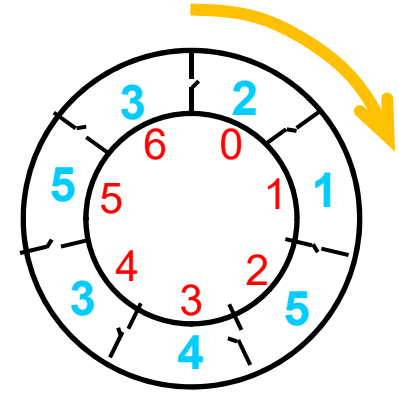
問題描述

- n 個房間排成環狀，編號由 **0** 到 **$n-1$**
- 房間之間有單向的路徑，由 **i** 到 **$i+1 \pmod n$**



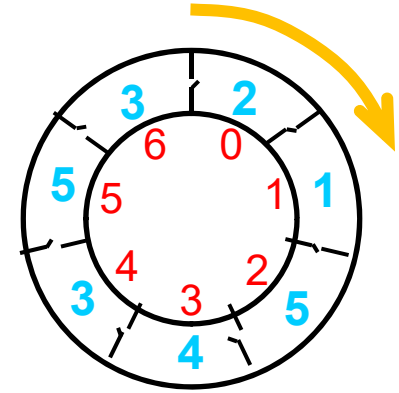
問題描述

- n 個房間排成環狀，編號由 0 到 $n-1$
- 房間之間有單向的路徑，由 i 到 $i+1 \pmod n$
- 進入編號 i 的房間可以獲得 p_i 點



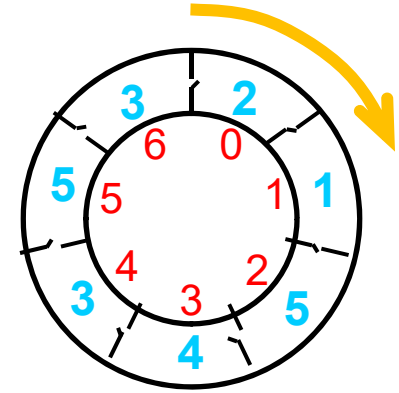
問題描述

- n 個房間排成環狀，編號由 0 到 $n-1$
- 房間之間有單向的路徑，由 i 到 $i+1 \pmod n$
- 進入編號 i 的房間可以獲得 p_i 點
- 依序有 m 個任務，第 j 個任務要求蒐集 q_j 點

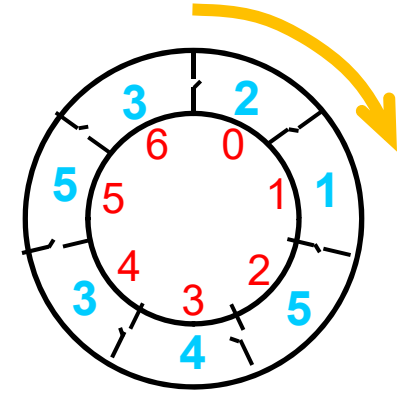


問題描述

- n 個房間排成環狀，編號由 0 到 $n-1$
- 房間之間有單向的路徑，由 i 到 $i+1 \pmod n$
- 進入編號 i 的房間可以獲得 p_i 點
- 依序有 m 個任務，第 j 個任務要求蒐集 q_j 點，由編號 s 的房間走到編號 t 的房間可以蒐集到 $p_s + p_{s+1 \pmod n} + \dots + p_t$ 點

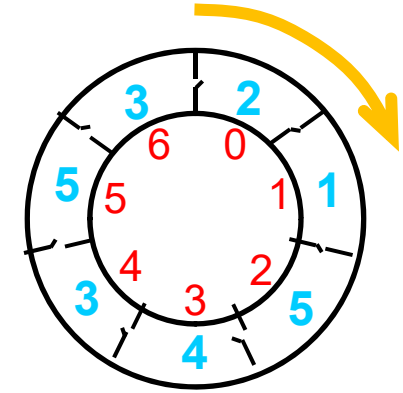


問題描述



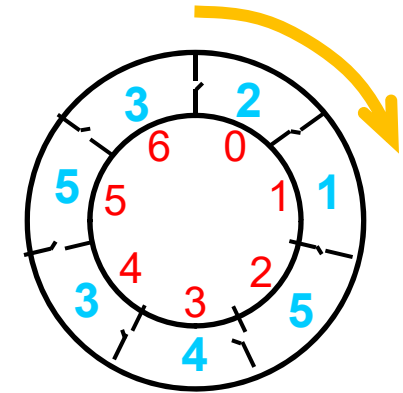
- n 個房間排成環狀，編號由 0 到 $n-1$
- 房間之間有單向的路徑，由 i 到 $i+1 \pmod n$
- 進入編號 i 的房間可以獲得 p_i 點
- 依序有 m 個任務，第 j 個任務要求蒐集 q_j 點，由編號 s 的房間走到編號 t 的房間可以蒐集到 $p_s + p_{s+1 \pmod n} + \dots + p_t$ 點，滿足 $p_s + p_{s+1 \pmod n} + \dots + p_{t-1} < q_j \leq p_s + p_{s+1 \pmod n} + \dots + p_t$

問題描述



- n 個房間排成環狀，編號由 0 到 $n-1$
- 房間之間有單向的路徑，由 i 到 $i+1 \pmod n$
- 進入編號 i 的房間可以獲得 p_i 點
- 依序有 m 個任務，第 j 個任務要求蒐集 q_j 點，由編號 s 的房間走到編號 t 的房間可以蒐集到 $p_s + p_{s+1 \pmod n} + \dots + p_t$ 點，滿足 $p_s + p_{s+1 \pmod n} + \dots + p_{t-1} < q_j \leq p_s + p_{s+1 \pmod n} + \dots + p_t$ ，停在編號 $t+1 \pmod n$ 的房間

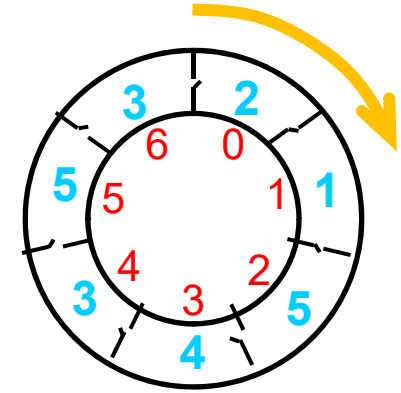
問題描述



- n 個房間排成環狀，編號由 0 到 $n-1$
- 房間之間有單向的路徑，由 i 到 $i+1 \pmod n$
- 進入編號 i 的房間可以獲得 p_i 點
- 依序有 m 個任務，第 j 個任務要求蒐集 q_j 點，由編號 s 的房間走到編號 t 的房間可以蒐集到 $p_s + p_{s+1 \pmod n} + \dots + p_t$ 點，滿足 $p_s + p_{s+1 \pmod n} + \dots + p_{t-1} < q_j \leq p_s + p_{s+1 \pmod n} + \dots + p_t$ ，停在編號 $t+1 \pmod n$ 的房間

範例一輸入
7 3 n m
2 1 5 4 3 5 3
8 9 12

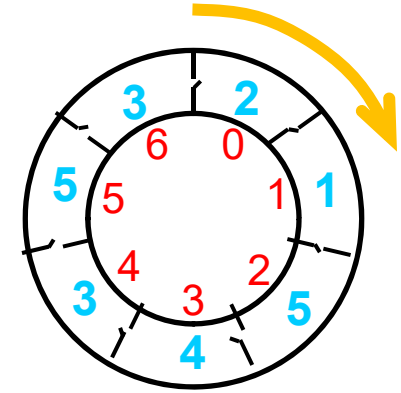
問題描述



- n 個房間排成環狀，編號由 0 到 $n-1$
- 房間之間有單向的路徑，由 i 到 $i+1 \pmod n$
- 進入編號 i 的房間可以獲得 p_i 點
- 依序有 m 個任務，第 j 個任務要求蒐集 q_j 點，由編號 s 的房間走到編號 t 的房間可以蒐集到 $p_s + p_{s+1 \pmod n} + \dots + p_t$ 點，滿足 $p_s + p_{s+1 \pmod n} + \dots + p_{t-1} < q_j \leq p_s + p_{s+1 \pmod n} + \dots + p_t$ ，停在編號 $t+1 \pmod n$ 的房間
- 一開始在編號 0 的房間

