5642. Count Good Meals

My Submissions (/contest/weekly-contest-222/problems/count-good-meals/submissions/) Back to Contest (/contest/weekly-contest-222/)

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* $10^9 + 7$.

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

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Medium

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 <= 10^5`
- `0 <= deliciousness[i] <= 2^20`

JavaScript

```javascript
/**
 * @param {number[]} deliciousness
 * @return {number}
 */
const MOD = BigInt(1e9 + 7);
const countPairs = (deliciousness) => {
    deliciousness = deliciousness.map(x => BigInt(x));
    let n = deliciousness.length;
    let cnt = 0n;
    for (let i = 0n; i < n; i++) {
        if (deliciousness[i] % 2n == 1n) {
            for (let j = i + 1n; j < n; j++) {
                if (deliciousness[i] == 1n && deliciousness[j] == 0n) cnt++;
                if (deliciousness[j] % 2n == 0n) continue;
                let sum = deliciousness[i] + deliciousness[j];
                if (isPowerTwo(sum)) {
                    cnt++;
                }
            }
        } else {
            for (let j = i + 1n; j < n; j++) {
                if (deliciousness[i] == 0n && deliciousness[j] == 1n) cnt++;
                if (deliciousness[j] % 2n == 1n) continue;
                let sum = deliciousness[i] + deliciousness[j];
                if (isPowerTwo(sum)) {
                    cnt++;
                }
            }
        }
    }
    return Number(cnt % MOD);
};

const isPowerTwo = (x) => {
```

```
35      return (x != 0n) && ((x & (x - 1n)) == 0n);
36  };
```

Custom Testcase Use Example Testcases

Run Submit

Submission Result: Time Limit Exceeded (/submissions/detail/437828299/) More Details > (/submissions/detail/437828299/)

Last executed input: [269731,778845,193336,330952,964,60,5,11,874888,173688,61,67,118079,12993,1,0,13594,19174,1,7,13…