2024“钉耙编程”中国大学生算法设计超级联赛（5）-题解

2024 年 8 月 2 日

目录

Problem 1　数表（二）

为什么出题人给的是一行 n 个格子，另一行 $n+1$ 个？
本题一个重要性质在于 $2n+1$ 是奇数，那么能导出一个很强的**性质**：如果不考虑第三条异或和为 0 的限制，构造出任意一组满足条件 1 和 2，而异或和为 x 状态，此时如果给每个元素异或上某数 y，条件 1、2 依然保持，而异或和变为 $x \oplus y$，就得到了一个满足条件 1、2，而异或和恰是 $x \oplus y$ 的状态。
因此容易说明，对所有满足条件 1、2 的状态按照异或和分成 2^k 类，任意两类之间存在**一一对应**关系，即每类的方案数相同。因此，可以直接不管异或和 $=0$ 的限制考虑，最终答案除以 2^k 即可！
接下来我们仅考虑每行每列的限制，我们可以先填好第二行，方案数是 $2^k \times (2^k-1) \times \cdots \times (2^k-n) = \frac{(2^k)!}{(2^k-n-1)!}$，对于第一行，我们考虑容斥：假设集合 U 为 $\{1,2,\ldots,n\}$，S 为 U 的子集。$f(S)$ 为仅 S 中元素与第二行相等的方案数；$g(S)$ 为 S 与第二行相等的方案数，则

$$f(S) = \sum_{S \subseteq T} (-1)^{|T|-|S|} g(T)$$

那么第一行的方案数为：

$$f(\emptyset) = \sum_{S \subseteq U} (-1)^{|S|} g(S) = \sum_{i=0}^{n} (-1)^i \binom{n}{i} \frac{(2^k-i)!}{(2^k-n+1)!}$$

$O(n)$ 预处理需要用到 $O(n)$ 个阶乘及其逆元，可以 $O(n)$ 计算答案。
[bonus] 如果两行都是 n 个格子呢？

Problem 2　Array Gift

操作次数的下界是 $n-1$；如果序列 a 中存在一个数为 1 ，那么对任意的 a_i 都有 $a_i \bmod 1 = 0$ ，而为了凑出一个 1，总是通过 $+x$ 和对另一数取模共 2 次操作得到，故操作次数的上界是 $n+1$. 故只要讨论清楚这三种情况。

- 对每个数要么是最后留下的那个，要么一定是要被操作的，因此次数是 $n-1$ 当且仅当存在某个数整除其他所有数，即 a 中最小值是 $\gcd_{i=1}^{n}(a_i)$ 的因子，不妨将 a 进行不降序的排序。
- 否则需要至少 n 次操作，考虑除了消掉一个数以外的这额外一次操作是什么，以及这一次额外操作发生在什么时候（有可能先消去某些数，再做这一次操作）：
- 如果是 $a_j \to a_j \bmod a_i = d$：（1）可能是 d 变成其他所有数的因子，我们断言，可以认为这个操作只会发生在最开始，因为这种情况 $d|a_i$ ，而 $a_j \bmod a_i = d$，那么 $d|a_j$，故 $a_j \to d$ 后能消去的数只会更多。（2）其余 $n-1$ 个数原本就能消掉 $n-2$ 个，除了 a_j 不行，假设原本 $n-1$ 个数的最小值是 d，那么操作 $a_j \to a_i = kd$ ，而 a_i 也要是 d 的倍数，故 a_j 也是 d 的倍数，故这种情况不需要考虑。

- 如果是 $a_i \to a_i + x$ ：（1）将 a_i 变大去删其他数，此时 a_i 一定是最小值。（2）让其他数的 gcd 整除 a_i.
- 否则，答案是 $n + 1$.

