2018-9-17 1955

今天 4小时算法时间，拟做15道题目，加油！至少要懂，然后写出来，不能复制粘贴。

39.combination sum

**题目**：给定一**组**候选数字（candidates）**（没有重复）**和目标数字（target），找到candidates 候选数字相加的所有唯一组合target。**相同**重复数目可以选自candidates 无限次数。

**思路**：有人说这是0/1背包问题，但是该题的思路是回溯法

回溯主要从第一个元素往后遍历，加起来等于Sum就return，不然就继续，遍历完还没有return 就 pop出来 进入下一步。没讲清楚，但是我觉得这个办法应该不是一个好办法。

40. Combination Sum II

**题目**：上一题的基础上，有了使用次数的限制，只能用一次，而且数字可能重复。

**思路：**在上一题的基础上，改一下条件就好了，但是存在一个问题，即因为数字可以重复，所以可能出现重复的组。考虑到这一点就好。

if( j && candidates[j]==candidates[j-1]&&j>i) continue;

具体体现在底层，for 往后遍历的意思是前面不取，那么当前面1不取的时候，后面有1也不取，判断FOR中，和前一个相同则跳过（第一个不跳，第一个即使相同也是选了的（迭代就是选，for 就是不选））

41. First Missing Positive

**题目**：寻找最小的缺失正整数

**思路**：利用类似于map的思想，将数字交换到他该在的位置去，不能交换就算了，最后遍历一遍，哪个数字不对，就是第一个最小的正整数。

nums[i]!=nums[nums[i]-1]

和 nums[i]

42. Trapping Rain Water

**题目**：n个非负整数，求能收集的雨水数目

**思路**：这是个好问题，可以用很多方法求解，遍历2次，从两端遍历一次，用栈，等等

48. Rotate Image

题目：nxn的矩阵转90度，不能用空间

思路：交换，一圈一圈的来，我做的时候参数太复杂了，直接利用a,b代表圈的范围[a][a]到[b][b]

53. Maximum Subarray

题目：求和最大的子串

思路：这一题求和主要要理解的是，加起来小于0那么就让他等于0.这个是sum ，sum大于 result 就替换result 不然就继续加，前面的sum 大于0 就用它，小于0就不用他，这样可以得到最大的。

54. Spiral Matrix

题目：将矩阵螺旋输出

思路：这一题和转90度差不多，都尽量一开始用 变量表示出矩阵的边界，上一题是正方形只要上下就好了，这一题要上下左右，4个变量。我因为没有用这个，导致最后边界出错，还是要注意。

55. Jump Game

题目：最多能跳n数值步，问能不能跳到数组最后

思路：我一开始用递归，但是超时了，所以改用普通的方法，reach 表示能到达的最大距离，i 数组不停的往后走，只能走到reach到的地方，然后更新reach 最后判断reach 到了最后吗。

56. Merge Intervals

题目：给定间隔的集合，合并所有重叠的间隔。

思路：有个好方法，引入了排序的初始化

可以使用的结构：sort(ins.begin(), ins.end(), [](Interval a, Interval b){return a.start < b.start;});

59. Spiral Matrix II

题目：给nxn矩阵,将整数从1到nxn螺旋打印进矩阵

思路：将矩阵的边界定义好，然后遍历，注意边界，我这里存在一个问题是没有将最后剩下一个的情况包含进去，最后使用了if 去包含，感觉有点不好，明天看看咋好。

62.Unique Paths

题目：求解从左上角到右下角的路线数目，只能往下和右

思路：一开始用递归，后来用乘法，然后用dp的方法，

62.Unique Paths 二

题目：求解从左上角到右下角的路线数目，只能往下和右，有些地方不能走

思路：用dp的方法，判断是不是能走，不能就是0（然后要用n+1 x m+1的矩阵，不然第一行第一列没法处理）

64.Minimum Path Sum

题目：求解从左上角到右下角的路线花费最小的代价

思路：主要考虑dp，然后注意的是第一行 第一列的数据处理。第一行的cost = 本身的cost+同行前一个的cost ,第一列的类似。

66.Plus One

题目：数组表示一个数，+1

思路：9往前进一位。可以先转换成数，+1再换回来。不然就要注意遇9化0，不然就加1 return 。到最后再赋1 ，最后加0；

73.Set Matrix Zeroes

题目：给定m xn矩阵，如果元素为0，则将其整行和列设置为0。

思路：不用空间的话，先行，再列

我的想法很简单：在该行的第一行中存储每个行的状态，并在该列的第一列中存储每个列的状态。因为ROW0的状态和Cyrn0的状态将占据同一个单元格，所以我将它作为ROW0的状态，并使用另一个变量“COL0”作为Caln0。在第一阶段，使用矩阵元素以自顶向下的方式设置状态。在第二阶段，使用状态以自下而上的方式设置矩阵元素。

74.Search a 2D Matrix

题目：怎么搜索，

思路：将2维当做1维来做，转换为 A[n] = B[n/x][n%x]

75.Sort Colors

题目：给定一个具有n个颜色为红色、白色或蓝色的对象的数组，将它们按位置排序，以便相同颜色的对象相邻，颜色顺序为红色、白色和蓝色。

思路：3个标志位，

2018-9-20 15:09:00

78.Subsets

题目：求一个集合的所有子集

思路：我采用了2维数组用一维数组表示，坐标转化的方式，用一个二进制数字，来表示子集各元素是否存在

79.Word Search

题目：给定一个二位矩阵和一个单词，查找该单词是否存在于网格中。

思路：遍历矩阵，对每一个点为起点，查找是不是这个单词，主要考虑全面，包括边界，遍历过的元素置0以及置回原数。

80.Remove Duplicates from Sorted Array II

题目：移除数组中的2个以上重复数字，最多2个重复的

思路：我写的思路是遍历，一个变量用来存现在重复了几次了，超过次数就删除。

这个用到了删除，但是其实可以不用，直接用两个变量，一个存现在数组遍历的位置，一个村数组存放的位置，在原数组中直接更新。

81.Search in Rotated Sorted Array II

题目：在一个循环左移的有序数组中找一个数

思路：一开始想分情况讨论，但是情况可能有点多。

这一题，采用二分法，先看左右两侧 哪一侧是顺序的，然后看目标是不是在顺序的这一侧范围内，在则搜索这一侧，不在则搜索另一侧（1.考虑可以判断的部分，不在这一部分就在那一部分）

