

Problem 798: Smallest Rotation with Highest Score

Problem Information

Difficulty: Hard

Acceptance Rate: 53.11%

Paid Only: No

Tags: Array, Prefix Sum

Problem Description

You are given an array `nums`. You can rotate it by a non-negative integer `k` so that the array becomes `[nums[k], nums[k + 1], ... nums[nums.length - 1], nums[0], nums[1], ..., nums[k-1]]`. Afterward, any entries that are less than or equal to their index are worth one point.

* For example, if we have `nums = [2,4,1,3,0]`, and we rotate by `k = 2`, it becomes `[1,3,0,2,4]`. This is worth `3` points because `1 > 0` [no points], `3 > 1` [no points], `0 <= 2` [one point], `2 <= 3` [one point], `4 <= 4` [one point].

Return `the rotation index k` that corresponds to the highest score we can achieve if we rotated `nums` by it. If there are multiple answers, return the smallest such index `k`.

Example 1:

Input: `nums = [2,3,1,4,0]` **Output:** `3` **Explanation:** Scores for each `k` are listed below:
`k = 0`, `nums = [2,3,1,4,0]`, score `2` `k = 1`, `nums = [3,1,4,0,2]`, score `3` `k = 2`, `nums = [1,4,0,2,3]`, score `3` `k = 3`, `nums = [4,0,2,3,1]`, score `4` `k = 4`, `nums = [0,2,3,1,4]`, score `3` So we should choose `k = 3`, which has the highest score.

Example 2:

Input: `nums = [1,3,0,2,4]` **Output:** `0` **Explanation:** `nums` will always have 3 points no matter how it shifts. So we will choose the smallest `k`, which is 0.

Constraints:

```
*`1 <= nums.length <= 105` *`0 <= nums[i] < nums.length`
```

Code Snippets

C++:

```
class Solution {  
public:  
    int bestRotation(vector<int>& nums) {  
  
    }  
};
```

Java:

```
class Solution {  
    public int bestRotation(int[] nums) {  
  
    }  
}
```

Python3:

```
class Solution:  
    def bestRotation(self, nums: List[int]) -> int:
```