Problem 805: Split Array With Same Average

Problem Information

Difficulty: Hard
Acceptance Rate: 26.51%
Paid Only: No
Tags: Array, Math, Dynamic Programming, Bit Manipulation, Bitmask

Problem Description

You are given an integer array `nums`.

You should move each element of `nums` into one of the two arrays `A` and `B` such that `A` and `B` are non-empty, and `average(A) == average(B)`.

Return `true` if it is possible to achieve that and `false` otherwise.

Note that for an array `arr`, `average(arr)` is the sum of all the elements of `arr` over the length of `arr`.

Example 1:

Input: nums = [1,2,3,4,5,6,7,8] **Output:** true **Explanation:** We can split the array into [1,4,5,8] and [2,3,6,7], and both of them have an average of 4.5.

Example 2:

Input: nums = [3,1] **Output:** false

Constraints:

* `1 <= nums.length <= 30` * `0 <= nums[i] <= 104`

Code Snippets

C++:

```
class Solution {
public:
bool splitArraySameAverage(vector<int>& nums) {

}
};
```

Java:

```
class Solution {
public boolean splitArraySameAverage(int[] nums) {

}
}
```

Python3:

```
class Solution:
def splitArraySameAverage(self, nums: List[int]) -> bool:
```