Problem 3 捆绑魔方

模拟 + 搜索

模拟部分：

- ~~简单的模拟即可~~
- 需要选择一种方式储存当前魔方的状态（6 个面 × 每面 8 个小块，或 12 棱块 × 2 种方向 +8 角块 × 3 种方向，等等）
- 需要手写每个面的旋转会对魔方状态的影响.
- 需要手动输入哪些面/块被捆绑，并在旋转前判断是否会影响捆绑关系
- 需要根据相邻颜色信息判断每个小面属于哪一个块

搜索部分：

- ~~简单的 bfs 即可，注意要输出最小字典序~~
- 在题面中有提到，捆绑魔方总共有 1108800 种状态，直接从终点（还原状态）开始反向搜索，记录下每个状态的还原路径。
- 但这样大概率会 TLE，由于 T 比较小，考虑使用双向搜索进行优化，先预处理搜索出距离终点一定范围内的所有状态。
- 对于每个询问，只需搜索到一个预处理过的中间状态 S，显然可以找到最短路径 (起点 $\to S \to$ 还原).
- ~~实际上这样跑的飞快，即使T 放大几十倍也能轻易通过。~~
- 由于题目还要求输出最小字典序，反向搜索时需要将相同距离的操作路径更新为其中字典序最小的一个，正向搜索则直接从小到大枚举即可。

Problem 4 树论（一）

首先注意到（具体见附录）$\sum_{i=1}^{n}\sum_{j=1}^{n}[lcm(i,j) \le n] = O(n \log^2 n)$，并且常数较小，可以尝试暴力枚举 lcm 的结果，处理出所有 $lcm(i,j) \le 10^5$ 的序对 i, j ，如果加上 $i \le j$ 的限制，只有 2×10^6 对左右。

将这些 $lcm(i,j) \le 10^5$ 的序对 (i,j) 根据其 lcm 的值排序，对于询问 r, x 离线：按照 x 从小到大排序，用双指针的技巧按顺序将 $lcm(i,j) \le x$ 的 (i,j) 的贡献打到 $LCA(i,j)$ 上，如果 $i = j$ 则贡献 +1，否则贡献 +2。查询则是子树查询，即 dfs 序上的区间查询。也就是单点修改，区间求和，用树状数组即可实现。

Problem 5　树论（二）

~~小清新数据结构~~

首先考虑枚举 gcd 的值 d，将所有 $O(n/d)$ 个 d 的倍数的点拿出来，这些点两两之间的路径的答案都至少是 d. 当然如果妄想直接建虚树对链做区间取 max 的操作肯定太暴力了会 TLE.

考虑从大到小枚举 gcd 的值，因为对每条边问的是最大值，因此一条边只有第一次覆盖到的时候需要计入答案，也就是一条边更新完就可以被“压缩”掉了，那么怎么实现这个“压缩”操作呢？那就是并查集!!

当更新完边 (u,v) 的答案时，假设 $dep_u > dep_v$，那就将并查集上 u 的父亲连接到 v，这样暴力在虚树上从叶子到根更新每条边，均摊下来每条边只会被跳一次，实现的好其实能通过本题。

如果仔细思考，其实建虚树是不需要的，因为这棵虚树上任意两点间的路径（如果没有在更早被覆盖到）迟早要被覆盖，以任意顺序合并 d 的那些倍数的点都是可以的，代码非常简洁：

```
for(int d=n/2;d>=1;--d){
    int x=find(d);
    for(int j=d+d;j<=n;j+=d){
        int y=find(j);
        while(x!=y){
            if(dep[x]>dep[y])swap(x,y);
            ans[idx[y]]=d;
            fa[y]=find(par[y]);
            y=find(par[y]);
        }
    }
}
```

时间复杂度就是枚举倍数，以及并查集（只有路径压缩）的均摊复杂度 $O(n\log n)$.

Problem 6　猫罐头游戏

- 最终局面显然是 $1,1,1$.
- 观察奇偶性，发现奇数、奇数、偶数和奇数、偶数、偶数的局面都能转移到奇数、奇数、奇数；而奇数、奇数、奇数的局面只能转移到奇数、奇数、偶数。
- 因此得到 a,b,c 都是奇数的情况下先手必输，有奇数也有偶数的情况先手必胜。
- 剩下 a,b,c 全是偶数的情况，通过类似的推理得到 $lowbit(a) = lowbit(b) = lowbit(c)$ 的情况下先手必输，其余情况先手必胜。

Problem 7　猫咪军团

我们称狗星的地区和狗星道路构成狗图（狗图是一棵树），和猫咪通道构成猫图。首先，如果猫图由多个连通块构成，那么每个连通块至少要有一只猫咪。分别考虑每个连通块，设某个连通块的点集为 U 。

