

Description | Accepted | Editorial | Solutions | Submissions

1588. Sum of All Odd Length Subarrays

Solved

Easy | Topics | Companies | Hint

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

The screenshot shows a code editor with a C# solution for the problem 'Sum of All Odd Length Subarrays'. The solution is accepted and shows a runtime of 67 ms, beating 40.00% of other solutions. The code iterates over all possible starting points and odd lengths, summing the subarrays.

```
public class Solution {
    public int SumOddLengthSubarrays(int[] arr) {
        int totalSum = 0;
        int n = arr.Length;

        // Iterate over all possible starting points
        for (int start = 0; start < n; start++) {
            // Iterate over all possible odd lengths
            for (int length = 1; start + length <= n; length += 2) {
                // Sum the subarray from start to start + length
                for (int i = start; i < start + length; i++) {
                    totalSum += arr[i];
                }
            }
        }

        return totalSum;
    }
}
```

Testcase: Accepted Runtime: 83 ms

Case 1 Case 2 Case 3

Input: `arr = [1,4,2,5,3]`

Output: 58

Expected: 58

Код:

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

```

int totalSum = 0;
int n = arr.Length;

// Iterate over all possible starting points
for (int start = 0; start < n; start++)
{
    // Iterate over all possible odd lengths
    for (int length = 1; start + length <= n; length += 2)
    {
        // Sum the subarray from start to start + length
        for (int i = start; i < start + length; i++)
        {
            totalSum += arr[i];
        }
    }
}

return totalSum;
}
}

```