

	Status ▾	Language ▾	Runtime	Memory	Notes
4	Accepted a day ago	C++	168 ms	115.1 MB	
3	Accepted a day ago	C++	188 ms	115.2 MB	
2	Accepted a day ago	C++	179 ms	115.1 MB	
1	Accepted Dec 23, 2024	C++	179 ms	115.2 MB	

```
1 class Solution {
2 public:
3     long long countOperationsToEmptyArray(vector<int>& nums)//метод принимает вектор целых чисел nums и возвращает количество операций, необходимых
        для опустошения массива.
4
5     {
6         map<int, int> mp;//ключ-число, значение-его индекс
7         int n = nums.size();//сохраняем размер массива
8         for(int i=0; i<n; i++) { //проходим по всем элементам массива
9             mp[nums[i]] = i;//заполняем map mp
10        }
11
12        int lastInd = -1, len = n, removed = 0, counter = 0;
13        //последний индекс, который мы обработали;длина;удаленные элементы;счетчик текущих элементов в текущей последовательности;переменная для
        хранения общего количества операций
14        long long ans = 0;
15        for(auto &[num, ind] : mp) { //цикл по всем элементам mp. Используем auto & для автоматического извлечения пар ключ-значение
16            if(ind < lastInd) { //такой элемент должен быть обработан после уже обработанных элементов
17                ans += len-removed;
18                removed += counter; //обновление количества удаленных элементов
19                counter = 0; //сбрасываем счетчик на 0, так как начинаем новую последовательность
20            }
21            lastInd = ind; //обновляем lastInd на текущий индекс
22
23            counter++; //увеличиваем счетчик на 1, так как мы обработали еще один элемент
24        }
25        ans += len-removed; //добавление оставшихся операций
26
27        return ans;
28    }
29 };
```