

# 杂题选讲

XJTU

Information and Computational Science

mg

[xjtumg.me](http://xjtumg.me)

[xjtumg1007@gmail.com](mailto:xjtumg1007@gmail.com)

- Codeforces 17C Balance

- 给定一个长度为 $n$ 的只含字母'a', 'b', 'c'的串，可以进行两种操作
- 选择相邻2个字符，把第二个字符替换成第一个
- 选择相邻2个字符，把第一个字符替换成第二个
- 问这个串进行任意多次变换后，能得到多少不同的串且满足三个字母出现次数两两差的绝对值不超过1

- $n \leq 150$

- 考虑所有可以被得到串的性质：
- 一段连续相同的字母必定对应原串中的某个字符
- 原串的某个字符必定对应可以得到的串中一段连续相同的字母
- 原串中两个字符的位置关系在新串中依然一致

- DP

- $f[l][a][b][c] \Rightarrow f[\text{next}[l][a]][a + 1][b][c]$

- $f[l][a][b][c] \Rightarrow f[\text{next}[l][b]][a][b + 1][c]$

- $f[l][a][b][c] \Rightarrow f[\text{next}[l][c]][a][b][c + 1]$

- $f[l][a][b][c] \Rightarrow \text{ans} \ (a + b + c == n \ \&\& \text{两两差的绝对值不大于1})$

- $O(N^4)$

- Codeforces 19E Fiary
- 给定 $n$ 个点 $m$ 条边的无向图，可以任意删除一条边，要求删边后的图是二分图
- 求可以删哪些边
- $n, m \leq 1e4$

- 暴力枚举 + 卡常数
- $O(m * (n + m))$
- 更好的方案？

- 一个图是二分图的充要条件是这个图中没有奇环 (如何证明? )
- 建立DFS树，非树边一定为返祖边
- 如果不存在奇环，则删任一条边均可以

- 如果有奇环，考虑一条返祖边构成的环
- 如果这个环是奇环则必须从这个环上删除一条边
- 如果有两个以上奇环，则必须删除树边(why?)
- 且树边必须满足：
  - 同时属于所有奇环
  - 不属于任何偶环



- DFS可以轻松统计每条树边属于多少奇环与偶环
- $O(n + m)$

- Codeforces 28D Do not fear, DravDe is kind
- 给定长度为 $n$ 的四元组序列 $(v_i, c_i, l_i, r_i)$
- 要求选出一个子序列，满足
- 子序列中所有的四元组 $c_i + l_i + r_i$ 均相等
- 第一个元素的 $l_i = 0$ , 最后一个元素的 $r_i = 0$
- 第 $i$ 个元素的 $l_i$ 等于前 $i - 1$ 个元素的 $c_i$ 之和
- 最大化 $v_i$ 的和
- $n \leq 1e5$

- 按照  $c_i + l_i + r_i$  归类
- $dp[i] = \max\{dp[j] \mid j < i \ \& \ l_i = l_j + c_i\} + v_i$
- 利用map维护
- $O(N \log N)$

- Codeforces 39E What Has Dirichlet Got to Do with That?
- 给定 $a, b, n$ ，两人轮流博弈操作，每次可以选择 $a++$ 或者 $b++$ ，如果某人操作完之后 $a^b \geq n$ 那么此人输了
- 问先手必胜、必败还是必和
- $1 \leq a \leq 1e4, 1 \leq b \leq 30, n \leq 1e9$

- 若 $a, b$ 均大于1,  $a^b < n$ 的二元组不多
- DP, 用map记忆化即可
- 若 $a = 1$ 可能出现平局, 即当 $2^b \geq n$ 时
- 若 $b = 1$ 时,  $a$ 可能会很大导致TLE
- 注意到当 $a^2 \geq n$ 时, 可以直接通过 $n - a$ 的奇偶性判断必胜必败
- 仍用map记忆化即可

- Codeforces 40E Number Table
- 给定 $n*m$ 矩阵，每个元素只能是1或-1，已经有严格小于 $\max(n, m)$ 个元素被确定了
- 问最少有多少种方案填充剩下的元素，使得满足任意一行或一列里所有元素的乘积都是-1
- $n, m \leq 1000$

- 注意因为最多只填了  $\max(n, m) - 1$  个格子
- 一定存在空行或空列
- 假设为空行，则不管其他格子如何填，最后填这一行总可以使得每列的乘积为1，且方案唯一
- 只需要关心对于某一行，有多少种方案使得这一行的乘积为-1
- 二维  $\rightarrow$  一维
- $O(n * m)$

