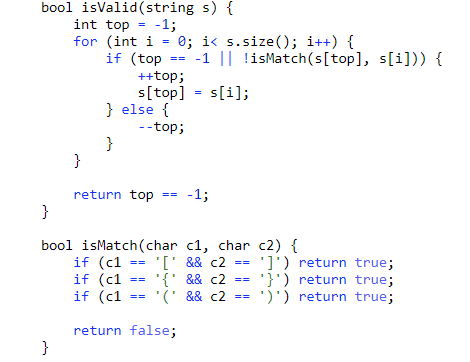
## Valid Parentheses

Given a string s containing just the characters '(', ')', '{', '}', '[' and ']', determine if the input string is valid.

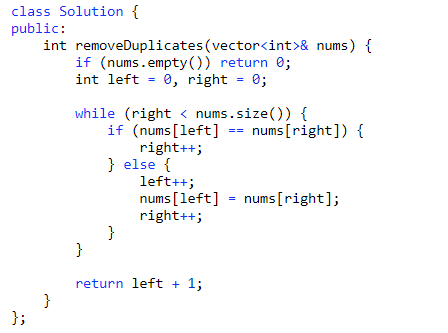
Explain：通过栈模拟，遇到左括号入栈，右括号判断栈顶是否为对应左括号。

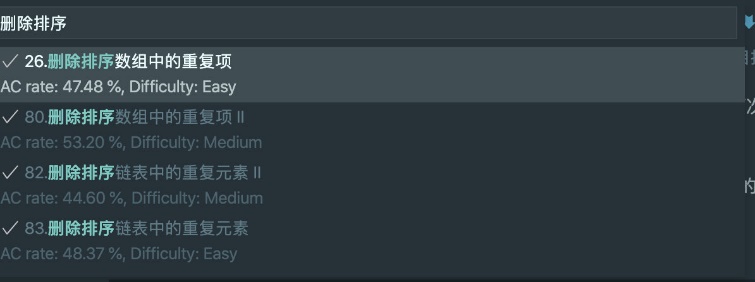
一种不用创建栈的方法：



## Remove duplicates from sorted array

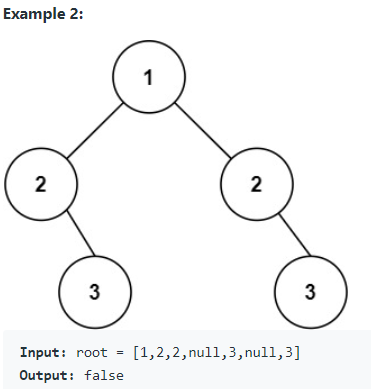
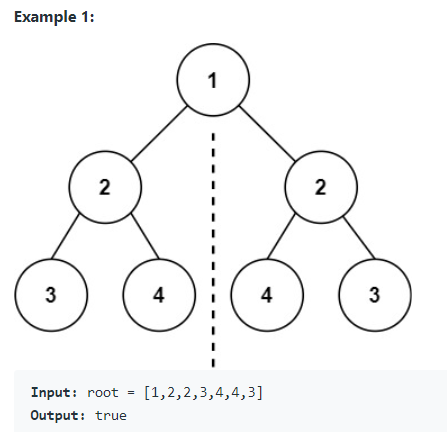
使用双指针的方法，快慢指针。



