15. 3Sum

Medium 19001 1826 Add to List Share

Given an integer array nums, return all the triplets `[nums[i], nums[j], nums[k]]` such that `i != j`, `i != k`, and `j != k`, and `nums[i] + nums[j] + nums[k] == 0`.

Notice that the solution set must not contain duplicate triplets.

Example 1:

```
Input: nums = [-1,0,1,2,-1,-4]
Output: [[-1,-1,2],[-1,0,1]]
```

Example 2:

```
Input: nums = []
Output: []
```

Example 3:

```
Input: nums = [0]
Output: []
```

Constraints:

- `0 <= nums.length <= 3000`
- `-10^5 <= nums[i] <= 10^5`

Accepted 2,054,574 Submissions 6,528,652

Seen this question in a real interview before? Yes No

Companies

Related Topics

Similar Questions

```
1  class Solut
2  public:
3      vector<
   threeSum(ve
4          vec
5          sor
   nums.end())
6          int
7
8          for
9
10
11
12
   nums[k] ==
13
    ans.push_b
   nums[k]});
14
   && nums[k]
15
   1 && nums[j
16
17
18
   nums[k] > -
19
   && nums[k]
20
21
22
23
   1 && nums[j
24
25
26
27
   nums[i] ==
28         }
29         ret
30     }
31 };
```

Your previous code was