Bowling 题解

题意：给定一个 n 个顶点的凸多边形与落在其外部的 m 个点，对于询问的 q 个方向，求凸多边形沿该方向平移会碰到多少点。

题解：题目可以视作凸多边形不动，点沿着反方向平移。单独研究一个点，可以发现如果要碰到凸多边形，其运动方向应当夹在点与凸多边形的两条切线之间。将两条切线之间的方向范围视作一段区间，那么题目可以转换成给出多个区间，询问一个方向向量被多少区间覆盖。一种处理方式是将所有切线与询问的方向向量离线存储并离散化，然后使用差分处理区间覆盖问题。本题的关键在于求出点到凸多边形的切线，使用二分的方式可以在 $O(\log n)$ 的复杂度求出一个点的切线，总复杂度为 $O(m\log n+q)$。

Independent Feedback Vertex Set 题解

题解：答案必须包含每个三元环中的恰好一个点，因为一个点都不选则会破坏森林约束，选至少两个则会破坏独立集约束。

同时对于一对有至少两个公共点的三元环，确定了答案包含其中一个的某个点之后另一个也随之确定了。

因此答案只可能有三种，分别对应图中唯一的三染色方案（去重后）中的每一种颜色的点。

Counting Stickmen 题解

为了方便描述, 姑且将火柴人划分为头, 手臂$\times 2$, 手掌$\times 2$, 身体, 腿$\times 2$ 的 8 条边.

对于一个火柴人, 我们枚举中间身体的那条边进行计数.

假设身体靠头一侧的端点为 x, 靠腿一侧的端点为 y,

我们在 x 处计算头和手的方案, 在 y 处计算腿的方案, 根据乘法原理, 相乘即可得到火柴人的种数.

首先考虑比较简单的腿的计数,

就是 $\binom{deg_y-1}{2}$, 减一是去掉身体本身这条边.

对于手的选择, 每种手掌外侧端点的选择就是一种方案.

这样的点距离 x 为 2, 显然个数为 $\sum_{u\in neighbor(x)} deg_u - deg_x$

由于这种点还不能与 y 相连 (否则就是腿了), 还要减去 deg_y-1

在这些点中选出两个点, 方案数就是 $\binom{(\sum_{u\in neighbor(x)} deg_u)-deg_x-(deg_y-1)}{2}$

但是还需要去掉两个手臂相同的方案数, 即 $(\sum_{u\in neighbor(x),u\neq y}\binom{deg_u-1}{2})$

选头的方案数显然是 deg_x-3 (减去身体和两只手臂的边).

将他们乘起来就是答案.

$O(n)$ 预处理 $deg_x, \sum_{u\in neighbor(x)}(deg_u-1), \sum_{u\in neighbor(x)}\binom{deg_u-1}{2}$, 再扫一遍所有边计算答案即可.

Black Magic 题解

题目描述

给出 n 个方块，每个方块的左/右都可能是黑或白。将这些方块排成一列，如果两个相邻方块相连接的面都是黑色，那么这两个方块会连在一起。求连通块的最大/最小数量。

数据范围：$1 \le n \le 10^5$

题解

分类讨论，以下的 L, R, E, B 含义和题目中一样。对于连通块最多的情况，要让能够相连的方块对尽量少。可以发现，所有 L 可以全放在最左边，所有 R 可以全放在最右边，这样他们就不会和中间的方块相连，之后尽可能用 E 把 B 隔开即可。对于连通块最少的情况，要让能够相连的方块对尽可能多。可以先把所有 B 拼在一起，之后可以把一个 L 和一个 R 分为一组拼起来。注意如果有至少一组 LR 并且也有至少一块 B 的话，可以把一组 LR 拆开拼在 B 的两边，这样可以再减少一个连通块。

同时存在一种线性 DP 的解法，在此不多赘述。

时间复杂度：$\mathcal{O}(1)$，空间复杂度：$\mathcal{O}(1)$。

Cyber Painter 题解

题目描述

有 $n \times m$ 个小正方形，每个小正方形的中心可能向四条边的中点连一条线段。将这些小正方形随机拼成 $n \times m$ 的大矩形，求期望构成的正方形个数（小正方形的边界不算，只计算由小正方形中心连向边界的线段构成的正方形）。

