

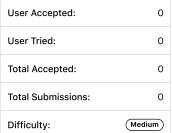


## 5642. Count Good Meals

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.



## 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<sup>5</sup>
- 0 <= deliciousness[i] <= 2<sup>20</sup>

```
JavaScript
                                                                                                                              \mathfrak{C}
1 - /**
     * @param {number[]} deliciousness
     * @return {number}
 3
 4
 5
    const MOD = BigInt(1e9 + 7);
    const countPairs = (deliciousness) => {
6
        deliciousness = deliciousness.map(x \Rightarrow BigInt(x));
 7
8
        let n = deliciousness.length;
        let cnt = 0n;
9
10
        for (let i = 0n; i < n; i++) {
             if (deliciousness[i] % 2n == 1n) {
11 ▼
12 •
                 for (let j = i + 1n; j < n; j++) {
                     if (deliciousness[i] == 1n && deliciousness[j] == 0n) cnt++;
13
                     if (deliciousness[j] % 2n == 0n) continue;
14
15
                     let sum = deliciousness[i] + deliciousness[j];
16
                     if (isPowerTwo(sum)) {
17
                         cnt++;
18
19
20 •
             } else {
                 for (let j = i + 1n; j < n; j++) {
21 •
22
                     if (deliciousness[i] == 0n && deliciousness[j] == 1n) cnt++;
                     if (deliciousness[j] % 2n == 1n) continue;
23
                     let sum = deliciousness[i] + deliciousness[j];
24
25 •
                     if (isPowerTwo(sum)) {
26
                          cnt++;
27
28
29
            }
30
31
        return Number(cnt % MOD);
32
    };
33
34 v const isPowerTwo = (x) => {
```

```
35
           return (x != 0n) && ((x & (x - 1n)) == 0n);
 36 };
☐ Custom Testcase
                       Use Example Testcases
                                                                                                                                           △ Submit
                                                                                                                                 Run
Submission Result: Time Limit Exceeded (/submissions/detail/437828299/) ?
                                                                                              More Details > (/submissions/detail/437828299/)
                                [269731, 778845, 193336, 330952, 964, 60, 5, 11, 874888, 173688, 61, 67, 118079, 12993, 1, 0, 13594, 19174, 1, 7, 13...]
    Last executed input:
Copyright © 2021 LeetCode
                         Help Center (/support) | Jobs (/jobs) | Bug Bounty (/bugbounty) | Students (/student) | Terms (/terms) | Privacy Policy (/privacy)
United States (/region)
```