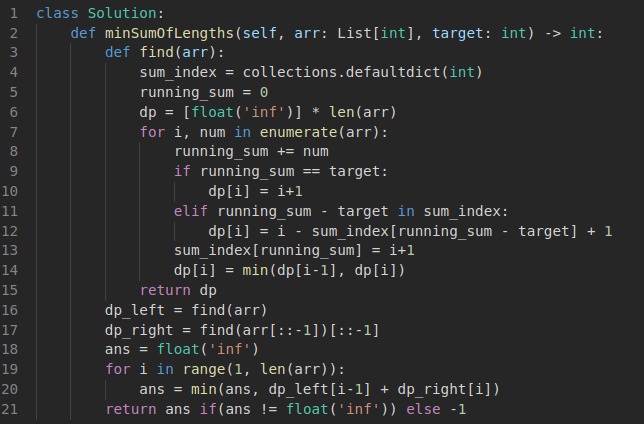
**Assignment 2 - HOPE**

**1.Find two non-overlapping subarrays with sum equal target. Find answer where the sum of length of two- sub array is minimum. -**

Input : arr=[3,2,2,4,3] Target=3

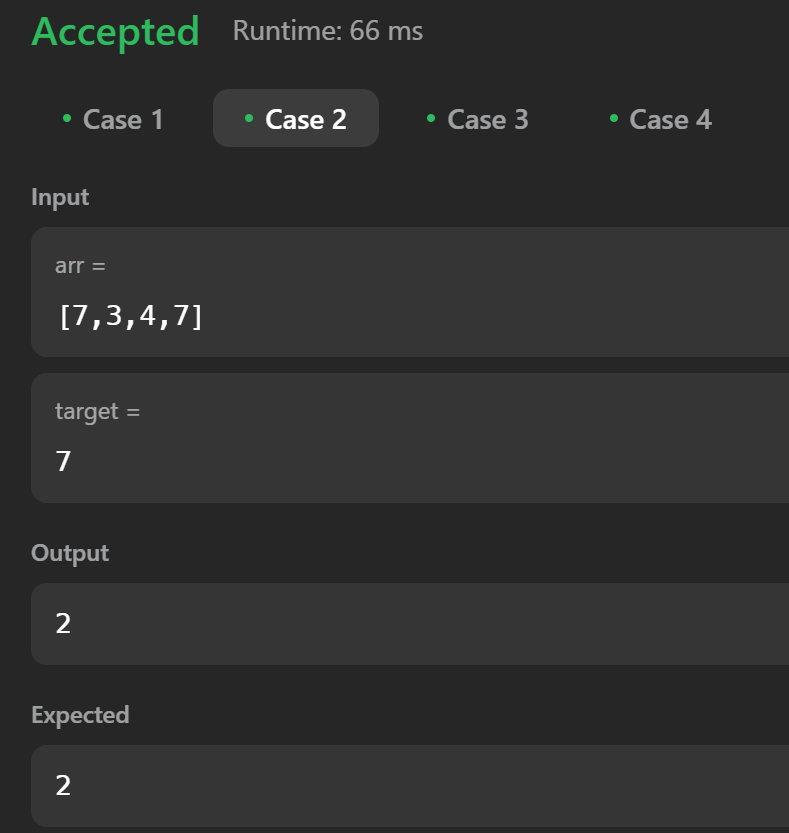
Output : 2

Explanation: Only two sub-arrays have sum=3([3] and [3]). The sum of their lengths is 2.

Output:

Explanation:

For each element, get the min length of target sum subarrays of its left and its right. Then add them to get the answer (minimum). Keep track of the running sum and the sum-index mapping, if the element == target from the start itself then put the length in dp array else if the target can be achieved from the middle also, then update the dp array by using the sum-index map. And also update the dictionary for running sum.



