# 构造，交互，通信

朱屹帆

洛谷

www.luogu.com.cn

# 自我介绍

长郡中学 朱屹帆（luogu uid 229378）。

noi2023 金牌， noi2024 金牌。

QQ：2457378930，欢迎大家交流学习。

# 前言

构造，交互，通信这三类题型作为非传统题的代表，在国内的正式比赛中出现次数越来越多，比如 NOIP2022 T2， WC2022 T3 和 NOI2024 d1t2 。

事实上这三者有很多方法和技巧是有共同点的，可以通过对一个题目的模型的总结得到一类题目的模型的处理技巧，这里将会简单介绍一些模型的技巧。

这一类题需要选手从多个角度进行思考,发挥想象力和创造力。

# 抽屉原理

抽屉原理，也称为鸽巢原理。

有 n+1 个物品和 n 个盒子，对于每一个物品，将其选择装入其中一个盒子中，那么至少会有一个盒子内的物品的数量大于等于 2 。

证明使用反证法即可。

# 抽屉原理

摩尔投票法

绝对众数：在可重集合中出现次数严格大于一半的元素。

给定一个长度为 $n$ 的序列 $a$，计算该序列的绝对众数，保证存在。

# 抽屉原理

CF643G Choosing Ads

给定一个长度为 n 的序列和整数 p， 有 m 次操作，操作有两种类型，支持 区间赋值 和 询问区间内出现次数至少占 p% 的数 。

对于每组询问，输出答案时，可以包含不满足条件的数，可以重复输出，但要求满足条件的数一定要在答案中，并且输出的数的个数小于等于 100/p （下取整）。

# 抽屉原理

将摩尔投票法进行扩展，定义 $q = 100/p$（下取整）$+1$， 每次选择互不相同的大小为 $q$ 的集合进行删除，根据抽屉原理，出现次数至少占 $p\%$ 的数一定不会被完全删除掉，因此会被保留到答案中。

使用线段树维护即可。

# 抽屉原理

CF1198C Matching vs Independent Set

给定一个 3n 个点和 m 条边的无向图，找到一个大小为 n 的点独立集或边独立集。

# 抽屉原理

依次考虑 m 条边，初始 3n 个节点均没有标记，对于边 (u , v) ，当点 u 和点 v 均没有标记时，将这条边加入边独立集中，并标记点 u 和点 v 。

当边独立集大小小于等于 n 时，可以找到一个合法的边独立集，否则考虑所有没有标记的节点构成的集合，集合内部显然是没有边的，并且根据抽屉原理，集合大小小于等于 n ，可以找到一个合法的点独立集。

# 归纳法

考虑解的充分必要条件，将问题转化到规模更小的子问题上。

# 归纳法

CF1470D Strange Housing

给定一个 $n$ 个点 $m$ 条边的无向图，将其中一些点进行标记使得满足不存在一条边的两个端点都被标记且仅保留有一端有标记的边之后原图联通，构造一组合法标记方案，或判断无解。

# 归纳法

首先考虑原图是二分图的情况，将其中一侧所有点进行标记即可。

将标记点放在左侧，未标记点放在右侧，则左侧内的点之间没有边，而将一个点加入到左侧点集的条件是其邻域中没有已经被加入到左侧点集中的点，此时会导致一些点被加入到右侧点集，而加入到右侧点集的点可以将其邻域中所有未被加入点集的点尽可能加入到左侧点集，这样形成一个类似 dfs 的过程，容易发现这样可以使一个原本联通的图在染色之后满足条件。

# 归纳法

agc031c Differ by 1 Bit

给定 n , a , b ， 尝试构造一个长度为 2^n 的值域在 [0， 2^{n} -1] 的排列满足任意相邻两项在二进制下恰好一位不同且排列的第一项为 a ， 最后一项为 b ， 或判断无解。

