1. Two Sum

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, and you may not use the same element twice.

Example: Given nums = [2, 7, 11, 15], target = 9,

Because nums[0] + nums[1] = 2 + 7 = 9,

return [0, 1].

1. Median of Two Sorted Arrays

There are two sorted arrays nums1 and nums2 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)).

You may assume nums1 and nums2 cannot be both empty.

Example 1:

nums1 = [1, 3] nums2 = [2]

The median is 2.0

Example 2:

nums1 = [1, 2] nums2 = [3, 4]

The median is (2 + 3)/2 = 2.5

1. Find First and Last Position of Element in Sorted Array

Given an array of integers nums sorted in ascending order, find the starting and ending position of a given target value.

Your algorithm's runtime complexity must be in the order of O(log n).

If the target is not found in the array, return [-1, -1].

Example 1:

Input: nums = [5,7,7,8,8,10], target = 8

Output: [3,4]

Example 2:

Input: nums = [5,7,7,8,8,10], target = 6

Output: [-1,-1]

1. First Missing Positive

Given an unsorted integer array, find the smallest missing positive integer.

Example 1:

Input: [1,2,0]

Output: 3

Example 2:

Input: [3,4,-1,1]

Output: 2

Example 3:

Input: [7,8,9,11,12]

Output: 1

Note: Your algorithm should run in O(n) time and uses constant extra space.

1. Plus One

Given a non-empty array of digits representing a non-negative integer, plus one to the integer. The digits are stored such that the most significant digit is at the head of the list, and each element in the array contain a single digit.

You may assume the integer does not contain any leading zero, except the number 0 itself. Example 1:

Input: [1,2,3]

Output: [1,2,4]

Explanation: The array represents the integer 123.

Example 2:

Input: [4,3,2,1]

Output: [4,3,2,2]

Explanation: The array represents the integer 4321.

1. Remove Duplicates from Sorted Array

Given a sorted array nums, 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 by modifying the input array in-place with O(1) extra memory.

Example 1: Given nums = [1,1,2],

Your function should return length = 2, with the first two elements of nums being 1 and 2 respectively.

It doesn't matter what you leave beyond the returned length.

Example 2:

Given nums = [0,0,1,1,1,2,2,3,3,4],

Your function should return length = 5, with the first five elements of nums being modified to 0, 1, 2, 3, and 4 respectively.

It doesn't matter what values are set beyond the returned length.

Clarification:

Confused why the returned value is an integer but your answer is an array?

Note that the input array is passed in by reference, which means modification to the input array will be known to the caller as well.

Internally you can think of this:

// nums is passed in by reference. (i.e., without making a copy)

int len = removeDuplicates(nums);

// any modification to nums in your function would be known by the caller.

// using the length returned by your function, it prints the first len elements.

for (int i = 0; i < len; i++) {

print(nums[i]);

}

1. Remove Nth Node From End of List

Given a linked list, remove the n-th node from the end of list and return its head.

Example :

Given linked list: 1->2->3->4->5 and n = 2.

After removing the second node from the end, the linked list becomes 1->2->3->5.

Note : Given n will always be valid.

Follow up:

Could you do this in one pass?

1. Merge k Sorted Lists

Merge k sorted linked lists and return it as one sorted list. Analyze and describe its complexity. Example:

Input:

[ 1->4->5,

1->3->4,

2->6

]

Output: 1->1->2->3->4->4->5->6

1. Swap Nodes in Pairs

Given a linked list, swap every two adjacent nodes and return its head.  
you may not modify the values in the list’s nodes, only nodes itself may be changed.

Example:

Given 1->2->3->4 you should return the list as 2->1->4->3.

1. Longest Substring Without Repeating Characters

Given a string, find the length of the longest substring without repeating characters.

Example 1:

Input: "abcabcbb"

Output: 3

Explanation: The answer is "abc", with the length of 3.

Example 2:

Input: "bbbbb"

Output: 1

Explanation: The answer is "b", with the length of 1.

Example 3:

Input: "pwwkew"

Output: 3

Explanation: The answer is "wke", with the length of 3.

Note that the answer must be a substring, "pwke" is a subsequence and not a substring.

1. Longest Palindromic Substring

Given a string s, find the longest palindromic substring in s. You may assume that the maximum length of s is 1000.

Example 1: Input: "babad"

Output: "bab"

Note: "aba" is also a valid answer.

Example 2:

Input: "cbbd"

Output: "bb"

1. String to Integer (atoi)

Implement atoi which converts a string to an integer.

The function first discards as many whitespace characters as necessary until the first non-whitespace character is found. Then, starting from this character, takes an optional initial plus or minus sign followed by as many numerical digits as possible, and interprets them as a numerical value.

The string can contain additional characters after those that form the integral number, which are ignored and have no effect on the behavior of this function.

If the first sequence of non-whitespace characters in str is not a valid integral number, or if no such sequence exists because either str is empty or it contains only whitespace characters, no conversion is performed.

If no valid conversion could be performed, a zero value is returned.

Note:

Only the space character ' ' is considered as whitespace character.

Assume we are dealing with an environment which could only store integers within the 32-bit signed integer range: [−231, 231 − 1]. If the numerical value is out of the range of representable values, INT\_MAX (231 − 1) or INT\_MIN (−231) is returned.

Example 1:

Input: "42"

Output: 42

Example 2:

Input: " -42"

Output: -42

Explanation: The first non-whitespace character is '-', which is the minus sign. Then take as many numerical digits as possible, which gets 42.

Example 3:

Input: "4193 with words"

Output: 4193

Explanation: Conversion stops at digit '3' as the next character is not a numerical digit.

Example 4:

Input: "words and 987"

Output: 0

Explanation: The first non-whitespace character is 'w', which is not a numerical digit or a +/- sign. Therefore no valid conversion could be performed.

Example 5:

Input: "-91283472332"

Output: -2147483648

Explanation: The number "-91283472332" is out of the range of a 32-bit signed integer.

Therefore INT\_MIN (−231) is returned.