

**题目：** 2

**题意：** 用链表给出两个数，并且是逆序表示的， $243+564=708$

**思路：**

- (1) 模拟，直到两个都为空结束
- (2) 只需要申请一个内存

**注意：**

- (1) 不需要申请内存
- (2) 注意怎么保存结果（第一个节点为预留节点，每次把值加到 l1 上，然后将 l1 连接在链表后面）
- (3) 建议仔细思考

**题目：** 3

**题意：** 给出一个字符串，求最长的没有重复字符的字符串

**思路：**

- (1) 用一个数组保存上一次该字符出现的位置，如果出现该字母，则用 startpos 和位置加一判断，谁更大，即更新 startpos

**注意：**

- (1) 对于遇见重复的字符，更新 startpos 应该是取 startpos 和位置加一的最大值

**题目：** 5

**题意：** 给出一个字符串，求最长回文子串

**思路：**

- (1) 回文串  $O(n)$  算法

**注意：**

- (1) 无

**题目：** 11

**题意：** 相当于给出 n 个挡板，求任意两个挡板所能装水的最大值

**思路：**

- (1) 模拟，用 L, R 两个指标，如果 L 高度小于 R 高度，显然是 L 往右移动

**注意：**

- (1) 无

**题目：** 14

**题意：** 求 n 个字符串的公共前缀

**思路：**

(1) 模拟

注意:

(1) 无

题目: 15

题意: 给出一个数组, 从中选三个数, 使其和等于 0 (K sum 问题)

思路:

• <http://tech-wonderland.net/blog/summary-of-ksum-problems.html>

注意:

(1) 第一层循环, 如果和前数相等, 跳过

(2) 在用两个指标时, 如果想加等于目标数, 两个指针同时跳过和他们相等的数

题目: 16

题意: 给出一个数组, 从中选三个数, 使其和最接近 target (K sum 问题)

思路:

(1) K sum 问题变形。在进行两指标运算时, 对每次循环进行判断, 其它一样

注意:

(1) 第一层循环, 如果和前数相等, 跳过

题目: 17

题意: 手机键盘, 每个数对应几个字母, 给出一个数字串, 求组成的字母串

思路:

(1) Dfs

注意:

(1) 如果输入为空, 那么输出也是空, 而不是一个长度为 0 的字符串

题目: 18

题意: 给出一个数组, 从中选四个数, 使其和为 target (K sum 问题)

思路:

(1) K sum 问题, 但是更好的方法不清楚, 还需思考

注意:

