Split Array In Three Equal Sum Subarrays

solved by	Senan
Platform	GeeksForGeeks
difficulty	Medium
tags	Logic
language	C++
solved on	@07/11/2024
link	https://www.geeksforgeeks.org/problems/split-array-in-three-equal-sum-subarrays/1
Completion	☑

Intuition

To split an array into three equal sum subarrays, we need to ensure that the sum of elements in each part is the same. If the total sum of the array is not divisible by 3, it's impossible to divide it as required, so we return an indication of failure. Otherwise, we look for two indices that divide the array into three parts with equal sums.

Approach

1. **Calculate Total Sum**: First, calculate the total sum of the array. If the total sum is not divisible by 3, return `{-1, -1}` as it's not possible to split the array as desired.
2. **Define Target Sum**: Divide the total sum by 3 to get the sum each part should have (let's call it `oneThird`).
3. **Traverse the Array**: Start from the beginning of the array and keep a running sum. Whenever the running sum matches `oneThird`, mark the index as a boundary of one part, reset the running sum to zero, and continue. Repeat this until we have two boundaries (since the third part is implicitly defined).
4. **Return Boundaries**: If we find exactly two valid boundaries, return them. If we can't find them, return `{-1, -1}`.

Complexity

- **Time Complexity**: **O(n)**, where **n** is the size of the array, as we only traverse the array once to calculate the total sum and once to find the boundaries.
- **Space Complexity**: **O(1)**, ignoring the output space, as we use only a few extra variables.

Code

```
class Solution {
  public:
    vector<int> findSplit(vector<int>& arr) {
        // Calculate total sum of the array
        int totalSum = accumulate(arr.begin(), arr.end(), 0);
        if (totalSum % 3 != 0) return {-1, -1};

        int oneThird = totalSum / 3;
        int currentSum = 0;
        vector<int> answer;

        for (int i = 0; i < arr.size(); i++) {
            currentSum += arr[i];

            if (currentSum == oneThird) {
                answer.push_back(i);
                currentSum = 0;

                if (answer.size() == 2) break;
            }
        }

        return answer.size() == 2 ? answer : vector<int>{-1, -1};
    }
};
```