1696. Jump Game VI

Medium 2073 76 Add to List Share

You are given a **0-indexed** integer array `nums` and an integer `k`.

You are initially standing at index `0`. In one move, you can jump at most `k` steps forward without going outside the boundaries of the array. That is, you can jump from index `i` to any index in the range `[i + 1, min(n - 1, i + k)]` **inclusive**.

You want to reach the last index of the array (index `n - 1`). Your **score** is the **sum** of all `nums[j]` for each index `j` you visited in the array.

Return *the **maximum score** you can get.*

Example 1:

```
Input: nums = [1,-1,-2,4,-7,3], k = 2
Output: 7
Explanation: You can choose your jumps forming the subsequence
[1,-1,4,3] (underlined above). The sum is 7.
```

Example 2:

```
Input: nums = [10,-5,-2,4,0,3], k = 3
Output: 17
Explanation: You can choose your jumps forming the subsequence
[10,4,3] (underlined above). The sum is 17.
```

Example 3:

```
Input: nums = [1,-5,-20,4,-1,3,-6,-3], k = 2
Output: 0
```

Constraints:

- `1 <= nums.length, k <= 10^5`
- `-10^4 <= nums[i] <= 10^4`

Accepted 62,551 Submissions 141,324

```
class Solut
public:
    int max
nums, int k
        int
        dec
        vec
        dp(
        q.p

        for
{
q.pop_front

dp[q.front(
dp[i] > dp(
q.pop_back(

        }
        ret
    }
};
```

Testcase Run Code Re

Accepted Runti

Your input [1, 2

Output 7

Expected 7