

Problem 3520: Minimum Threshold for Inversion Pairs Count

Problem Information

Difficulty: Medium

Acceptance Rate: 55.22%

Paid Only: Yes

Tags: Array, Binary Search, Binary Indexed Tree, Segment Tree

Problem Description

You are given an array of integers `nums` and an integer `k`.

An inversion pair with a **threshold** `x` is defined as a pair of indices `(i, j)` such that:

`i < j` * `nums[i] > nums[j]` * The difference between the two numbers is **at most** `x` (i.e. `nums[i] - nums[j] <= x`).

Your task is to determine the **minimum** integer `min_threshold` such that there are **at least** `k` inversion pairs with threshold `min_threshold`.

If no such integer exists, return `-1`.

Example 1:

Input: `nums = [1,2,3,4,3,2,1], k = 7`

Output: 2

Explanation:

For threshold `x = 2`, the pairs are:

1. `(3, 4)` where `nums[3] == 4` and `nums[4] == 3`. 2. `(2, 5)` where `nums[2] == 3` and `nums[5] == 2`. 3. `(3, 5)` where `nums[3] == 4` and `nums[5] == 2`. 4. `(4, 5)` where `nums[4]`

== 3` and `nums[5] == 2`. 5. `(1, 6)` where `nums[1] == 2` and `nums[6] == 1`. 6. `(2, 6)` where `nums[2] == 3` and `nums[6] == 1`. 7. `(4, 6)` where `nums[4] == 3` and `nums[6] == 1`. 8. `(5, 6)` where `nums[5] == 2` and `nums[6] == 1`.

There are less than `k` inversion pairs if we choose any integer less than 2 as threshold.

Example 2:

Input: nums = [10,9,9,9,1], k = 4

Output: 8

Explanation:

For threshold `x = 8`, the pairs are:

1. `(0, 1)` where `nums[0] == 10` and `nums[1] == 9`. 2. `(0, 2)` where `nums[0] == 10` and `nums[2] == 9`. 3. `(0, 3)` where `nums[0] == 10` and `nums[3] == 9`. 4. `(1, 4)` where `nums[1] == 9` and `nums[4] == 1`. 5. `(2, 4)` where `nums[2] == 9` and `nums[4] == 1`. 6. `(3, 4)` where `nums[3] == 9` and `nums[4] == 1`.

There are less than `k` inversion pairs if we choose any integer less than 8 as threshold.

Constraints:

* `1 <= nums.length <= 104` * `1 <= nums[i] <= 109` * `1 <= k <= 109`

Code Snippets

C++:

```
class Solution {
public:
    int minThreshold(vector<int>& nums, int k) {

    }

};
```

Java:

```
class Solution {  
    public int minThreshold(int[] nums, int k) {  
  
    }  
}
```

Python3:

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