(1) 第一层循环, 如果和前数相等, 跳过

题目: 20

**题意：**判断括号是否匹配

**思路：**

(1) 使用栈模拟

**注意：**

(1) 当遇见反括号时，不能用栈顶元素和该反括号相等符比较，只能是配对，不能是相等

**题目：21**

**题意：**合并两个有序链表

**思路：**

(1) 模拟

(2) 递归

**注意：**

(1) 空指针不能有下一个指针

(2) 可以不用归排的思想,只要有一个为 **NULL** 就退出,然后直接把剩下的赋给结果链表

**题目：22**

**题意：**一共有  $n$  对括号，问有哪些合法的括号序列

**思路：**

(1) 递归，如果还可以加左括号，加上。如果右括号少于左括号，可以加右括号

**注意：**

(1) 无

**题目：24**

**题意：**给出一个链表，交换每两个相邻的链表

**思路：**

(1) 对于  $p$  和  $p \rightarrow next$  结点都非空，交换两个结点

**注意：**

(1) 头结点要更换

(2) 对于交换后的第二个结点的下一个结点，如果后两个非空，则为后第二个，否则为后第一个

**题目：26**

**题意：**去除已排序数组中的重复数

**思路：**

(1) 标记法

(2) 模拟

**注意：**

(1) 无

**题目：27**

**题意：**从数组中移除指定元素

**思路：**

(1) 模拟

(2) 遇见指定的数与最后一个数交换，然后长度减一

**注意：**

(1) 无

**题目：28**

**题意：**求一个字符串在另一个字符串中出现的位置

**思路：**

(1) KMP

**注意：**

(1) 如果没有找到返回-1

(2) 字符串为空的情况

**题目：31**

**题意：**求一个序列的下一个序列

**思路：**

(1) 从后往前，对于每个数，再从后到它的前一个位置，出现比它大的第一个数，两个交换，然后翻转这个数后面的数

**注意：**

(1) 是大于，不能是等于

(2) 注意原数组是最大的数（全逆序）

**题目：33**

**题意：**给出一个有序数组的旋转序列，求目标数的位置

**思路：**

(1) 先找出最小数的位置，然后判断是在左边还是右边二分

(2) 直接二分

**注意：**

(1) 二分姿势，可以依照（1,2）和（2,1）来写二分

(2) 对于 mid，如果 mid 大于 L，那么说明 L 到 mid 是递增的：如果 t 在这之间，那么 R 向

右移；如果不在这之间，L 向左移。如果 mid 小于 L，说明，mid 在第二区间，L 在第一区间，如果 t 大于 L 或者小于 mid，R 都应往左移

**题目：34**

**题意：**给出一个有序数组与目标数，求该目标数的起始位置和终点位置

**思路：**

(1) 利用 lower\_bound 和 upper\_bound

(2) 手写二分

**注意：**

(1) 第一种，后者要减一

(2) 判断不存在情况：后者位置小于前者位置

(3) 注意二分的姿势

**题目：35**

**题意：**给出一个有序数组和一个目标数，问目标数应该插在哪个位置

**思路：**

(1) 二分

**注意：**

(1) 二分姿势

**题目：36**

**题意：**判断一个数独是否合法

**思路：**

(1) 模拟

**注意：**

(1) 无

**题目：37**

**题意：**数独填写，保证有唯一解

**思路：**

(1) Dfs

**注意：**

(1) 如果获取哪个九宫格

(2) 如果找到答案，应该再 dfs 后也应该 return

**题目：**38

**题意：**1,11,21,1211,111221,……求第 n 个

**思路：**

·模拟

**注意：**

(1) 无

**题目：**43

**题意：**高精度乘法

**思路：**

(1) 模拟乘法，用一个数组保存每一位与另一个数的每一位的结果（不需要进位），最后再进位

**注意：**

(1) 包含 0 的情况，结果不要为空或者有前导 0

**题目：**46

**题意：**给出一个序列，求其所有能组成的序列

**思路：**

(1) Dfs

**注意：**

(1) 无

**题目：**47

**题意：**给出一个序列，求其所有能组成的序列 (含有重复元素)

**思路：**

(1) 利用 next\_permutation 思想

**注意：**

(1) 无

**题目：**48

**题意：**将一个矩阵顺时针旋转 90 度

**思路：**

(1) 从外到内，每一层进行旋转

(2) 对于每层旋转，对于一边的任何数，与三边的数，进行旋转

(3) 反正时间复杂度为  $O(n*n)$

**注意：**

(1) 写旋转交换的时候要仔细

**题目：** 50

**题意：** 求  $x$  的  $n$  次方

**思路：**

(1) 快速幂

**注意：**

(1) 注意  $n$  为负数的情况。在所有将 `int` 的负数变成正数的时候，要注意 `int` 的最小值

**题目：** 51

**题意：** 求  $n$  皇后问题的解

**思路：**

(1) dfs

**注意：**

(1) 判断列重复，可以用一个整数，二进制保存

(2) `String` 不要每次都重新定义一个

**题目：** 52

**题意：** 求  $n$  皇后问题的解的个数

**思路：**

(1) Dfs

(2) 打表

**注意：**

**题目：** 53

**题意：** 求和最大的子数组

**思路：**

(1) 略

**注意：**

(1) 注意全是非正数情况

**题目：** 54

**题意：** 输出矩阵，按螺旋形

**思路：**

(1) 每次输出剩下矩阵的最外圈

**注意：**

(1) 当输出下面和右面一行时，要注意

(2) 错了许多次，再写一次

**题目：58**

**题意：**求最后一个单词的长度

**思路：**

(1) 模拟

**注意：**

(1) 要是空格,必须跳过;

(2) For 循环后的  $i++$  可以不要

**题目：59**

**题意：**填写螺旋数

**思路：**

(1) 和 54 题一样

**注意：**

(1) 见 54 题

**题目：60**

**题意：**求  $1-n$  全排列的第  $k$  个

**思路：**

(1) 模拟，对于第  $i$  位，用  $k / (i-1)! + 1$ ，代表还剩下数的第几个，最后一个特殊处理

**注意：**

(1)  $K$  首先减一，是还剩下数的第几个，不是具体的哪一个数

**题目：61**

**题意：**将链表的后  $k$  个，移到链表前

**思路：**

(1) 翻转前  $len-k$ ，再翻转后面  $k$  个，整体再翻转

(2) 直接找到  $len-k-1$ ，把最后一个的  $next$  赋为  $head$ ， $len-k-1$  的  $next$  为空，返回  $len-k$  节点