判断条件是 left<=right，

88.Merge Sorted Array

题目：合并两个有序数组

思路：一开始想的用了额外的空间，后来想办法不使用空间一时忘记了，后来看了代码才发现，可以通过从后往前的方式，避免覆盖原来数据。

90.Subsets II

题目：写出集合的所有子集（集合有重复数字）

思路：和以前不一样的是要考虑相同数字的重复的，未完

再写1个半小时吧~

2018-9-20 22:52:01

90.Subsets II

题目：写出集合的所有子集（集合有重复数字）

思路：和以前不一样的是要考虑相同数字的重复的，这个主要考虑一般情况：如果一个数出现K次，那么它子集数目为（K1+1）(K2+1)(K3+1)(K4+1)，K+1 是只这个数在子集中出现0到K次

no, 他的思路和我不一样，他是先考虑1个数，可能有可能没有2种情况，然后，这两种情况加上第而个数就有4中情况了，加上第3个就8种了，这样子的话，先排序，然后重复的a就用push\_back(a)，加入集合，再pushback(a) 加入集合，即用1次 ，用2次，这样子，就可以使用以上子集数目的思路了。

105.Construct Binary Tree from Preorder and Inorder Traversal

题目：给定前序遍历和中序遍历求树

思路：我想的是采用递归的方式？或者说是分治法 利用前序中序 找根，和 左右子树部分

106.Construct Binary Tree from Inorder and Postorder Traversal

题目：中序，后序求解

思路：和上一题一样，但是这里有个收获，就是多加4个变量记录目前的子串是从哪到哪，这样子就不用为重新构造数组进行传递了。

2018-9-21

118.Pascal's Triangle

题目：

**Input:** 5 **Output:** [ [1], [1,1], [1,2,1], [1,3,3,1], [1,4,6,4,1] ]

思路：从上往下遍历，有什么特殊的方法吗？应该没有吧，不然也不是easy了

119.Pascal's Triangle 2

题目：输出第n行，空间限制O（n）

思路：原行变化，遍历过来。

121：Best Time to Buy and Sell Stock

题目：给价格，什么时候买什么时候卖最划算

思路：记录，更新最小值，更新差值（该数减最小）

122.Best Time to Buy and Sell Stock II

题目：可以卖出N次

思路：记录并采用所有的上升阶段？但是怎么又出现了runtime error？

加上这个if(prices.size()==0)

return 0;就没事了

152.Maximum Product Subarray

题目：找出最大的乘积的连续子集

思路：这里要记录两个值，一个是乘积的最大，一个是最小，还有一个是result，记录目前最大，对于每一个元素，都看在，他本身，他本身x最大，他本身x最小中找出新的最大和最小，更新 目前最大。

153.Find Minimum in Rotated Sorted Array

题目：找出右移有序数组中最小的数

思路：我的想法和找target一样，先判断哪部分是有序的，然后再有序中先找，不对的话就在无序的部分找。

2018-9-23 10:06:24

154.Find Minimum in Rotated Sorted Array II

题目：和上题一样，但是加了一项，可以是重复数组

思路：注意，判别条件

162.Find Peak Element

题目：找局部的顶点，相邻元素不相同，末尾是负无穷

思路：也就是说找一个有斜率正有斜率负的区域，前后末尾有负无穷，也就是我们找中间，看斜率，符合哪边走哪边，主要是要注意，最后咋取

167.Two Sum II - Input array is sorted

题目：求两个数的和为TARGET 的数组下标

思路：两个指针，一个在0出，一个在size-1处

169.Majority Element

题目：主要的元素，超过n/2个，必定存在

思路：使用两个变量一个存该元素，一个存该元素减去其他元素得的值

189.Rotate Array

题目：循环右移K个位置

思路：我按照他的思想来，直接K次，每次移一位，应该有更加快的方法

1、Reverse the first n - k elements, the last k elements, and then all the n elements.

2、用前k个元素交换最后的k个元素。

209.Minimum Size Subarray Sum

题目：找到最短的连续子集和超过target的长度

思路：两个变量记录目前遍历的情况，sum 记录目前的和，res 记录最佳子串长度。

一个一个遍历过去，这个时候要考虑情况多，一个是 遍历完，和都小于TARGET，第二个情况是遍历到最后的时候才满足>=target，可以减去前面的部分，但是循环结束了

2018-9-24 13:15:43

216.Combination Sum III

题目：K个数字 和为N 写出所有数字（约束数字1到9）

思路：循环递归，递归中 i到9,一个个插入递归删。

217.Contains Duplicate

题目：找数组中有没有重复的数字

思路：1、sort 在比较前后是否一样

2、map<int, bool> myMap;

228.Summary Ranges

题目：**Input:** [0,1,2,4,5,7] **Output:** ["0->2","4->5","7"]

思路：遍历，循环看后面是否了连续，连续就继续循环，然后看是否循环了，来输出

res.push\_back(to\_string(nums[start]) + "->" + to\_string(nums[end]));

229.Majority Element II

题目：超过n/3的元素

思路：还没有看，得想想

路上好好看discuss?