数据范围：$1 \le n \times m \le 10^5$

题解

求期望的正方形个数等价于求所有方案下的正方形个数之和。可以先枚举形成的正方形边长，有 $\sqrt{nm}$ 种，之后再枚举四个角的小正方形种类，有 $4^4 = 256$ 种。之后可以将剩余的小正方形分为三类：能拼竖边的，能拼横边的，以及都能拼的（即十字）。可以考虑枚举大正方形的竖边中有几个是十字小方块，然后用剩余的十字小方块和只能拼横边的小方块去拼大正方形的横边。用组合数计算答案即可这个可能的十字小方块个数也是 $\sqrt{nm}$ 种。

时间复杂度：$\mathcal{O}(256nm)$，空间复杂度：$\mathcal{O}(nm\sqrt{nm})$ 或 $\mathcal{O}(nm)$。

Sumire 题解

数位DP，用 $dp[i][j][0/1][0/1]$ 表示已经统计了前 i 位，其中有 j 个位上是 d，是否取数字上界，以及是否有前导零即可。用记忆化搜索维护转移。转移时需分四种情况考虑：

1、当前位卡住上界，且之前也卡住数字上界。

2、当前位为0，且目前还在维护前导零。

3、当前位为 d

4、以上都不满足

要注意上述几种情况可能也会合并成一种。用这种讨论的形式能够优化转移的复杂度。最终复杂度为 $O(\log_B^2 r)$。

Weighted Beautiful Tree 题解

普通树DP，用 $dp[u][0/1]$ 代表处理完以节点 u 为根的子树，并且当前节点的 weight 小于等于其父边/大于等于其父边时的最小代价。转移的时候需要把当前节点的所有子边按照边的权值排序后，再扫一遍以统计花费。设当前节点为 u，连接到的子节点为 v，两者之间连接的边权值为 w_e 当前枚举到的花费是 j ，父边的权值是 fw 。则有：

$$val = \begin{cases} dp[v][0] & j > w_e \\ \min(dp[v][0], dp[v][1]) & j = w_e \\ dp[v][1] & j < w_e \end{cases}$$

$$\begin{cases} dp[u][0] = \min(dp[u][0], val) & j \le fw \\ dp[u][1] = \min(dp[u][1], val) & j \ge fw \end{cases}$$

在对权值进行扫描的时候需要对相同权值子节点的进行批量处理。最终复杂度 $O(n \log n)$。

Triangle Game 题解

题目描述

一个非退化三角形，三边边长分别为 a, b, c。现 Kate 和 Emilico 二人做游戏，每轮需要令三角形的一边长度减去一正整数，使这个三角形退化的一方负。Kate 先手，双方均采用最优策略，问 Kate 是否会获胜。

题解

结论是：Kate 获胜当且仅当 $(a-1) \oplus (b-1) \oplus (c-1) \neq 0$。其中 $\oplus$ 为异或运算。

不妨令 x, y, z 分别为三角形最短，次短和最长的边长。由于 $x + y > z$ 且 x, y, z 都是正整数，则有 $(x-1) + (y-1) \ge (z-1)$。

不妨定义如下状态：

- （L 态）$(x-1) \oplus (y-1) \oplus (z-1) = 0$
- （W 态）$(x-1) \oplus (y-1) \oplus (z-1) \neq 0$

当玩家目前处于 L 态时，由于 $(x-1) \oplus (y-1) = (z-1)$，则有 $(x-1) + (y-1) \ge (x-1) \oplus (y-1) = (z-1)$，此时一定为一个非退化三角形。

在 L 态时，转移有如下情况：

- $x = 1$，则 $y = z$。这种情况任何玩家移动都会判负，因此为必败态。
- $x > 1$，则 $1 < x < y < z$，此时一定存在一种减少边长的方案，使得减少边长后三角形不退化。不妨考虑 x 减少为 x'，则由于减去的是正整数，会有 $(y-1) \oplus (z-1) = (x-1) \neq (x'-1)$。则 $(x'-1) \oplus (y-1) \oplus (z-1) \neq 0$。如果改变的是其他边，也可以类似地利用此式。

