待写p9241，p9232字符串相关，p8624前缀和，p8749杨辉三角，规律

2024/2/26 p2580 字典树，示范案例通过了，洛谷上不通过

\*2024/3/19 p9241 飞机降落，深搜，先标记一个A，接着深搜剩下的；然后取消标记A，重复上述的（分割小问题，标记）

经验：判断n个是否可以放入盒子，有顺序要求或者放入了B要取出前面的B(用不了贪心)，用深搜

2024/3/19 p9242 接龙数列，删掉某个纯数字字符串使得字符串数列能接龙；线性dp

\*\*\*2024/3/19 p9234 买瓜，深搜爆搜，剪枝；删除

2024/3/17 p1352 上司排斥，树状dp，最大可能性。

2024/3/20 p9234 有限制的迷宫，常规bfs，用队列代替递归的dfs可以不用判断最少步数

\*\*\*2024/3/24 p8613 小盆友交换位置，排序， 树状数组

2024/3/26 p8669 水题，巩固基础，判断奇偶来简化，在哪里需要mod（所有可能爆的地方）

2024/3/26 p8625 生命之树，树形dp，没有限制根节点，需要指定跟节点，深搜子节点

2024/3/27 p9421水题，利用桶

2024/3/27 p8781 水题花盆裁剪mid=(n%2)?(n>>1)+1:(n>>1)；对称位置(n-i)----(i-1)

2024/3/27 p8748 水题longlong最大存9\*10(18)，1秒=1000毫秒

2024/3/27 p8649 k倍区间，计算前缀和。注意从0开始还是从1开始循环；用前缀和取余数法省时间；注意是否为负数；注意ans开long long;

\*\*\*2024/3/27 p8624 前缀和，实质是si和si-1的调换，使得最大的（si-si-1）的值尽可能最小

2024/3/29 p8683后缀表达式，默认有括号

2024/3/30 p8667 三个升序数组abc的排序比较，计算可能性，用中间的数组b作为循环，双指针指向两边的数组，使用(n-cntc)的方式获得比b[i]大的结果的数量

运行前：return ans，自定义的函数的参数；

刷题记录：

2023/5/16 Leetcode55，做了取巧的，回头得看贪心

2023/5/17 Leetcode49 看了题解：sort（string.begin(),string.end()）vector.emplace\_back()

迭代器 auto it=map.begin(),;it!=map.end();it++;

It->second

Leetcode11 思路正确，有不足，没考虑到筛选高的一边来优化

2023/5/19 Leetcode230 直接中序遍历存储，按k输出(可)

Leetcode 199 层序遍历，queue.front();

2023/5/20 Leetcode 54 螺旋矩阵, 双层循环模拟 不用递归

Leetcode 2 两数相加,关键是最后一次进位,次要的还要指针初始化

今天学了markdown **[https://orcid.org/0009-0000-8653-6014](https://orcid.org/0009-0000-8653-6014" \t "_blank)**

2023/5/21 Leetcode 24 两两交换 指针past，cur，head。先创造答案节点，让past为答案节点，保持状态head和cur是先后要交换的节点，past在交换后指向head，在让head和cur前往下一个状态。在前往下一个状态是判断下面是否有两个节点，没有就退出。这样while可以一直循环，不用判断。

2023/5/23 Leetcode15 三数之和，双指针。第一个大于零时可以返回。先存一个，在跳过重复值。根据和值的大小（最后都要算）来判断左指针还是右指针移动。注意每次移动左右指针时要重新判断。后面部分有些缺漏，需要补。Num[left++]，这里应该改为left+1，不然指针移动。

2023/5/25 Leetcode56 合并区间，先排序，再比较区间是否能合并，注意相等情况

Leetcode189复习。1.自己方法，指针存储位置，cur存储需要转移的值，遇到size=k\*2会有问题。不懂，还得回头看

2023/5/31 Leetcode46 全排列，回溯模板：

void backtrack(vector<vector<int>> &ans,int first,int length,vector<int> &nums){

if(first==length){

ans.push\_back(nums);

return;

} //返回条件

for(int i=first;i<length;++i){

swap(nums[i],nums[first]);

backtrack(ans,first+1,length,nums);

swap(nums[i],nums[first]);

} //考虑的条件

}

Leetcode78 子集，回溯，考虑的条件：有和没用两种

Leetcode198 动态规划初学，直接看题解

Leetcode279完全背包或动态规划，看题解

Leetcode736划分字母区间，新技巧，

Leetcode300 动态规划，看题解技巧

Leetcode139 划分字母，拆分状态，找状态方程(核心)，dp初始条件和结束（次要）

Leetcode322 零钱 使用leetcode139的方法会超时，

Leetcode62 不同路径，二维动态规划

Leetcode64 路径和最小，二维动态规划，注意边界细节

23/9/23 Leetcode841钥匙和房间，深度优先搜索

Leetcode547“省份”，深度优先搜索，自己写出来

Leetcode 1466重新规划路线，不懂

23/9/24 Leetcode 994腐烂橘子，bfs复盘，队列一轮一轮的思路

23/9/26 22蓝桥2659 和不超过k的小矩阵，前缀和指针，自己的思路只有7分

\*23/10/13 洛谷蓝桥，（bfs记录走过的点，防止重复）（前缀和简化bfs5\*5体积移动问题）（二维前缀和：b[i][j]=b[i][j-1]+b[i-1][j]-b[i-1][j-1]+a[i][j]）

23/10/28 洛谷蓝桥9420 冶炼金属，

23/11/8 p8661蓝桥 结构体暴力排序，每个节点重新计算

\*23/11/9 p8777 蓝桥 极角排序叉乘法，求象限

23/11/10 p8650 蓝桥 递归分析正则表达式，while(cin>>c)

23/11/10 p8687用M位二进制整数表示状态，（1<<n）来设置；动态规划dp[i]=min(dp[i],dp[...]+1)；位运算符|，

\*23/11/10 p8756国际象棋 动态规划状态压缩plus，判断二进制数中1的个数；\*没搞懂转移方程

\*23/11/19 p9736 异或和经典题

二叉查找树：二叉平衡树，红黑树

平衡树：不平衡第一种（单旋转LL）

有TreeNode\* root,k1,k2;

1,root==k1;

2.k2->left=k1->right;

3.k1->right=k2;

第二种（双旋转LR）

重复第一种两次

第三种RR，第四种RL类似

红黑树

核心性质

最长子树（最多）是最短的两倍，左子树黑节点个数等于右子树黑节点个数

红节点没有个数限制，红节点父节点不能为红节点。

插入新节点（默认插入在一颗确定是红黑树的树上）：