301 Remove Invalid Parentheses

Review III

301-1 think about how to DFS.设定last\_j last\_i 来减少之后遍历的内容

301-2 开始count++ count—当count<0 才需要进行异常处理

301-3 注意中断条件 什么时候return

301-4 正面遍历完了没有中断 需要反面遍历。这种情况，char[] {‘(‘, ‘)’}数组需要反过来。

注意运用char[]数组来储存()等经常出现的char

**302 Smallest RectangleEnclosingBlackPixels**

**Review III**

302-1 1 0数组要看清到底是char[]数组还是int[]数组

302-2 bfs图的时候要防止revisit，visit过的点需要更改

302-3 Matrix的最优解就是binary search！！！

本题是利用一个方程同时处理两个维度上binary search的典范！O(nlogm)+O(mlogn)

303 304 Range Sum Query

Review III

303-1 既然关心sum那么就把数组变成sum的数组！

304-1 一维变二维 在写dp关系时候注意用三值运算符排出边界条件

305 Number Of Islands II

Review III

本题是union finding的典型例题。当在矩阵或者图中有交汇合并的情节的时候请使用union finding

UnionFind是一个单独的object 里面有一个id数组用于对数据进行分类。有find函数找dependency，有isConnected() 最重要的是有union函数整合数据。

306 AdditiveNumber

Review I

306-1 Long.parseLong() String.valueOf(integer/long/char/)

307 Range Sum Query – Mutable

Review I

307-1 前n项和：binary index tree 用binary index tree

307-2 如何求一个数以二进制表示最低的一位1的位置i&(-i)

307 308 Binary Index Tree跳过先不看 revisit later

tree = new int[nums.length+1];

for(int i=1; i<tree.length; i++)

{

sum = 0;

lowbit = i&(-i);

for(int j=i; j>i-lowbit; j++)

sum += nums[j-1];

tree[i] = sum;

}

public int sumRange(int i, int j)

{

return sum[j+1] – sum(i);

}

private int sum

{

int sum = 0;

for(int i=k; i>0; i= I – i&(-i)

sum += tree[i];

return sum;

}

309 Best Time to Buy and Sell Stock with Cooldown\

Review III

309-1 DP with multiple arrays at the same time。三个dp数组对每一个位置有三种描述 互相影响，求得最后结果。

310 Minimum Height Trees

ReviewIII

310-1 从叶子往中间跑，首当其冲就想到了edgeArrayList或者hashMap记录neigbor关系。通过数neighbor个数判断是不是叶子，进一步决定是否删除不断删除不断删除直到hashMap/ArrayList中只剩两个元素

311 Sparse Matrix Multiplication

311-1 注意C[i][k] += A[i][j] \* B[j][k];中的加号！

312 Burst Balloons

ReviewII

312-1 dp[i][j] = Max(dp[i][j], dp[i][x-1] + nums[i-1]\*nums[x]\*nums[j+1] + dp[x+1][j])

312-2 边界条件的处理 可以考虑扩充数组：arr[len] 🡪 arr[len+2] 收尾各加一个1

313 Super Ugly Number

Review II

313-1 dp不断的生成新数，注意remove duplicate.去重非常重要！！！

314 BT Vertical Order Traversal

Review II

314-1 本题水平方向 垂直方向都有顺序。必然想到 1 如果存储用map一定用treemap可以保证key的有序性。 2 重新定义class TreeNode 本身 让其有权重。如果用dfs左右方向的顺序可以在argument中表示。如果用bfs上下方向的权重可以用queue的round表示。所以两种方法均可以

314-2 如果用queue可以每次输入两个值：第一次为值本身 第二次为权重，这样就不需要重新定义TreeNode Class了

315 Count Of Smaller Numbers After Self

315-1 Arrays.asList() 是把Array转化成List的工具。但是如果Array的元素必须是对象。所以如果是int类型需要以Integer类型加入

315-2在过程中找数：有没有顺序 ？ 一定是binary search

316 Remove Duplicate Letters

Review III

316-1 肯定从最难着手的字母考虑：建立一个map表示每个字母的最后一次出现的位置。这样for loop的时候就考虑从开始到这个点之间的最小值！！！想清楚为什么

317 Shortest Distance From All Buildings

Review III

317-1 从1点开始bfs。当某个点被遍历过k次（k是已知building的个数）并且distance总和最小，则取该点。所以需要有int[][] distance 表示到达该点的所有building的距离总和，int[][] touched 表示到达过该点的building个数。注意，当每一个building开始bfs时候，visited=true需要写在进入queue的时候而不是出queue的时候！！！想清楚为什么！！！

318 Maximum Product of Word Lengths

Review II

318-1 起初想到hashMap但是hashMap存储的都是Integer而且是位置关系，并且用于后边，所以想到了bit manipulation

318-2 单比特运算符的最终结果一定要加括号！！！例如：if ( (a&b)<c )

319 Bulb Switcher

Review II

319-1 不断的翻转，想到位操作

320 Generalized Abbreviation

320-1 substring backtrack的时候注意特殊情况substring 为”” 需要用三值运算符对值为0的情况特殊处理

320-2 bit operation本题是另一种用法。每个位置都有两种可能：归入数字或者显示。分别用1或者0表示。所以一个n位数的所有情况可以用从1 到1<<(n-1)的所有数来表示。