# 归纳法

解的充分必要条件是当 n>0 时 a 和 b 在二进制下的 1 的个数的奇偶性不同，当 n=0 时 a=b=0 。

首先考虑 n=1 时显然成立，然后考虑 n>1 时首先可以将整个排列全部异或上 a 使得排列第一项为 0 ，然后显然 b 不为 0 ，选择其中的一位作为 b 的最高位，然后在正中间选择一个合适的数满足其满足 n-1 状态下的条件，中间在最高位上进行衔接即可。

# 归纳法

arc122e Increasing LCMs

给定一个长度为 n 的序列 a， 尝试对序列 a 进行重排使得其前缀 lcm 序列 b[i] = lcm{ a[1] , a[2] , ⋯ , a[i-1]， a[i] } 单调递增，或判断无解。

# 归纳法

考虑倒过来做，每次尝试对序列 a 的最后一项选择其中一个合法的 a 中元素，然后一直重复这个过程即可。

证明可以考虑如果 n 的问题有解，那么根据 n 的解可以推导出 n-1 的解，因此做法正确。

# 贪心与动态规划

可以对策略进行贪心来构造方案，使用动态规划来维护有解的状态并进行转移，运用其相关知识来分析。

# 贪心与动态规划

abc166f Three Variables Game

有三堆石子和一个长度为 n 的操作序列，每次操作给出三堆中的其中两堆的编号，要求在这两堆中选择其中一堆的一个石子并移动到另一堆中，尝试构造一组合法的选择方案，或判断无解。

# 贪心与动态规划

当石子总数分别为 0 或 1 或 2 时，可以直接用动态规划来构造一组合法的选择方案。

当石子总数 >2 时，当至少有两堆石子均不为空时，每次操作将其中较大的一堆石子移动到较小的一堆石子，可以发现这样可以保证任意时刻均存在至少两堆石子不为空。

# 经典算法

一些经典的排序算法，经典的分治算法，经典的分块算法，都可以进行运用。

# 经典算法

arc154d A+B>C?

交互题，给定 n，有一个长度为 n 的排列 p，可以向交互库询问 i , j , k，交互库会判断 p[i] + p[j] > p[k] 是否成立，其中 i , j , k 可以相同，复原排列 p 。

# 经典算法

令 p[x] = 1， 则对于所有 k 不等于 x， 均有 p[x] + p[x] > p[k] 的返回值为 0， 因此可以找到 x 。

确定 x 之后， 由于 p 是一个排列， 因此 p[i] + 1 > p[j] 的返回值相当于 p[i] > p[j] 的返回值， 使用归并排序即可。

# 转化图论关系

将问题的模型转化成图论关系，建立出图论模型，再用常见的图论算法来处理。

# 转化图论关系

CF1270G Subset with Zero Sum

给定一个长度为 n 的序列 a， 满足 i-n <= a[i] <= i-1， 选择一个非空子集使得子集内的元素和为 0， 并构造一组合法方案。

# 转化图论关系

如果选择的子集为 s， 则 \sum_{x \in s} a[x] = 0， 注意到限制范围和 i 相关，不妨令 a[i] 变成 i-a[i]， 则 \sum_{x \in s} (a[x] - x) = 0 。

对 (x , a[x]) 连双向边，由于 n 个点有 n 条边，且每个点均有边连接，因此图中一定存在环，而环一定是一组合法的方案，因此找到一个环即可。

# 转化图论关系

CF1404D Game of Pairs

交互题，给定 1 , ⋯ , 2n ， A 和 B 进行游戏， A 将这 2n 个数分成恰好 n 组，每个组恰好两个数，然后 B 在每个组中选择恰好一个数，若选出的数的权值和为 2n 的倍数则 B 获胜，否则 A 获胜。

选择扮演其中的一方，并取得胜利。

# 转化图论关系

当 n 为偶数时考虑将 (i , n+i) 设为一组，此时无论 B 怎么选，选出的数的权值和一定为 n(n+1)/2 + kn ，可以发现其一定不为 2n 的倍数，此时先手必胜。