- 若狗图中有某条边连接了 U 中的两个点，且猫图中不含有这条边，那么这个连通块需要 2 只猫咪，否则需要 1 只猫咪。（做完了）

必要性：当狗仔只在猫图的一个连通块的点（点集为 V ）里行动，不经过不属于这个连通块的点。

A. 若狗图中**没有**某条边 e 连接了 V 中的两个点，且猫图中不含有这条边:

- 可以当作猫咪也能走狗图中的边，因为狗图是一棵树，所以我们在这个连通块只需要 1 只猫咪走最短路去抓狗仔。

B. 若狗图中**存在**某条边 e 连接了 V 中的两个点，且猫图中不含有这条边:

- 那么如果这个连通块只有一只猫咪，那么狗仔可以在 e 连接的两个地区上反复横跳或者原地不动，所以这个连通块至少 2 只猫咪。当这个连通块有两只猫咪后，一只猫咪沿着狗图上的最短路追狗仔，当狗图上的边猫图上没有时，让另一只猫咪走猫图到达下一个点，这样就类似情况 A。

充分性：当我们分别对猫图的每个连通块考虑之后并且分别在每个连通块投放了 $1 \sim 2$ 只猫咪后，证明其充分性。

- 任取一只猫咪作为追捕狗仔的主猫，狗仔会被主猫限制在狗图的一个子树内，派遣一只猫咪（也可以是主猫）走到狗图上主猫到狗仔的最短路上的第一个地区，然后这只被派遣的猫咪作为主猫。
- 可以看出狗仔被限制在的子树越来越小，最终被抓住。

Problem 8　猫咪们狂欢

最大权值闭合子图模型：

- 1. 将所有猫先安排到树 1 上。当前狂欢值先统计进答案里。
- 2. 然后开 k 个点，第 i 个点表示” 狂欢猫 u_i 从树 1 换到树 2”，点权为在树 1 相邻有效树边的狂欢值之和的相反数（有效边相邻的两个点对应编号都是狂欢猫），意思是如果只将它从树 1 换到树 2 会减少这些狂欢值。
- 3. 若树 1 上有某条边连接 u_i , u_j 两点，则新建一个权值为该边的狂欢值的点，向 i ，j 各连一条边。因为如果 i，j 都被选中，这条树边的狂欢值会被重复扣除一次，所以新建这个点来补偿。
- 4. 若树 2 上有某条边连接 u_i , u_j 两点，则新建一个权值为该边的狂欢值的点，向 i，j 各连一条边。因为如果 i ，j 都换到树 2，则在树 2 上它们相邻，狂欢值增加。
- 5. 然后转为最小割即可。

Problem 9　用心感受（三）

对每一项只有 $\mu(k) \neq 0$ 时才需要考虑，这意味着 k 中无平方因子，即 k 一定是若干互不相同的素数乘积，不妨设 $k = \prod_{i=1}^{t} p_i$，此时 $\varphi(k) = \prod_{i=1}^{k}(p_i - 1)$.

对于素数 p，由费马小定理有 $a^p \equiv a(\mathrm{mod}p)$，那么：

- 若 $k|a$，则 $a^{\varphi(k)+1} \equiv 0^{\varphi(k)+1} \equiv 0 \equiv a(\mathrm{mod}k)$.
- 否则，a 在 $\mathrm{mod}p$ 下有逆元，$a^{p-1} \equiv 1(\mathrm{mod}p)$. 对任一 k 的素因子 p_j 考虑，$a^{p_j-1} \equiv 1(\mathrm{mod}p_j)$，两边取 $\prod_{i\neq j}(p_i - 1)$ 次幂，则 $a^{\varphi(k)} \equiv 1(\mathrm{mod}p_j)$. 对每个素因子考虑，可以得到 t 个这样的同余式，由于模数两两互质，可以用中国剩余定理合并同余方程组，得到 $\mathrm{mod}k$ 下的结果即 $a^{\varphi(k)} \equiv 1(\mathrm{mod}k)$

综上，如果 k 无平方因子，则 $a^{\varphi(k)+1} \equiv a(\mathrm{mod}k)$ 成立，因此问题就变成了求

$$\sum_{k=1}^{n} \mu(k) \cdot (a \bmod k) = a \cdot \sum_{k=1}^{n} \mu(k) - \sum_{k=1}^{n} \mu(k) \cdot k \cdot \lfloor \frac{a}{k} \rfloor$$

