Problem 1588: Sum of All Odd Length Subarrays

Problem Information

Difficulty: Easy
Acceptance Rate: 83.70%
Paid Only: No
Tags: Array, Math, Prefix Sum

Problem Description

Given an array of positive integers `arr`, return _the sum of all possible**odd-length subarrays** of _`arr`.

A **subarray** is a contiguous subsequence of the array.

Example 1:

Input: arr = [1,4,2,5,3] **Output:** 58 **Explanation:** The odd-length subarrays of arr and their sums are: [1] = 1 [4] = 4 [2] = 2 [5] = 5 [3] = 3 [1,4,2] = 7 [4,2,5] = 11 [2,5,3] = 10 [1,4,2,5,3] = 15 If we add all these together we get 1 + 4 + 2 + 5 + 3 + 7 + 11 + 10 + 15 = 58

Example 2:

Input: arr = [1,2] **Output:** 3 **Explanation:** There are only 2 subarrays of odd length, [1] and [2]. Their sum is 3.

Example 3:

Input: arr = [10,11,12] **Output:** 66

Constraints:

* `1 <= arr.length <= 100` * `1 <= arr[i] <= 1000`

Follow up:

Could you solve this problem in O(n) time complexity?

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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