範例一輸入
7 3 n m
2 1 5 4 3 5 3
8 9 12

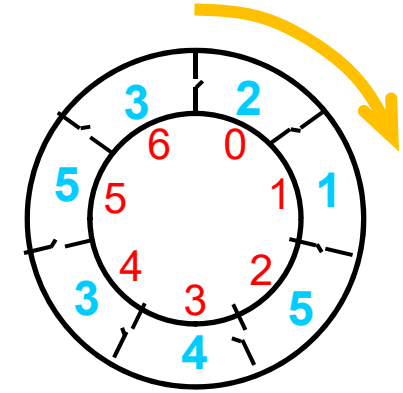
問題描述



- n 個房間排成環狀，編號由 0 到 $n-1$
- 房間之間有單向的路徑，由 i 到 $i+1 \pmod n$
- 進入編號 i 的房間可以獲得 p_i 點
- 依序有 m 個任務，第 j 個任務要求蒐集 q_j 點，由編號 s 的房間走到編號 t 的房間可以蒐集到 $p_s + p_{s+1 \pmod n} + \dots + p_t$ 點，滿足 $p_s + p_{s+1 \pmod n} + \dots + p_{t-1} < q_j \leq p_s + p_{s+1 \pmod n} + \dots + p_t$ ，停在編號 $t+1 \pmod n$ 的房間
- 一開始在編號 0 的房間
- 依據指定的 m 個任務，請問任務完成時停在哪一間房間？

範例一輸入
7 3 n m
2 1 5 4 3 5 3
8 9 12

問題描述



- n 個房間排成環狀，編號由 0 到 $n-1$
- 房間之間有單向的路徑，由 i 到 $i+1 \pmod n$
- 進入編號 i 的房間可以獲得 p_i 點
- 依序有 m 個任務，第 j 個任務要求蒐集 q_j 點，由編號 s 的房間走到編號 t 的房間可以蒐集到 $p_s + p_{s+1 \pmod n} + \dots + p_t$ 點，滿足 $p_s + p_{s+1 \pmod n} + \dots + p_{t-1} < q_j \leq p_s + p_{s+1 \pmod n} + \dots + p_t$ ，停在編號 $t+1 \pmod n$ 的房間
- 一開始在編號 0 的房間
- 依據指定的 m 個任務，請問任務完成時停在哪一間房間？

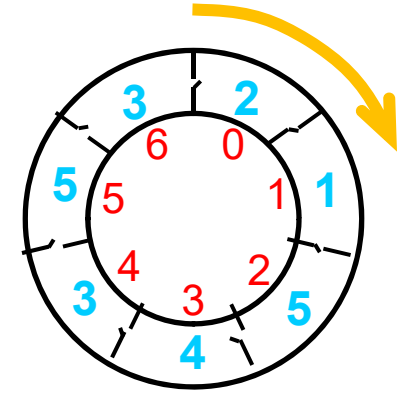
範例一輸入

```
7 3      n m
2 1 5 4 3 5 3
8 9 12
```

範例一輸出

4

問題描述



- n 個房間排成環狀，編號由 0 到 $n-1$
- 房間之間有單向的路徑，由 i 到 $i+1 \pmod n$
- 進入編號 i 的房間可以獲得 p_i 點
- 依序有 m 個任務，第 j 個任務要求蒐集 q_j 點，由編號 s 的房間走到編號 t 的房間可以蒐集到 $p_s + p_{s+1 \pmod n} + \dots + p_t$ 點，滿足 $p_s + p_{s+1 \pmod n} + \dots + p_{t-1} < q_j \leq p_s + p_{s+1 \pmod n} + \dots + p_t$ ，停在編號 $t+1 \pmod n$ 的房間
- 一開始在編號 0 的房間
- 依據指定的 m 個任務，請問任務完成時停在哪一間房間？

範例一輸入

```
7 3      n m
2 1 5 4 3 5 3
8 9 12
```

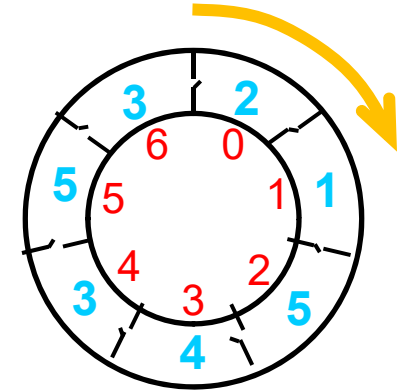
範例一輸出

4

範例二輸入

```
4 3
1 3 5 7
4 2 2
```

問題描述



- n 個房間排成環狀，編號由 0 到 $n-1$
- 房間之間有單向的路徑，由 i 到 $i+1 \pmod n$
- 進入編號 i 的房間可以獲得 p_i 點
- 依序有 m 個任務，第 j 個任務要求蒐集 q_j 點，由編號 s 的房間走到編號 t 的房間可以蒐集到 $p_s + p_{s+1 \pmod n} + \dots + p_t$ 點，滿足 $p_s + p_{s+1 \pmod n} + \dots + p_{t-1} < q_j \leq p_s + p_{s+1 \pmod n} + \dots + p_t$ ，停在編號 $t+1 \pmod n$ 的房間
- 一開始在編號 0 的房間
- 依據指定的 m 個任務，請問任務完成時停在哪一間房間？

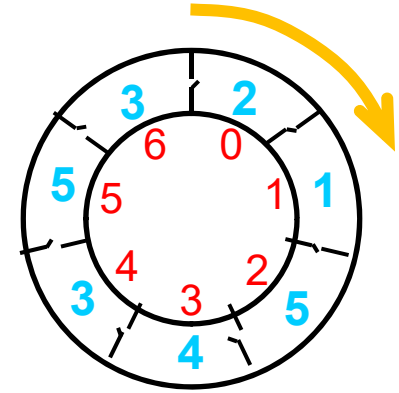
範例一輸入
7 3 n m
2 1 5 4 3 5 3
8 9 12

範例一輸出
4

範例二輸入
4 3
1 3 5 7
4 2 2

範例二輸出
0

問題描述



- n 個房間排成環狀，編號由 0 到 $n-1$
 - 房間之間有單向的路徑，由 i 到 $i+1 \pmod n$
 - 進入編號 i 的房間可以獲得 p_i 點
 - 依序有 m 個任務，第 j 個任務要求蒐集 q_j 點，由編號 s 的房間走到編號 t 的房間可以蒐集到 $p_s + p_{s+1 \pmod n} + \dots + p_t$ 點，滿足 $p_s + p_{s+1 \pmod n} + \dots + p_{t-1} < q_j \leq p_s + p_{s+1 \pmod n} + \dots + p_t$ ，停在編號 $t+1 \pmod n$ 的房間
 - 一開始在編號 0 的房間
 - 依據指定的 m 個任務，請問任務完成時停在哪一間房間？
- $$p_0 + p_1 + \dots + p_{n-1} \leq 10^9$$

```
範例一輸入
7 3      n m
2 1 5 4 3 5 3
8 9 12
```

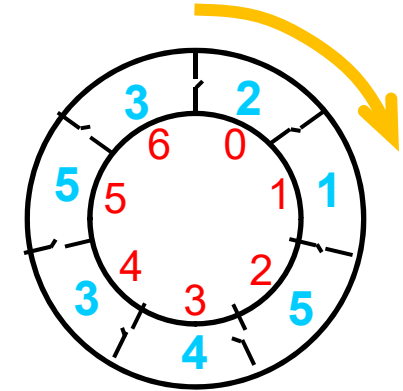
範例一輸出

範例二輸入

```
4 3
1 3 5 7
4 2 2
```

範例二輸出

問題描述



- n 個房間排成環狀，編號由 0 到 $n-1$
- 房間之間有單向的路徑，由 i 到 $i+1 \pmod n$
- 進入編號 i 的房間可以獲得 p_i 點
- 依序有 m 個任務，第 j 個任務要求蒐集 q_j 點，由編號 s 的房間走到編號 t 的房間可以蒐集到 $p_s + p_{s+1 \pmod n} + \dots + p_t$ 點，滿足 $p_s + p_{s+1 \pmod n} + \dots + p_{t-1} < q_j \leq p_s + p_{s+1 \pmod n} + \dots + p_t$ ，停在編號 $t+1 \pmod n$ 的房間
- 一開始在編號 0 的房間
- 依據指定的 m 個任務，請問任務完成時停在哪一間房間？