综上，L 态一定转移到某必败态或 W 态。

当玩家处于 W 态时，有 $(x-1) \oplus (y-1) \neq (z-1)$。令 $r = (x-1) \oplus (y-1) \oplus (z-1)$，则 $((x-1) \oplus r) + 1 < x$ 或 $((y-1) \oplus r) + 1 < y$ 或 $((z-1) \oplus r) + 1 < z$，三式中必有一式成立。考虑 r 的二进制中每一位 1 都一定出现了奇数次，对于最高位的 1，将其异或 r 后一定会变小。因此可以将成立的不等式作为这一步操作进行代换，即可转为 L 态。

由此，每次移动三角形某边长均会减小，W 态会转化为 L 态，而 L 态均转化为某些必败态和 W 态。由此，所有 W 态均为必胜态，L 态均为必败态。结论证毕。

时间复杂度：$\mathcal{O}(1)$，空间复杂度：$\mathcal{O}(1)$。

Counting Good Arrays 题解

考虑固定 n 和 m，此时数组中都是 m 的因子，且最后一个位置为 m。将每个 a_i 除以 a_{i-1} 之后对每个位置上的数做质因子分解，可以观察到答案对于每个质因子是互相独立的，即固定 n 时答案是一个关于 m 的积性函数。固定 m 时，设 m 中质因子 p 的幂为 e，则方案数等于将 e 个 p 分给 $n-1$ 个位置的方案数。显然这是一个组合数，且可以视作关于 n 的一个 e 次多项式。因此任选 $e+1$ 个 n，对每个 n 用 min25 算一次即可将答案插值出来。单次询问复杂度为 $O(m^{3/4})$。

Connectivity of Erdős-Rényi Graph 题解

题意：n 个点的完全图，每条边以 p 的概率留下，$1-p$ 的概率断掉，最终形成图 G，求 G 的连通块个数的期望模 998244353。

题解：$p=1$ 时答案恒为 1。

当 $p\neq 1$ 时：考虑使用生成函数统计有 c 个连通块的无向图有多少个，并通过给边加权来计算概率。

考虑将两张点数分别为 k_1,k_2 的图 G_1,G_2 合并为图 H，则 G_1 和 G_2 对 H 的概率的贡献长成这样：

$$p_{G_1}p_{G_2}(1-p)^{k_1k_2}\binom{k_1+k_2}{k_1}$$

考虑从p_G中提一个$(1-p)^{k(k-1)/2}$出来，即令$(1-p)^{k(k-1)/2}p'_G=p_G$，则有

$$(1-p)^{k_1(k_1-1)/2}(1-p)^{k_2(k_2-1)/2}(1-p)^{k_1k_2}\binom{k_1+k_2}{k_1}p'_{G_1}p'_{G_2}=(1-p)^{(k_1+k_2)(k_1+k_2-1)/2}\binom{k_1+k_2}{k_1}p'_{G_1}p'_{G_2}$$

因此我们将所有点数为k的图的概率提一个 $(1-p)^{k(k-1)/2}$ 出来之后就可以用 EGF 计算了。

令 $(1-p)^{k(k-1)/2}a_k$ 为 G 是 k 个点的连通图的概率，设 a_k 的生成函数为 $F(x)=\sum_k a_k\dfrac{x^k}{k!}$。

类似地，令 $(1-p)^{k(k-1)/2}a_k^{(c)}$ 为 G 是 k 个点 c 个连通分量的图的概率，设 $a_k^{(c)}$ 的生成函数为 $F^{(c)}(x)$。