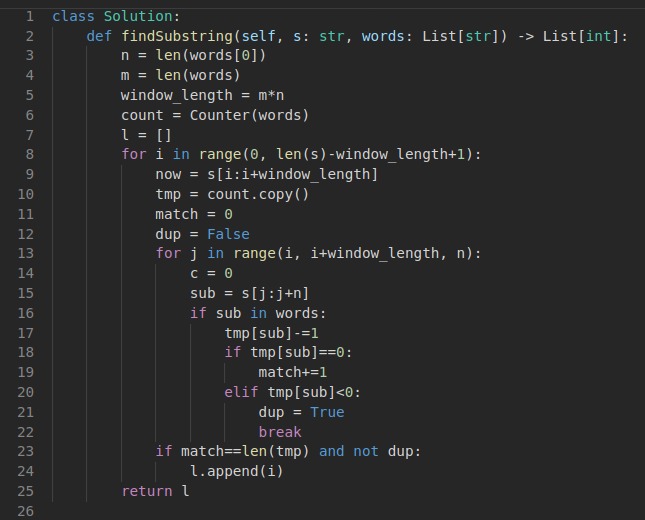
**2. Concatenated Substring in s is a substring that contains all the strings of any permutation of words concatenated - 30**

Input :s = "barfoothefoobarman", words = ["foo","bar"]

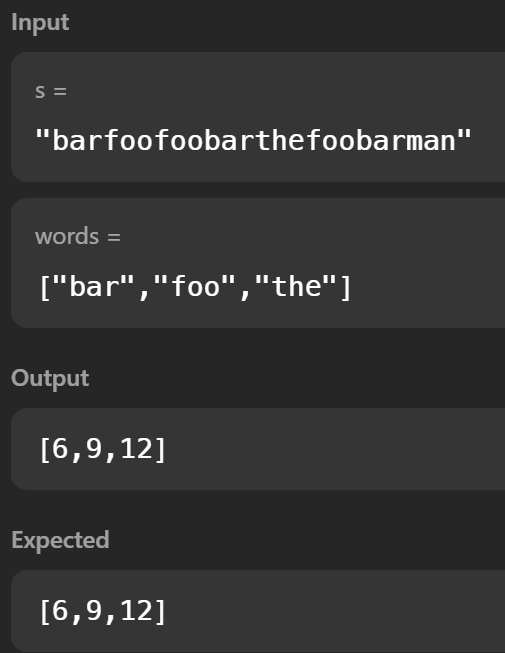
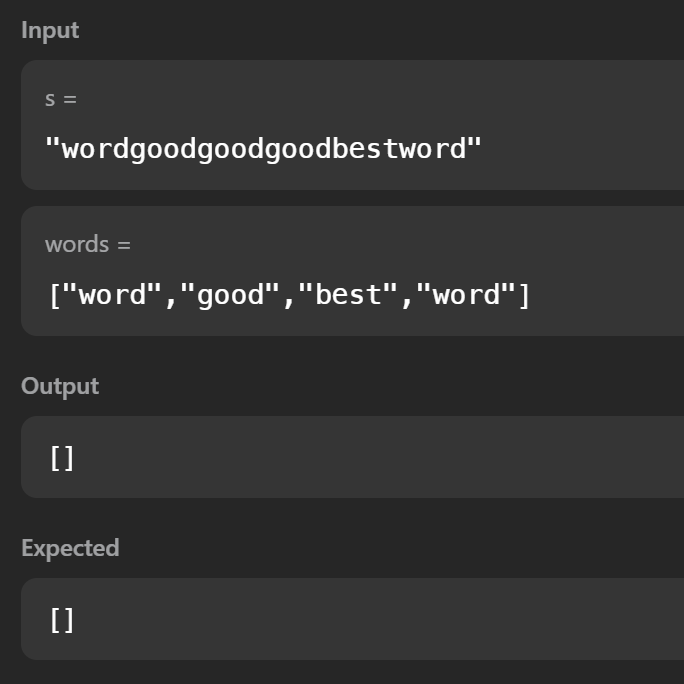
**Output:** [0,9]

**Explanation:** Since words.length == 2 and words[i].length == 3, the concatenated substring has to be of length 6. The substring starting at 0 is "barfoo". It is the concatenation of ["bar","foo"] which is a permutation of words. The substring starting at 9 is "foobar". It is the concatenation of ["foo","bar"] which is a permutation of words. The output order does not matter. Returning [9,0] is fine too.

# Solution:



# Output:



# Explanation:

Start from the first element. All the words are equal. So calculate window size. Split the entire string in all possible ways and have a dictionary to check if the currect dictionary values equals the first formed dictionary using the words given.