Frequency Queries ☆

Your Frequency Queries submission got 40.00 points. Share Tweet

Proceed to Interview Preparation Kit

Problem Submissions Leaderboard Editorial

You are given q queries. Each query is of the form two integers described below:

- $1\ x$: Insert x in your data structure.
- $2\ y$: Delete one occurence of y from your data structure, if present.
- $3\ z$: Check if any integer is present whose frequency is exactly z. If yes, print 1 else 0.

The queries are given in the form of a 2-D array $queries$ of size q where $queries[i][0]$ contains the operation, and $queries[i][1]$ contains the data element. For example, you are given array $queries = [(1,1),(2,2),(3,2),(1,1),(1,1),(2,1),(3,2)]$. The results of each operation are:

```
Operation   Array   Output
(1,1)       [1]
(2,2)       [1]
(3,2)                0
(1,1)       [1,1]
(1,1)       [1,1,1]
(2,1)       [1,1]
(3,2)                1
```

Return an array with the output: $[0,1]$.

Function Description

Complete the freqQuery function in the editor below. It must return an array of integers where each element is a 1 if there is at least one element value with the queried number of occurrences in the current array, or 0 if there is not.

freqQuery has the following parameter(s):

- queries: a 2-d array of integers

Input Format

The first line contains of an integer q, the number of queries.
Each of the next q lines contains two integers denoting the 2-d array $queries$.

Constraints

- $1 \leq q \leq 10^5$
- $1 \leq x, y, z \leq 10^9$
- All $queries[i][0] \in \{1, 2, 3\}$
- $1 \leq queries[i][1] \leq 10^9$

Output Format

Return an integer array consisting of all the outputs of queries of type 3.

Sample Input 0

```
8
1 5
```

```
1 6
3 2
1 10
1 10
1 6
2 5
3 2
```

Sample Output 0

```
0
1
```

Explanation 0

For the first query of type 3, there is no integer whose frequency is 2 ($array = [5, 6]$). So answer is 0.
For the second query of type 3, there are two integers in $array = [6, 10, 10, 6]$ whose frequency is 2 (integers = 6 and 10). So, the answer is 1.

Sample Input 1

```
4
3 4
2 1003
1 16
3 1
```

Sample Output 1

```
0
1
```

Explanation 1

For the first query of type 3, there is no integer of frequency 4. The answer is 0. For the second query of type 3, there is one integer, 16 of frequency 1 so the answer is 1.

Sample Input 2

```
10
1 3
2 3
3 2
1 4
1 5
1 5
1 4
3 2
2 4
3 2
```

Sample Output 2

```
0
1
1
```

Explanation 2

When the first output query is run, the array is empty. We insert two 4's and two 5's before the second output query, $arr = [4, 5, 5, 4]$ so there are two instances of elements occurring twice. We delete a 4 and run the same query. Now only the instances of 5 satisfy the query.

Change Theme C++

```
vector<int> freqQuery(vector<vector<int>> queries) {
    int q, type, current_count;
    vector<int> query;
    vector<int> query_results;
    unordered_map<long, int> count_map;
    unordered_map<int, long> frequency_map;
    q = queries.size();
    for(int i =0 ; i < q ;i++){
        query = queries[ i ];
        type = query[0];
        switch(type){
            case 1:
                frequency_map[count_map[ query[1] ]]--;
                count_map[ query[1] ]++;
                frequency_map[count_map[ query[1] ]]++;
                break;
            case 2:
                current_count = count_map[ query[1] ];
                if(current_count > 0){
                    frequency_map[current_count]--;
                    count_map[ query[1] ]--;
                    frequency_map[count_map[ query[1] ]]++;
                }
                break;
            case 3:
                query_results.push_back(( frequency_map[query[1]] > 0 )? 1:0);
                break;
            default:
                break;
        }
    }
    return query_results;
```

Line: 11 Col: 17

Upload Code as File Test against custom input Run Code Submit Code

Congratulations

You solved this challenge. Would you like to challenge your friends?

Proceed

Test case 0

Test case 1

Test case 2

Test case 3

Test case 4

Compiler Message

Success

Input (stdin) Download

```
8
1 5
1 6
3 2
```

Test case 5

Test case 6

```
1 10
1 10
1 6
2 5
3 2
```