

1. Remove Duplicates from Sorted Array

Given a sorted array, remove the duplicates in place such that each element appear only once and return the new length.

Do not allocate extra space for another array, you must do this in place with constant memory.

For example,

Given input array $A = [1, 1, 2]$,

Your function should return $\text{length} = 2$, and A is now $[1, 2]$.

2. Remove Duplicates from Sorted Array II

Follow up for "Remove Duplicates" :

What if duplicates are allowed at most twice?

For example,

Given sorted array $A = [1, 1, 1, 2, 2, 3]$,

Your function should return $\text{length} = 5$, and A is now $[1, 1, 2, 2, 3]$.

3. Search in Rotated Sorted Array

Suppose a sorted array is rotated at some pivot unknown to you beforehand.

(i.e., $0\ 1\ 2\ 4\ 5\ 6\ 7$ might become $4\ 5\ 6\ 7\ 0\ 1\ 2$).

You are given a target value to search. If found in the array return its index, otherwise return -1.

You may assume no duplicate exists in the array.

4. Search in Rotated Sorted Array II

Follow up for "Search in Rotated Sorted Array" :

What if duplicates are allowed?

Would this affect the run-time complexity? How and why?

Write a function to determine if a given target is in the array.

5. Median of Two Sorted Arrays

There are two sorted arrays A and B of size m and n respectively. Find the median of the two sorted arrays. The overall run time complexity should be $O(\log(m+n))$.

6、Longest Consecutive Sequence

Given an unsorted array of integers, find the length of the longest consecutive elements sequence.

For example,

Given [100, 4, 200, 1, 3, 2],

The longest consecutive elements sequence is [1, 2, 3, 4]. Return its length: 4.

Your algorithm should run in $O(n)$ complexity.

7、Two Sum

Given an array of integers, 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.

Input: numbers={2, 7, 11, 15}, target=9

Output: index1=1, index2=2

8、3Sum

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:

- Elements in a triplet (a,b,c) must be in non-descending order. (ie, $a \leq b \leq c$)
- 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)

9、3Sum Closest

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. ($-1 + 2 + 1 = 2$).

10、 Remove Element

Given an array and a value, remove all instances of that value in place and return the new length.

The order of elements can be changed. It doesn't matter what you leave beyond the new length.

11、 Next Permutation

Implement next permutation, which rearranges numbers into the lexicographically next greater permutation of numbers.

If such arrangement is not possible, it must rearrange it as the lowest possible order (ie, sorted in ascending order).

The replacement must be in-place, do not allocate extra memory.

Here are some examples. Inputs are in the left-hand column and its corresponding outputs are in the right-hand column.

1, 2, 3 \rightarrow 1, 3, 2

3, 2, 1 \rightarrow 1, 2, 3

1, 1, 5 \rightarrow 1, 5, 1

12、 Permutation Sequence

The set $[1, 2, 3, \dots, n]$ contains a total of $n!$ unique permutations.

By listing and labeling all of the permutations in order,

We get the following sequence (ie, for $n = 3$):

- "123"
- "132"
- "213"
- "231"
- "312"
- "321"

Given n and k , return the k^{th} permutation sequence.

Note: Given n will be between 1 and 9 inclusive.

13、Valid Sudoku

Determine if a Sudoku is valid, according to: [Sudoku Puzzles – The Rules](#).

The Sudoku board could be partially filled, where empty cells are filled with the character '.'.

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

A partially filled sudoku which is valid.

Note:

A valid Sudoku board (partially filled) is not necessarily solvable. Only the filled cells need to be validated.

14、Trapping Rain Water

Given n non-negative integers representing an elevation map where the width of each bar is 1, compute how much water it is able to trap after raining.

For example,

Given $[0, 1, 0, 2, 1, 0, 1, 3, 2, 1, 2, 1]$, return 6.



The above elevation map is represented by array `[0,1,0,2,1,0,1,3,2,1,2,1]`. In this case, 6 units of rain water (blue section) are being trapped. **Thanks Marcos** for contributing this image!

15、 Rotate Image

You are given an $n \times n$ 2D matrix representing an image.

Rotate the image by 90 degrees (clockwise).

Follow up:

Could you do this in-place?

16、 Plus One

Given a non-negative number represented as an array of digits, plus one to the number.

The digits are stored such that the most significant digit is at the head of the list.

17、 Climbing Stairs

You are climbing a stair case. It takes n steps to reach to the top.

Each time you can either climb 1 or 2 steps. In how many distinct ways can you climb to the top?

18、 Gray Code

The gray code is a binary numeral system where two successive values differ in only one bit.

Given a non-negative integer n representing the total number of bits in the code, print the sequence of gray code. A gray code sequence must begin with 0.

For example, given $n = 2$, return `[0, 1, 3, 2]`. Its gray code sequence is:

00 - 0

01 - 1

11 - 3

10 - 2

Note:

For a given n , a gray code sequence is not uniquely defined.

For example, $[0, 2, 3, 1]$ is also a valid gray code sequence according to the above definition.

For now, the judge is able to judge based on one instance of gray code sequence. Sorry about that.

19、Set Matrix Zeroes

Given a $m \times n$ matrix, if an element is 0, set its entire row and column to 0. Do it in place.

Follow up:

Did you use extra space?

A straight forward solution using $O(mn)$ space is probably a bad idea.

A simple improvement uses $O(m + n)$ space, but still not the best solution.

Could you devise a constant space solution?

20、Gas Station

There are N gas stations along a circular route, where the amount of gas at station i is $gas[i]$.

You have a car with an unlimited gas tank and it costs $cost[i]$ of gas to travel from station i to its next station $(i+1)$. You begin the journey with an empty tank at one of the gas stations.

Return the starting gas station's index if you can travel around the circuit once, otherwise return -1.

Note:

The solution is guaranteed to be unique.

21、Candy

There are N children standing in a line. Each child is assigned a rating value.

You are giving candies to these children subjected to the following requirements:

- Each child must have at least one candy.
- Children with a higher rating get more candies than their neighbors.

What is the minimum candies you must give?

22、 Single Number

Given an array of integers, every element appears twice except for one. Find that single one.

Note:

Your algorithm should have a linear runtime complexity. Could you implement it without using extra memory?

23、 Single Number II

Given an array of integers, every element appears three times except for one. Find that single one.

Note:

Your algorithm should have a linear runtime complexity. Could you implement it without using extra memory?