Problem 1711: Count Good Meals

Problem Information

Difficulty: Medium
Acceptance Rate: 32.21%
Paid Only: No
Tags: Array, Hash Table

Problem Description

A **good meal** is a meal that contains **exactly two different food items** with a sum of deliciousness equal to a power of two.

You can pick **any** two different foods to make a good meal.

Given an array of integers `deliciousness` where `deliciousness[i]` is the deliciousness of the `i■■■■■■th■■■■`■■■■ item of food, return _the number of different**good meals** you can make from this list modulo_ `109 + 7`.

Note that items with different indices are considered different even if they have the same deliciousness value.

Example 1:

Input: deliciousness = [1,3,5,7,9] **Output:** 4 **Explanation:** The good meals are (1,3), (1,7), (3,5) and, (7,9). Their respective sums are 4, 8, 8, and 16, all of which are powers of 2.

Example 2:

Input: deliciousness = [1,1,1,3,3,3,7] **Output:** 15 **Explanation:** The good meals are (1,1) with 3 ways, (1,3) with 9 ways, and (1,7) with 3 ways.

Constraints:

* `1 <= deliciousness.length <= 105` * `0 <= deliciousness[i] <= 220`

Code Snippets

C++:

```
class Solution {
public:
int countPairs(vector<int>& deliciousness) {

}
};
```

Java:

```
class Solution {
public int countPairs(int[] deliciousness) {

}
}
```

Python3:

```
class Solution:
def countPairs(self, deliciousness: List[int]) -> int:
```