July 2025, CSE 106

Online on Arrays

Section: B1+B2 Time: 40 minutes

Given an array of integers A, your goal is to find any triplet of **unique indices `(i, j, k)`** such that **A[i] + A[j] + A[k] == 0**. You must implement this in **$O(n^2)$** time and space complexity. If there are multiple such triplets, you may return any one of them. You can sort the array if needed (using library functions or on your own, whichever you prefer).

You are free to tackle the problem however you like, but that approach will earn at most 60%. To receive full marks, you need to meet the stated time and space constraints.

```
=== Test 1 ===
Input: [-1, 0, 1, 2, -1, -4]
Output: [1, 2, 5]
Explanation: A[1] + A[2] + A[5] = 0 + 1 + -1 = 0

=== Test 2 ===
Input: [0, 0, 0, 0]
Output: [0, 1, 3]

=== Test 3 ===
Input: [-2, -1, 0, 1, 2, 3]
Output: [0, 1, 5]

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

=== Test 5 ===
Input: [1, 2, -2, -1]
Output: No triplets found

=== Test 6 ===
Input: [0, 1]
Output: No triplets found
```