观察 $F^{(c)}(x)$ 与 $F^{(c-1)}(x)$ 的关系。我们枚举 n 个点的图中的一个大小为 k 的点子集 S，并统计 G 恰好有 c 个连通块且 S 在 G 中的导出子图是一个的连通块的概率，有

$$\begin{aligned}(1-p)^{n(n-1)/2}a_n^{(c)}&=\frac{1}{c}\sum_{k,l}\binom{n}{k}(1-p)^{k(k-1)/2}a_k(1-p)^{(n-k)(n-k-1)/2}a_{n-k}^{(c-1)}(1-p)^{k(n-k)}\\&=(1-p)^{n(n-1)/2}\frac{1}{c}\sum_k\binom{n}{k}a_ka_{n-k}^{(c-1)}\end{aligned}$$

其中 $1/c$ 来源于每张满足条件的图都被统计了恰好 c 次（每个连通块都被枚举了一次）。因此有

$$F^{(c)}(x)=\frac{1}{c}F(x)F^{(c-1)}(x)=\frac{1}{c!}F(x)^c$$

令 $(1-p)^{k(k-1)/2}b_k=1$ 为 G 为 k 个点的图的概率，考虑 $b_{k,l}$ 的生成函数 $G(x)=\sum_k b_k\dfrac{x^k}{k!}$，枚举连通分量个数 c 即得

$$G(x)=\sum_c F^{(c)}(x)=\sum_c\frac{F(x)^c}{c!}=\exp F(x)$$

~~以上证明可简化为“由exp的组合意义易得”。~~我们最终想要的答案为

$$\sum_{m,c}c(1-p)^{n(n-1)/2}a_{n,m}^{(c)}=(1-p)^{n(n-1)/2}\left[\frac{x^n}{n!}\right]\sum_c cF^{(c)}(x)$$