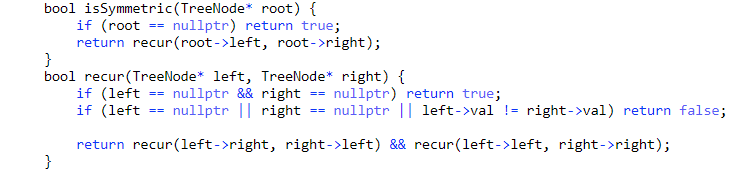


## Symmetric Tree

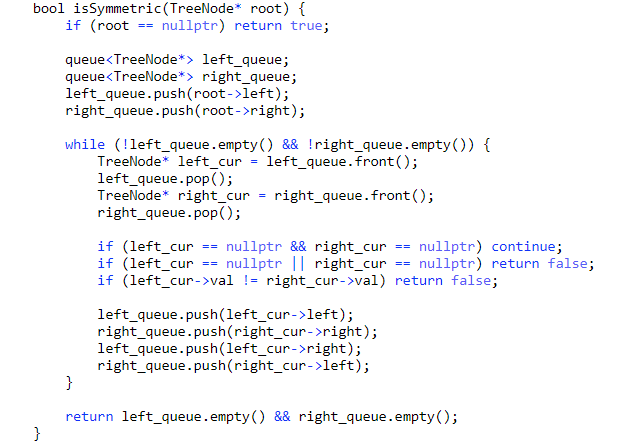
Given the root of a binary tree, check whether it is a mirror of itself (i.e., symmetric around its center).



1. 递归解法：



1. 非递归解法：队列模拟递归

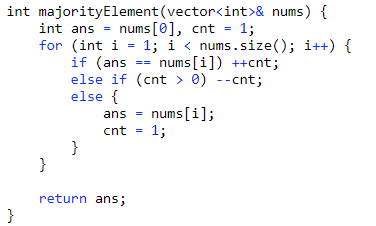


## Majority Element

Given an array nums of size n, return *the majority element*.

The majority element is the element that appears more than ⌊n / 2⌋ times. You may assume that the majority element always exists in the array.

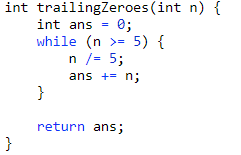
（使用投票法）



## Factorial Tailing Zeros

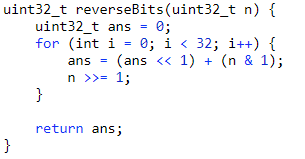
Given an integer n, return *the number of trailing zeroes in*n!.

Follow up: Could you write a solution that works in **logarithmic** time complexity?



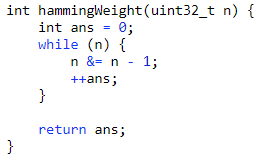
## Reverse Bits (190)

Reverse bits of a given 32 bits unsigned integer.



## Number of 1 Bits (191)

Write a function that takes an unsigned integer and returns the number of '1' bits it has (also known as the [Hamming weight](http://en.wikipedia.org/wiki/Hamming_weight)).

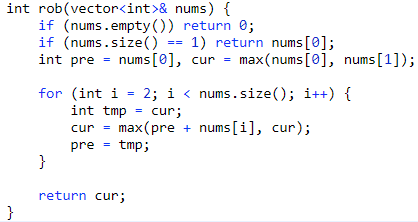


## House Robber (198)

You are a professional robber planning to rob houses along a street. Each house has a certain amount of money stashed, the only constraint stopping you from robbing each of them is that adjacent houses have security systems connected and it will automatically contact the police if two adjacent houses were broken into on the same night.

Given an integer array nums representing the amount of money of each house, return the maximum amount of money you can rob tonight without alerting the police.

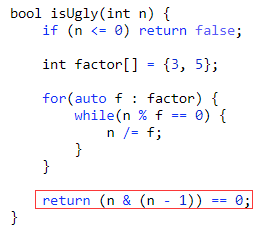
(考虑动态规划的思路，抢或者不抢当前房子的价值取舍)



## Ugly Number (263)

An ugly number is a positive integer whose prime factors are limited to **2, 3, and 5**.

Given an integer n, return true if n is an **ugly number**.



**(n & (n - 1)) == 0 用来判断n是否是一个2指数倍的数**

## Power of Four (342)

Given an integer n, return *true if it is a power of four. Otherwise, return false*.

An integer n is a power of four, if there exists an integer x such that n == 4x.

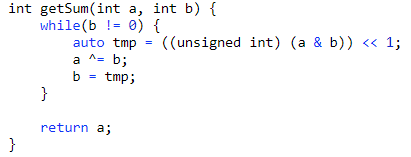


4的幂次方特点：是2的倍数，减去1是3的倍数，（二进制中1在奇数位）

(n - 1) % 3 == 0 ：用来判断减1后是否为3的倍数

## Sum of Two Intergers (371)

Given two integers a and b, return the sum of the two integers without using the operators + and -.



