5761. Finding Pairs With a Certain Sum

My Submissions (/contest/weekly-contest-241/problems/finding-pairs-with-a-certain-sum/submissions/)

Back to Contest (/contest/weekly-contest-241/)

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

You are given two integer arrays `nums1` and `nums2`. You are tasked to implement a data structure that supports queries of two types:

1. **Add** a positive integer to an element of a given index in the array `nums2`.
2. **Count** the number of pairs `(i, j)` such that `nums1[i] + nums2[j]` equals a given value (`0 <= i < nums1.length` and `0 <= j < nums2.length`).

Implement the `FindSumPairs` class:

- `FindSumPairs(int[] nums1, int[] nums2)` Initializes the `FindSumPairs` object with two integer arrays `nums1` and `nums2`.
- `void add(int index, int val)` Adds `val` to `nums2[index]`, i.e., apply `nums2[index] += val`.
- `int count(int tot)` Returns the number of pairs `(i, j)` such that `nums1[i] + nums2[j] == tot`.

Example 1:

```
Input
["FindSumPairs", "count", "add", "count", "count", "add", "add", "count"]
[[[1, 1, 2, 2, 2, 3], [1, 4, 5, 2, 5, 4]], [7], [3, 2], [8], [4], [0, 1], [1, 1], [7]]
Output
[null, 8, null, 2, 1, null, null, 11]

Explanation
FindSumPairs findSumPairs = new FindSumPairs([1, 1, 2, 2, 2, 3], [1, 4, 5, 2, 5, 4]);
findSumPairs.count(7);  // return 8; pairs (2,2), (3,2), (4,2), (2,4), (3,4), (4,4) make 2 + 5 and pairs (5,1),
findSumPairs.add(3, 2); // now nums2 = [1,4,5,4,5,4]
findSumPairs.count(8);  // return 2; pairs (5,2), (5,4) make 3 + 5
findSumPairs.count(4);  // return 1; pair (5,0) makes 3 + 1
findSumPairs.add(0, 1); // now nums2 = [2,4,5,4,5,4]
findSumPairs.add(1, 1); // now nums2 = [2,5,5,4,5,4]
findSumPairs.count(7);  // return 11; pairs (2,1), (2,2), (2,4), (3,1), (3,2), (3,4), (4,1), (4,2), (4,4) make
```

Constraints:

- `1 <= nums1.length <= 1000`
- `1 <= nums2.length <=` 10^5
- `1 <= nums1[i] <=` 10^9
- `1 <= nums2[i] <=` 10^5
- `0 <= index < nums2.length`
- `1 <= val <=` 10^5
- `1 <= tot <=` 10^9
- At most `1000` calls are made to `add` and `count` **each**.

JavaScript

```
const counter = (a_or_s) => { let map = new Map(); for (const i of a_or_s) map.set(i, map.get(i) + 1 ||
1); return map; };

const pr = console.log;
function FindSumPairs(a1, a2) {
    // let m1 = counter_value_indexA_in(nums1);
    // let m2 = counter_value_indexA_in(nums2);
    let m1 = counter(a1);
    let m2 = counter(a2);
    // pr(m1, m2);
    return { add, count }
    function add(index, val) {
        let pre = a2[index];
        let cur = pre + val;
        a2[index] = cur;
        let occ = m2.get(pre);
        if (occ == 1) {
            m2.delete(pre);
        } else if (occ > 1) {
            m2.set(pre, occ - 1);
        }
        m2.set(cur, m2.get(cur) + 1 || 1);
    }

    function count(tot) {
        // pr()
        let u1 = Array.from(m1.keys());
        let res = 0;
        // pr("u1", u1);
        // pr(m1, m2);
        for (const x of u1) {
            let y = tot - x;
            if (m2.has(y)) {
                let occ1 = m1.get(x);
                let occ2 = m2.get(y);;
                //pr("x", x, "occ1", occ1, "y", y, "occ2", occ2);
                res += occ1 * occ2;
                //pr("res", res)
            }
        }
        return res;
    }
}
```

☑ **Custom Testcase** Use Example Testcases

```
["FindSumPairs","count","add","count","count","add","add","count"]
[[[1,1,2,2,2,3],[1,4,5,2,5,4]],[7],[3,2],[8],[4],[0,1],[1,1],[7]]
```

How to create a testcase

Run Submit

Submission Result: Accepted (/submissions/detail/493784034/) More Details (/submissions/detail/493784034/)

Share your acceptance!

1