|  |  |
| --- | --- |
|  |  |
|  | E278. First Bad Version |
|  | M34. Search for a Range |
|  |  |
|  |  |

#### E278. First Bad Version

You are a product manager and currently leading a team to develop a new product. Unfortunately, the latest version of your product fails the quality check. Since each version is developed based on the previous version, all the versions after a bad version are also bad.

Suppose you have n versions [1, 2, ..., n] and you want to find out the first bad one, which causes all the following ones to be bad.

You are given an API bool isBadVersion(version) which will return whether version is bad. Implement a function to find the first bad version. You should minimize the number of calls to the API.

**Tags：**Binary Search

**Similar Problems：**(M) Search for a Range (M) Search Insert Position (E) Guess Number Higher or Lower

|  |
| --- |
| 注意：如果你把ver = (low + high)/2; 那就会出现超时错误，因为直接相加可能溢出，仅此一点，剩下的很简单  //取中位数的正确方式：int iMid = iLow + (iHigh - iLow)/2;  //错误方式：若使用iMid = (iHigh + iLow)/2;则可能导致溢出 |

#### M34. Search for a Range

Given a sorted array of integers, 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].

For example,

Given [5, 7, 7, 8, 8, 10] and target value 8,

return [3, 4].

Tags：Binary Search Array

Similar Problems：(E) First Bad Version

#### M35. Search Insert Position

Given a sorted array and a target value, return the index if the target is found. If not, return the index where it would be if it were inserted in order.

You may assume no duplicates in the array.

Here are few examples.

[1,3,5,6], 5 → 2

[1,3,5,6], 2 → 1

[1,3,5,6], 7 → 4

[1,3,5,6], 0 → 0

Tags：Array Binary Search

Similar Problems：(E) First Bad Version

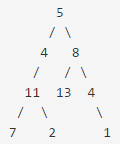
### 二叉树题目

#### E112. Path Sum

Given a binary tree and a sum, determine if the tree has a root-to-leaf path such that adding up all the values along the path equals the given sum.

**For example:**

Given the below binary tree and sum = 22,



return true, as there exist a root-to-leaf path 5->4->11->2 which sum is 22.

**Tags：**Tree, Depth-first Search

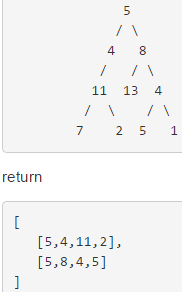
**Similar Problems：** Path Sum II, Binary Tree Maximum Path Sum, Sum Root to Leaf Numbers, Path Sum III

#### E113. Path Sum II

Given a binary tree and a sum, find all root-to-leaf paths where each path's sum equals the given sum.

**For example:**

Given the below binary tree and sum = 22,



**Tags：**Tree, Depth-first Search

**Similar Problems：** Path Sum, Binary Tree Paths, Path Sum III

**Example：**

#### E437. Path Sum III

You are given a binary tree in which each node contains an integer value.

Find the number of paths that sum to a given value.

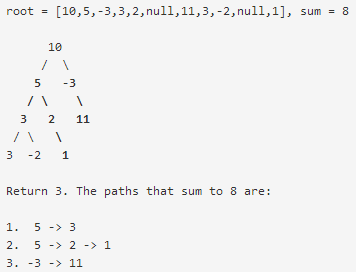
The path does not need to start or end at the root or a leaf, but it must go downwards (traveling only from parent nodes to child nodes).

The tree has no more than 1,000 nodes and the values are in the range -1,000,000 to 1,000,000.

**Tags：**Tree

**Similar Problems：** (E) Path Sum (M) Path Sum II

**Example：**



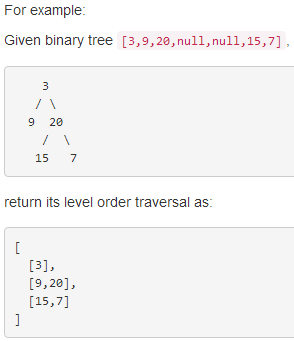
#### 逐层遍历

##### 102. Binary Tree Level Order Traversal

Given a binary tree, return the level order traversal of its nodes' values. (ie, from left to right, level by level).