$$q_j \leq p_0 + p_1 + \dots + p_{n-1} \leq 10^9$$

範例一輸入

7 3 n m
2 1 5 4 3 5 3
8 9 12

範例一輸出

4

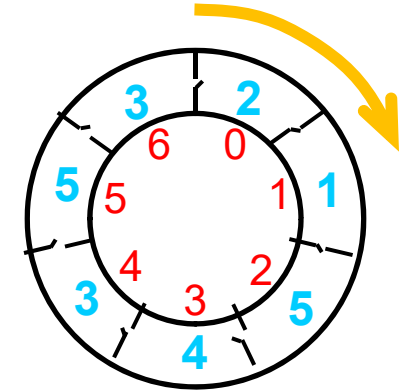
範例二輸入

4 3
1 3 5 7
4 2 2

範例二輸出

0

問題描述



- n 個房間排成環狀，編號由 0 到 $n-1$
- 房間之間有單向的路徑，由 i 到 $i+1 \pmod n$
- 進入編號 i 的房間可以獲得 p_i 點
- 依序有 m 個任務，第 j 個任務要求蒐集 q_j 點，由編號 s 的房間走到編號 t 的房間可以蒐集到 $p_s + p_{s+1 \pmod n} + \dots + p_t$ 點，滿足 $p_s + p_{s+1 \pmod n} + \dots + p_{t-1} < q_j \leq p_s + p_{s+1 \pmod n} + \dots + p_t$ ，停在編號 $t+1 \pmod n$ 的房間
- 一開始在編號 0 的房間
- 依據指定的 m 個任務，請問任務完成時停在哪一間房間？

$$q_j \leq p_0 + p_1 + \dots + p_{n-1} \leq 10^9$$

- $1 \leq n \leq 200000, 1 \leq m \leq 20000$

範例一輸入

7 3 n m
2 1 5 4 3 5 3
8 9 12

範例一輸出

4

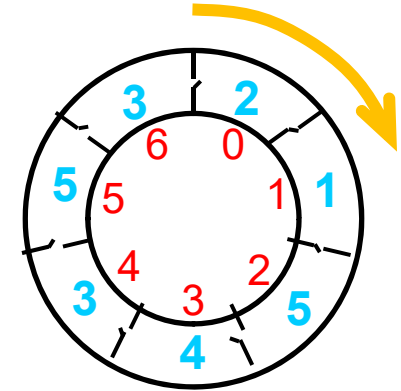
範例二輸入

4 3
1 3 5 7
4 2 2

範例二輸出

0

問題描述



- n 個房間排成環狀，編號由 0 到 $n-1$
- 房間之間有單向的路徑，由 i 到 $i+1 \pmod n$
- 進入編號 i 的房間可以獲得 p_i 點
- 依序有 m 個任務，第 j 個任務要求蒐集 q_j 點，由編號 s 的房間走到編號 t 的房間可以蒐集到 $p_s + p_{s+1 \pmod n} + \dots + p_t$ 點，滿足 $p_s + p_{s+1 \pmod n} + \dots + p_{t-1} < q_j \leq p_s + p_{s+1 \pmod n} + \dots + p_t$ ，停在編號 $t+1 \pmod n$ 的房間
- 一開始在編號 0 的房間
- 依據指定的 m 個任務，請問任務完成時停在哪一間房間？

$$q_j \leq p_0 + p_1 + \dots + p_{n-1} \leq 10^9$$

- $1 \leq n \leq 200000, 1 \leq m \leq 20000$
- 時間限制 2 sec

範例一輸入

7 3 n m
2 1 5 4 3 5 3
8 9 12

範例一輸出

4

範例二輸入

4 3
1 3 5 7
4 2 2

範例二輸出

0

問題分析

- 看到這樣子一步一步描述的問題，基本的作法是模擬，藉由電腦的速度與正確性把結果暴力解出來

問題分析

- 看到這樣子一步一步描述的問題，基本的作法是模擬，藉由電腦的速度與正確性把結果暴力解出來，不過看到 n 可以到達 200000， m 到達 20000，這種數值範圍可以想像模擬時需要盡量節省時間，否則沒有辦法在指定的時間限制裡完成

問題分析

- 看到這樣子一步一步描述的問題，基本的作法是模擬，藉由電腦的速度與正確性把結果暴力解出來，不過看到 n 可以到達 200000， m 到達 20000，這種數值範圍可以想像模擬時需要盡量節省時間，否則沒有辦法在指定的時間限制裡完成
- 資料的數量到達 10^5 時，執行的方法如果是 $O(n^2)$ 以上的，像是兩層的暴力迴圈、氣泡排序等等，就會超過時間限制了

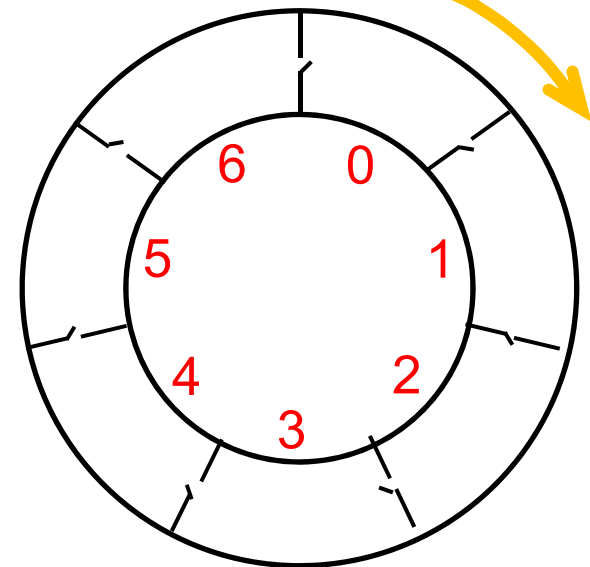
問題分析

- 看到這樣子一步一步描述的問題，基本的作法是**模擬**，藉由電腦的速度與正確性把結果**暴力**解出來，不過看到 **n** 可以到達 200000，**m** 到達 20000，這種數值範圍可以想像模擬時需要盡量節省時間，否則沒有辦法在指定的時間限制裡完成
- 資料的數量到達 **10^5** 時，執行的方法如果是 **$O(n^2)$** 以上的，像是兩層的暴力迴圈、氣泡排序等等，就會超過時間限制了
- 既然是**模擬**，最基本的解題步驟就是拿一個不太大的測資，**手動模擬**看看，可以讓你弄清楚題目的特性

問題分析

- 看到這樣子一步一步描述的問題，基本的作法是**模擬**，藉由電腦的速度與正確性把結果**暴力**解出來，不過看到 **n** 可以到達 200000，**m** 到達 20000，這種數值範圍可以想像模擬時需要盡量節省時間，否則沒有辦法在指定的時間限制裡完成
- 資料的數量到達 **10^5** 時，執行的方法如果是 **$O(n^2)$** 以上的，像是兩層的暴力迴圈、氣泡排序等等，就會超過時間限制了
- 既然是**模擬**，最基本的解題步驟就是拿一個不太大的測資，**手動模擬**看看，可以讓你弄清楚題目的特性

Sample Input
7 3



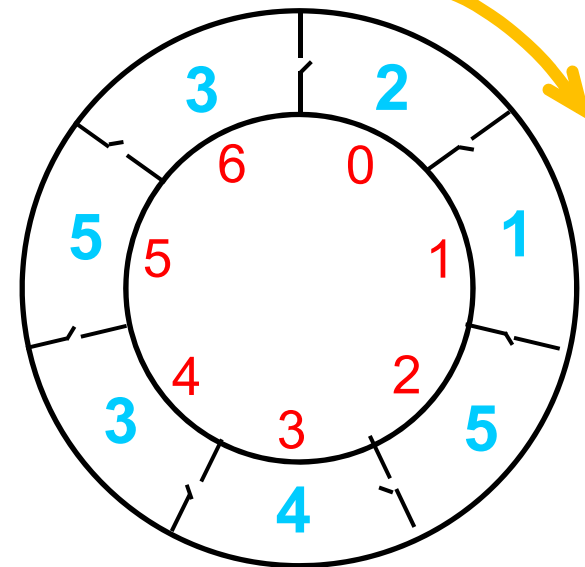
問題分析

- 看到這樣子一步一步描述的問題，基本的作法是**模擬**，藉由電腦的速度與正確性把結果**暴力**解出來，不過看到 **n** 可以到達 200000，**m** 到達 20000，這種數值範圍可以想像模擬時需要盡量節省時間，否則沒有辦法在指定的時間限制裡完成
- 資料的數量到達 **10^5** 時，執行的方法如果是 **$O(n^2)$** 以上的，像是兩層的暴力迴圈、氣泡排序等等，就會超過時間限制了
- 既然是**模擬**，最基本的解題步驟就是拿一個不太大的測資，**手動模擬**看看，可以讓你弄清楚題目的特性

