You have a long strip of paper with integers written on it in a single line from left to right. You wish to cut the paper into exactly three pieces such that each piece contains at least one integer and the sum of the integers in each piece is the same. You cannot cut through a number, i.e. each initial number will unambiguously belong to one of the pieces after cutting. How many ways can you do it?

It is guaranteed that the sum of all elements in the array is divisible by 3.

**Example**

For a = [0, -1, 0, -1, 0, -1], the output should be  
threeSplit(a) = 4.

**Input/Output**

* **[time limit] 20000ms (swift)**
* **[input] array.integer a**

*Constraints:*  
5 ≤ a.length ≤ 104,  
-108 ≤ a[i] ≤ 108.

* **[output] integer**

It's guaranteed that for the given test cases the answer always fits signed 32-bit integer type.

<https://codefights.com/arcade/code-arcade/well-of-integration/QmK8kHTyKqh8xDoZk>

|  |
| --- |
| func threeSplit(a: [Int]) -> Int { |
|  | var sum = 0 |
|  | for num in a { |
|  | sum += num |
|  | } |
|  | sum /= 3 |
|  | var cut1 = 0, cut2 = 0, cut3 = 0, count = 0 |
|  | var sum1 = 0, sum2 = 0, sum3 = 0 |
|  | for cut1 in 0..<a.count { |
|  | sum1 += a[cut1] |
|  | if sum1 == sum { |
|  | sum2 = 0 |
|  | if cut1 + 1 >= a.count { |
|  | continue |
|  | } |
|  | for cut2 in (cut1 + 1)..<a.count { |
|  | sum2 += a[cut2] |
|  | if sum2 == sum { |
|  | sum3 = 0 |
|  | if cut2 + 1 >= a.count { |
|  | continue |
|  | } |
|  | for cut3 in (cut2 + 1)..<a.count { |
|  | sum3 += a[cut3] |
|  | } |
|  | if sum3 == sum { |
|  | count += 1 |
|  | } |
|  | } |
|  | } |
|  | } |
|  | } |
|  | return count |
|  | } |