2023“钉耙编程”中国大学生算法设计超级联赛（5）题解

1 1001

对于每个点，计算其到所有线段的距离取 min 即可，复杂度 $O(nm)$ 。

2 1002

首先有结论 $(2^i-1,2^j-1)=2^{(i,j)}-1$ ，证明如下（假设 $j>i$ ）：

$(2^i-1,2^j-1)=(2^i-1,2^j-2^i)=(2^i-1,2^i(2^{j-i}-1))$

$\because (2^i-1,2^i)=1$

$\therefore (2^i-1,2^i(2^{j-i}-1))=(2^i-1,2^{j-i}-1)$

$\therefore (2^i-1,2^j-1)=2^{(i,j)}-1$

所以原式子（定义 $f(x)=(2^x-1)^K$ ）可以变为一个经典问题：

$$\begin{aligned}\sum_{i=1}^{n}\sum_{j=1}^{n}f[(i,j)]&=\sum_{d=1}^{n}\sum_{i=1}^{n}\sum_{j=1}^{n}[(i,j)=d]f(d)\\&=\sum_{d=1}^{n}f(d)\sum_{i=1}^{\lfloor\frac{n}{d}\rfloor}\sum_{j=1}^{\lfloor\frac{n}{d}\rfloor}\sum_{t|(i,j)}\mu(t)\\&=\sum_{d=1}^{n}f(d)\sum_{t=1}^{\lfloor\frac{n}{d}\rfloor}\lfloor\frac{n}{dt}\rfloor^2\mu(t)\\&=\sum_{T=1}^{n}\lfloor\frac{n}{T}\rfloor^2\sum_{d|T}f(d)\mu(\frac{T}{d})\end{aligned}$$

令 $g=f*\mu$ ，则问题转化为对 T 进行除法分块，求 g 的前缀和 $S(n)$ ，可以利用杜教筛（将 g 卷上 1 ）解决：

$$\sum_{i=1}^{n}f(i)=\sum_{i=1}^{n}(g*1)(i)=\sum_{i=1}^{n}S(\lfloor\frac{n}{i}\rfloor)S(n)=\sum_{i=1}^{n}f(i)-\sum_{i=2}^{n}S(\lfloor\frac{n}{i}\rfloor)$$

最后就是 $\sum_{i=1}^{n}f(i)$ 的求解：

$$\sum_{i=1}^{n}(2^i-1)^K=\sum_{i=1}^{n}\sum_{k=0}^{K}\binom{K}{k}2^{ik}(-1)^{K-k}=\sum_{k=0}^{K}\binom{K}{k}(-1)^{K-k}\sum_{i=1}^{n}2^{ik}$$

只需要等比数列求和即可，注意 $k=0$ 的情况。

杜教筛预处理前 $n^{\frac{2}{3}}$ ，时间复杂度 $O(n^{\frac{2}{3}}K)$ 。

3 1003

可以考虑在回文中心统计答案，对于 i 和 $i+1$ 中间的回文中心，如果 $S[i-k+1,i]$ 和 $S[i-k+1,i+k]$ 都是回文串，就找到了一个符合的答案。

先用 Manacher 预处理每个位置 i 能向两边延伸的最长回文串长度 $d[0][i]$ 与 $d[1][i]$（奇数与偶数）。

枚举中心 i 后，如果某个位置 p 满足 $p+d[1][p]>i$ 或 $p+d[0][p]\geq i$ ，并且 $i-d[0][i]<p\leq i$（p 在 i 中心的大回文串内），就找到了一个答案。

所以只需离线二维数点 / 主席树等方法处理此二维限制即可计算出答案。

时间复杂度 $O(n\log n)$ 。

4 1004

4.1 做法一

easy 版本的做法没办法将答案统计到每个子串的右端点，因此不适用于 hard 版本。

求每个前缀的答案实际上就是求每个右端点的符合子串个数，也就是说对于右端点 R 和长度 len ，$[R-2len+1,R]$ 和 $[R-len+1,R]$ 都是回文串。

枚举每个右端点的回文串我们会想到回文自动机，但是回文自动机对每个右端点暴力枚举复杂度是不能接受的，因此需要使用一个回文自动机的科技：Palindrome Series。

简单描述这个科技，就是回文自动机上的节点 x 朝着 Parent 树往上跳，串长序列可以划分为不超过 $\log_2 n$ 个等差数列（科技的详细信息可以自行搜索一下，这里不展开）。因此我们实际上是可以通过枚举等差数列，来枚举每个右端点的所有回文串的。