2018-9-24 22:12:29

229.Majority Element II

题目：超过n/3的元素

思路：方法1：先找最多的两个元素，然后判断这两个是不是比1/3多。用a，b表示最多的两个元素，。

238.Product of Array Except Self

题目：找所有子集乘积的集合（不能）

思路：1、front las 两边来，乘

268.Missing Number

题目：找0到N中 有哪个数缺失了

思路：我想的是先排序，再...，然后那些解答的还没有看懂

283.Move Zeroes

题目：把0移到数组后面

思路：i , j i遍历 j来写非0数，最后 j之后补上0

2018-9-25 00:23:20

287.Find the Duplicate Number

题目：找到n+1个数中重复的数字（数字在1到n之间）

思路：1、有重复的数字的话，例如11个数，1到10，按道理来说大于5.5的5个，小于5.5的也是5个，但是存在重复的就意味着，这里有偏差，我们就是通过这一点去折半查找。2、第二种方法是通过有没有圆来判断，这里首先要找出圆的起点，然后判断圆。

289.Game of Life

题目：细胞生死

思路：本来0表示无，1表示有，现在可以利用第二位来表示变化后的。这里学会了用>>= &1 &2 这些符号。

380. Insert Delete GetRandom O(1)

题目：insert 等等操作 在O(1)中完成

思路：这要考什么呢？map map的操作 find 返回迭代的指针 =map.end()说明没找到 erase（）删除

414.Third Maximum Number

集合：插入3之后虽然插入了一个1，但是我们发现set中最后一个值仍然是3哈，这就是set 。

题目：找第3大的数字

思路：记录第一第二第三大

if (num <= c || num == b || num == a) continue;

c = num;

if (c > b) swap(b, c);

if (b > a) swap(a, b);

比我的分3中情况简洁多了

442.Find All Duplicates in an Array

题目：1到n范围内的数字，有n个 可以出现1次或者2次 找出出现2次的，On

思路：把 x 放到位置 num[x-1]上 2、遇到了 num[x]就把他变成相反数，然后判断你是不是正的，是正的就是结果之一

448.Find All Numbers Disappeared in an Array

题目：找到所有的不存在的数字和上题一样的约束

思路：把 x 放到位置 num[x-1]上 ， 怎么放呢？ 交换法，

485.Max Consecutive Ones

题目：找最多有连续多少个1

思路：我想的是，有一个max来记录，res 记录

max = max\*nums[i]+nums[i];

if(max>res)

res=max;

不用if 用运算来代替会加速！

2018-9-25 22:57:52

70.Climbing Stairs

题目：爬N步，一次可以爬1步或者2步，问有多少种爬法

思路：这是动态规划问题，从一步开始计算，N步的方法是 n-1步+ n-2步之和

91.Decode Ways

题目：给一组String 数字，看有多少种编码方式，编码 1到26

思路：这是上一题的一个加条件，那么我是把条件放到了代码中，导致过于复杂，那么可以把条件提出来

bool isValid(char a,char b){

return a == '1'||(a == '2' && b <='6');

}

bool isValid(char a){

return a != '0';

}

2018-10-5 17:30:38

96.Unique Binary Search Trees

题目：本题主要求N层的二叉排序树的个数

思路：和以前做的不一样的地方是这个不用求树的集合，以前要求树的集合主要的问题是：

139.Word Break

题目：找有是否能够由分词组成

思路：这个没弄懂

198.House Robber

题目：找到小偷可以抢的最多的钱，不能抢邻居

思路：使用dp ,抢 第i个 的话，就是+ 前i-2个 不抢 就等于 前i-1个

213.House Robber II

题目：找小偷，这里房子是一个圈

思路：我的思路是把圈剪开，即分情况，第一个房子抢或者不抢。

264.Ugly Number II

题目：找第N大的丑数

思路：用3个指针来指向2\* 3\* 5\* 的乘数，用过就+1

279.perfect-squares

题目：找一个数由多少个n2组成，n2 包括 1,4,9,16 ,,,

思路：我一开始想先算出小于该数的所有n2 存储起来，然后从后往前减去，可以减就减，不能减就跳过，但是超时了。

方法1：从1开始一直到目标数字每个值先赋值为最大值INT\_MAX ，然后从1开始，他的值等于本值与本值减n2 的值+1 中的最小值，遍历到最后就得到了

方法2：同样的方法 只是没有初始值，不初始定义vector的长度，就是不停的push\_back

2018-10-8 09:18:55

300.longest-increasing-subsequence

题目：在无序的数组中，找出有序排列的元素个数

思路：使用dp O（n2），对于每一个元素，找他以前的元素，如果该元素大于以前的元素，那么该元素的X就等于以前元素的X+1（选择原值与X+1中较大者）。

303.Range Sum Query - Immutable

题目：找第i个元素到第j个元素之和

思路：求出每一个元素到0元素的和，i j = j - i

304.range-sum-query-2d-immutable

题目：求2维数组中(i,j) 到 （n,m）矩形中元素和

思路：其等于 （n,m）+（i,j）-（n,j）-（i,m）

309.best time to buy and sell stock with cooldown

题目：买卖，求最大收益，买卖不能再同一天，卖了要冷却一天才能买

思路：3个状态？

2018-10-9 21:19:30

322.Coin Charge

题目：换硬币，硬币面值存储在coin中

思路：这个不能使用贪心，因为 例如 要凑 11块钱 有 10块，5块，2块，3块，1毛，你先选了10块就要用11个银币，而最佳3个就够了

所以用的是DP，从目标1块开始，后面的等于后面 -coin面值的前面的 +1 第一个是0

338.Counting Bits

题目：计算1到num中二进制表示1的个数

思路：这里要先找规律，二进制表示中，a的数字等于 a/2 后面+ 0 或者1 也就是说，我们找到了dp的规律，初始化a=0 时 a =1的情况 即可

343.Integer Break

题目：10 分成 3 3 2 2 相乘最大 输出最大乘积

思路：所有的数都要分成3 3 3 3 ...(2/22)组成的，可以用贪心，就是能3就3，等于4 就用2 等于 3就用 3 等于 2 就用 2 。 或者dp 前一个等于后面的A[i-2]\*2 A[i-3]\*3中的较大值

357.Count Numbers with Unique Dights

题目：找到数字不重复的数字的个数

思路：一位一位的算，2位的等于 1位 + 2位 2位 = 1位 \* 9

2018-10-9 23:45:24

375.Guess Number Higher or Lower

题目：1到n猜一个数，猜的代价是你猜的那个数，问最少要准备多少代价

思路：这里要采用的是2维的DP ，构造一个二维的数组，i j = min(max(i x , x+1 j))

