Problem 862: Shortest Subarray with Sum at Least K

Problem Information

Difficulty: Hard
Acceptance Rate: 32.42%
Paid Only: No
Tags: Array, Binary Search, Queue, Sliding Window, Heap (Priority Queue), Prefix Sum, Monotonic Queue

Problem Description

Given an integer array `nums` and an integer `k`, return _the length of the shortest non-empty**subarray** of _`nums` _with a sum of at least_`k`. If there is no such **subarray** , return `-1`.

A **subarray** is a **contiguous** part of an array.

Example 1:

Input: nums = [1], k = 1 **Output:** 1

Example 2:

Input: nums = [1,2], k = 4 **Output:** -1

Example 3:

Input: nums = [2,-1,2], k = 3 **Output:** 3

Constraints:

* `1 <= nums.length <= 105` * `-105 <= nums[i] <= 105` * `1 <= k <= 109`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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