Problem 996: Number of Squareful Arrays

Problem Information

Difficulty: Hard
Acceptance Rate: 50.91%
Paid Only: No
Tags: Array, Hash Table, Math, Dynamic Programming, Backtracking, Bit Manipulation, Bitmask

Problem Description

An array is **squareful** if the sum of every pair of adjacent elements is a **perfect square**.

Given an integer array nums, return _the number of permutations of_`nums` _that are**squareful**_.

Two permutations `perm1` and `perm2` are different if there is some index `i` such that `perm1[i] != perm2[i]`.

Example 1:

Input: nums = [1,17,8] **Output:** 2 **Explanation:** [1,8,17] and [17,8,1] are the valid permutations.

Example 2:

Input: nums = [2,2,2] **Output:** 1

Constraints:

* `1 <= nums.length <= 12` * `0 <= nums[i] <= 109`

Code Snippets

C++:

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

}
};
```

Java:

```
class Solution {
public int numSquarefulPerms(int[] nums) {

}
}
```

Python3:

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