Sample Input

7 3

2 1 5 4 3 5 3



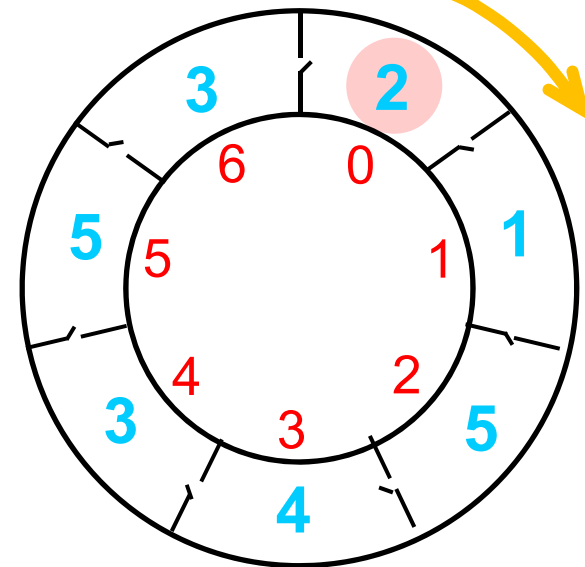
問題分析

- 看到這樣子一步一步描述的問題，基本的作法是**模擬**，藉由電腦的速度與正確性把結果**暴力**解出來，不過看到 **n** 可以到達 200000，**m** 到達 20000，這種數值範圍可以想像模擬時需要盡量節省時間，否則沒有辦法在指定的時間限制裡完成
- 資料的數量到達 **10^5** 時，執行的方法如果是 **$O(n^2)$** 以上的，像是兩層的暴力迴圈、氣泡排序等等，就會超過時間限制了
- 既然是**模擬**，最基本的解題步驟就是拿一個不太大的測資，**手動模擬**看看，可以讓你弄清楚題目的特性

Sample Input

7 3

2 1 5 4 3 5 3



問題分析

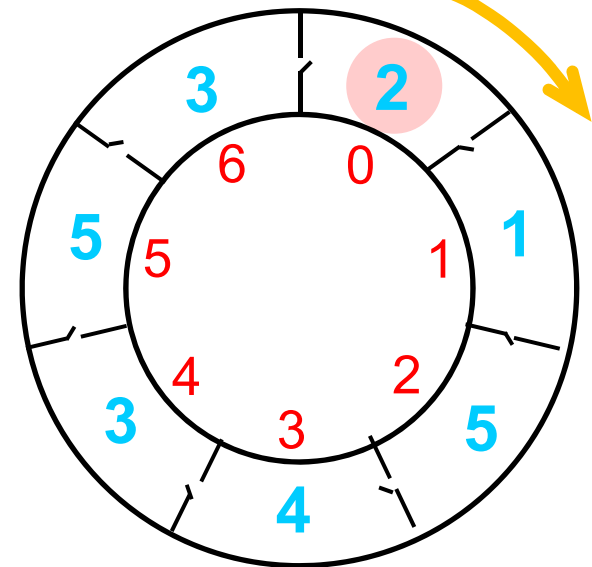
- 看到這樣子一步一步描述的問題，基本的作法是**模擬**，藉由電腦的速度與正確性把結果**暴力**解出來，不過看到 **n** 可以到達 200000，**m** 到達 20000，這種數值範圍可以想像模擬時需要盡量節省時間，否則沒有辦法在指定的時間限制裡完成
- 資料的數量到達 **10^5** 時，執行的方法如果是 **$O(n^2)$** 以上的，像是兩層的暴力迴圈、氣泡排序等等，就會超過時間限制了
- 既然是**模擬**，最基本的解題步驟就是拿一個不太大的測資，**手動模擬**看看，可以讓你弄清楚題目的特性

Sample Input

7 3

2 1 5 4 3 5 3

8



問題分析

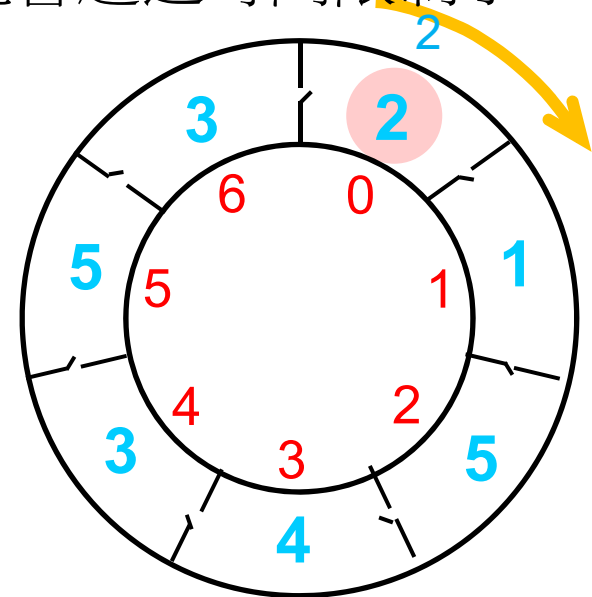
- 看到這樣子一步一步描述的問題，基本的作法是**模擬**，藉由電腦的速度與正確性把結果**暴力**解出來，不過看到 **n** 可以到達 200000，**m** 到達 20000，這種數值範圍可以想像模擬時需要盡量節省時間，否則沒有辦法在指定的時間限制裡完成
- 資料的數量到達 **10^5** 時，執行的方法如果是 **$O(n^2)$** 以上的，像是兩層的暴力迴圈、氣泡排序等等，就會超過時間限制了
- 既然是**模擬**，最基本的解題步驟就是拿一個不太大的測資，**手動模擬**看看，可以讓你弄清楚題目的特性

Sample Input

7 3

2 1 5 4 3 5 3

8



問題分析

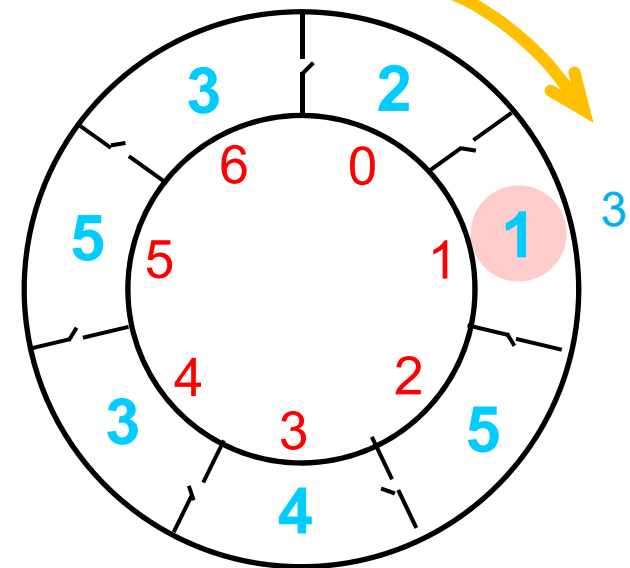
- 看到這樣子一步一步描述的問題，基本的作法是**模擬**，藉由電腦的速度與正確性把結果**暴力**解出來，不過看到 **n** 可以到達 200000，**m** 到達 20000，這種數值範圍可以想像模擬時需要盡量節省時間，否則沒有辦法在指定的時間限制裡完成
- 資料的數量到達 **10^5** 時，執行的方法如果是 **$O(n^2)$** 以上的，像是兩層的暴力迴圈、氣泡排序等等，就會超過時間限制了
- 既然是**模擬**，最基本的解題步驟就是拿一個不太大的測資，**手動模擬**看看，可以讓你弄清楚題目的特性

