

Problem 3409: Longest Subsequence With Decreasing Adjacent Difference

Problem Information

Difficulty: Medium

Acceptance Rate: 15.91%

Paid Only: No

Tags: Array, Dynamic Programming

Problem Description

You are given an array of integers `nums``.

Your task is to find the length of the **longest** subsequence `seq`` of `nums``, such that the **absolute differences** between `_consecutive_` elements form a **non-increasing sequence** of integers. In other words, for a subsequence `seq0`, seq1`, seq2`, ..., seqm`` of `nums``, $|seq1 - seq0| \geq |seq2 - seq1| \geq \dots \geq |seqm - seqm - 1|$.

Return the length of such a subsequence.

Example 1.

Input: `nums = [16,6,3]`

Output: 3

Explanation:

The longest subsequence is `[16, 6, 3]` with the absolute adjacent differences `[10, 3]`.

Example 2.

Input: `nums = [6,5,3,4,2,1]`

Output: 4

****Explanation:****

The longest subsequence is `[6, 4, 2, 1]` with the absolute adjacent differences `[2, 2, 1]`.

****Example 3:****

****Input:**** nums = [10,20,10,19,10,20]

****Output:**** 5

****Explanation:****

The longest subsequence is `[10, 20, 10, 19, 10]` with the absolute adjacent differences `[10, 10, 9, 9]`.

****Constraints:****

$2 \leq \text{nums.length} \leq 104$ $1 \leq \text{nums}[i] \leq 300$

Code Snippets

C++:

```
class Solution {
public:
    int longestSubsequence(vector<int>& nums) {

    }
};
```

Java:

```
class Solution {
    public int longestSubsequence(int[] nums) {

    }
}
```

Python3:

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