Problem 3487: Maximum Unique Subarray Sum After Deletion

Problem Information

Difficulty: Easy
Acceptance Rate: 40.40%
Paid Only: No
Tags: Array, Hash Table, Greedy

Problem Description

You are given an integer array `nums`.

You are allowed to delete any number of elements from `nums` without making it **empty**. After performing the deletions, select a subarray of `nums` such that:

1. All elements in the subarray are **unique**. 2. The sum of the elements in the subarray is **maximized**.

Return the **maximum sum** of such a subarray.

Example 1:

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

Output: 15

Explanation:

Select the entire array without deleting any element to obtain the maximum sum.

Example 2:

Input: nums = [1,1,0,1,1]

Output: 1

Explanation:

Delete the element `nums[0] == 1`, `nums[1] == 1`, `nums[2] == 0`, and `nums[3] == 1`. Select the entire array `[1]` to obtain the maximum sum.

Example 3:

Input: nums = [1,2,-1,-2,1,0,-1]

Output: 3

Explanation:

Delete the elements `nums[2] == -1` and `nums[3] == -2`, and select the subarray `[2, 1]` from `[1, 2, 1, 0, -1]` to obtain the maximum sum.

Constraints:

* `1 <= nums.length <= 100` * `-100 <= nums[i] <= 100`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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