**注意：**

(1) 链表为空的时候

(2)  $K$  要先对  $len$  取余



**题目：**62

**题意：**从  $m \times n$  矩阵的左上角，走到右下角，有多少种走法

**思路：**

(1)  $dp[i][j] = dp[i-1][j] + dp[i][j-1]$

**注意：**

(1) 无

**题目：**63

**题意：**从  $m \times n$  矩阵的左上角，走到右下角，有多少种走法，其中有障碍，不能走

**思路：**

(1) 和上题一样，只是如果遇见障碍，那么那个位置的  $dp$  就为 0

**注意：**

(1) 上和左边界特殊处理

**题目：**64

**题意：**从左上到右下的路径中最小权值和

**思路：**

(1)  $dp[i][j] = \min(dp[i-1][j], dp[i][j-1]) + grid[i][j]$

**注意：**

(1) 无

**题目：**66

**题意：**用数组给出一个数，然后将这个数加 1

**思路：**

- 高精度模拟

**注意：**

- 注意全是 9 的情况就好了

**题目：**67

**题意：**给出两个二进制字符串，求它们的和

**思路：**

- 高精度模拟

**注意：**

• 无

**题目：**69

**题意：**求一个数的平方根

**思路：**

(1) 调用 `sqrt` 函数，然后将其减少一个定值，再加挨着判断

**注意：**

(1) 减去一个定值后，应该大于等于 0；

(2) 注意相乘越界

（如果是判断是否为一个平方根，注意负数的情况）

**题目：**70

**题意：**求第  $n$  项斐波那数列

**思路：**

(1) 模拟，使用  $O(1)$  复杂度

**注意：**

(1) 无

**题目：**71

**题意：**给出 `unix` 的操作方式，求最终的目录文件

**思路：**

(1) 模拟，用栈

(2) 就 `pop`，/多余的跳过，...都跳过，具体见代码

**注意：**

(1) 无

**题目：**73

**题意：**给出一个矩阵，如果某个数为 0，那么将其行和列都全部设置为 0

**思路：**

(1) 用两个数组，记录每行没列是否有 0（空间复杂度为  $O(n)$ ，也可只记录为 0 的行和列）

(2) 对于  $O(1)$  时间复杂度的算法：先找一个为 0 的位置为  $i, j$ 。对于其它为 0 的，比如，坐标为  $x, y$  的，也就是把  $x$  行和  $y$  列变为 0，那么可以分为两个，一个列为  $j$ ，行为  $x$ （和  $ij$  同列），一个列为  $y$ ，行为  $i$ （和  $ij$  同行）。最后再全部赋值为 0

**注意：**

(1) 不能直接修改，因为修改了，就会影响后面的判断

(2) 如果行和列数据比较小，可以用二进制记录

- (3) 对于第二种方法，要注意，在对每行赋值为 0 的时候，不能将 i 行赋值为 0，最后才将 i 行赋值为 0

**题目：74**

**题意：**给出一个矩阵，每一行是递增的，每行的第一个大于上行的最后一个

**思路：**

- (1) 二分，该矩阵相当于一个  $m*n$  的有序数组

**注意：**

- (1) 无

**题目：75**

**题意：**对一个只包含 0,1,2 三个数的数组排序

**思路：**

- (1) 标记每个出现次数，再赋值，但是需要扫描两遍  
(2) （经典）假如 0,1,2 在数组中存的结束位置为 x, y, z。如果遇见 0，那么先将 2,1 的位置往后移动；如果遇见 1，先将 2 的位置往后移动（具体见代码）