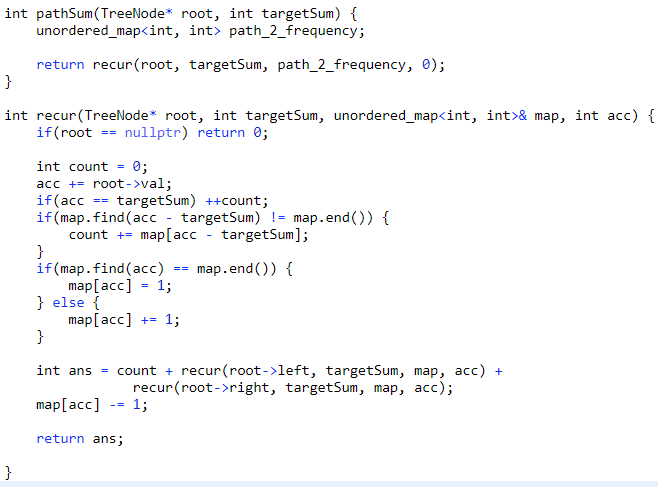
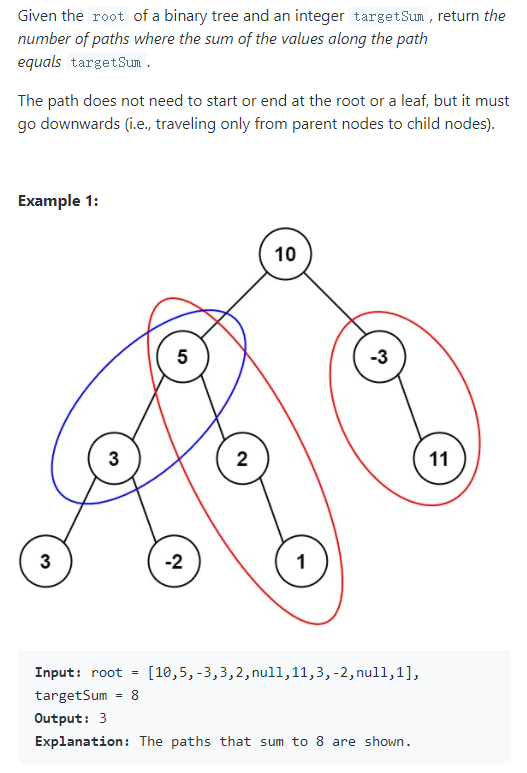
1. (a & b) << 1 ：获取那些位置需要进位，用unsigend int因为避免负数左移时报错。
2. a ^= b : 使用异或操作，模拟一个没有进位操作的相加。

## Binary Watch (401)

<https://leetcode-solution-leetcode-pp.gitbook.io/leetcode-solution/easy/401.binary-watch>

1. bitset<>
2. to\_string()

## Path Sum Ⅲ (437)



（一种空间换时间的思路，创建了hashmap来存储每个节点的累计值和此累计值出现的次数。 1）： count += map[acc - targetSum]：用于减去开头到某一结点的长度，如蓝框减去了10. 2）： 记得当前递归节点结束之前使用map[acc] -= 1，消除当前已经处理完节点的影响。）

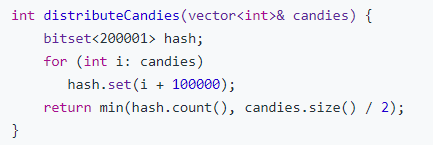
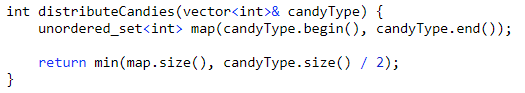
## Distribute Candies

Alice has n candies, where the ith candy is of type candyType[i]. Alice noticed that she started to gain weight, so she visited a doctor.

The doctor advised Alice to only eat n / 2 of the candies she has (n is always even). Alice likes her candies very much, and she wants to eat the maximum number of different types of candies while still following the doctor's advice.

Given the integer array candyType of length n, return the ***maximum*** number of different types of candies she can eat if she only eats n / 2 of them.