转换成莫比乌斯函数前缀和，可以用杜教筛解决。

Problem 10　世末农庄

本题有至少三种做法，这里只介绍在线做法中较为简单的一种。

首先简述做法：

- 对于操作一，进行树上倍增的预处理，但是是在线的，每次添加一个点就需要进行一次预处理；
- 对于操作二，进行倍增查询，跳至离根最近的，且还有剩余农作物的节点，获取它，直到背包装满；
- 对于操作三，将对应节点的剩余农作物量打上标记 -1，操作二倍增时当作这个节点还有剩余，而当父节点已经清空或者通过操作二找到该节点时，置为 0。

复杂度分析：

记询问数为 Q，节点总数为 n，显然 $n \leq Q$，操作二总数为 m，操作三总数为 p；

对于操作一，单次操作复杂度为 $O(\log n)$；

对于操作二，考虑不存在操作三的情况下，单次操作复杂度可能达到 $O(n \times \log n)$，但是实际上，对于所有的操作二，最多进行 $n + m$ 次查询，对于每个节点最多被操作二询问 m 次，故综合复杂度为 $O((n + m) \times \log n)$；

对于操作二，考虑存在操作三的情况下，最多使得操作二多做 p 次倍增查询该节点，并清空标记的动作，综合复杂度为 $O((n + m + p) \times \log n)$ 即 $O(Q \times \log n)$；

对于操作三，$O(1)$ 时间内完成。

特别地，对于 T 个测试用例，需要清空数组，此处由于 $T \leq 10$，无需动态开空间。

Problem 11　开关灯

首先答案肯定不是 2^n ——例如当 $n=2$ 时两盏灯只能同开/关，只有 2 种状态。~~特判$n=2$交一发，还是 wa 了~~

我会找规律： 2^n 是上限，写个爆搜会发现，如果 $n\equiv 2(\mathrm{mod}3)$ 时答案是 2^{n-1}，否则是 2^n ，快速幂计算即可，复杂度 $O(T\log n)$.

大聪明验题人的思路：注意到在 $1,2$ 两个位置进行操作，可以只翻转 3 号灯而不影响其他位置，以此类推，所有标号 $\equiv 0(\bmod 3)$ 的灯和所标号 $\equiv n-2(\bmod 3)$ 的灯都可以看成是可以任意操作，如果两个同余类不相交，则剩下一个同余类也可以任意操作，答案是 2^n；否则如果两个同余类是同一个同余类，即 $n\equiv 2(\mathrm{mod}3)$，可以把这些位置单独拎出来，剩下的位置可以看成是“在位置 i 操作会翻转 i 和 $i+1$ 的状态”，答案是 2^{n-1}.

线性代数爆算：

- 首先对于前 $n-1$ 盏灯总是可以调整到任何一个状态：从左往右考虑，如果第 i 盏灯（$1\le i<n$）不是我希望的状态，那么一定可以通过第 $i+1$ 盏灯来控制。
- 对于每个位置，只关心在这里操作了奇数次还是偶数次。

不妨设列向量 $x=(x_1,x_2,\ldots,x_n)^T\in\mathbb{Z}_2^n$ 表示每个位置操作次数的奇偶（x_i 为 1 表示奇数次，为 0 表示偶数次）。列向量 $b=(b_1,b_2,\ldots,b_n)^T\in\mathbb{Z}_2^n$ 表示要到达的状态（0 表示关，1 表示开）。那么原问题就转化为，问：对于多少个列向量 $b=(b_1,b_2,\ldots,b_n)$，如下方程有解:

$$\begin{bmatrix}1&1&0&0&\cdots&0&0&0\\1&1&1&0&\cdots&0&0&0\\0&1&1&1&\cdots&0&0&0\\\vdots&\vdots&\vdots&\vdots&\ddots&\vdots&\vdots&\vdots\\0&0&0&0&\cdots&1&1&1\\0&0&0&0&\cdots&0&1&1\end{bmatrix}\begin{bmatrix}x_1\\x_2\\\vdots\\x_n\end{bmatrix}=\begin{bmatrix}b_1\\b_2\\\vdots\\b_n\end{bmatrix}$$

