Problem 1186: Maximum Subarray Sum with One Deletion

Problem Information

Difficulty: Medium
Acceptance Rate: 45.80%
Paid Only: No
Tags: Array, Dynamic Programming

Problem Description

Given an array of integers, return the maximum sum for a **non-empty** subarray (contiguous elements) with at most one element deletion. In other words, you want to choose a subarray and optionally delete one element from it so that there is still at least one element left and the sum of the remaining elements is maximum possible.

Note that the subarray needs to be **non-empty** after deleting one element.

Example 1:

Input: arr = [1,-2,0,3] **Output:** 4 **Explanation:** Because we can choose [1, -2, 0, 3] and drop -2, thus the subarray [1, 0, 3] becomes the maximum value.

Example 2:

Input: arr = [1,-2,-2,3] **Output:** 3 **Explanation:** We just choose [3] and it's the maximum sum.

Example 3:

Input: arr = [-1,-1,-1,-1] **Output:** -1 **Explanation:** The final subarray needs to be non-empty. You can't choose [-1] and delete -1 from it, then get an empty subarray to make the sum equals to 0.

Constraints:

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

Code Snippets

C++:

```
class Solution {
public:
int maximumSum(vector<int>& arr) {

}
};
```

Java:

```
class Solution {
public int maximumSum(int[] arr) {

}
}
```

Python3:

```
class Solution:
def maximumSum(self, arr: List[int]) -> int:
```