Tags：Tree Breadth-first Search

Similar Problems：(M) Binary Tree Zigzag Level Order Traversal (E) Binary Tree Level Order Traversal II (E) Minimum Depth of Binary Tree (M) Binary Tree Vertical Order Traversal

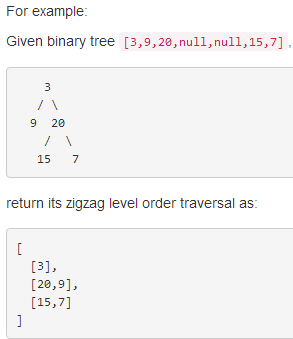


##### 103. Binary Tree Zigzag Level Order Traversal

Given a binary tree, return the zigzag level order traversal of its nodes' values. (ie, from left to right, then right to left for the next level and alternate between).

Tags：Tree Breadth-first Search Stack

Similar Problems： (E) Binary Tree Level Order Traversal



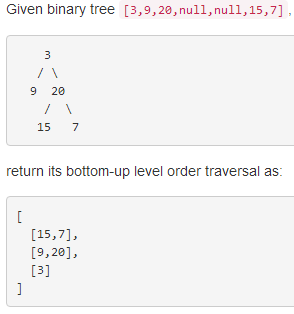
##### 107. Binary Tree Level Order Traversal II

Given a binary tree, return the bottom-up level order traversal of its nodes' values. (ie, from left to right, level by level from leaf to root).

Tags： Tree Breadth-first Search

Similar Problems： (E) Binary Tree Level Order Traversal

For example：



##### 104. Maximum Depth of Binary Tree

Given a binary tree, find its maximum depth.

The maximum depth is the number of nodes along the longest path from the root node down to the farthest leaf node.

Tags：Tree Depth-first Search

Similar Problems：(E) Balanced Binary Tree (E) Minimum Depth of Binary Tree

##### 111. Minimum Depth of Binary Tree

Given a binary tree, find its minimum depth.

The minimum depth is the number of nodes along the shortest path from the root node down to the nearest leaf node.

Tags： Tree Depth-first Search Breadth-first Search

Similar Problems： (E) Binary Tree Level Order Traversal (E) Maximum Depth of Binary Tree

##### 110. Balanced Binary Tree

Given a binary tree, determine if it is height-balanced.

For this problem, a height-balanced binary tree is defined as a binary tree in which the depth of the two subtrees of every node never differ by more than 1.

Tags： Tree Depth-first Search

Similar Problems： (E) Maximum Depth of Binary Tree

### 位操作

#### E461. Hamming Distance

|  |
| --- |
| **E461. Hamming Distance** |
| The Hamming distance between two integers is the number of positions at which the corresponding bits are different.  Given two integers x and y, calculate the Hamming distance.  Note:  0 ≤ x, y < 231.  Example:  Input: x = 1, y = 4  Output: 2  Explanation:  1 (0 0 0 1)  4 (0 1 0 0)  ↑ ↑  The above arrows point to positions where the corresponding bits are different.  **Tags：**Bit Manipulation  **Similar Problems：** (E) Number of 1 Bits (M) Total Hamming Distance |

#### M477. Total Hamming Distance

|  |
| --- |
| **M477. Total Hamming Distance** |
| The Hamming distance between two integers is the number of positions at which the corresponding bits are different.  Now your job is to find the total Hamming distance between all pairs of the given numbers.  **Example:**  Input: 4, 14, 2  Output: 6  **Explanation:** In binary representation, the 4 is 0100, 14 is 1110, and 2 is 0010 (just showing the four bits relevant in this case). So the answer will be:  HammingDistance(4, 14) + HammingDistance(4, 2) + HammingDistance(14, 2) = 2 + 2 + 2 = 6.  **Note:**  Elements of the given array are in the range of 0 to 10^9  Length of the array will not exceed 10^4.  **Tags：**Bit Manipulation  **Similar Problems：** (E) Hamming Distance |

#### 371. Sum of Two Integers

Calculate the sum of two integers a and b, but you are not allowed to use the operator + and -.

**Example:**

Given a = 1 and b = 2, return 3.

**Tags：**Bit Manipulation

**Similar Problems：** (M) Add Two Numbers

#### 226. Invert Binary Tree

Invert a binary tree.

4

/ \

2 7

/ \ / \

1 3 6 9

to

4

/ \

7 2

/ \ / \

9 6 3 1

Tag: Tree

/\*\*

\* Definition for a binary tree node.