376. Wiggle Subsequese

题目：前一个数减去后一个数得到差值，求一个数组中差值正负交替的子数组元素个数

思路：这里采用2个数组来存储，一个存储最后两元素差值为负的，一个存储为正的，然后dp的时候就容易了， 如果现在新的数比原数要大 正 i = 负 i +1

377.Combination Sum IV

题目：求由数组中元素组成的target的方式有多少种，不同顺序算不同种类

思路：一个数组来存储target 1 到 n 的方式

392.Is Subsequence

题目：找到子串

思路：没有用dp

413.Arithmetic Slices

题目：找一个数组中，等差子数组的个数

思路：我的方法是用NxN矩阵存储，n长度的子串是否是等差。结果内存溢出了。

2018-10-13 12:26:24

416.partition-equal-subset-sum

题目：求一个数组是否能分为和相等的两个子数组

思路：就是求数组中元素和为sum./2的子数组是否存在，这里可以用01背包问题，是否用这个数，

464，Can i win

题目：game 谁会赢？

思路：这个问题今天晚上一定要搞完（放一下）

这个的话，其实可以不用想，怎么样是最优的，就是存储所有的状态，然后计算所有的状态

STRING :

20.Valid Parentheses

题目：括号问题

思路：栈

22.Geberate Parentheses

题目：括号的所有组合

思路：利用子函数，不断生成新的

43.Multiply String

题目：string 的乘法

思路：模拟乘法运算，过程，逐位相乘

58. Length of last word

题目：求一个句子中最后一个单词的长度

思路：从后往前，直接了当 循环不一定要用for while等等都可以

67：Add Binary

题目：字符串 二进制加法

思路：和乘法一样，每一位由-‘0’操作转换为int 进行操作，符号运算要多用，？x ： xx

71. Simplify Path

题目：路径简化

思路：这里主要利用了stringstream 以及 getline(ss,tmp,"/"); 来获得 / / 之间的sTring 然后分情况处理

93.Restore IP Addresses

题目：给出一串数字，求其所有可能代表的的IP字符串

思路：dfs 典型 （下一个问题的时候着重处理）

125.Valid Palindrome

题目：只考虑数字和字母，不论大小写，问是否为回文

思路：先去其他元素，然后转换大小写，然后看其与倒置是否相同，超时了。

更好的方法是什么呢？直接从两头遍历，遇到非数字字母就跳过，然后对比

165.Compare Version Numbers

题目：比较版本前后

思路：读取数字，然后对比

344. Reverse String

题目：反转

思路：前后对换

**MATH PROBLEm :**

29.Divive two integers;

题目：两个数相除，不能用除法和乘法（二进制移位不算乘除）

思路：一开始想要用到方法是，不断的减，看减多少次就是等于多少，当然先要判断符号，

但是超时了，于是看了discuss 其采用的方法是也是减法，但是减的数不是除数，而是尽可能多的移位，减去除数的2n倍，同时结果+2n 2n通过2进制移位得到

50 Pow(x,n)

题目：求解x的n次方

思路：如果一次一次乘的话，时间复杂度太高了，于是，使用上次类似的方法，缩减计算步骤。

Pow(x,n) = P(x\*x,n/2)

60.permutation-sequence

题目：求第k个组合 组合由1到n组成

思路：找规律，先锁定第k个的位置，然后构造结果

69.sqrt(x)

题目：求平方根

思路：二分法找平方根 判断条件为：x/mid>=mid

明天能否完成27斩杀？ 挑战极限的时间来了！

166

题目：分数表示为小数（string）

思路：这个的话就是逐位相除，然后用一个表来记录出现过的数字以获取循环的开始处。

168

题目：求用字母表示数字，26进制与10进制的转换

思路：存储进制的每一位，然后转换为char输出

171：

题目：用数字表示26进制的字母表示的数字

思路：这个简单一些，不用转换，直接使用 char - 'A' 来表示每位的数。

172：

题目：求n! 末尾0的个数

思路：这个就是找规律了，显然末尾零的个数只与因子5有关，就算因子中5的个数

while(n>=5){ n/=5; res+=n; }

202.

题目：判断该数是否是快乐数字（123 = 1 + 4 + 9 ...最后结果为1）

思路：循环，两个终止条件，一个是1，一个数出现重复数字

204

题目：计算素数的个数

思路：用一个数组来存储已知的素数，当其他数除以素数数组中的数都不能整除时，该数也是素数。

223.

题目：求解两个矩形覆盖面积

思路：主要的要求解两个矩形相交的部分，这个要找规律（两个左侧边中靠右的那一条，上侧边考下的那一条，...四条边组成，然后可以求）

231.

题目：求一个数是否是2的n次幂

思路：不断除以2

258

题目：123 = 6 一直循环直到为个位数

思路：找规律，9为循环的节点，123 = 6 124 = 7 125 =8 126 = 9 127 =1 128 =2

263

题目：判断是否是丑数，丑数是只含有2,3,5因子的数

思路：就是一个个除，看最后是否为1（注意负数不行）

313

题目：超级丑数，即求因子为目标数组的第k个数

思路：这个的话，set 第一个是最小一个？ set 先初始化为数组，然后第一个数为第一个，删去第一个，加上第一个乘所有数，继续删第一个，加...

319.

题目：求开灯关灯 最后的几个灯开，一开始全开，然后关2的倍数，然后转换3的倍数

思路：？？

326

题目：求某数是否是3的幂

思路：不断除以3，但是这种方法不好

1、 用最大的那个3的幂%目标数 是否为0，是就是对的

2、用x = logn % log3 看是不是整数 （int）x == x ?

365.

题目：水和杯子的问题

思路：清空与倒水， 所以是只要目标z = ax + by

368

题目：目标是否是x2

思路：1、 x2 = 1+3+5+7+9+... 2、 x2 求x low =0, high =x2

397.

