

Problem 1477: Find Two Non-overlapping Sub-arrays Each With Target Sum

Problem Information

Difficulty: Medium

Acceptance Rate: 36.67%

Paid Only: No

Tags: Array, Hash Table, Binary Search, Dynamic Programming, Sliding Window

Problem Description

You are given an array of integers `arr` and an integer `target`.

You have to find **two non-overlapping sub-arrays** of `arr` each with a sum equal `target`. There can be multiple answers so you have to find an answer where the sum of the lengths of the two sub-arrays is **minimum**.

Return the minimum sum of the lengths of the two required sub-arrays, or return `-1` if you cannot find such two sub-arrays.

Example 1:

Input: `arr = [3,2,2,4,3]`, `target = 3` **Output:** `2` **Explanation:** Only two sub-arrays have sum = 3 (`[3]` and `[3]`). The sum of their lengths is 2.

Example 2:

Input: `arr = [7,3,4,7]`, `target = 7` **Output:** `2` **Explanation:** Although we have three non-overlapping sub-arrays of sum = 7 (`[7]`, `[3,4]` and `[7]`), but we will choose the first and third sub-arrays as the sum of their lengths is 2.

Example 3:

Input: `arr = [4,3,2,6,2,3,4]`, `target = 6` **Output:** `-1` **Explanation:** We have only one sub-array of sum = 6.

****Constraints:****

*`1` <= arr.length <= 105` *`1` <= arr[i] <= 1000` *`1` <= target <= 108`

Code Snippets

C++:

```
class Solution {  
public:  
    int minSumOfLengths(vector<int>& arr, int target) {  
  
    }  
};
```

Java:

```
class Solution {  
    public int minSumOfLengths(int[] arr, int target) {  
  
    }  
}
```

Python3:

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