

배열을 입력받아 합으로 0을 만들 수 있는 3개의 엘리먼트를 출력하라.

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

Output: [[-1,-1,2],[-1,0,1]]

### 1.브루트 포스

class Solution:

```
def threeSum(self, nums: List[int]) -> List[List[int]]:
    nums.sort()
    results = []
    for i in range(len(nums) - 2):
        if i > 0 and nums[i] == nums[i - 1]:
            continue
        for j in range(i + 1, len(nums) - 1):
            if j > i + 1 and nums[j] == nums[j - 1]:
                continue
            for k in range(j + 1, len(nums)):
                if k > j + 1 and nums[k] == nums[k - 1]:
                    continue
                if nums[i] + nums[j] + nums[k] == 0:
                    results.append([nums[i], nums[j], nums[k]])
    return results
```

▸  $O(n^3)$ 으로 시간 초과

### 2. $O(n^2)$

class Solution:

```
def threeSum(self, nums: List[int]) -> List[List[int]]:
    results = []
    nums.sort()
    for i in range(len(nums) - 2):
        if i > 0 and nums[i] == nums[i - 1]:
            continue
        left, right = i + 1, len(nums) - 1
        while left < right:
            sum = nums[i] + nums[left] + nums[right]
            if sum < 0:
                left += 1
            elif sum > 0:
                right -= 1
            else:
                results.append([nums[i], nums[left], nums[right]])
                while left < right and nums[left] == nums[left + 1]:
                    left += 1
                while left < right and nums[right] == nums[right - 1]:
                    right -= 1
                left += 1
                right -= 1
    return results
```

- $O(n^2)$
- 투포인터를 활용해서 복잡도를 줄인다.