题目：如果n为偶数则n变为n/2,n为奇数则变为n+1 -1,

思路：递归的话就是比较慢，那么就用dp

tree

94、inary-tree-inorder-traversal

题目：描述一个树，按照中序的步骤

思路：用dfs 深度遍历

98.validate-binary-search-tree

题目：判断树是不是一个BST 二叉搜索树

思路：先用dfs 把所有数存起来，然后看数组是否递增。

100.same tree

题目：判断树是否相同

思路：判断是否为空都为空返回1，一个为空返回0，都不为空则判断值是否相同，相同则判断子树

，不同则直接返回0

101 symmetric tree

题目：判断树是否为镜像的

思路：如上题，但是这里是对比左子树和 另外的右子树

102 Binary Tree Level Order Traversal

题目：分层的输出树

思路：广度遍历树，这里用队列存储树，同时记得要带上层的标记。每一次插入的新元素的救援数层+1。

2018-10-29 14:27:00

103 Binary Tree Zigzag Level Order Traversal

题目：分层的输出树，但是锯齿状输出

思路：广度遍历，每次遍历的时候用一个记录，来记录应该正向还是反向进行

104 Maxium Depth of Binary Tree

题目：树的深度

思路：递归求解，树的深度等于子树深度+1

107 Binary Tree Level Order Traversal II

题目：输出树，分层，从底层到顶层

思路：利用insert(vector.begin(),sub);

108 Convert Sorted Array to Binary Search Tree

题目：将顺序数组转换为平衡搜索树

思路：将n/2作为树的父节点，前半部分作为左子树...

110 Balanced Binary Tree

题目：平衡二叉树的判断

思路：判断他是不是平衡二叉树（左右子树深度相差1以内），然后判断其子树是否是

111 Minimum Depth of Binary Tree

题目：求树根节点到叶子节点的最短路径

思路：这里要清楚叶子节点的意义，叶子节点就是没有子树的节点，然后递归+1，要分要1子树，有2子树的情况进行处理，避免处理空指针的情况。

112 Path Sum

题目：判断路径和是否为x

思路：和上一题类似递归，这个时候只要root = null 就可以返回 false 所以不用分情况了

113 Path Sum II

题目：判断路径和是否为x，如果为x则存下路径

思路：这里和上面主要难点在于数据的传递，如何在递归过程中

path.push\_back(node->val);

判断 + 更深层递归

path.pop\_back();

2018-10-29 20:41:41

（今天还要看三角剖分代码）

114 Flatten Binary Tree to Linked List

题目：将树存为类似链表的结构

思路：把右边的放到左边的底下，然后把左边变成右边

116 Populating Next Right Pointers in Each Node

题目：将树存为类似链表的结构（116 117 颠倒了）

思路：比下一题这里需要多加2个变量，一个记录下一个的头，一个记录现在算的一个

117 Populating Next Right Pointers in Each Node II

题目：加next 进入树结构

思路：利用父节点，左子树的next = 右， 右 = 父->next->左 父 = 父->next

129 Sum Root to Leaf Numbers

题目：求每一条路表示的数的和

思路：一个私有变量sum，每次递归到叶子结点就+

144 Binary Tree Preorder Traversal

题目：求先序数组

思路： &vec 代替 vec 传引用

2018-11-1

199 Binary Tree Right Side View

题目：找从右侧视角看到的节点

思路：深度遍历，如果插入的节点个数和比现在所在的层数少，就插入该节点（先遍历右，再左）

222 Count Complete Tree Nodes

题目：计算完整二叉树的节点个数

思路：遍历的方法可能有点慢，这里采用的是先判断是否是完全二叉树，是的话就得解，不是就为1+两个子树的节点个数

226 Invert Binary Tree

题目：左右颠倒树

思路：循环进行左右颠倒，终止判断其是否是叶子节点，其有左子树就颠倒左，有右则右

230 Kth Smallest Element in a BST

题目：找到二叉搜索树的第k小的数

思路：这里用辅助空间，存储走过的路径，先左左左左，再右 继续循环， 存入 链表，然后读链表判断是不是第k个，不是的话，k--。

235 Lowest Common Ancestor of a Binary Search Tree

题目：求最低层的共同祖先节点

思路：判断大小，如果两个数在节点的两侧（或者一个在节点上） 该节点就是了，不然就分情况找子树

2018-11-2 16:38:29

236 lowest-common-ancestor-of-a-binary-tree

题目：求最低层的共同祖先节点

思路：当root == p q null 返回root，左子树，右子树递归，左右子树是否为null 判断返回值

257 Binary Tree Paths

题目：输出二叉树所有的路径

思路：遍历，如果是叶子节点就输出，不然的话继续向左右子树搜索

337 House Robber III

题目：小偷偷东西，不能连续偷两家，树性

思路：tryRob(TreeNode\* root, int& l, int& r) 下一级在这里求出来，通过

l = tryRob(root->left, ll, lr); 求解下两层和一下层

404 Sum of Left Leaves

题目：左叶子的和

思路：就是要找到左叶子，空集判断 return 0，左叶子判断条件（） return val + right 其他条件 return left + right

429 N-ary Tree Level Order Traversal

题目：存储每一层的

思路：广度遍历，利用队列

437 Path Sum III

题目：求路径或者路径中的一段求和为x 的数目

思路：

449 Serialize and Deserialize BST

题目：

思路：

450 Delete Node in a BST

题目：

思路：

hash map

36 Valid Sudoku

题目：判断目前输入的9x9网络是否可能组成数独

思路：对于每一个元素记录其出现的位置，判断其与其他相同元素的位置关系。

136 Single Number

题目：求唯一 一个单个的数，其他都是2个

思路：使用 XOR ^ 进行处理，把所有的数都 ^ 起来， 然后结果是啥就是啥。

138 Copy List with Random Pointer

题目：

思路：

187 Repeated DNA Sequences

题目：

思路：

2018-11-6 16:43:29

205 isomorphic-strings

题目：判断两个字符串是不是同构

思路：1 1 对应的变换，用两个vector（256） int 记录 每个字符串出现的位置，当 aab ccd匹配时，第一次匹配，不存在则，让 int1[a] = int2[c] = 1 第二次匹配 判断int1[a] == int2[c] 然后赋值为2

