

題目 B 蚯蚓疊積木

執行時間限制: 5 秒

還記得老蚯蚓的那些寶物嗎？隨著蚯蚓們挖到的寶物越來越多，蚯蚓的公用倉庫越疊越高，老蚯蚓發現自己竟愛上了疊東西。為了滿足自己疊東西的欲望，老蚯蚓從寶物堆中找出了許多大小不一的積木，不停的疊來疊去。

然而，在疊了七七四十九年後，單純的疊積木再也無法滿足老蚯蚓了。為了讓疊積木變得更有趣，老蚯蚓決定出個考題考考自己：依照某個順序每次拿起一個積木，一一決定是否將這個積木疊到積木塔上，並使得最後積木塔上疊的積木最多。當然，為了要疊出一個穩固的積木塔，任何一個積木都只能疊在一個比他自己大的積木上。

幸運，也不幸的，因為老蚯蚓天生對積木的直覺，這個問題在老蚯蚓嘗試玩了三次以後就變得簡單無聊。所以老蚯蚓決定問問自己一個更有挑戰性的問題：有哪些積木如果被規定了不准疊到積木塔上，會使得能疊到積木塔上的最多積木數量減少？

■ 輸入檔說明

輸入檔的第一行有一個正整數 T ($T \leq 6000$)，表示接下來總共有幾筆測試資料。

每組測試資料的第一行的開頭有一個正整數 N ($1 \leq N \leq 200000$) 代表蚯蚓打算依序拿起 N 個積木，同一行接著有 N 個整數 S_i ($1 \leq S_i \leq N$)，代表那 N 個積木的大小，所有積木的大小都是不同的。我們保證有 99% 的測試資料 $N \leq 1000$ 。

■ 輸出檔說明

為了輸出方便，我們先定義一個雜湊函數 *hash*，能夠將一個長度為 *size* 的整數序列轉換成一個整數。

```
int hash(int numbers[], int size){
    int value = 0, i;
    for(i = 0; i < size; i++){
        value ^= (numbers[i] + i + 1);
    }
    return value;
}
```

對於每一筆測試資料，請輸出中間用一個空白分隔的兩個整數 A, B 。 A 代表有幾個積木若被規定不准放到積木塔上，會使得能疊到積木塔上的最多積木數量減少。 B 代表將那 A 個積木的索引值 (依據輸入順序從 1 開始編號到 N) 從小排到大後雜湊的結果。

■ 範例輸入

```
3
6 6 5 4 3 2 1
6 1 2 4 3 5 6
6 2 3 5 4 6 1
```

■ 範例輸出

```
0 0
4 4
3 14
```

■ 範例說明

第三筆測試資料中，2 3 5 6 或 2 3 4 6 是唯二能疊四個積木的疊法。但拔掉重量為 4 或 5 的積木都還會剩下一種疊法，所以只有第 1、2、5 個積木是重要的。因此答案為將 [1, 2, 5] 這個序列雜湊後的結果，即 $(1 + 1) \oplus (2 + 2) \oplus (5 + 3) = 14$ 。