Problem 1695: Maximum Erasure Value

Problem Information

Difficulty: Medium
Acceptance Rate: 64.01%
Paid Only: No
Tags: Array, Hash Table, Sliding Window

Problem Description

You are given an array of positive integers `nums` and want to erase a subarray containing **unique elements**. The **score** you get by erasing the subarray is equal to the **sum** of its elements.

Return _the**maximum score** you can get by erasing **exactly one** subarray._

An array `b` is called to be a subarray of `a` if it forms a contiguous subsequence of `a`, that is, if it is equal to `a[l],a[l+1],...,a[r]` for some `(l,r)`.

Example 1:

Input: nums = [4,2,4,5,6] **Output:** 17 **Explanation:** The optimal subarray here is [2,4,5,6].

Example 2:

Input: nums = [5,2,1,2,5,2,1,2,5] **Output:** 8 **Explanation:** The optimal subarray here is [5,2,1] or [1,2,5].

Constraints:

* `1 <= nums.length <= 105` * `1 <= nums[i] <= 104`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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