（两种思路：（1）使用unordered\_set。（2）使用bitset）

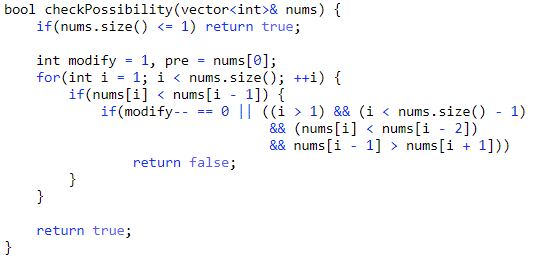


（因为type范围是-100000~100000，所以创建了200001大小的bitset，同时i+100000）

## Non-decreasing Array

Given an array nums with n integers, your task is to check if it could become non-decreasing by modifying **at most one element**.

We define an array is non-decreasing if nums[i] <= nums[i + 1] holds for every i (**0-based**) such that (0 <= i <= n - 2).

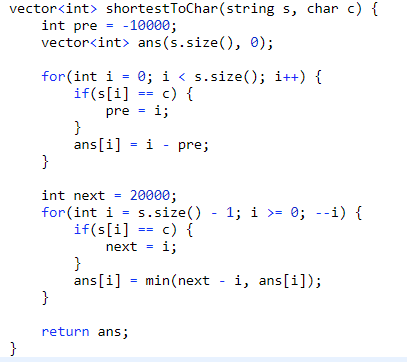


（重点是判断什么情况进行修改后还是不能满足不减）

## Shortest Distance to a Character

Given a string s and a character c that occurs in s, return an array of integers answer where answer.length == s.length and answer[i] is the ***distance*** from index i to the ***closest*** occurrence of character c in s.

The **distance** between two indices i and j is abs(i - j), where abs is the absolute value function.



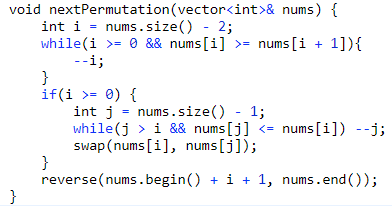
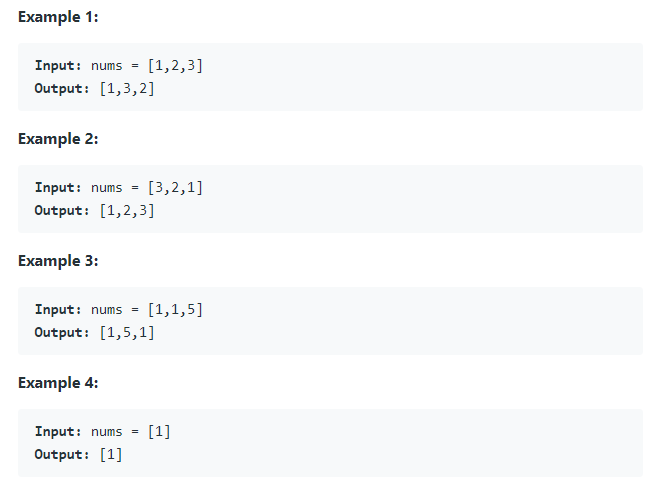
（从前往后和从后往前遍历两次string并计算每个char到前一个目标char的距离，取最小值即可）

## Next Permutation

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

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

The replacement must be [in place](http://en.wikipedia.org/wiki/In-place_algorithm) and use only constant extra memory.



（理解如何才能组合到下下一排序）

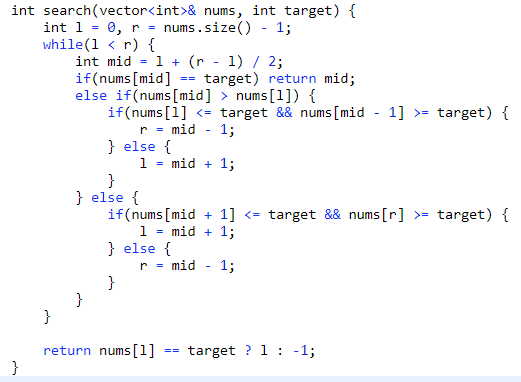
## Search in Rotated Sorted Array

There is an integer array nums sorted in ascending order (with distinct values).