而

$$\sum_c cF^{(c)}(x)=\sum_c \frac{F(x)^c}{(c-1)!}=F(x)\exp F(x)=G(x)\ln G(x)$$

因此答案为

$$(1-p)^{n(n-1)/2}\left[\frac{x^n}{n!}\right]G(x)\ln G(x)$$

其中 $G(x)$ 中 $\frac{x^n}{n!}$ 的系数为 $(1-p)^{-k(k-1)/2}$。

因此对于一组 p 相同且最大点数为n的询问，答案可在 $O(n\log n)$ 内求出。

Rock Tree 题解

题目描述

给定一颗带点权的树，求点权和最大的直径不超过 k 的联通块权值和。

数据范围：$1\le k\le n\le 10^5$

题解

一个显然的想法是先点分治，求经过分治中心的联通块，只需要对于每个子树求出深度不超过
$i=0,1,\dots k-1$

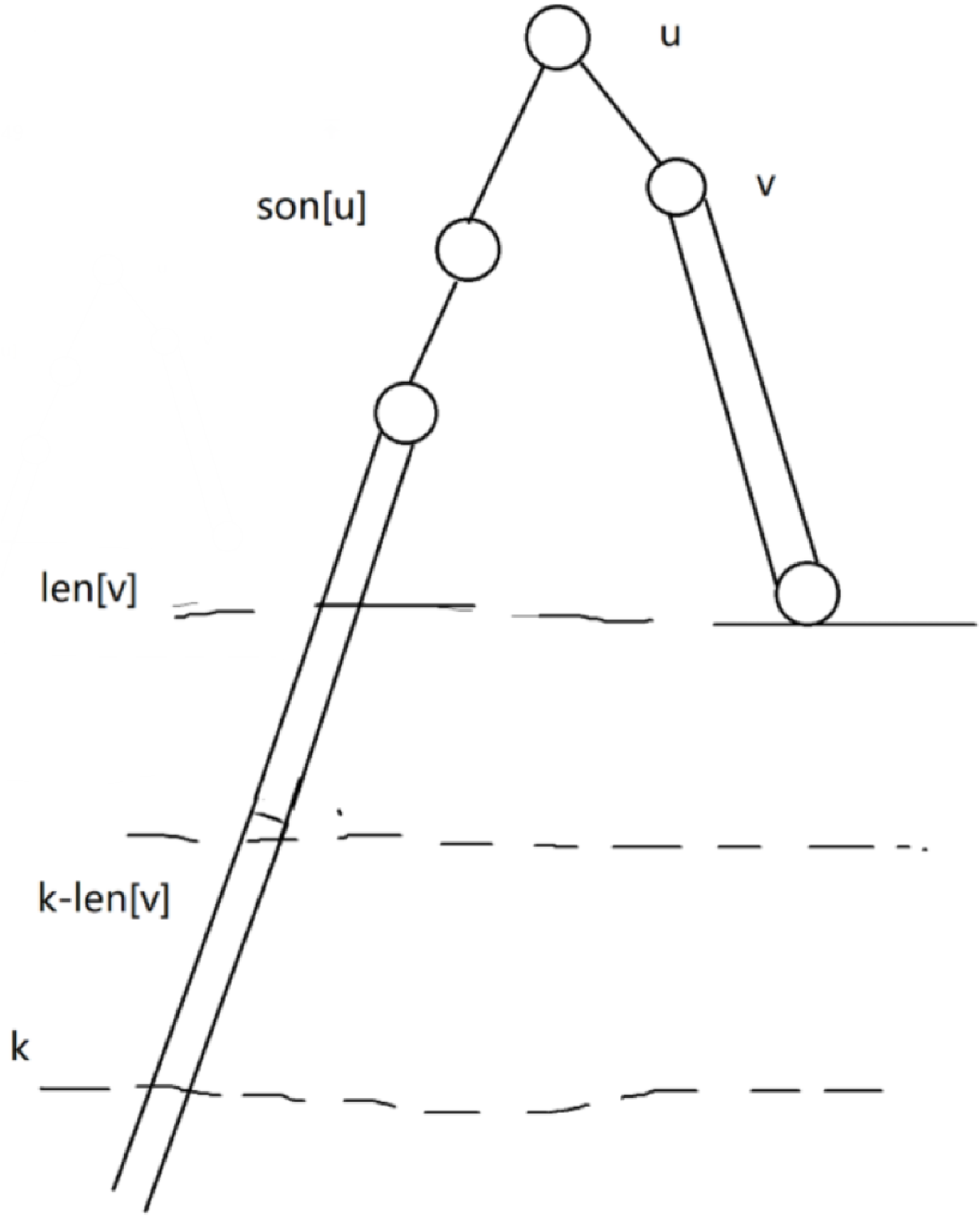
的最大的包含根的联通块，最后在分治中心合并各子树并计算答案即可。

套用下文的长链剖分同样可以通过。

但可以直接长链剖分，对于每个点，我们考虑计算其为联通块最高点的答案。

设 $f_{u,i}$ 表示以 u 为根且包含 u 的联通块，深度不超过 i 的最大权值和，显然 f_u 单调不降。

只需要考虑如何将短链答案合并到长链上：



对于 u 和其轻儿子 v，现在将 f_v 合并至 f_u。len_v 表示 v 的轻链链长。假设 $2 \cdot len_v \le k$

首先对于 $i \le len_v$ 可以直接合并答案。

对于 $len_v < i \le k - len_v$，这一部分 $f_{u,i}$ 可以直接加上 f_{v,len_v} (即 f_v 最大值)。

对于 $k - len_v < i \le k$，由于要满足直径不超过 k，所以合并时两链长度之和不能超过 k，对于这一部分每个 $f_{u,i}$ 只能和 $f_{v,k-i}$ 合并。

对于 $k < i$，显然不符合要求，直接舍弃即可。

注意 $f_{u,i}$ 表示的是深度不超过 i 的答案，即前缀 $\max$，故合并后需要重新保证单调不降。

同时还要单点修改和区间加，考虑利用线段树维护 f

对于第一和第三部分的更新，直接暴力 $O(len_v)$ 次线段树操作即可，第二部分区间加即可。

同时再记录前缀最大值依次更新第一第三部分即可。

当 u 计算完后再加上 a_u 的贡献，全部减去 a_u 后二分出最前面小于 0 的位置赋成 0 (一个点都不取)。

对于 $2 \cdot len_v > k$ 的类似讨论。

这样总复杂度为 $O(n \log n)$，可以轻松通过此题。