

NO. 15

3 SUM

PROBLEM DESCRIPTION

- Given an array S of n integers, are there elements a, b, c in S such that $a + b + c = 0$? Find all unique triplets in the array which gives the sum of zero.
- Note: The solution set must not contain duplicate triplets.
- For example, given array $S = [-1, 0, 1, 2, -1, -4]$,
- A solution set is: $[[-1, 0, 1], [-1, -1, 2]]$

IDEAS

- Similar to 2 sum except
 - There are repeated number in the input array
 - Ask to return value instead of indices
 - SORTING!
 - Head-tail pointers!
- Slip redundant elements while traversing head tail pointers

```
for i in array[0: len(array)] {  
    2_sum();  
}
```

SOLUTION

- https://github.com/Brady31027/leetcode/tree/master/15_3Sum

NO. 16

3 SUM CLOSEST

PROBLEM DESCRIPTION

- Given an array S of n integers, find three integers in S such that the sum is closest to a given number, target. Return the sum of the three integers. You may assume that each input would have exactly one solution.
- For example, given array $S = \{-1\ 2\ 1\ -4\}$, and target = 1.
- The sum that is closest to the target is 2. $(-1 + 2 + 1 = 2)$.

IDEAS

- Similar to 3 sum except
 - Need to maintain a local minimal difference

```
min_diff = +INF
for i in array[0: len(array)] {
    diff = 2_sum();
    min_diff = min(min_diff, diff)
}
```

SOLUTION

- https://github.com/Brady31027/leetcode/tree/master/16_3Sum_Closest

NO. 18

4 SUM

PROBLEM DESCRIPTION

- Given an array S of n integers, are there elements a , b , c , and d in S such that $a + b + c + d = \text{target}$? Find all unique quadruplets in the array which gives the sum of target.
- Note: The solution set must not contain duplicate quadruplets.
- For example, given array $S = [1, 0, -1, 0, -2, 2]$, and target = 0.
- A solution set is: $[[-1, 0, 0, 1], [-2, -1, 1, 2], [-2, 0, 0, 2]]$

IDEAS

- Similar to 3 sum except
 - Two level for loop
- K-Sum problems can be formalized using recursive functions
 - <https://github.com/Brady31027/leetcode/blob/master/18.py>
- "To Iterate is Human, to Recurse, Divine"

```
for i in array[0: len(array)] {  
    for j in array[ i + 1: len(array) ] {  
        2_sum();  
    }  
}
```

SOLUTION

- https://github.com/Brady31027/leetcode/tree/master/18_4Sum

NO. 11

CONTAINER WITH MOST WATER

PROBLEM DESCRIPTION

- Given n non-negative integers a_1, a_2, \dots, a_n , where each represents a point at coordinate (i, a_i) . n vertical lines are drawn such that the two endpoints of line i is at (i, a_i) and $(i, 0)$. Find two lines, which together with x-axis forms a container, such that the container contains the most water.
- Note: You may not slant the container and n is at least 2.

IDEAS

- Amount of water is determined by
 - Distance of the boundaries
 - Heights of the boundaries
 - $\min(\text{boundary1.y}, \text{boundary2.y}) * (\text{boundary2.x} - \text{boundary1.x})$
 - Use head-tail pointers to traverse and find the maximum value

SOLUTION

- https://github.com/Brady31027/leetcode/tree/master/11_Container_With_Most_Water