Three Sum

Given an array of integers, determine if there are three numbers that sum up to zero.  
  
Input: Array of integers  
Output: Boolean

# Example

Input: [1, -1, 3 ,0, 2] => Output: True

Input: [1, 3, 9, -2] => Output: False

# Constraints

Time Complexity: O(N2)  
Auxiliary Space Complexity: Intermediate O(N), Advanced O(1).

An element may not be used more than once in a set of three that are summed.

If asked, assume an efficient in-place sorting method is provided.

# Solution

Intermediate:

1. Loop through the array and create a hash table of all the elements as keys.
2. Performed a nested loop on the array with two pointers
3. Sum the two values at the pointers and return true if the negative of that sum is in the hash table.
4. Outside the nested loop, return false if Step. 3 is never triggered

Advanced:

1. Sort the array using an efficient in-place sorting algorithm (e.g., in-place quicksort)
2. Loop through sorted array with an index i
3. For each value, set two pointers: j starting at i+1, k at the last element
4. Check to see if the sum of i, j, and k equals to zero
5. If the sum is less than zero, increment j. If the sum is greater than zero decrement k and if the sum equals zero, return true.
6. Continue to the next i if ever j and k are equal
7. Outside the for loop, return false if the return statement in Step 5. was never triggered.

# Notes

Recommended to ask the intermediate version with O(N) auxiliary space allowed. Then ask the advanced version if the interviewee is able to solve it within the constraint.

Common first round technical screen question for Facebook.

# 

# Resources

https://leetcode.com/problems/3sum/