Sample Input

7 3

2 1 5 4 3 5 3

8



問題分析

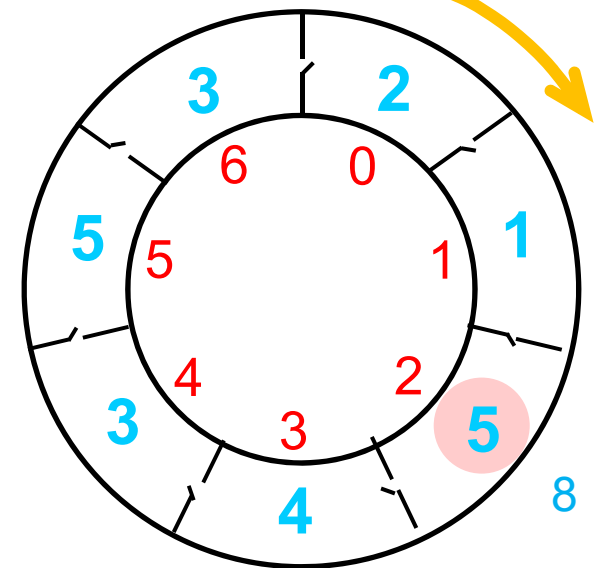
- 看到這樣子一步一步描述的問題，基本的作法是**模擬**，藉由電腦的速度與正確性把結果**暴力**解出來，不過看到 **n** 可以到達 200000，**m** 到達 20000，這種數值範圍可以想像模擬時需要盡量節省時間，否則沒有辦法在指定的時間限制裡完成
- 資料的數量到達 **10^5** 時，執行的方法如果是 **$O(n^2)$** 以上的，像是兩層的暴力迴圈、氣泡排序等等，就會超過時間限制了
- 既然是**模擬**，最基本的解題步驟就是拿一個不太大的測資，**手動模擬**看看，可以讓你弄清楚題目的特性

Sample Input

7 3

2 1 5 4 3 5 3

8



問題分析

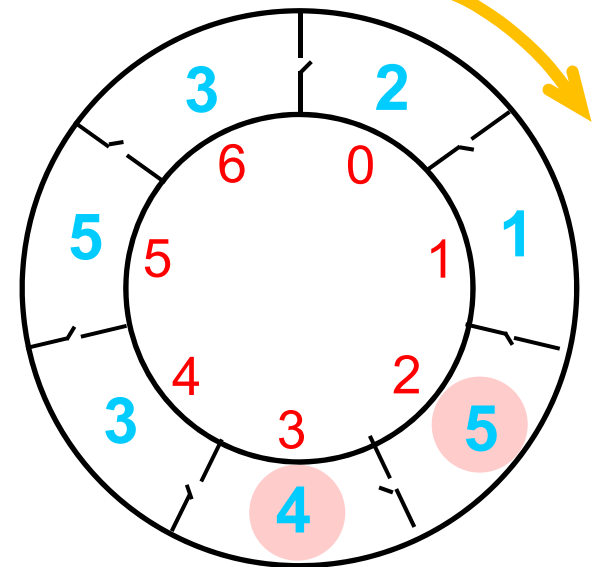
- 看到這樣子一步一步描述的問題，基本的作法是**模擬**，藉由電腦的速度與正確性把結果**暴力**解出來，不過看到 **n** 可以到達 200000，**m** 到達 20000，這種數值範圍可以想像模擬時需要盡量節省時間，否則沒有辦法在指定的時間限制裡完成
- 資料的數量到達 **10^5** 時，執行的方法如果是 **$O(n^2)$** 以上的，像是兩層的暴力迴圈、氣泡排序等等，就會超過時間限制了
- 既然是**模擬**，最基本的解題步驟就是拿一個不太大的測資，**手動模擬**看看，可以讓你弄清楚題目的特性

Sample Input

7 3

2 1 5 4 3 5 3

8



問題分析

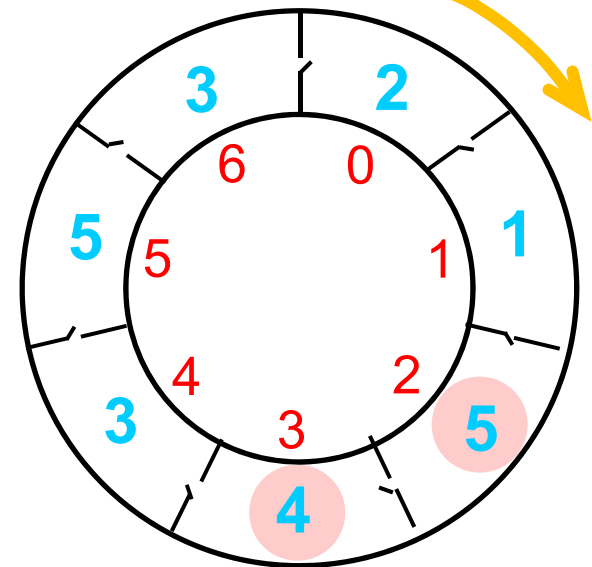
- 看到這樣子一步一步描述的問題，基本的作法是**模擬**，藉由電腦的速度與正確性把結果**暴力**解出來，不過看到 **n** 可以到達 200000，**m** 到達 20000，這種數值範圍可以想像模擬時需要盡量節省時間，否則沒有辦法在指定的時間限制裡完成
- 資料的數量到達 **10^5** 時，執行的方法如果是 **$O(n^2)$** 以上的，像是兩層的暴力迴圈、氣泡排序等等，就會超過時間限制了
- 既然是**模擬**，最基本的解題步驟就是拿一個不太大的測資，**手動模擬**看看，可以讓你弄清楚題目的特性

Sample Input

7 3

2 1 5 4 3 5 3

8 9



問題分析

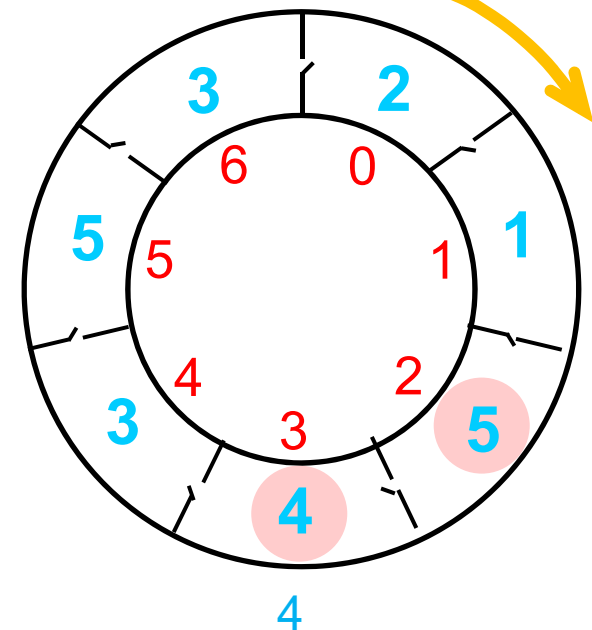
- 看到這樣子一步一步描述的問題，基本的作法是**模擬**，藉由電腦的速度與正確性把結果**暴力**解出來，不過看到 **n** 可以到達 200000，**m** 到達 20000，這種數值範圍可以想像模擬時需要盡量節省時間，否則沒有辦法在指定的時間限制裡完成
- 資料的數量到達 **10^5** 時，執行的方法如果是 **$O(n^2)$** 以上的，像是兩層的暴力迴圈、氣泡排序等等，就會超過時間限制了
- 既然是**模擬**，最基本的解題步驟就是拿一個不太大的測資，**手動模擬**看看，可以讓你弄清楚題目的特性

Sample Input

7 3

2 1 5 4 3 5 3

8 9



問題分析

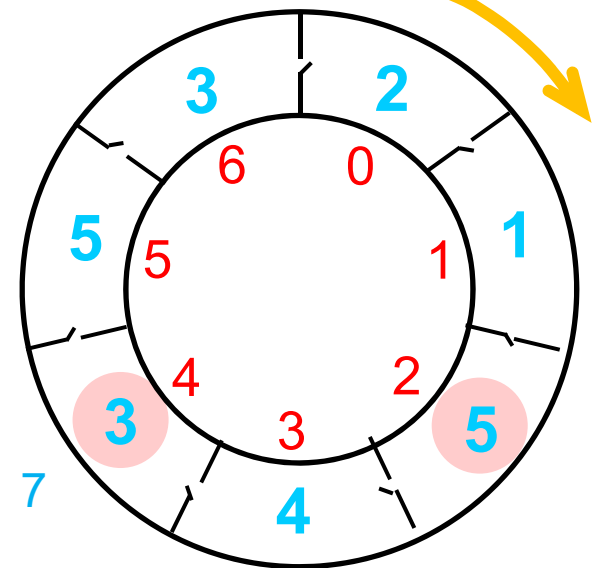
- 看到這樣子一步一步描述的問題，基本的作法是**模擬**，藉由電腦的速度與正確性把結果**暴力**解出來，不過看到 **n** 可以到達 200000，**m** 到達 20000，這種數值範圍可以想像模擬時需要盡量節省時間，否則沒有辦法在指定的時間限制裡完成
- 資料的數量到達 **10^5** 時，執行的方法如果是 **$O(n^2)$** 以上的，像是兩層的暴力迴圈、氣泡排序等等，就會超過時間限制了
- 既然是**模擬**，最基本的解題步驟就是拿一個不太大的測資，**手動模擬**看看，可以讓你弄清楚題目的特性