Prior to being passed to your function, nums is rotated at an unknown pivot index k (0 <= k < nums.length) such that the resulting array is [nums[k], nums[k+1], ..., nums[n-1], nums[0], nums[1], ..., nums[k-1]] (0-indexed). For example, [0,1,2,4,5,6,7] might be rotated at pivot index 3 and become [4,5,6,7,0,1,2].

Given the array nums after the rotation and an integer target, return the index of target if it is in nums, or -1 if it is not in nums.

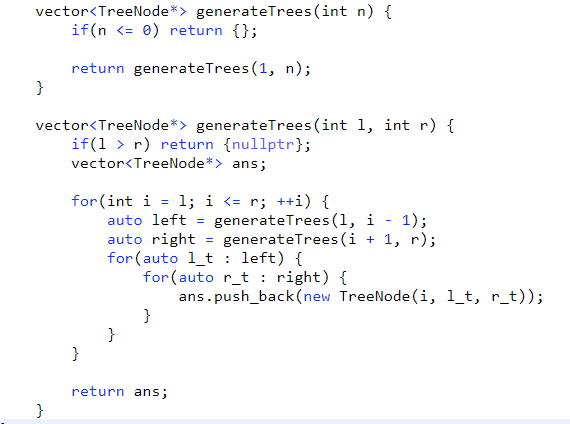
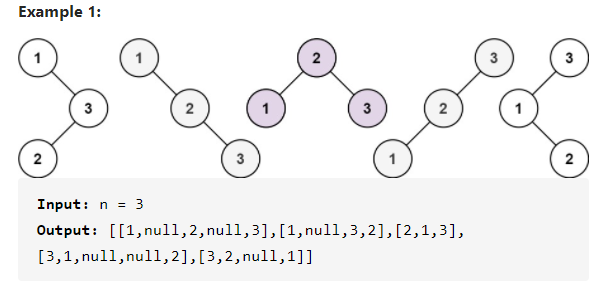
You must write an algorithm with O(log n) runtime complexity.



（注意，二分查找的思路是去除）

## Unique Binary Search Trees Ⅱ

Given an integer n, return all the structurally unique BST's (binary search trees), which has exactly n nodes of unique values from 1 to n. Return the answer in any order.

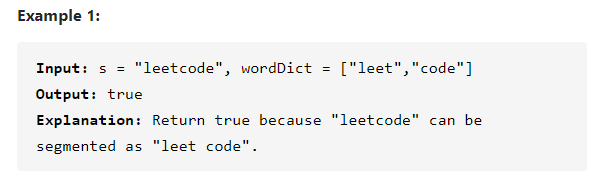


（使用分治法，先分后治）

## Word Break

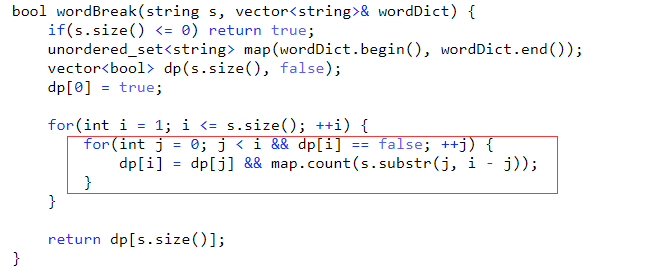
Given a string s and a dictionary of strings wordDict, return true if s can be segmented into a space-separated sequence of one or more dictionary words.

Note that the same word in the dictionary may be reused multiple times in the segmentation.



**最初想法使用递归，类似 131.分割回文串 来解决，结果发现时间超时了。看了其他人的解决方法才发觉这题可以用dp思路进一步优化时间复杂度。**

**具体思路是：创建dp状态转移，dp[i]代表从1到第i个字符组成的string是否可以分割后存在于wordDict中。**



## Insertion Sort List

Given the head of a singly linked list, sort the list using insertion sort, and return the sorted list's head.

The steps of the insertion sort algorithm:

1. Insertion sort iterates, consuming one input element each repetition and growing a sorted output list.
2. At each iteration, insertion sort removes one element from the input data, finds the location it belongs within the sorted list and inserts it there.
3. It repeats until no input elements remain.

The following is a graphical example of the insertion sort algorithm. The partially sorted list (black) initially contains only the first element in the list. One element (red) is removed from the input data and inserted in-place into the sorted list with each iteration.