242 valid-anagram

题目：是否组成字母一样

思路：+- 最后看是不是都是0

2018-11-6 19:50:03

274 H-Index

题目：求科学家引用指数H， H由最好的N篇论文中 最差的那篇引用数M ，NM最小值构成

思路：1排序，然后看di i 篇大于i+1 否，不大于的时候 输出 i

2 hash 用数组替代hash 表，hash 表示引用数为 i 的论文数目，然后 从后往前 +论文数，比较引用数即可。

290 Word Pattern

题目：同构，不过是字母与字符串的同构

思路：同 但是 用的 两个 map () ，<string,char> <char,string> 记录

2018-11-7 15:00

299 Bulls and Cows

题目：公牛母牛问题？也就是求猜密码的过程中，位置和数值都对的数目和位置不对数字对的数目

思路：利用一个vector<int> a(10)记录0 到 1 在密码出现的次数，位置一致的容易判断就直接相等++，a--，位置不一样时（相同位置不是同一个值）利用a(10) -- 来统计数目

347 Top K Frequent Elements

题目：找到最常出现的k个元素

思路：先统计数字数目，然后利用二维数组 记录 ，数目为 x 的 数字有哪些，然后从后遍历插入结果即可

349 Intersection of Two Arrays

题目：两个数组的公共数字

思路：利用map 计算每一个元素的个数，然后遍历另一个数组，遇到个数>0 则存为结果并初始化为0

350 Intersection of Two Arrays II

题目：两个数组的公共元素

思路：同上 只是初始化不是赋值为0 而是--

355 Design Twitter

题目：设计一个推特，可以发布信息，可以关注，取关，可以查看最新的10条关注人的信息

思路：主要是设计数据结构，利用hashmap进行数据的关联，

std::unordered\_map<int, std::vector<Tweet>> tweets; // [u] = array of tweets by u

std::unordered\_map<int, std::unordered\_set<int>> following; // [u] = array of users followed by u

381 Insert Delete GetRandom O(1) - Duplicates allowed

389 Find the Difference

题目：找两个字符串中不同的那个元素

思路：统计字母个数，判断哪一个不同。 ... 异或也可以用，另外统计数目奇偶 更好一点、

409 Longest Palindrome

题目：求能组成的最长回文串长度

思路：找规律，偶数个的都可以参与组合回文，奇数个的取出偶数个组成回文，还能有一个奇数个的全部参与

438 Find All Anagrams in a String

题目：找所有包含所有元素x的起始位置

思路：先找0到n个，然后推进 1到n+1 优化 记录元素及个数的vector 当这个vector等于 要找的，就输出

447 Number of Boomerangs

题目：回力标？？？

思路：这一题有点歧义

451 Sort Characters By Frequency

题目：按照出现次数给字符串排序

思路：先统计出现次数，然后按照统计次数，统计每种次数的元素，然后倒序输出

454 4Sum II

题目：求4个数组里面取数，4个数 和为0 的个数。

思路：两两合并计算，先算AB两数组，然后用map统计和的数值 以及该数值和的个数，CD相加，判断map中是否有其相反数，有结果就+计数

463 Island Perimeter

题目：找岛屿的边界

思路：找规律 发现岛屿的边界即1旁边0或者边界的个数，遍历即可，但是速度一般

另外一种方法是减法累，不需要考虑太多情况，先统计1的个数，\*4 就是边的个数，然后减去土地的公共边\*2 就是土地和水的边数了。

500 Keyboard Row

题目：就是找字符串是否能由键盘中的一行输出

思路：3个set 用find 去判断即可。

depth-first search

109 Convert Sorted List to Binary Search Tree

题目：将一个链表构造成一个平衡搜索二叉树

思路：主要是将链表构造，那么找到mid就是本题的一个点，利用 mid推进一格 end推进两格；来找中间节点

130 Surrounded Regions

题目：水由0表示，陆地由X表示，找到所有被X包围的0 然后把0变成X。从

思路：即保留边界相连的0，其他0都变成X，遍历4个边界，对边界的0标记为1,

2018-11-12 11:16:56

目标：做到 417题

133 Clone Graph

题目：图的克隆

思路：深度遍历 复制root,复制root的neighbor 一直下去，用map判断是否存在了，同时记录克隆图与原图的一一对应关系

200. Number of Islands

题目：0代表水 1代表陆地，求岛的数目

思路：就是求 1 有几个不相连的块，找到一个块，摧毁一个快，遍历一遍就好了，找到一个 就用dfs从4个方向遍历dfs摧毁，res++

207.Course Schedule

题目：选课问题

思路：

如提示所示，此问题等同于检测由先决条件表示的图形中的循环。 BFS和DFS都可以使用拓扑排序的思想来解决它。如果您发现自己不熟悉这些概念，可以参考他们的维基百科页面。具体来说，您可能只需要参考第三个提示中的链接来解决此问题。

由于pair <int，int>对于图算法的实现不方便，我们首先将其转换为图形。如果课程u是课程v的先决条件，我们将添加从节点u到节点v的有向边。

对于DFS，它将首先访问一个节点，然后访问它的一个邻居，然后访问该邻居的一个邻居......依此类推。如果它遇到在当前DFS访问过程中访问过的节点，则会检测到一个循环并返回false。否则它将从另一个未访问的节点开始并重复此过程，直到访问了所有节点。请注意，您应该创建两个记录：一个是记录所有访问过的节点，另一个是在当前DFS访问中记录访问过的节点。

代码如下。我们使用vector <bool>访问记录所有访问过的节点，使用另一个vector <bool> onpath记录当前DFS访问的访问节点。当前访问完成后，我们将起始节点的onpath值重置为false。

210. Course Schedule II

题目：。。

思路：。。

2018-11-13 19:21:44

332. Reconstruct Itinerary

题目：给一沓机票，求旅程

思路：先把机票 pair关系 重新存为 map<string,mutiset<string>> 然后从开始的地方开始遍历，

，visit,然后找这个map 可以到的地方，取第一个目的地，删除该目的地，visit

394. Decode String

题目：s = "3[a]2[bc]", return "aaabcbc".

