

# 1. TEBOW IT

## Homework

Week 3 - Arrays - 5-08-2017

- Array S of n integers, 3 elements abc such that  $a+b+c=0$
- Find unique triplets
- Solution must not contain duplicate triplets

## TEBOW IT

### T-TALK (Questions / Clarifications)

- What is the input?  $\rightarrow$  Vector of ints
- Array sorted?  $\rightarrow$  No
- Contain Duplicate Values?  $\rightarrow$  Yes
- What is returned?  $\rightarrow$  Vector of Vectors of ints
- In the output array, should the elements be sorted?  $\rightarrow$  Yes (ascending)
- Need to validate input
- Return empty vector if invalid input
- If there is no correct answer return empty vector
- Assume we can use any library

### E-EXAMPLES

Sample Input	Class	Out
$[]$	Empty vector	$[]$
$[1, 0], [1]$	< 3 elements	$[]$
$[10, 7, 3]$	Array with no solution	$[]$
$[-1, 0, 1, 2, 3]$	Array with 1 solution	$[[-1, 0, 1]]$
$[2, -1, 0, 1, 1, 2]$	Array with +1 solutions	$[[-2, 0, 2], [-1, 0, 1], [-2, 1, 1]]$

### B-BRUTE FORCE

- Form all possible 3 elements combinations
- $\rightarrow$  Time complexity of  $O(n^3)$
- Check which 3 elements add up to 0
- Return an array of the arrays with the possible combinations

### O-OPTIMIZE

- Sort the array  $\rightarrow O(n \log n)$
- Use different pointers to go through the array (loop once)
- Check for Duplicates

### W-WALKTHROUGH

Example

$S = \{1, 2, -1, -4\}$

Solution set

$\{1, 0, 1\}$

$\{-1, -1, 2\}$

①  $[-1, 0, 2, -1, -4]$

②  $[-1, -1, 0, 2, 4]$

③  $[-1, -1, 0, 1, 2, 4]$

$\uparrow \quad \uparrow \quad \uparrow \quad \uparrow \quad \uparrow$   
 $i \quad j \quad k \quad l \quad m$

$[-1, -1, 0, 1, 2, 4] \rightarrow -1 + 1 + 2 = 2$   
 $\uparrow \quad \uparrow \quad \uparrow \quad \uparrow \quad \uparrow$   
 $i \quad j \quad k \quad l \quad m$

$[-1, -1, 0, 1, 2, 4] \rightarrow -1 + 0 + 1 = 0$   
 $\uparrow \quad \uparrow \quad \uparrow \quad \uparrow \quad \uparrow$   
 $i \quad j \quad k \quad l \quad m$

$[-1, -1, 0, 1, 2, 4] \rightarrow -1 + 0 + 1 = 0$   
 $\uparrow \quad \uparrow \quad \uparrow \quad \uparrow \quad \uparrow$   
 $i \quad j \quad k \quad l \quad m$

## J-IMPLEMENT

```
#include <vector>
#include <algorithm>
using namespace std;

vector<vector<int>> threeSum(vector<int> &list) {
    vector<vector<int>> result;
    if (list.size() < 3 || list.empty()) { return result; }
    sort(list.begin(), list.end());
    for (int i = 0; i < list.size() - 2; i++) {
        if (i == 0 || list[i] > list[i-1]) {
            int j = i + 1;
            int k = list.size() - 1;
            while (j < k) {
                if (list[i] + list[j] + list[k] == 0) {
                    result.push_back(vector<int>({list[i], list[j], list[k]}));
                    j++;
                    k--;
                    // check for duplicates
                    while (j < k && list[j] == list[j-1]) {
                        j++;
                    }
                    while (j < k && list[k] == list[k+1]) {
                        k--;
                    }
                }
                else if (list[i] + list[j] + list[k] < 0) {
                    j++;
                }
                else {
                    k--;
                }
            }
        }
    }
    return result;
}
```

## T-TEST

```
[ ] = [ ] ✓
[1] = [ ] ✓
[10, 7, 3] = [ ] ✓
```

## 2. CODE IMPLEMENTATION

```
#include<algorithm>
#include<vector>
using namespace std;
```

```

class Solution {
public:
    vector<vector<int> > threeSum(vector<int> &list) {
        vector<vector<int> > result;
        if(list.size() < 3) return result;

        sort(list.begin(), list.end());

        for (int i = 0; i < list.size() - 2; i++) {
            if(i == 0 || list[i] > list[i-1]) {

                int j = i + 1;
                int k = list.size() - 1;
                while (j < k) {
                    if (list[j] + list[k] == 0){
                        result.push_back(vector<int> ({ list[i], list[j], list[k] }));
                        j++;
                        k--;

                        //handle duplicate here
                        while(j < k && list[j] == list[j-1])
                            j++;
                        while(j < k && list[k] == list[k+1])
                            k--;
                    }
                    else if(list[i]+list[j]+list[k] < 0){
                        j++;
                    }
                    else{
                        k--;
                    }
                }
            }
        }

        return result;
    }
};

```