321 Create Maximum Number

Review III

321-1 要最大的n个元素还要用顺序。所以当取到当前最大的元素的时候，前面比他小的一定会舍去，只有这样才能保证相对顺序

321-2 合并的时候首先取length的最大值，然后for循环的时候len>len1? 用三值运算符表示此轮该数组有数可取还是已经过界（设为0）。Merge函数的同时可以配一个compare函数，参数是arr1, arr1 current index, arr2, arr2 current index. 每轮均可以调用

322 Coin Change

Review II:

1 为什么会想到dp！！！在recursion的过程中 可选的值是非常有限的，仅限于coins的length。并且结果只需要最短的极端值，不需要所有的解。解决极端值的问题，dp是唯一方法！

323 Number Of Connected Components In An Undirected Graph

323-1 连图问题 一看就是unionfinding 。union finding 中 parent数组，初始化函数，section个数，union函数，find函数，请回忆

解法二 pending

325 Maximum Size Subarray Sum Equals k

Review II

324-1 如何算在数组中一段连续的sum from i to j: sum(start->i) – sum(start->j)

324-2 TreeSet 复习：

treeSet.ceiling(E e);

treeSet.floor(E e);

treeSet.contains()

treeSet.iterator()

treeSet.first()

treeSet.last()

treeSet.pollFirst()

treeSet.pollLast()

325 Binary Index Tree

Tushar Video!

327 Count Of Range Sum

Review III

327-1 方法1：在for loop中不断填充（put）hashmap并且用containsKey对以前的hashMap进行比较。相当于减少了一层hashMap。如果对值有要求可以用subMap来约束key value

327-2 : System. arraycopy(Object src, int srcPos, Object dest, int destPos, int length)

329 Longest Increasing Path In A Matrix

Review I:

329-1 标准dfs注意用record数组避免重复

330 Patching Array

Review II

330-1 善于用于数学归纳法其实是dp思想.前面的先解决再加一个数 看看后面的可不可以由前面的结论得出。

331 Verify Preorder Serialization of a BT

Review II

331-1 validation必须要找一些特性。在树里面。每个非叶子点都有一个parent两个child，每个叶子点都有一个parent所有点的连线和为 0。设立一个diff每当非叶子节点-1叶子检点+2，无论何时此值不可以小于0

332 Reconstruct Iterneraray

Review II

332-1 对于当前起飞点，要找字典序最小的机场最为降落点。所以肯定用hashMap。而hashMap的value需要一个priorityQueue

333 Largest BST Subtree

333-1 My solution ! good! 每个点有三个信息需要传递，最大值最小值，总的和。我把返回值用于表示总的和，正负代表是否所有点均valid. Int[]数组代表最值参数传递。

334 Increasing Triplet Subsequence

334-1 n = 3一看就是dp

336 Palindrome Pairs

Review III

336-1 最简单的方法肯定是n平方。如果要减维度。必然用map并且，要遍历string的长度的substrings。Revisit！看subString中非palindrome的部分reverse是否在map中

336-2 注意空集的情况

337 House Robber III

Review III

337-1 隔层问题一定想到recursion！

338 Counting Bits

Review II

338-1 DP 或者greedy类似于数学归纳法。n==0 n==1 等基数情况一定要单独考虑

338-2 JAVA里面的bitcount函数: int count = Integer.bitCount(num);

338-3 sol3 last set bit在bit operation 中是一个很重要的概念。

340 Longest Substring with At Most K Distinct Characters

Review II

340-1 对于characters的题上来就要问是不是只针对字母还是都包括uppper case, lower case and special characters

340-2 new int[256]

341 Flatten Nested List Iterator

在不同的level中轮换，想到了什么！递归嵌套的调用

Review III

341-1 Discussion Solution 1:如何用一个stack！！！倒着遍历存入stack，如需进一步下一层，把下一层继续遍历加在本层的上面！！！

345 Reverse Vowels of a String

345-1 check official solution: discussion1 : 把所有字母组成一个单独string aeiouAEIOU。 然后用String.indexOf() 判断。

345-2 reverse一定想到 双指针站肩！

346 Moving Average From Data Stream

Review II

346-1 一看是有个有限长度序列，想到linkedlist和queue由于只需要不断读入和取出。可以用int[] array 数组也可以，keep一个指针表示当前位置，如果越界原来的结果自动抹掉

346-2 如果结果是一个double 在过程中做/运算的时候前面一定不要忘了加double (double) sum/n

346-3 本题discussion做法，也可用一个数组加一个指针位置的办法取代linkedlist 想想why

347 Top K Frequent Elements

Review II

347-1 sol1 : My Sol: PriorityQueue: 注意不是list。是一种单独的数据结构(heap)。 Compare 函数必须是object的argument。所以int必须写成Integer

347-2 List 可以为数组的类型: List<Integer>[] bucket = new List[n];

348 Design Tic-Tac-Toe

348-1 分析变化！当加入一个新点后可能产生的变化

349 350 Intersection Of Two Arrays