**注意：**

- (1) 无

**题目：77**

**题意：**1-n 取 k 个数，所有取的种类

**思路：**

- (1) Dfs

**注意：**

- (1) 无

**题目：78**

**题意：**求一个集合的子集

**思路：**

- (1) Dfs

- (2) 位运算

**注意：**

- (1) 无

**题目：**82

**题意：**删除一个链表中的重复数（只要有重复的就全部删除）

**思路：**

(1) 先保存一个头结点，然后遇见一个没有重复的，就加在该头结点的链表中

**注意：**

(1) 怎么判断以及跳过重复的元素

(2) 注意细节问题

**题目：**83

**题意：**删除有序链表中重复的数

**思路：**

(1) 当前数等于下一个数的时候，删除下一个数

**注意：**

(2) head 节点为空的时候

(3) 删除的节点要是释放空间

**题目：**86

**题意：**把一个链表中，小于  $x$  的数放在前面，否则放在后面

**思路：**

(1) 用两个头结点保存两个链表，最后合并即可

**注意：**

(1) 无

**题目：**88

**题意：**合并两个有序数组到第一个数组

**思路：**

(1) 由于按照从前面合并会覆盖，所以从后面覆盖即可

**注意：**

(1) 无

**题目：**91

**题意：**给出一个数列，问有多少种用字母表示的方法（如，12，可以为“AB”，“L”）

**思路：**

(1)  $Dp[i]=dp[i-1]+dp[i-2]$

**注意：**

(1) 01 这种有前导 0 的不能组合，遇见 0，则必须和前面组合，所以遇见 0，单独处理

- (2) 当  $i$  等于 1 时，特别处理
- (3) 组合起来大于 26 小于 1 都不行

**题目：** 94

**题意：** 二叉树的中序遍历

**思路：**

(1) Dfs

**注意：**

(1) 无

**题目：** 96

**题意：** 给出  $n$  个结点，求能组成的二叉树

**思路：**

(1) 卡特兰数

**注意：**

(1) 无

**题目：** 100

**题意：** 判断两个二叉树是否相同

**思路：**

(1) dfs 思想

**注意：**

(1) 注意判断，不能用  $p==q$

**题目：** 101

**题意：** 给出一个二叉树，判断它是否为镜子二叉树

**思路：**

(1) Dfs

**注意：**

(1) Root 为空，以及如何写 dfs，和判断什么时候返回 false

**题目：** 102

**题意：** 给出一个二叉树，存储每一层的元素

**思路：**

(1) Dfs，对于每个深度，存入相应的数组

注意：

(1) 无

题目：104

题意：给出一个二叉树，求该二叉树的最大深度（根节点到叶节点的最长距离）

思路：

(1) dfs

注意：

(1) 如何判断叶节点

题目：105

题意：给出一个二叉树的先序遍历和中序遍历，然后构造这棵二叉树

思路：

(1) 先序遍历的第一个是中间的那个数，然后将中序遍历分成两拨，然后先序遍历剩下的前部分与中序遍历的左边，进行递归，剩下的进行递归

注意：

(1) 无

题目：106

题意：给出一个二叉树的后序遍历和中序遍历，然后构造这棵二叉树

思路：

(1) 和 105 类似

注意：

(1) 无

题目：107

题意：给出一个二叉树，存储每一层的元素

思路：

(1) Dfs，和 102 一样，只是结果翻转

注意：

(1) 无

题目：110

**题意：**判断一颗二叉树是不是平衡树

**思路：**

(1) Dfs,对于每个节点判断，左右最深节点深度是否差值的绝对值小于等于 1

**注意：**

(1) 该怎么写 dfs

**题目：111**

**题意：**给出一个二叉树，求该二叉树的最小深度（根节点到叶节点的最短距离）

**思路：**

(1) dfs

**注意：**

(1) 如何判断叶节点

**题目：112**

**题意：**给出一个二叉树，求是否存在从根节点到叶节点的和为 sum

**思路：**

(1) dfs

**注意：**

(1) 根节点为空

(2) 只有一个节点

(3) 判断是否为叶节点，需要判断它的两个孩子都为空

(4) 不能剪枝（因为加负数会越来越小）

(5) 建议重新写一遍

**题目：113**

**题意：**给出一个二叉树，求从根节点到叶节点的和为 sum 的所有路径

**思路：**

(1) dfs

**注意：**

(1) 根节点为空

(2) 只有一个节点

(3) 判断是否为叶节点，需要判断它的两个孩子都为空

(4) 不能剪枝（因为加负数会越来越小）

(5) 建议重新写一遍

**题目：118**

**题意：**输出前  $n$  行杨辉三角

**思路：**

- 求和

**注意：**

- (1) 输入为 0
- (2)  $V$  的大小一定要是 `numRows`

**题目：**120

**题意：**三角形的最短路径

**思路：**

- (1) dp

**注意：**

- (1) 空间复杂度可以为  $O(1)$

**题目：**121

**题意：**给出一行股票价格，问最多一次买进，一次卖出，最大收益多少

**思路：**

- (1) 遍历数组，一个数记录当前位置以前最小的数，然后与当前数作差

**注意：**

- (1) 数组为空的情况

**题目：**125

**题意：**给出一个字符串,判断是否为回文串,只判断其中的字母和数字

**思路：**

·p 从起始位置记录,q 从最后的位置记录,遇见非法就跳过.否则判断是否相等

·将合法字符保存起来(可以就保存在 `s` 中),然后判断是否为回文

**注意：**

- (1) 不区分大写字母和小写字母;
- (2) 有数字和字母两种

**题目：**129

**题意：**给出一个二叉树，求每个根节点到叶节点组成的数的和

**思路：**

- (1) dfs

**注意：**

- (1) 注意细节，尤其是部分先后顺序



**题目：**134

**题意：**在圆环上有很多个加气站，在  $i$  加气站，有  $gas[i]$  的油，从  $i$  到  $i+1$  要消耗  $cost[i]$  的油，问从哪个加气站出发，能跑完整个一圈。保证解最多一个，如果不存在返回 -1

**思路：**

(1) 暴力可解

(2) 所有的油加起来小于消耗的油，则无解，否则有解。从起点开始，用个  $sum$  记录油箱剩余油的数量。如果小于 0，则代表前面任意一个为起点都不行（可以证明一下，如果之前一个点  $i$  可以为起点，那么所剩的油一定不会比现在的油多，如果比现在油多，则证明，在  $i$  之前有油不够的情况，导致后面补充，而不够的情况，肯定会判断为其中一个点不能走到下一个点），只能以该点的下一个点为起点，并将  $sum$  赋值为 0

**注意：**

(1) 无

**题目：**136

**题意：**一个数组中，有一个元素出现了一次，其他元素都出现了两次，求这个元素

**思路：**

(1) 利用异或运算，异或运算满足交换律，所以两个相同的异或将为 0，

**注意：**

(1) 无

**拓展：**

- (1) 有一个元素出现一次，而其它元素出现  $k$  次（ $k$  为偶数时，一样的思路； $k$  为奇数时，二进制的每一位 1 的个数求出来，然后  $\text{mod } k$ ，不为 0 的，就代表该元素在这位贡献 1，最后还原该数（注意，不要用除以 2 的办法，应该用 &，因为有负数））
- (2) 有两个元素只出现一次，其它元素都出现两次（先将所有数进行异或，结果不为 0，然后选择该结果的一位为 1 的位置，将该数组分为两组，然后对每组异或，得到的两个结果就是答案（相同的数将会被分为一组））

$K$  为 3 的代码：

```
ans2|=ans1&x;
```

```
ans1^=x;
```

```
ans3=ans1&ans2;//由于 ans3 的 1 为 ans1 和 ans2 贡献的，所以，ans3 为 1 的地方，ans1 和 ans2 为 0
```

```
ans3=~ans3;
```

```
ans1&=ans3;
```

```
ans2&=ans3;
```