思路：这里用的深度递归很巧妙，首先判断是不是数字，如果不是数字就一直读到末尾或者到 " ] "是数字就存下数字，然后过掉【 继续递归 ，这个时候遇到的是不是数字的部分，返回的是【】内部的string,然后存下string 过掉】 循环继续，看是不是数字

417. Pacific Atlantic Water Flow

题目：求能够同时到左上和右下的点，只能从高走到低或者平着走

Pacific ~ ~ ~ ~ ~ ~ 1 2 2 3 (5) \* ~ 3 2 3 (4) (4) \* ~ 2 4 (5) 3 1 \* ~ (6) (7) 1 4 5 \* ~ (5) 1 1 2 4 \* \* \* \* \* \* Atlantic

思路：这里我用的是两个vector来记录，可以走到左上的和可以走到右下的，然后找都能走到的点，具体找是从左上的点出发，深度遍历找能走到的地方。

430. Flatten a Multilevel Doubly Linked List

题目：

**Input:** 1---2---3---4---5---6--NULL | 7---8---9---10--NULL | 11--12--NULL **Output:** 1-2-3-7-8-11-12-9-10-4-5-6-NULL

思路：一个 一个的处理，一个for 遇到有子节点的节点 就把子节点插入到当前节点的后面。而next插在子节点的next 到空的后面，然后继续 for next 就到了子节点的链表了，继续处理有子节点的节点。

二分搜索

240. Search a 2D Matrix II

题目：矩阵[1, 4, 7, 11, 15], [2, 5, 8, 12, 19], [3, 6, 9, 16, 22], [10, 13, 14, 17, 24], [18, 21, 23, 26, 30]

思路：找目标数，一个方法是从左下开始判断，如果目标小于左下则不再最后一行 ，目标大于左下则不在第一列 第二个方法用广度遍历，

275. H-Index II

题目：H指数2，这里给出的论文引用就是按照升序排列的，这里就是普通的找值了

思路：就是普通的二分搜索

278.

二分搜索题目都差不多，主要看如何进行评定边界和选择最终的输出。

374. Guess Number Higher or Lower

题目：猜数，每次告诉你猜低了还是高了

思路： m = (up-down)/2+down; 要这样子才不出问题

441. Arranging Coins

题目：求n能够由几行1,2,3,4,5,6...组成 例如 7 = 1,2,3,1 所以是3行

思路：这个列公式求解，或者-1 -2 -3 减到不能减就是结果都可以

475. Heaters

题目：有一堆暖气装置，有一堆房子，我们知道他们的位置，求能让所有房子都能覆盖暖气的，单个暖气的最小覆盖范围

思路：求房子的右侧最近的暖气，以及左侧最近的暖气，然后取最小值，为该房子的暖气需要，然后整个房子列表中，最大的需要就是结果

two point

19. Remove Nth Node From End of List

题目：删除链表的倒数第N个点

思路：先走一个nodeN步，然后走另一个，

142. Linked List Cycle II

题目：找循环链表的入口

思路：这个题目用快慢指针去做，第一次判断是否有圈，第二次从不同起点出发判断入口

234. Palindrome Linked List

题目：判断链表是否是回文数

思路：要用O(1)空间，那么做两件事，第一找到中间，第二转置中间到最后那半段，然后对比判断

stack

150. Evaluate Reverse Polish Notation

题目：给出+-\*/以及数字，按照后序的方法运算，给出运算结果

思路：stack存储数字，遇到符号就提取数字进行运算，运算结果继续存进去

155. Min Stack

题目：一个能够O(1)时间输出最小值的 stack

思路：利用两个stack 一个stack存放正常数，一个stack存放当前最小数 ，然后维护即可

225. Implement Stack using Queues

题目：用队列作栈

思路：在push的时候将插入的元素插入到队列的最前端每次都是

232. Implement Queue using Stacks

题目：用栈做队列

思路：在pop和top的时候 判断s2是否为空，然后从s1倒数据到s2

331. Verify Preorder Serialization of a Binary Tree

题目：判断一个序列是否是一棵二叉树的先序，用#表示空指针

思路：任何一个节点需要有2个空指针，利用stack 从后往前，遇到# 就i++ 遇到数字键判断 i-2是否大于0 然后i-- ， 找规律

402. Remove K Digits

题目：删除一个字符串数字中的k个数字，使得所得数字最小

思路：这里就要找到规律， 要使得最小，那么则先删除高位比低位大的数，然后没删完再从后往前删除，最后还要判断是否将所有元素都删除完了，如果删除完了要返回“0”而不是NULL

496. Next Greater Element I

题目：找元素的后面的更大的值

思路：先找所有元素中后面较大的值，遍历列表，西安判断 当前值list[i]是否比栈存下来的大，大的话，当前值就是栈中值的更大值，存在map中，然后继续判断栈中值是否小于当前值，最后把当前值存入栈中，进入下一个值的操作。

贪心

贪心算法（又称贪婪算法）是指，在对[问题求解](https://baike.baidu.com/item/%E9%97%AE%E9%A2%98%E6%B1%82%E8%A7%A3/6693186)时，总是做出在当前看来是最好的选择。也就是说，不从整体最优上加以考虑，他所做出的是在某种意义上的局部[最优解](https://baike.baidu.com/item/%E6%9C%80%E4%BC%98%E8%A7%A3/5208902)。

贪心算法不是对所有问题都能得到整体最优解，关键是贪心策略的选择，选择的贪心策略必须具备无后效性，即某个状态以前的过程不会影响以后的状态，只与当前状态有关。

134. Gas Station

题目：一个序列表示加油站的油量，一个序列表示从这个加油站到下一个的消耗，问是否能走一圈，从哪出发好

思路：这里用贪心算法，每一个加油站是亏是赚，用赚开始亏的和来统计，我们找到和函数的最低点，这个最低点以后肯定都是赚的，不会低于这个点，然后如果最后的结果是>0 那么就是可以的。