问题转化为：对于两个等差数列 $a_1+d_1i(0\leq i\leq c_1),a_2+d_2j(0\leq j\leq c_2)$ ，统计多少个 i,j 满足：$a_1+d_1i=2(a_2+d_2j),d_1i-2d_2j=2a_2-a_1$

这是一个不定方程的问题，可以通过 exgcd 求解。

时间复杂度 $O(n\log^3 n)$ ，但实际上一般数据下等差数列根本到不了 $\log_2 n$ 的级别，因此跑得飞快。

4.2 做法二

考虑边建 PAM 边求出 PAM 上每个点是否是两个回文拼起来，树上倍增树上前缀和即可。

复杂度 $O(n\log n)$ 。

5 1005

可以有很多做法，一种做法是用生成函数。

考虑只有一条蛇的情况，则长度为 n 的情况数为 $n!$, 指数型生成函数为 $\frac{x}{1-x}$.

另外因为长度不能超过 k, 所以最终的指数型生成函数为 $\frac{(x^k-1)x}{1-x}$.

由此可以得出有 m 条蛇的情况书为 $\frac{(\frac{(x^k-1)x}{1-x})^m}{m!}$ 的第 n 项.

展开得到答案为 $(x^k-1)^m(1-x)^{-m}$ 的第 $n-m$ 项, 二项式展开即可.

6 1006

简单 DP，定义 $f_{i,a,b}$ 表示考虑完前 i 个飞碟，手里先拿着 a 颜色，后拿着 b 颜色的最大分数。

把题目中的转移都考虑一下即可。

7 1007

问题的关键在于对于任意 $0 \le k \le n$, 求出最终赢 k 次的概率。

假设每次赢的概率是 p, 则最后赢 k 次的概率就是 $(1+px)^n$ 的第 k 项。

8 1008

令 $g_n = \sum_{i=1}^n i^m$, 存在 $m+1$ 次多项式 $F(x), g_n = F(n)$

令 $G(x) = (1-p+px)^n$

$$\text{ans } = \sum_{i=1}^{n} \left[x^i\right] G(x) \times F(i)$$

注意到

$$G^{(t)}(1) = \sum_{i=1}^{n} \left[x^i\right] G(x) \times i^{\underline{t}}$$

设

$$G(x) = \sum_{i=0}^{m} a_i \times x^{\underline{i}}$$

已知点值求数列 $\{a\}$ 可以参考：清华集训 2016 如何优雅地求和

注意到 $G^{(t)}(1) = n^{\underline{t}} \times p^t$

答案即为

$$\sum_{i=0}^{m} a_i \times G^{(i)}(1)$$

9　1009

树形动态规划。把每个重链顶端所在的子树的答案求出来。

转移的时候，深度等于重链本身所形成的树的深度，加上重链沿伸出的子树的深度的最大值。

10　1010

让 1 为根，删掉每条边之后的两部分就是一个子树和全体除掉这个子树。

问题在于如何求出子树内两两异或最大值和子树外两两异或最大值。

子树内两两异或最大值可以通过 dsu on tree + 01trie 实现，复杂度 $O(n \log n \log v)$ 。

子树外两两异或最大值做法很多，这里只讲一个 $O(n \log v)$ 的做法：

用 01trie 找到任意一对 x, y ，使得 $w_x \oplus w_y$ 是最大的。

对于任意一个 u ，如果 u 不是 x 或 y 的祖先（x, y 均在子树外），那么 $ans[u] = w_x \oplus w_y$ 。

对于 1 到 x ，1 到 y 的两个链，沿着链向下走，每次把链外的子树都加入 01Trie 然后查一下即可。

总复杂度 $O(n \log n \log v)$ 。

11　1011

注意到仙人掌的桥如果断了就没办法用别的代替了，首先答案和所有桥的边权取 min 。

剩下的若干个环，任意时刻都要有 $len - 1$ 条边是可以传输的，因此将环上所有的边权从小到大排序，答案和 $\min(w_1 + w_2, w_3)$ 取 min ，其中 w_1, w_2, w_3 代表前三小。特殊注意两条边的环答案和 $w_1 + w_2$ 取 min 。

由于只关注环上前三小，时间复杂度为 $O(m)$ ，当然排序也可以通过。

12　1012

假设一个点的度为 deg_u ，那么他对 ans_k $(2 \le k \le deg_u)$ 的贡献为 $\binom{deg_u}{k}$ 。

$O(n)$ 预处理阶乘和阶乘的逆元，直接循环计算每个点的贡献即可。

复杂度为 $O(n + \sum deg_u) = O(n + m)$ 。