**题目：**141

**题意：**判断一个链表中是否存在环

**思路:**

(3) 一个结点一次走两步，另一个结点一次走一步，最后看两个节点是否相等

**注意:**

(1) 在走的过程中，判断是否为空的情况

**题目: 142**

**题意:** 给出一个链表，如果存在环，返回环的首节点，否则返回空

**思路:**

(1) 列出一个等式，会发现，头结点到环首节点与相遇结点到环首节点距离相同

**注意:**

(1) 无

**题目: 143**

**题意:** 重新排列一个链表，原链表为  $L_1, L_2, L_3, \dots, L_n$ , 重排后变成  $L_1, L_n, L_2, L_{n-1}, L_3, \dots$

**思路:**

(1) 取出后半部分的链表，然后翻转后面的链表。然后再合并

**注意:**

(1) 当链表为或者只有一个节点的时候

(2) 合并的时候注意一些细节问题

**题目: 144**

**题意:** 求二叉树的先序遍历

**思路:**

(1) 用栈，先入右孩子，再入左孩子

(2) Dfs

**注意:**

(1) 三种遍历，最好都用非递归解法写一下

(2) 非递归解法空间复杂度可以为  $O(n)$  或  $O(1)$

**题目: 147**

**题意:** 链表的插入排序

**思路:**

(1) 先申请一个空间，表示头结点。然后遍历该链表，从前向后进行插入排序

**注意:**

(1) 每次插入后，都应该将当前链表的尾节点赋值为空

**题目：148**

**题意：**对一个链表排序

**思路：**

- (1) 归并排序，每次找到中间节点（一个走两步，一个走一步），然后分治
- (2) 快速排序，小于该节点放在左边，否在放在右边

**注意：**

- (1) 找到中间节点，左边从头结点，右边从中间结点的下一个，中间结点下一个赋值为空
- (2) 归并排序的合并：返回合并链表后的头结点，历尽一个节点就可以停止，将另一个节点复制在后面
- (3) 建议重写

**题目：150**

**题意：**给出后缀表达式，求结果

**思路：**

- (1) 用栈，模拟

**注意：**

- (1) 字符串转化为整数时，注意负数

**题目：151**

**题意：**翻转一句话，但是不翻转单词

**思路：**

- (1) 首先翻转整句话，再翻转每个单词
- (2) 将每个单词放入 `stack` 中，然后取出

**注意：**

- (1) 当遇见多个空格时，变成一个，句首和句尾没有空格

**题目：152**

**题意：**求一个数组的连续最大乘积

**思路：**

- (1) 由于可能存在负数，所以保存一个最小值（小于等于 0，如果大于等于 0，则赋值为 1），保存一个最大值（大于等于 0，如果小于等于 0，则赋值为 1）。因为最小值乘以负数会变成最大值
- (2) 对于第  $i$  个的所有可能有三种方式得到，一是本身，而是前面最大值乘以当前数，三是前面最小数乘以当前数

**注意：**

- (1) 如何证明其正确性

(2) 注意后面要使用的数，会在前面覆盖，所以预先保存

**扩展：**

如果包含有小数，那么需要改变一个条件，那就是最大值小于 1，就应该赋值为 1

**题目：153**

**题意：**给出一个数组是另一个已排序数组的旋转，求旋转后的数组中的最小值

**思路：**

(1) 遍历求最小值

(2) 用二分搜索，如果 mid 大于 L，说明，最小值在后面

**注意：**

(1) 在二分搜索中，注意如果该数组为已排序好的数组情况

**题目：155**

**题意：**求一个栈不断入栈出栈过程中的最小元素

**思路：**

(1) 使用两个栈，一个栈，正常操作；另一个栈：入栈，当该栈为空或者元素小于等于栈顶元素。出栈，当两个栈顶相等时，单调栈出栈

**注意：**

(1) 无

**题目：160**

**题意：**给出两个链表，求它们的相交节点

**思路：**

(1) 先遍历两个链表，判断最后一个是否相等，然后求出两个链表长度差，使长的那个链表首先跑差度，然后一起跑

**注意：**

(1) 每次使用链表的时候，都应该判断头结点是否为空

**题目：162**

**题意：**求一个数组中的坏数（比左右两边的数都大）

**思路：**

(1) 遍历整个数组即可

**注意：**

(1) 注意边界情况

**题目：**165

**题意：**给出两个版本号，比较两个版本的大小

**思路：**

(1) 模拟，由于版本中有许多‘.’，所以以其为分界线，将每个部分转化成整数，然后比较

**注意：**

(1) 无

**题目：**166

**题意：**求一个数除以另一个数的结果，循环部分用括号括起来

**思路：**

(1) 模拟，如果余数出现相同的，就为循环节，由于余数可能太大，所以用 map 标记

**注意：**

(1) 可能存在负数，且转化为正数，可能越界，不应该输出-0

(2) 自然数转化为 string，要注意最后要翻转以及为 0 的时候

(3) 整除以及分子大于分母情况

**题目：**169

**题意：**给出一个数组,有个数出现了超过一半的次数。求出这个数

**思路：**

(2) 每次去掉两个不同的数，最后剩下的数就是答案

(3) 先排序，然后用一个间隔一半的两个指标移动，如果这两个数相等就是答案

(4) 求中位数，中位数肯定是一半以上

**注意：**

(1) 代码的实现

**题目：**179

**题意：**给出一个数组，求它们所能组成的最大数

**思路：**

(1) 每个数转化成 string，然后排序

**注意：**

(1) 全部为 0 的情况

**题目：**189

**题意：**给出一个数组将整个数组顺移 k 位

**思路：**

(5) 先将前  $n-k$  个数翻转，再将后  $k$  个数翻转，最后将整个数组翻转

(6) 借用  $k$  个空间

**注意：**

(1)  $k$  的值可能比  $n$  大

(2) 字符串长度为 0

**题目：198**

**题意：**给出一个非负数组，从中取一些数，使其和最大，且这些数不相邻

**思路：**

(4) 保存两个最大值，一个从起点开始，另一个从第一个和第二个最大值开始，遍历整个数组，维护这两个值。（意思：如果现在进行到  $i$ ，第一个值保存到  $i-1$  的最大值，第二个值，保存到  $i$ ）