406. Queue Reconstruction by Height

题目：已知队列中人的身高以及在其前面身高比其高的人数，还原队列顺序

思路：选出最高的一群人并在子阵列（S）中对它们进行排序。 由于没有其他人群比他们高，因此每个人的指数将与他的k值相同。

对于第二高的组（以及其余组），将每一组插入（S）中的k值。 等等等等。

例如。

输入：[[7,0]，[4,4]，[7,1]，[5,0]，[6,1]，[5,2]]

步骤1后的子阵列：[[7,0]，[7,1]]

第2步之后的子阵列：[[7,0]，[6,1]，[7,1]]

前面有几个比你高的，你就插入到第几个去，从高的排到低的

435. Non-overlapping Intervals

题目：**Input:** [ [1,2], [2,3], [3,4], [1,3] ] **Output:** 1 最少去掉几个区间 使得区间不重复

思路：也就是和 最多可以上几门课是一样的，都是排序结束时间，然后一个一个选。

452. Minimum Number of Arrows to Burst Balloons

题目：最佳射击次数，一个气球 从 1 到 4漂 一个从2 到 5漂 ，那么在2到4区间内就可以一枪打两个

思路：就是找公共区间数，先排序，再处理

transback

排列问题，用swap

组合问题用push\_back

89. Gray Code 利用 bitset表示二进制，进行相关操作，flip,取反某一位， to\_ulong 返回值

131. Palindrome Partitioning 加一个判断函数，一个一个判断，如果是就加入，然后进行后续操作，如果不是就跳过

211. Add and Search Word - Data structure design 可以利用新的结构哦

401. Binary Watch 遍历，条件判断，

做任何事情应该都要有规矩，散漫，逃避，不是解决办法，面对问题的方式。

游戏，综艺，剧，小说，路，你的时间应该放在哪里？

现制定规则

link list

143. Reorder List

Given a singly linked list *L*: *L*0→*L*1→…→*Ln*-1→*L*n,

reorder it to: *L*0→*Ln*→*L*1→*Ln*-1→*L*2→*Ln*-2→…

//find the middle of the list, and split into two lists.

//reverse from the middle to the end

//merge these two list

147. Insertion Sort List

Sort a linked list using insertion sort.

//对于每个p->next节点，如果他小于其上一个p节点，pre往前遍历，找到一个比他大的，然后插入

148. Sort List

**Input:** 4->2->1->3 **Output:** 1->2->3->4

// 本题主要的难点在于如何实现O(n log n)的时间复杂度，我们需要利用二分法的思想，但是传统的二分法排序难以实现，所以用分治的方法。把问题分为2个小问题，然后解决2个小问题后再合并两个问题的结果。

160. Intersection of Two Linked Lists

A: a1 → a2 ↘ c1 → c2 → c3 ↗ B: b1 → b2 → b3 找c1

// 我采用的是比较传统的方法，先比较两个链表的长度差异，然后长的先出发，短的后出发，使得其有公共点时，同时位于同一公共点上

//更加好的方法是将两个链表在不改变结构的情况连接起来，相对更加巧妙

203. Remove Linked List Elements

**Input:** 1->2->6->3->4->5->6, ***val*** = 6 **Output:** 1->2->3->4->5

//比较普通的遇到一样的就删除。

206. Reverse Linked List

**Input:** 1->2->3->4->5->NULL **Output:** 5->4->3->2->1->NULL

//基本的倒置方法，通过将未倒置的节点插入在首节点位置，不断的对链表进行转置。

while(cur&&cur->next){

ListNode\* temp = pre->next;

pre->next = cur->next;

cur->next = cur ->next->next;

pre ->next->next = temp;

}

return dummy ->next;

237. Delete Node in a Linked List

给你node然后让你把他从链表中删去

class Solution {

public:

void deleteNode(ListNode\* node) {

\*node = \*(node->next);

}

};

//这个题目有点意思？

328. Odd Even Linked List

**Input:** 2->1->3->5->6->4->7->NULL **Output:** 2->3->6->7->1->5->4->NULL

//这个题目主要就是考虑如何将奇数与偶数分开，可以再原链表的基础上构造2个子链表再拼接，不用多的内存

445. Add Two Numbers II

**Input:** (7 -> 2 -> 4 -> 3) + (5 -> 6 -> 4) **Output:** 7 -> 8 -> 0 -> 7

//普遍的算法是利用vector stack 等数据结构存储然后进行相加

**bit manipunation**

137. Single Number II

Given a **non-empty** array of integers, every element appears *three* times except for one, which appears exactly once. Find that single one.

**Input:** [2,2,3,2] **Output:** 3

//将目标函数用2进制存储起来，然后每位想加，除去重复的次数，除了不重复的数字的肯定不会能整除。那么所取余即为结果。

187. Repeated DNA Sequences

...

位运算不是很会，明天看看博客。

看了差不多了，基本上都是巧妙利用移位，| & >> <<

191. Number of 1 Bits

找目标数的二进制中 有多少个1,

while (n) {

n &= (n - 1);

count++;

}

巧妙的方法

260. Single Number III

描述：给定一组数字nums，其中恰好两个元素只出现一次而所有其他元素恰好出现两次。 找到只出现一次的两个元素

思路：这里，先用xor 算出两个数的xor的结果，通过这个结果，可以知道这两个数，哪位不相同，然后通过这一位不同的判断，把所有数分成两组，分别进行xor，即可得到两个结果

sort：

sort(num.begin(), num.end(), [](int a, int b){

return to\_string(a)+to\_string(b) > to\_string(b)+to\_string(a);

});

这个要记住怎么写

324. Wiggle Sort II

**Input:** nums = [1, 5, 1, 1, 6, 4] **Output:** One possible answer is [1, 4, 1, 5, 1, 6].

需要O(n)的空间复制数组，sort排序，然后再做最后的操作。

215. Kth Largest Element in an Array

找第几大的元素，第一种方法就是快排，然后直接找

第二种是堆排序？这个不太明白看看博客。