\* struct TreeNode {

\* int val;

\* TreeNode \*left;

\* TreeNode \*right;

\* TreeNode(int x) : val(x), left(NULL), right(NULL) {}

\* };

\*/

class Solution {

public:

TreeNode\* invertTree(TreeNode\* root) {

}

};

#### 100. Same Tree

Given two binary trees, write a function to check if they are equal or not.

Two binary trees are considered equal if they are structurally identical and the nodes have the same value.

Subscribe to see which companies asked this question

Tags: Tree Depth-first Search

/\*\*

\* Definition for a binary tree node.

\* struct TreeNode {

\* int val;

\* TreeNode \*left;

\* TreeNode \*right;

\* TreeNode(int x) : val(x), left(NULL), right(NULL) {}

\* };

\*/

class Solution {

public:

bool isSameTree(TreeNode\* p, TreeNode\* q) {

}

};

#### 237. Delete Node in a Linked List

Write a function to delete a node (except the tail) in a singly linked list, given only access to that node.

Supposed the linked list is 1 -> 2 -> 3 -> 4 and you are given the third node with value 3, the linked list should become 1 -> 2 -> 4 after calling your function.

Subscribe to see which companies asked this question

Tags: Linked List

Similar Problems: (E) Remove Linked List Elements

/\*\*

\* Definition for singly-linked list.

\* struct ListNode {

\* int val;

\* ListNode \*next;

\* ListNode(int x) : val(x), next(NULL) {}

\* };

\*/

class Solution {

public:

void deleteNode(ListNode\* node) {

}

};

#### 349. Intersection of Two Arrays

Given two arrays, write a function to compute their intersection.

Example:

Given nums1 = [1, 2, 2, 1], nums2 = [2, 2], return [2].

Note:

Each element in the result must be unique.

The result can be in any order.

Subscribe to see which companies asked this question

Tags: Binary Search Hash Table Two Pointers Sort

Similar Problems: (E) Intersection of Two Arrays II

class Solution {

public:

vector<int> intersection(vector<int>& nums1, vector<int>& nums2) {

}

};

#### 283. Move Zeroes

Given an array nums, write a function to move all 0's to the end of it while maintaining the relative order of the non-zero elements.

For example, given nums = [0, 1, 0, 3, 12], after calling your function, nums should be [1, 3, 12, 0, 0].

Note:

You must do this in-place without making a copy of the array.

Minimize the total number of operations.

Credits:

Special thanks to @jianchao.li.fighter for adding this problem and creating all test cases.

Subscribe to see which companies asked this question

Tags: Array Two Pointers

Similar Problems: (E) Remove Element

class Solution {

public:

void moveZeroes(vector<int>& nums) {

}

};

#### 104. Maximum Depth of Binary Tree

Given a binary tree, find its maximum depth.

The maximum depth is the number of nodes along the longest path from the root node down to the farthest leaf node.

Tags: Tree Depth-first Search

Similar Problems: (E) Balanced Binary Tree (E) Minimum Depth of Binary Tree

/\*\*

\* Definition for a binary tree node.

\* struct TreeNode {

\* int val;

\* TreeNode \*left;

\* TreeNode \*right;

\* TreeNode(int x) : val(x), left(NULL), right(NULL) {}

\* };

\*/

class Solution {

public:

int maxDepth(TreeNode\* root) {

}

};

#### 292. Nim Game

You are playing the following Nim Game with your friend: There is a heap of stones on the table, each time one of you take turns to remove 1 to 3 stones. The one who removes the last stone will be the winner. You will take the first turn to remove the stones.

Both of you are very clever and have optimal strategies for the game. Write a function to determine whether you can win the game given the number of stones in the heap.

For example, if there are 4 stones in the heap, then you will never win the game: no matter 1, 2, or 3 stones you remove, the last stone will always be removed by your friend.

Hint:

If there are 5 stones in the heap, could you figure out a way to remove the stones such that you will always be the winner?

Tags: Brainteaser

Similar Problems: (M) Flip Game II

class Solution {

public:

bool canWinNim(int n) {

}

};

#### 258. Add Digits

Given a non-negative integer num, repeatedly add all its digits until the result has only one digit.

For example:

Given num = 38, the process is like: 3 + 8 = 11, 1 + 1 = 2. Since 2 has only one digit, return it.

Follow up:

Could you do it without any loop/recursion in O(1) runtime?

Hint:

A naive implementation of the above process is trivial. Could you come up with other methods?

What are all the possible results?

How do they occur, periodically or randomly?

You may find this Wikipedia article useful.