**注意：**

(1) 数组的长度为 0、1 的时候

**题目：199**

**题意：**给出一个二叉树，求能从右边看见的序列

**思路：**

(1) Dfs，先遍历右子数，再遍历左子树，如果当前深度大于等于 `vector` 的大小，说明这层还没有元素被看见（被加进 `vector`）

(2) 层次遍历，每一层加一个进去

**注意：**

(1) 不能因为有右孩子就不遍历左孩子，因为左孩子可能比右孩子的深度大

**题目：200**

**题意：**和油田一样，问有多少个不连在一起的区域

**思路：**

(1) Dfs

**注意：**

(1) 注意，用引用，且想想不改变原图怎么做

**题目：201**

**题意：**求  $[m, n]$  区间所有数取与运算的结果

**思路：**

(1) 所有数中，只要第  $i$  位有一个为 0，那么该位就为 0。 $m$  和  $n$  不断除以 2，如果两个相等，那么所有数的前面部分都是这两个，而后面的全部为 0

**注意：**

(1) 无

**题目：202**

**题意：**给出一个数,不断循环计算它的各位的平方之和,问最后能否出现 1

**思路：**

(1) 模拟

**注意：**

(1) 无

**题目：203**

**题意：**从链表中删除值为 `val` 的

**思路：**

(1) 模拟

**注意：**

(1) 注意代码

**题目：204**

**题意：**求小于  $n$  有多少个素数

**思路：**

(1) 筛选法

**注意：**

(1) 无

**题目：205**

**题意：**判断两个字符串是否同构

**思路：**

·对于  $s$  标记每种字符对应  $t$  中的哪个，对于  $t$  标记每种字符对应  $t$  中的哪个，注意必须要两个都标记

**注意：**

(1) 无

**题目：209**

**题意：**求一个数组中最短的区间，使其和大于等于  $s$

**思路:**

(5) 遍历整个数组, 用两个指标记录现在的区间, 当大于等于  $s$  时, 左指标向右移动

**注意:**

- 无

**题目: 213**

**题意:** 198 的加强版。在圆上有一些点, 取其中一些点, 这些点不能相邻, 求最大值

**思路:**

(1) 转化成 198, 从第一个到倒数第二个, 从第一个到最后一个, 分求最大值, 再求最大值

**注意:**

(1) 无

**题目: 215**

**题意:** 求一个数组中的第  $k$  大元素

**思路:**

(1) 利用快排思想, 每次进行一轮快排, 返回关键字位置, 如果该位置右边有  $k-1$  个数, 则代表该位置的数就是第  $k$  大的数

**注意:**

(1) 注意快排的写法

**题目: 217**

**题意:** 给出一个数组, 判断其中有没有重复的元素

**思路:**

(1) 利用 hash, 时间复杂度为  $O(n)$ , 空间复杂度为  $O(n)$

(2) 先排序, 再比较, 时间复杂度  $O(n\log n)$

**注意:**

(1) 无

**题目: 219**

**题意:** 给出一个数组, 判断其中有没有相等的元素, 且它们的下标之差不大于  $k$

**思路:**

(1) 先排序, 再比较

(2) 用 hash

(3) 标记法

(4) 维护一个长度为  $k$  的二叉搜索树, 每次进行查找和删除, 时间复杂度为  $n\log(k)$ 。利用 `treemap` 的 `subSet` 函数



**注意：**

(1) 无

**题目：221**

**题意：**给出一个矩阵，求最大的正方形，使其中的每个元素为 1

**思路：**

(1)  $dp[i][j] = \min(dp[i-1][j-1], dp[i-1][j], dp[i][j-1])$

**注意：**

(1) 对于左、上边界，要特殊判断

**题目：222**

**题意：**求完全二叉树的节点个数

**思路：**

(1) 对于某个节点，如果一直左孩子的高度等于一直右孩子的高度，则说明是满二叉树；否则，就用递归， $1 + \text{左孩子节点数} + \text{右孩子节点数}$