当 n 为奇数时考虑沿用上面的思路，尝试构造使得 i 和 n+i 恰好只能选择其中一个，因此可以对于同一组的两个数进行连边，并对 i 和 n+i 之间连边，有连边表示两个端点中应当恰好选择一个，由于每个点的度数恰好为 2 ，因此会恰好构成若干个偶环，注意到此时所有数的和为 2n(2n+1)/2 ，因此该二分图存在一侧的点的对应数的权值和满足条件。

# 转化图论关系

欧拉回路

CF547D Mike and Fish

给定平面上的 n 个整点，要求给每个点染成红色或蓝色，使得对于同一水平线或垂直线上的点的两种颜色数量相差至多为 1，构造一组合法的染色方案。

# 转化图论关系

对每一行和每一列均建立点，对于一个点 (x[i] , y[i])， 将行 x[i] 和列 y[i] 的对应点连双向边。

建立虚拟点，当一个点度数为奇数时将其与虚拟点连边，此时每个点度数均为偶数，构造欧拉回路，在欧拉回路中若一条边从行向列，则设置为红色，否则设置为蓝色。

# 转化图论关系

*agc018f Two Trees

给定两棵树 A 和 B， 节点均从 1 到 n 编号，给每个编号设置一个权值使得这两棵树上均满足任意节点的子树权值和为 1 或 -1， 构造一组合法方案，或判断无解。

# 转化图论关系

当树上有相同编号的节点的儿子数量不同时无解，否则可以证明，每个点权值为 -1 或 0 或 1 时可以构造一组合法方案。

首先建立 2n 个点，点 1 到点 n 有树 A 上的边，点 n+1 到点 2n 有树 B 上的边，然后若编号 x 有偶数个儿子则将其在树 A 和树 B 上对应节点连边，两棵树的根需要向虚拟点连边，然后跑欧拉回路，根据对应编号在树 A 和树 B 之间的边的走向决定其权值。

# 转化图论关系

差分约束

*P4926 倍杀测量者

本题中定义选手 x 有 k (k>0) 倍杀选手 y 为 c[x] >= k*c[y] 。

有 n 名选手，给定若干条限制，设定一个正数常数 T ，此时限制 (x , y , k) 有两种类型，分别为选手 x 有 k-T 倍杀选手 y ，或选手 y 没有 k+T 倍杀选手 x ，其中有 m 名选手的分数固定。

找到一个最大的常数 T 使得无论如何设置选手分数，一定存在至少一个限制被满足，或判断无解。

# 转化图论关系

将分数取 log， 此时"倍杀"由乘法转化为加法，使用差分约束，当存在负环时说明一定存在至少一个限制被满足。

差分约束中形如 x+c >= y 的信息，可以理解为变量 x 给变量 y 设置了一个 x+c 的上界，而变量 y 需要取的是所有上界中最小的上界，使用最短路维护，反之使用最长路维护。

# 最优化交互

可以先分析交互库能够提供的有效信息，然后可以尝试用一些状态来描述当前已有的信息，并分析交互库提供的信息将会给当前状态带来的变化，可以理解成在此基础上进行的状态的转移，而最优化即要求得到当前状态的最大代价最小，可以考虑使用一些策略或动态规划来进行最优化的过程。

在一些最优化交互的题目中，有时可以充分发扬人类智慧，缩小动态规划或者策略考虑的范围，从而提升效率。

# 最优化交互

uoj52 元旦激光炮

交互题，有长度分别为 n1 , n2 , n3 的三个单调不降序列 a , b , c ，每次可以询问其中一个序列的其中一个位置的值，计算三个序列归并后的第 k 小的值。

# 最优化交互

归并后的第 k 小值可以写为 min_{k1+k2+k3=k} { max(a[k1] , b[k2] , c[k3]) }， 可以对 k1 和 k2 进行二分，注意到三个变量中取到 max 的变量可以缩小右边界，而取到 min 的变量可以缩小左边界。

也可以类似倍增的增大 k1 , k2 和 k3， 取到 min 的变量进行增大。

# 最优化交互

CF1746E2 Joking (Hard Version)

