5243. Tuple with Same Product

My Submissions (/contest/weekly-contest-224/problems/tuple-with-same-product/submissions/) | Back to Contest (/contest/weekly-contest-224/)

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

Given an array `nums` of **distinct** positive integers, return *the number of tuples* `(a, b, c, d)` *such that* `a * b = c * d` *where* `a`, `b`, `c`*, and* `d` *are elements of* `nums`*, and* `a != b != c != d`.

Example 1:

```
Input: nums = [2,3,4,6]
Output: 8
Explanation: There are 8 valid tuples:
(2,6,3,4) , (2,6,4,3) , (6,2,3,4) , (6,2,4,3)
(3,4,2,6) , (3,4,2,6) , (3,4,6,2) , (4,3,6,2)
```

Example 2:

```
Input: nums = [1,2,4,5,10]
Output: 16
Explanation: There are 16 valids tuples:
(1,10,2,5) , (1,10,5,2) , (10,1,2,5) , (10,1,5,2)
(2,5,1,10) , (2,5,10,1) , (5,2,1,10) , (5,2,10,1)
(2,10,4,5) , (2,10,5,4) , (10,2,4,5) , (10,2,4,5)
(4,5,2,10) , (4,5,10,2) , (5,4,2,10) , (5,4,10,2)
```

Example 3:

```
Input: nums = [2,3,4,6,8,12]
Output: 40
```

Example 4:

```
Input: nums = [2,3,5,7]
Output: 0
```

Constraints:

- `1 <= nums.length <= 1000`
- `1 <= nums[i] <=` 10^4
- All elements in `nums` are **distinct**.

JavaScript

```
/**
 * @param {number[]} nums
 * @return {number}
 */
const tupleSameProduct = (nums) => {
    let n = nums.length;
    nums.sort((a, b) => a - b);
    let m = new Map();
    for (let i = 0; i < n; i++) {
        for (let j = i + 1; j < n; j++) {
            let p = nums[i] * nums[j];
            let tmp = JSON.stringify([nums[i], nums[j]]);
            if (m.has(p)) {
                m.set(p, m.get(p).add(tmp));
            } else {
                m.set(p, new Set([tmp]));
            }
        }
    }
    let res = 0;
    for (const [k, v] of m) {
        let t = v.size;
        if (t > 1) {
            res += Number(combination(t, 2)) * 8;
        }
    }
    return res;
};

const combination = (m, n) => {
    return factorial(m, n) / factorial(n, n);
};

const factorial = (m, n) => {
    let num = BigInt(1);
    let cnt = 0;
    for (let i = BigInt(m); i > 0; i--) {
        if (cnt == n) break;
        num = num * i;
        cnt++;
    }
    return num;
};
```

☐ **Custom Testcase** Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/443948297/) More Details (/submissions/detail/443948297/)

Share your acceptance!

Copyright © 2021 LeetCode Help Center (/support) | Jobs (/jobs) | Bug Bounty (/bugbounty) | Students (/student) | Terms (/terms) | Privacy Policy (/privacy)

United States (/region)