**注意：**

(1) 无

**题目：223**

**题意：**求两个矩形围成的总面积（矩形的各边平行于坐标轴）

**思路：**

(1) 虽然可能会分许多情况，但是只需要两种：1，两个矩形不相交，有 x 和 y 轴共四个判断；2，将每种情况画出来（一共 3 种），会发现可以用一种方法解决

**注意：**

(1) 无

**题目：224**

**题意：**表达式求值，只有加减括号

**思路：**

(1) 如果没有括号，就计算，有括号就先算括号里的，再计算

**注意：**

(1) 建议再写一遍

**题目：225**

**题意：**使用队列模拟栈

**思路：**

(1) 入栈：进入一个空队列，然后把另一个队列移到这个队列；出栈：取出非空队列的首元素

(2) 使用一个队列，入栈：入队；出栈：除最后一个元素，从前往后挨着加入队列

**注意：**

(1) 无

**题目：**226

**题意：**翻转二叉树

**思路：**

(1) Dfs，翻转每个节点的左右孩子，然后继续 dfs

(2) 使用栈，每次取栈顶，翻转左右孩子，如果孩子非空，入栈

**注意：**

(1) 建议所有用递归方法解决的，使用非递归解决

**题目：**227

**题意：**表达式求值，只有加减乘除

**思路：**

(1) 一个栈保存数，另一个栈保存操作符

**注意：**

(1) 如果当前操作符大于栈内操作符，则入栈；否则计算

**题目：**228

**题意：**给出一个排序后的数组,输出连续数的区间

**思路：**

(1) 模拟

**注意：**

(2) 输入的数可能存在负数

(3) 负数变成正数的时候注意最小的负数

(4) 字符串翻转的时候，注意负数时不一样的

**题目：**229

**题意：**求一个数组中，至少出现  $n/k$  次的数（该题  $k$  为 3）

**思路：**

(1) 该数最多有  $k-1$  个，那么保存这  $k-1$  个数，如果和所有的不同，则全部减一，如果还没

有保存够  $k-1$  个数，那么就保存

注意：

(1) 求出来的每个数，最后代入看是否出现  $n/k$  次

(2)  $k-1$  个数不能相同

**题目：230**

**题意：**求搜索二叉树的第  $k$  小元素

**思路：**

(1) dfs，中序遍历，第  $k$  个，计数器在中间加

**注意：**

(1) 无

**题目：231**

**题意：**判断一个数是否为 2 的次方

**思路：**

(1)  $n > 0 \ \&\& \ ! (n \& (n - 1))$

(2)  $n$  中 1 的个数

**注意：**

(1) 注意  $n$  大于 0

**题目：232**

**题意：**使用栈模拟队列

**思路：**

(1) 进队，进入 B 栈，出队，从 A 栈出（A 栈为空，B 栈转移进来）

**注意：**

(1) 无

**题目：233**

**题意：**求 0- $n$  之间 1 出现的个数

**思路：**

(1) 对于每一位 1 贡献的次数

**注意：**

(1) 注意  $n$  小于 0 的情况

**题目：**234

**题意：**判断一个链表是否为回文

**思路：**

(1) 将链表中的数据全部拷贝出来，然后判断，空间复杂度为  $O(n)$

(2) 将链表分成两部分，然后翻转前半部分，然后判断，空间复杂度为  $O(1)$

**注意：**

(1) 注意逻辑，建议重写一次

**题目：**235

**题意：**在二叉搜索树中，求两个结点的公共祖先

**思路：**

(1) 由于是二叉搜索树，所以，对于每个节点，左边的都比它小，右边的都比它大。因此，从根节点开始，如果遇见两个点某一个，则就是该点；否则两个点都比该节点大或小，继续搜索；如果一个在左边一个在右边，则就是该点

**注意：**

(1) 无

**题目：**236

**题意：**在二叉树中，求两个结点的公共祖先

**思路：**

(1) Dfs 搜索，保存根节点到两个结点的路径，然后求路径的最后一个相同的

**注意：**

(1) 判断结点相等，不能用值判断，直接用指针判断

(2) 可以把两个 dfs 写在一起

**题目：**237

**题意：**删除一个单向链表的当前节点

**思路：**

(1) 由于不能将上个节点的 next 赋值为下个节点，所以不能删除当前节点。将下个节点的值赋值给当前节点，然后删除下一个节点就行了

**注意：**

(1) 无

**题目：**238

**题意：**给出一个数组，求一个数组，使得第  $i$  位的值为该数组所有元素的乘积除以该位的数

**思路：**

(1) 不能用除法，那么用一个数组保存从 0 到 n 位的乘以，然后再从最后一个元素倒回来，用一个变量保存倒回来的乘积，那么将对应的两个相乘即为答案

**注意：**

(1) 第一位数要特殊处理，无论是正着还是倒回

**题目：240**

**题意：**给出一个矩阵，每一行为升序，每一列为升序，问是否能找到目标数

**思路：**

(1) 对于每一列，如果目标数不小于第一个，不大于最后一个，则二分

(2) 从右上角开始，如果该数大于目标数，则该列不可能有，列号减一；如果该数小于目标数，该行不可能有，行号减一

**注意：**

(1) 无

**题目：242**

**题意：**判断一个字符串是否可以由另一个字符串组成

**思路：**

(1) 对两个字符串排序，看是否相等

(2) 计数器统计字符串的各字母出现个数

**注意：**

(1) 无

**题目：257**

**题意：**求一个二叉树的所有路径

**思路：**

(1) Dfs

**注意：**

(1) ->什么时候需要（当前面有数的时候）

(2) 注意负数，然后转化成字符串时，注意符号

**题目：258**

**题意：**不断对一个数求各个数位的和，最后小于 10 停止

**思路：**

(1) 模拟

**注意：**

(1) 如果有负数的话，要注意，该题没有负数

**题目：**260

**题意：**数组中有两个数出现了一次，而其它数出现了两次，求这两个数

**思路：**

- (1) 对所有的取异或运算，则结果  $sum$  就是那两个数  $a, b$  的异或。然后找到  $sum$  的为 1 的为，那么就代表  $a, b$  在该位一个为 0，一个为 1。那么将该数组分成两部分，一部分为该位为 1，另一个部分该位为 0，相同的数会被分在一组，再对两组分别取异或，则两个的结果就是  $a, b$