交互题，给定整数 n ，有一个整数 x \in [1 , n] ，可以向交互库询问一个集合 s ，交互库会返回整数 x 是否在 s 中出现过。

交互库可能会返回错误的信息，但是相邻连续两次询问中交互库至少会返回一次正确的信息，有两次提交整数 x 的机会，其中一次正确即可。

# 最优化交互

当 x 的待选集合大小 >= 4 时，将其划分为 a,b,c,d 四份，两次分别查询 a,b 和 a,c ，此时每次询问均可以转化为 x 在查询集合或者查询集合的补集中，由于连续两次之间必然存在一次真话，因此在这两个部分中均未出现的集合即可以排除，进行到集合大小 <=3 时搜索出一组合法的操作即可。

# 最优化交互

考虑上述算法分析的本质，注意到我们可以用两个状态的集合来描述待选集合，即 a 为在上一次询问中符合的所有元素， b 为在上一次询问中不符合但是仍然在 x 待选集合中的所有元素，则当查询 a 中元素时如果不符合这次询问则归属到集合 b' 中，否则归属到集合 a' 中，当查询 b 中元素时如果不符合这次询问则可以排除，否则归属到集合 a' 中。因此每次可以将当前状态归属于这样两个集合来描述。

观察上述过程，发挥人类智慧，可以发现每次的最优查询策略大概是将集合 a 和集合 b 均采用分成恰好两半然后进行查询，当 n <= 500 时可以动态规划预处理一下对于 |a| 和 |b| 采用的最优的划分方案和划分点是多少，当 n > 500 时直接分成恰好两半即可。

# 最优化交互

*P10786 [NOI2024] 百万富翁

# 二进制分组及其拓展

CF1365G Secure Password

交互题，有一个长度为 n 的序列 a ，定义序列 b ，其中 b[i] 表示序列 a 除去 a[i] 后的二进制或的结果，每次可以向交互库询问一个集合 s 的二进制或的结果，计算序列 b 。

# 二进制分组及其拓展

每一个位置 x 可以用二进制来描述，维护出每一个下标在某一位为 0 或 1 的所有数的或的和，询问次数为 2 log n 。

考虑给每一个位置重新设置一个编号，不妨让所有编号的 popcount 都相同，这样任意两个不同编号之间一定可以找到 (0 , 1) 和 (1 , 0) 的不同的位，因此询问次数为 c ( {c \choose (c/2)} >= n) 。

# 二进制分组及其拓展

思考：上述做法的本质是什么？

# 随机化

CF1562F Tubular Bells

交互题，有一个长度为 n 的序列 a ，序列中的数互不相同，且存在 0 <= l <= 2*10^5 - n 使得 a[i] \in [l+1 , l+n] 。

每次可以向交互库询问 x , y ，交互库将返回 lcm(a[x] , a[y]) ，计算序列 a 。

# 随机化

当 n 比较小的时候可以将每一对 x , y 的 lcm 都算出来，先找到最大的 n+l-1 和 n+l，然后再从大往小依次计算。

当 n 比较大的时候，当能够找到一个数 x 使得 x 是质数且 2x > l+n 时，其他位置上的数就可以通过一次查询得到了，打表之后发现这样的数的密度是比较高的，查询一个位置的数可以将其与多个位置进行询问，将所有询问的结果取 gcd，最终的结果大概率是这个位置的数。

# 随机化

uoj454 打雪仗

通信题，小 A 有一个长度为 2n 的 01 串，小 B 有 1 到 2n 中的恰好 n 个位置，小 B 希望知道这些位置在小 A 的字符串中分别是多少，通信代价为双方传输信息的较大值。

# 随机化

考虑这样一个过程，小 A 传输给小 B 当前位的信息，然后小 B 传输给小 A 表示下一个需要传输的位是当前位的后一位还是后两位，此时可以发现代价最大为 1.5 n。

将 01 串的每一位的位置随机打乱，再做上面这个过程即可。

# 谢谢大家

祝大家学业有成