Sample Input

7 3

2 1 5 4 3 5 3

8 9



問題分析

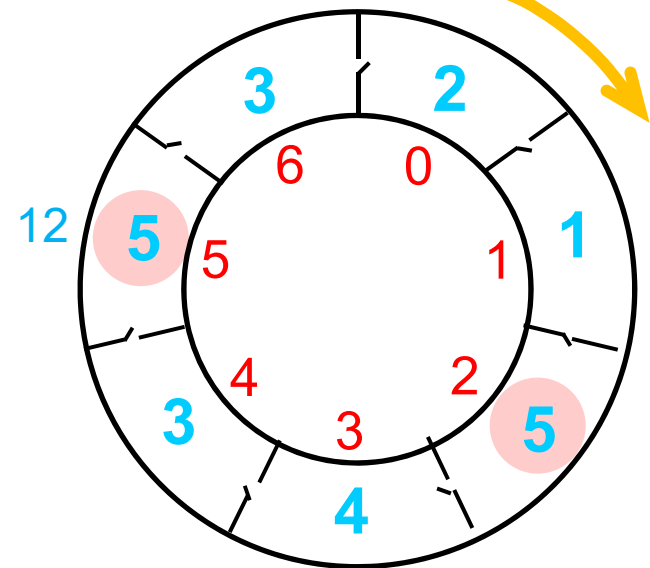
- 看到這樣子一步一步描述的問題，基本的作法是**模擬**，藉由電腦的速度與正確性把結果**暴力**解出來，不過看到 **n** 可以到達 200000，**m** 到達 20000，這種數值範圍可以想像模擬時需要盡量節省時間，否則沒有辦法在指定的時間限制裡完成
- 資料的數量到達 **10^5** 時，執行的方法如果是 **$O(n^2)$** 以上的，像是兩層的暴力迴圈、氣泡排序等等，就會超過時間限制了
- 既然是**模擬**，最基本的解題步驟就是拿一個不太大的測資，**手動模擬**看看，可以讓你弄清楚題目的特性

Sample Input

7 3

2 1 5 4 3 5 3

8 9



問題分析

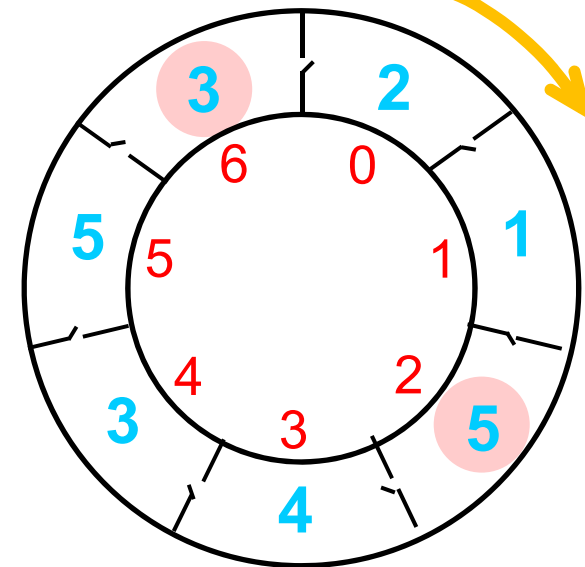
- 看到這樣子一步一步描述的問題，基本的作法是**模擬**，藉由電腦的速度與正確性把結果**暴力**解出來，不過看到 **n** 可以到達 200000，**m** 到達 20000，這種數值範圍可以想像模擬時需要盡量節省時間，否則沒有辦法在指定的時間限制裡完成
- 資料的數量到達 **10^5** 時，執行的方法如果是 **$O(n^2)$** 以上的，像是兩層的暴力迴圈、氣泡排序等等，就會超過時間限制了
- 既然是**模擬**，最基本的解題步驟就是拿一個不太大的測資，**手動模擬**看看，可以讓你弄清楚題目的特性

Sample Input

7 3

2 1 5 4 3 5 3

8 9



問題分析

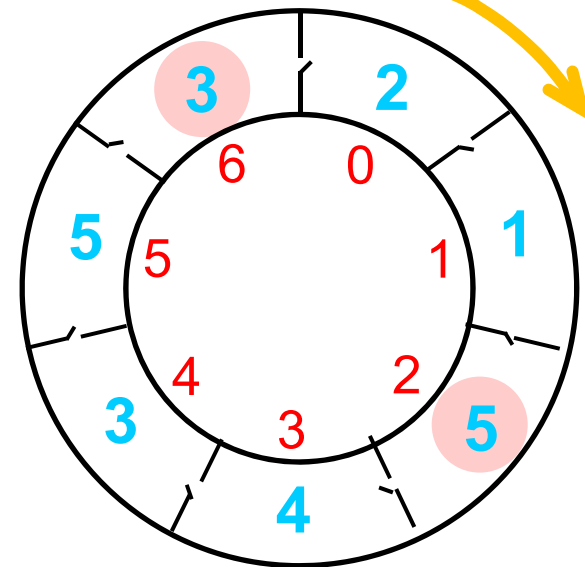
- 看到這樣子一步一步描述的問題，基本的作法是**模擬**，藉由電腦的速度與正確性把結果**暴力**解出來，不過看到 **n** 可以到達 200000，**m** 到達 20000，這種數值範圍可以想像模擬時需要盡量節省時間，否則沒有辦法在指定的時間限制裡完成
- 資料的數量到達 **10^5** 時，執行的方法如果是 **$O(n^2)$** 以上的，像是兩層的暴力迴圈、氣泡排序等等，就會超過時間限制了
- 既然是**模擬**，最基本的解題步驟就是拿一個不太大的測資，**手動模擬**看看，可以讓你弄清楚題目的特性

Sample Input

7 3

2 1 5 4 3 5 3

8 9 12



問題分析

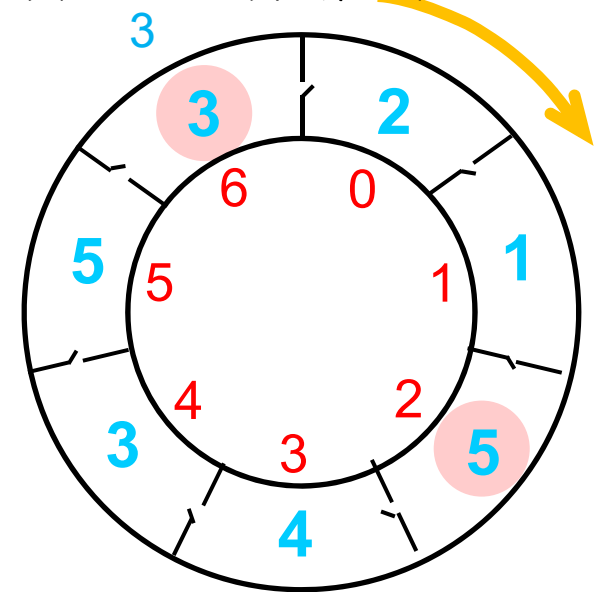
- 看到這樣子一步一步描述的問題，基本的作法是**模擬**，藉由電腦的速度與正確性把結果**暴力**解出來，不過看到 **n** 可以到達 200000，**m** 到達 20000，這種數值範圍可以想像模擬時需要盡量節省時間，否則沒有辦法在指定的時間限制裡完成
- 資料的數量到達 **10^5** 時，執行的方法如果是 **$O(n^2)$** 以上的，像是兩層的暴力迴圈、氣泡排序等等，就會超過時間限制了
- 既然是**模擬**，最基本的解題步驟就是拿一個不太大的測資，**手動模擬**看看，可以讓你弄清楚題目的特性