记那个 n 阶方阵为 A_n，根据线性代数的知识，答案是 $2^{rank(A_n)}$，于是我们去算矩阵 A_n 的秩，而根据我们上面的分析，$rand(A_n)\ge n-1$，因此其实只需要判断 $rank(A_n)$ 是否为 n ，即判断 A_n 的行列式 $|A_n|$ 是否非零即可，对第一列做 Laplace 展开，再对第一行展开：

$$|A_n|=|A_{n-1}|-\begin{vmatrix}1&0&0&\cdots&0&0&0\\1&1&1&\cdots&0&0&0\\\vdots&\vdots&\vdots&\ddots&\vdots&\vdots&\vdots\\0&0&0&\cdots&1&1&1\\0&0&0&\cdots&0&1&1\end{vmatrix}=|A_{n-1}|-|A_{n-2}|$$

根据 $|A_1|=1$ 和 $|A_2|=0$ ，可以得到 $\{|A_n|\}$ 的前几项为：$1,0,-1,-1,0,1,1,0\ldots$

因为是二阶递推，所以根据前几项可以肯定 $|A_n|$ 有周期 6，故 $|A_n|=0$ 即 $rank(A_n)=n-1$ 当且仅当 $n\equiv 2(\mathrm{mod}3)$.

Problem 12　串串

考虑 n^2 的 dp 做法，设 $dp(l,r)$ 为操作子串 $s[l:r]$ 为空的最小花费，$dp(l,r)=\min(dp(l+1,r),dp(l,r-1))+occ(s[l:r])$

考虑 $occ(s[l:r])$ $n\times n$ 的矩阵的性质。

基本子串结构给出了如下性质：

- 1. 对于 occ 相等的连通块，长度最大的串仅有一个，由此可知每个连通块都只能是一个梯形结构
- 2. 把每个连通块长度最大的串称为这个连通块中元素的代表元，本质不同的代表元对应的连通块的周长是 $O(n)$ 的

根据这些性质，维护本质不同的代表元对应的连通块周长上的 dp 值便可正确转移

Problem 13　飞行棋

在 0 号格时有 $\frac{1}{n}$ 的概率直接进入终点。

对另外 $\frac{n-1}{n}$ 概率发生的情况，可以发现从 1 号格到 $n-1$ 号格花费 1 的代价进入终点的期望是相同的，都是 $\frac{1}{n-1}$，另外 $\frac{n-2}{n-1}$ 的概率继续停留在 $1\sim n-1$ 号格内，故在这个问题上，位于 $1\sim n-1$ 号格子上的状态都是等价的。

所以期望花费的代价为 $1+\frac{n-1}{n}\times(n-1)$.

Appendix A　对 Problem4 的补充

分析 $\sum_{i=1}^{n}\sum_{j=1}^{n}[lcm(i,j)\le n]$ 的上界：

$$S=\sum_{i=1}^{n}\sum_{j=1}^{n}[\frac{ij}{\gcd(i,j)}\le n]=\sum_{d=1}^{n}\sum_{i=1}^{n/d}\sum_{j=1}^{n/d}[\gcd(i,j)=1][\frac{id\times jd}{d}\le n]=\sum_{k=1}^{n}\mu(k)\sum_{d=1}^{n}\sum_{i=1}^{n/kd}\sum_{j=1}^{n/kd}[ijk^2d\le n]$$

$$\le n\sum_{k=1}^{n}\frac{\mu(k)}{k^2}\sum_{d=1}^{n}\frac{1}{d}\sum_{i=1}^{n/kd}\frac{1}{i}\le n\log^2 n\cdot\sum_{d=1}^{n}\frac{\mu(k)}{k^2}=O(n\log^2 n).$$

对于最后的 $\sum_{k=1}^{n}\frac{1}{k^2}\le\zeta(2)=\frac{\pi^2}{6}\approx 1.64$，而我们的分子是 $\mu(k)$ 只会让结果更小更小。
实际上 $\sum_{k=1}^{\infty}\frac{\mu(k)}{k^2}$ 就是 μ 的狄利克雷生成函数 $M(x)$ 在 $x=2$ 的取值，有 $M(x)=\frac{1}{\zeta(x)}$，那么 $M(2)$ 是一个约为 0.6 的常数。
而在这个问题中，我们也只要处理 (i,j) 的有序对，因此这个常数还可以再少一半左右。