NO. 1

TWO SUM

PROBLEM DESCRIPTION

- Given an array of integers, return indices of the two numbers such that they add up to a specific target. You may assume that each input would have exactly one solution. Example:

- Given nums = [2, 7, 11, 15], target = 9. Because nums[0] + nums[1] = 2 + 7 = 9, return [0, 1].

IDEAS

- Sorted input | return values(e.g. 167. Two Sum II)
  - Head-tail pointers
- Unsorted input & return indices (e.g. 1 Two Sum)
  - Hash

```
input: numbers, target
output: a pair of indices
let hash_table = {} // key: number, value: index
for i in numbers {
    if (target - numbers[i] in hash_table ){
        return [i, match_item.value ]
    } else {
        add (numbers[i], i) to hash_table
    }
}
```

SOLUTIONS

- https://github.com/Brady31027/leetcode/tree/master/1_Two_Sum

NO. 167

TWO SUM II

PROBLEM DESCRIPTION

- Given an array of integers that is already sorted in ascending order, find two numbers such that they add up to a specific target number.
- The function twoSum should return indices of the two numbers such that they add up to the target, where index1 must be less than index2. Please note that your returned answers (both index1 and index2) are not zero-based.
- You may assume that each input would have exactly one solution and you may not use the same element twice.
- Input: numbers={2, 7, 11, 15}, target=9 Output: index1=1, index2=2

IDEAS

- Head-tail pointers

```
input: sorted numbers, target
output: a pair of indices

let head = 0; let tail = len(sorted numbers) - 1
while ( head < tail ) {
    if (numbers[head] + numbers[tail] == target ) {
        return [ head, tail]
    } else if ( numbers[head] + numbers[tail] > target) {
        tail - -
    } else {
        head ++
    }
}
```

SOLUTIONS

- https://github.com/Brady31027/leetcode/tree/master/167_Two_Sum_II