Sample Input

7 3

2 1 5 4 3 5 3

8 9 12



問題分析

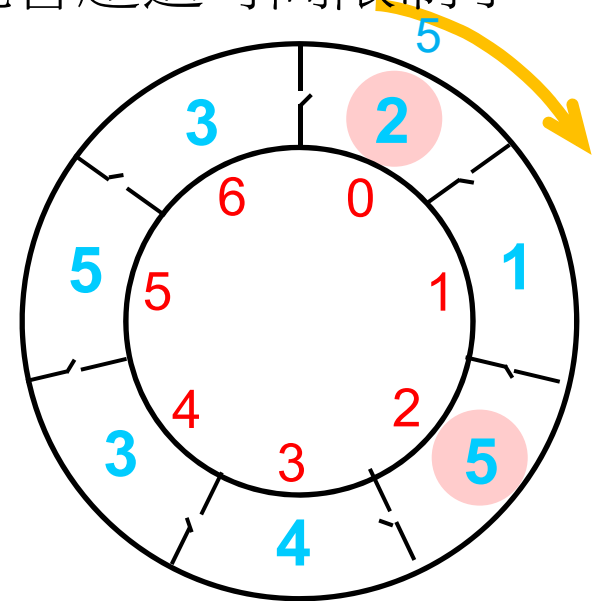
- 看到這樣子一步一步描述的問題，基本的作法是**模擬**，藉由電腦的速度與正確性把結果**暴力**解出來，不過看到 **n** 可以到達 200000，**m** 到達 20000，這種數值範圍可以想像模擬時需要盡量節省時間，否則沒有辦法在指定的時間限制裡完成
- 資料的數量到達 **10^5** 時，執行的方法如果是 **$O(n^2)$** 以上的，像是兩層的暴力迴圈、氣泡排序等等，就會超過時間限制了
- 既然是**模擬**，最基本的解題步驟就是拿一個不太大的測資，**手動模擬**看看，可以讓你弄清楚題目的特性

Sample Input

7 3

2 1 5 4 3 5 3

8 9 12



問題分析

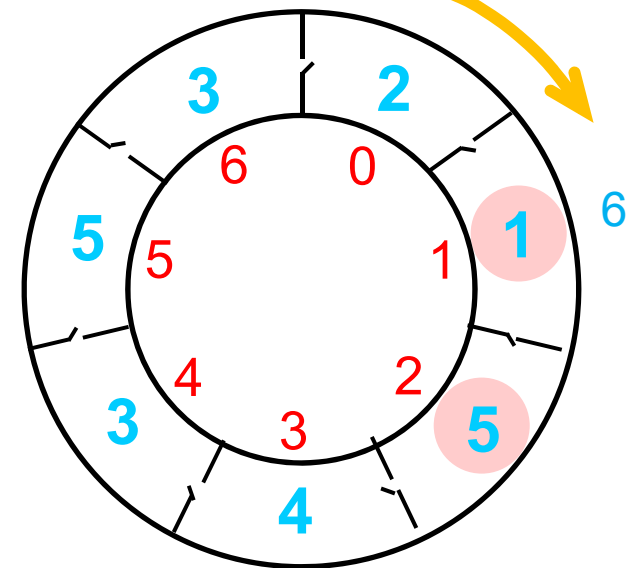
- 看到這樣子一步一步描述的問題，基本的作法是**模擬**，藉由電腦的速度與正確性把結果**暴力**解出來，不過看到 **n** 可以到達 200000，**m** 到達 20000，這種數值範圍可以想像模擬時需要盡量節省時間，否則沒有辦法在指定的時間限制裡完成
- 資料的數量到達 **10^5** 時，執行的方法如果是 **$O(n^2)$** 以上的，像是兩層的暴力迴圈、氣泡排序等等，就會超過時間限制了
- 既然是**模擬**，最基本的解題步驟就是拿一個不太大的測資，**手動模擬**看看，可以讓你弄清楚題目的特性

Sample Input

7 3

2 1 5 4 3 5 3

8 9 12



問題分析

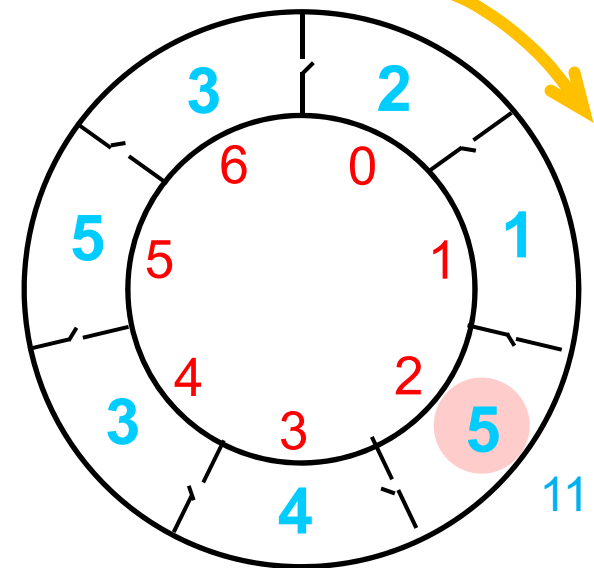
- 看到這樣子一步一步描述的問題，基本的作法是**模擬**，藉由電腦的速度與正確性把結果**暴力**解出來，不過看到 **n** 可以到達 200000，**m** 到達 20000，這種數值範圍可以想像模擬時需要盡量節省時間，否則沒有辦法在指定的時間限制裡完成
- 資料的數量到達 **10^5** 時，執行的方法如果是 **$O(n^2)$** 以上的，像是兩層的暴力迴圈、氣泡排序等等，就會超過時間限制了
- 既然是**模擬**，最基本的解題步驟就是拿一個不太大的測資，**手動模擬**看看，可以讓你弄清楚題目的特性

Sample Input

7 3

2 1 5 4 3 5 3

8 9 12



問題分析

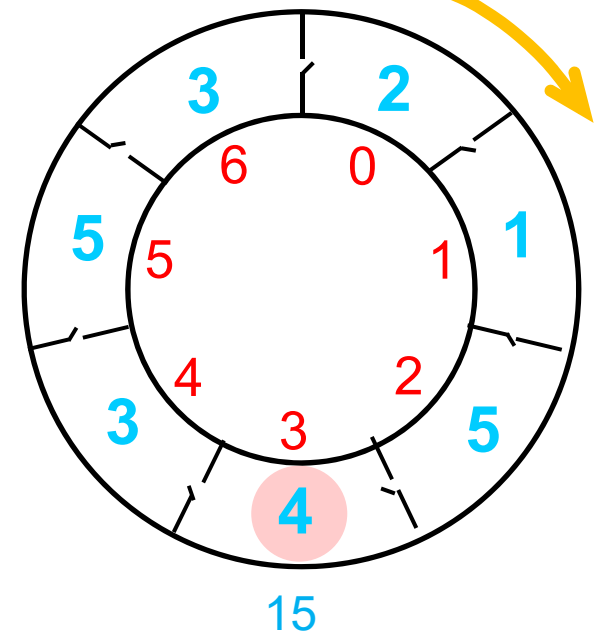
- 看到這樣子一步一步描述的問題，基本的作法是**模擬**，藉由電腦的速度與正確性把結果**暴力**解出來，不過看到 **n** 可以到達 200000，**m** 到達 20000，這種數值範圍可以想像模擬時需要盡量節省時間，否則沒有辦法在指定的時間限制裡完成
- 資料的數量到達 **10^5** 時，執行的方法如果是 **$O(n^2)$** 以上的，像是兩層的暴力迴圈、氣泡排序等等，就會超過時間限制了
- 既然是**模擬**，最基本的解題步驟就是拿一個不太大的測資，**手動模擬**看看，可以讓你弄清楚題目的特性