**注意：**

- (1) 它的推广，见 136

**题目：**263

**题意：**判断一个数是否为丑数

**思路：**

- (1) 判断是否只含 2, 3, 5 三个约数

**注意：**

- (1) 无

**题目：**264

**题意：**求第  $n$  个丑数

**思路：**

- (1) 用三个变量记录应该乘以 2, 3, 5 的位置，如果最小值等于第  $x$  个乘以 2，那么  $x++$ ，其它同理

**注意：**

- (1) 无

**题目：**268

**题意：**给出一列数，求出其中缺少的一个数

**思路：**

- (1) 找出最大的一个数，并且求出所有数的和  $sum$ ，如果 1 到最大数的和大于  $sum$ ，那么其差值就是缺少的那个数；否则就是最大的那个数加 1

**注意：**

- (1) 缺少 0 的情况  
(2) 不缺数的情况

题目：

题意：给出两个大小都为  $n$  的有序数组，求这两个数组的中位数

思路：

- (1) 求出第一个数组和第二个数组的中位数，如果相等，则就是该数；否则，假设数组 1 的大于数组 2 的，那么中位数存在与数组 1 的前半部分，和数组 2 的后半部分。再次进行计算。注意，为了保证两个区间始终相同，那么如果不等，则变为相等（长的区间变小）

代码：

```
int len1 = arr1.size(), len2 = arr2.size();
int L1 = 0, L2 = 0, R1 = len1 - 1, R2 = len2 - 1;
while(true)
{
    if(L1 == R1)
    {
        return min(arr1[L1], arr2[L2]);
    }
    int mid1 = (L1 + R1) >> 1;
    int mid2 = (L2 + R2) >> 1;
    if(arr1[mid1] == arr2[mid2])
        return arr1[mid1];
    if(arr1[mid1] > arr2[mid2])
    {
        R1 = mid1;
        L2 = mid2;
        if(R2 - L2 > R1 - L1)
            L2++;
    }
    else
    {
        L1 = mid1;
        R2 = mid2;
        if(R2 - L2 < R1 - L1)
            L1++;
    }
}
```

题意：给出两个有序数组，求第  $k$  小的数

思路：

- (1) 假如一个长度为  $m$ ，一个为  $n$ ，且  $(m \leq n)$ 。如果  $k$  小于等于  $m$ ，那么直接求两个数组的前  $k$  个的中位数。如果大于  $n$ ，那么可以缩小两个数组的查询范围，比如一个为  $L1$ ， $R1$ ，另一个为  $L2$ ， $R2$ 。虽然它们范围相同，但是不能直接求中位数，因为求出的中位数，是第  $k-1$  个（模拟可知），所以  $L1$ ， $L2$  特殊判断，然后都加 1，然后求中位数。如果在其之间，求出范围，会发现第二种多一个，所以特殊判断，然后  $L2$  加 1，再求中位数。

(2) 求中位数注意，看中位数是否为第  $k$  个

代码：

```
int findKthNum(vector<int> arr1, vector<int> arr2, int kth)
{
    int n = arr1.size(), m = arr2.size(), k = kth;
    if(k < 1 || k > m + n)
        return 0;
    int mi = min(n, m);
    int ma = max(n, m);
    if(k <= mi)
    {
        return Mid(arr1, arr2, 0, 0, k - 1, k - 1);
    }
    if(k > ma)
    {
        int L1 = k - ma - 1, R1 = mi - 1;
        int L2 = k - mi - 1, R2 = ma - 1;
        if(n > m)
        {
            if(arr1[L2] > arr2[mi - 1])
                return arr1[L2];
            if(arr2[L1] > arr1[ma - 1])
                return arr2[L1];
            L1++;
            L2++;
            return Mid(arr1, arr2, L2, L1, R2, R1);
        }
        if(arr1[L1] > arr2[ma - 1])
            return arr1[L1];
        if(arr2[L2] > arr1[mi - 1])
            return arr2[L2];
        L1++;
        L2++;
        return Mid(arr1, arr2, L1, L2, R1, R2);
    }
    int L1 = 0, R1 = mi - 1;
    int L2 = k - mi - 1, R2 = k - 1;
    if(n > m)
    {
        if(arr1[L2] > arr2[mi - 1])
            return arr1[L2];
        L2++;
        return Mid(arr1, arr2, L2, L1, R2, R1);
    }
}
```



```
    if(arr2[L2] > arr1[mi - 1])  
        return arr2[L2];  
    L2++;  
    return Mid(arr1, arr2, L1, L2, R1, R2);  
}
```