Tags: Math

Similar Problems: (E) Happy Number

class Solution {

public:

int addDigits(int num) {

}

};

#### 344. Reverse String

Write a function that takes a string as input and returns the string reversed.

Example:

Given s = "hello", return "olleh".

Subscribe to see which companies asked this question

Tag: Two Pointers String

Similar Problems: (E) Reverse Vowels of a String

class Solution {

public:

string reverseString(string s) {

}

};

#### 242. Valid Anagram

Given two strings s and t, write a function to determine if t is an anagram of s.

For example,

s = "anagram", t = "nagaram", return true.

s = "rat", t = "car", return false.

Note:

You may assume the string contains only lowercase alphabets.

Follow up:

What if the inputs contain unicode characters? How would you adapt your solution to such case?

Subscribe to see which companies asked this question

Tag: Hash Table Sort

Similar Problems: (M) Group Anagrams (E) Palindrome Permutation

#### 171. Excel Sheet Column Number

Related to question Excel Sheet Column Title

Given a column title as appear in an Excel sheet, return its corresponding column number.

For example:

A -> 1

B -> 2

C -> 3

...

Z -> 26

AA -> 27

AB -> 28

Credits:

Special thanks to @ts for adding this problem and creating all test cases.

Subscribe to see which companies asked this question

Tag: Math

Similar Problems: (E) Excel Sheet Column Title

#### 169. Majority Element

Given an array of size n, find the majority element. The majority element is the element that appears more than ⌊ n/2 ⌋ times.

You may assume that the array is non-empty and the majority element always exist in the array.

Tags: Array Divide and Conquer Bit Manipulation

Similar Problems: (M) Majority Element II

class Solution {

public:

int majorityElement(vector<int>& nums) {

}

};

#### 350. Intersection of Two Arrays II

Given two arrays, write a function to compute their intersection.

Example:

Given nums1 = [1, 2, 2, 1], nums2 = [2, 2], return [2, 2].

Note:

Each element in the result should appear as many times as it shows in both arrays.

The result can be in any order.

Follow up:

What if the given array is already sorted? How would you optimize your algorithm?

What if nums1's size is small compared to nums2's size? Which algorithm is better?

What if elements of nums2 are stored on disk, and the memory is limited such that you cannot load all elements into the memory at once?

Subscribe to see which companies asked this question

Tag: Binary Search Hash Table Two Pointers Sort

Similar Problems: (E) Intersection of Two Arrays

class Solution {

public:

vector<int> intersect(vector<int>& nums1, vector<int>& nums2) {

}

};

#### 206. Reverse Linked List

Reverse a singly linked list.

Hint:

A linked list can be reversed either iteratively or recursively. Could you implement both?

Tag: Linked List

Similar Problems: (M) Reverse Linked List II (M) Binary Tree Upside Down (E) Palindrome Linked List

#### 217. Contains Duplicate

Given an array of integers, find if the array contains any duplicates. Your function should return true if any value appears at least twice in the array, and it should return false if every element is distinct.

Subscribe to see which companies asked this question

Tag: Array Hash Table

Similar Problems: (E) Contains Duplicate II (M) Contains Duplicate III

#### 235. Lowest Common Ancestor of a Binary Search Tree

Given a binary search tree (BST), find the lowest common ancestor (LCA) of two given nodes in the BST.

According to the definition of LCA on Wikipedia: “The lowest common ancestor is defined between two nodes v and w as the lowest node in T that has both v and w as descendants (where we allow a node to be a descendant of itself).”

\_\_\_\_\_\_\_6\_\_\_\_\_\_

/ \

\_\_\_2\_\_ \_\_\_8\_\_

/ \ / \

0 \_4 7 9

/ \

3 5

For example, the lowest common ancestor (LCA) of nodes 2 and 8 is 6. Another example is LCA of nodes 2 and 4 is 2, since a node can be a descendant of itself according to the LCA definition.

Subscribe to see which companies asked this question

Tag: Tree

Similar Problems: (M) Lowest Common Ancestor of a Binary Tree

/\*\*

\* Definition for a binary tree node.

\* struct TreeNode {

\* int val;

\* TreeNode \*left;

\* TreeNode \*right;

\* TreeNode(int x) : val(x), left(NULL), right(NULL) {}

\* };

\*/

class Solution {

public:

TreeNode\* lowestCommonAncestor(TreeNode\* root, TreeNode\* p, TreeNode\* q) {

}

};