- 给定一个 $n*m$ 的矩形，初始时所有元素都为0
- 给定 $q$ 个操作
- 将第 $x$ 行所有元素异或1
- 将第 $y$ 列所有元素异或1
- 输出一个矩形区域内1的个数
- $n, m, q \leq 1e6$



- 两个树状数组维护行列
- 因为始终是对整行整列操作，故矩形区域内1的位置仍然是“矩形”
- 利用树状数组求出矩形区域内被操作奇数次的有多少行多少列即可
- $O(q \log nm)$

- Codeforces 201E Thoroughly Bureaucratic Organization
- 有一个 $1 \sim n$ 的排列A，目标是通过尽量少次询问得到A
- 每一次询问你可以给出一个长度为 $k$  ( $0 < k \leq m$ )的序列B，满足 $1 \leq B_i \leq n$ 且B中没有相同元素
- 系统会根据序列B生成一个长度为 $k$ 的序列C，满足 $A[C_i] == B_i$ ，然后系统会将序列C随机打乱后返回给你
- $n, m \leq 1e9$
- 1000组数据

- 答案显然满足二分性质
- 只需要求出对于一个答案 $s$ (询问次数)，它最长能判定出的长度是多少

- 因为C被随机洗过了，所以如果两个数字在S个询问中出现情况完全一致，那么必然不可行
- 于是对于一个n，我们要选出长度为S的01字符串，使得每个字符串两两不同，且竖着看每一列1的个数不超过m
- 结论：存在一个方案的充分必要条件是所有1的个数不超过 $S*m$
- 贪心的选择1出现次数尽量少的字符串S即可

- Codeforces 176E Archaeology
- 给定一棵 $n$ 个点的树(带边权)，每个点都是黑色或者白色，最初所有点都是白色，要求支持两种操作
- 把一个点反色
- 用最少的边把所有的黑点都连通起来的树的边权和
- $n, Q \leq 1e5$

- 设树上有k个黑点，按dfs序排序，不妨设为 $v_1, v_2, \dots, v_k$
- 边权和 =  $\frac{1}{2} * (\text{path}(v_1, v_2) + \text{path}(v_2, v_3) + \dots + \text{path}(v_{k-1}, v_k) + \text{path}(v_k, v_1))$
- set维护即可
- $O(Q \log n)$

- Codeforces 167E Wizards and Bets
- 给定一个 $n$ 个点 $m$ 条边的有向无环图，入度为0的点视为源点，出度为0的点视为汇点。保证源点汇点个数相同。
- 现在考虑用两两不相交的路径将源汇配对
- 对于某一方案，若将源点从小到大排序后对应的汇点的逆序对个数是奇数，则此方案贡献为-1，否则为1
- 求所有方案的贡献和
- $n \leq 600$

- 考虑如果没有“两两不相交”的条件
- 令 $A[i][j]$ 为第 $i$ 个入度为0的点到第 $j$ 个出度为0的点的路径数量
- 则行列式 $A$ 的值即为答案



- 考虑两条相交的路径
- 若把相交点后面的路径对调，则会对应到另一种相交的方案
- 贡献抵消
- 故行列式 $A$ 的值即为答案

- Codeforces 125E MST Company
- 给定一个 $n$ 个点 $m$ 条边的无向带权图
- 求点1度数为 $K$ 的最小生成树
- $n \leq 5000$
- $m \leq 100000$

- 二分一个delta
- 将点1相连的边权全部+delta
- 求最小生成树判断点1度数与K的关系
- 随着delta的增大点1的度数单调减少

- Codeforces 191D Metro Scheme
- 给定一个 $n$ 个点 $m$ 条边的仙人掌图，即每个点每条边均至多属于一个简单环
- 要求用最少的简单链和简单环去覆盖所有的边，每条边均仅能覆盖一次
- 求最少的简单链和简单环的数量和
- $n \leq 1e5$
- $m \leq 3 * 1e5$

- 如果是树

- 简单贪心

- 先做dfs树
- 因为每个点至多只在某一个环内，我们可以分开考虑每一个环
- 先将所有简单环从图中删掉(点保留，仅删除边)，然后对森林贪心求答案

- 对于某一个简单环
- 若其深度最低的点不是根节点，则只要环上存在一个点的孩子有一个(孩子不包含环上的点)，则必定有方案使得答案不会变差
- 若其深度最低的点是根节点，则只要环上存在两个点的孩子有一个(孩子不包含环上的点)，则必定有方案使得答案不会变差
- 否则必须用一个简单环覆盖