Sample Input

7 3

2 1 5 4 3 5 3

8 9 12



問題分析

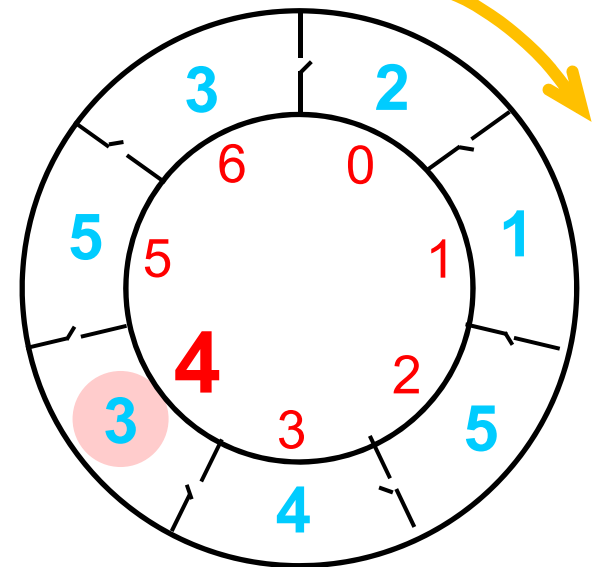
- 看到這樣子一步一步描述的問題，基本的作法是**模擬**，藉由電腦的速度與正確性把結果**暴力**解出來，不過看到 **n** 可以到達 200000，**m** 到達 20000，這種數值範圍可以想像模擬時需要盡量節省時間，否則沒有辦法在指定的時間限制裡完成
- 資料的數量到達 **10^5** 時，執行的方法如果是 **$O(n^2)$** 以上的，像是兩層的暴力迴圈、氣泡排序等等，就會超過時間限制了
- 既然是**模擬**，最基本的解題步驟就是拿一個不太大的測資，**手動模擬**看看，可以讓你弄清楚題目的特性

Sample Input

7 3
2 1 5 4 3 5 3
8 9 12

Sample Output

4

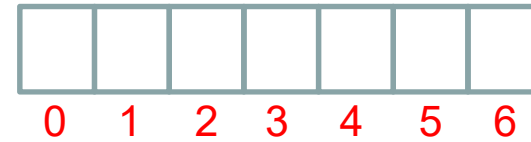


簡單迴圈基準實作

- 以陣列存放 $p_0, p_1, \dots, p_{n-1}$

簡單迴圈基準實作

- 以陣列存放 $p_0, p_1, \dots, p_{n-1}$

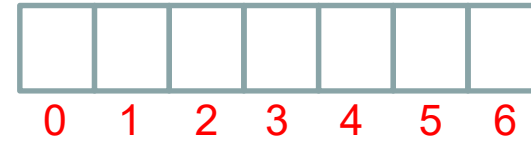


```
01 int n, m, i, j, r, q, p[200000];
```

簡單迴圈基準實作

- 以陣列存放 $p_0, p_1, \dots, p_{n-1}$

```
01 int n, m, i, j, r, q, p[200000];
```



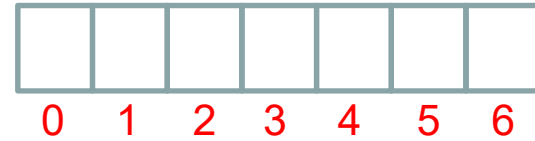
範例一輸入
7 3

簡單迴圈基準實作

- 以陣列存放 $p_0, p_1, \dots, p_{n-1}$

```
01 int n, m, i, j, r, q, p[200000];
```

```
02 scanf("%d%d", &n, &m);
```



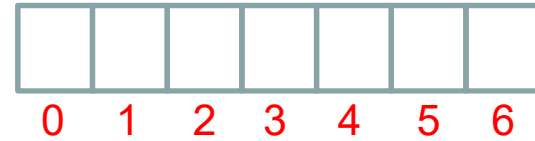
範例一輸入
7 3

簡單迴圈基準實作

- 以陣列存放 $p_0, p_1, \dots, p_{n-1}$

```
01 int n, m, i, j, r, q, p[200000];
```

```
02 scanf("%d%d", &n, &m);
```



範例一輸入
7 3
2 1 5 4 3 5 3

簡單迴圈基準實作

- 以陣列存放 $p_0, p_1, \dots, p_{n-1}$

```
01 int n, m, i, j, r, q, p[200000];  
02 scanf("%d%d", &n, &m);  
03 for (i=0; i<n; i++)  
04     scanf("%d", &p[i]);
```

2	1	5	4	3	5	3
0	1	2	3	4	5	6

範例一輸入
7 3
2 1 5 4 3 5 3

簡單迴圈基準實作

- 以陣列存放 $p_0, p_1, \dots, p_{n-1}$

```
01 int n, m, i, j, r, q, p[200000];
```

```
02 scanf("%d%d", &n, &m);
```

```
03 for (i=0; i<n; i++)
```

```
04     scanf("%d", &p[i]);
```

2	1	5	4	3	5	3
0	1	2	3	4	5	6

範例一輸入

7 3

2 1 5 4 3 5 3

8 9 12

簡單迴圈基準實作

- 以陣列存放 $p_0, p_1, \dots, p_{n-1}$

2	1	5	4	3	5	3
0	1	2	3	4	5	6

```
01 int n, m, i, j, r, q, p[200000];
02 scanf("%d%d", &n, &m);
03 for (i=0; i<n; i++)
04     scanf("%d", &p[i]);
05 for (i=j=0; j<m; j++) {
06     scanf("%d", &q);

09 }
```

範例一輸入

7 3

2 1 5 4 3 5 3

8 9 12

簡單迴圈基準實作

- 以陣列存放 $p_0, p_1, \dots, p_{n-1}$

2	1	5	4	3	5	3
0	1	2	3	4	5	6

```
01 int n, m, i, j, r, q, p[200000];
02 scanf("%d%d", &n, &m);
03 for (i=0; i<n; i++)
04     scanf("%d", &p[i]);
05 for (i=j=0; j<m; j++) {
06     scanf("%d", &q);
07     for (r=0; r<q; i=(i+1)%n)
08         r+=p[i];
09 }
```

範例一輸入

7 3

2 1 5 4 3 5 3

8 9 12

- 以迴圈模擬一間一間房間前進
 r 是蒐集到的點數

簡單迴圈基準實作

- 以陣列存放 $p_0, p_1, \dots, p_{n-1}$

2	1	5	4	3	5	3
0	1	2	3	4	5	6

```
01 int n, m, i, j, r, q, p[200000];
02 scanf("%d%d", &n, &m);
03 for (i=0; i<n; i++)
04     scanf("%d", &p[i]);
05 for (i=j=0; j<m; j++) {
06     scanf("%d", &q);
07     for (r=0; r<q; i=(i+1)%n)
08         r+=p[i];
09 }
```

範例一輸入

7 3

2 1 5 4 3 5 3

8 9 12

- 以迴圈模擬一間一間房間前進
 r 是蒐集到的點數
- 到達第 n 間房間的時候藉由除以 n 的餘數折返陣列最前端

簡單迴圈基準實作

- 以陣列存放 $p_0, p_1, \dots, p_{n-1}$

2	1	5	4	3	5	3
0	1	2	3	4	5	6

```
01 int n, m, i, j, r, q, p[200000];
```

```
02 scanf("%d%d", &n, &m);
```

```
03 for (i=0; i<n; i++)
```

```
04     scanf("%d", &p[i]);
```

```
05 for (i=j=0; j<m; j++) {
```

```
06     scanf("%d", &q);
```

```
07     for (r=0; r<q; i=(i+1)%n) •
```

```
08         r+=p[i];
```

```
09 }
```

```
10 printf("%d\n", i);
```

範例一輸入

7 3

2 1 5 4 3 5 3

8 9 12

範例一輸出

4

以迴圈模擬一間一間房間前進
 r 是蒐集到的點數

- 到達第 n 間房間的時候藉由除以 n 的餘數折返陣列最前端

簡單迴圈基準實作

- 以陣列存放 $p_0, p_1, \dots, p_{n-1}$

2	1	5	4	3	5	3
0	1	2	3	4	5	6

```
01 int n, m, i, j, r, q, p[200000];
```

```
02 scanf("%d%d", &n, &m);
```

```
03 for (i=0; i<n; i++)
```

```
04     scanf("%d", &p[i]);
```

```
05 for (i=j=0; j<m; j++) {
```

```
06     scanf("%d", &q);
```

```
07     for (r=0; r<q; i=(i+1)%n) • 以迴圈模擬一間一間房間前進
```

r 是蒐集到的點數

```
08         r+=p[i];
```

```
09 }
```

```
10 printf("%d\n", i);
```

範例一輸入

7 3

2 1 5 4 3 5 3

8 9 12

範例一輸出

4

- 內層迴圈最多把陣列 p 走過一次， n 個元素，外層迴圈執行 m 次， m 個 q 值，所以時間需要 4×10^9 ，會超過 2 秒限制