20240924题解

A

注意S,P的范围，先枚举和再判断积显然不可行，所以正确做法是sqrt(P)枚举P的因数，再判断加起来是不是S。时间复杂度O(sqrt(P))。

B

学过KMP的同学可以用KMP写（见luogu P4824 Censoring S），但是没学过也能做。

注意到要删除的字符串的长度为3，因此可以用栈判断，遇到除了x以外的字母就入栈，遇到x就判断栈顶的两个字母是不是o和f，是的话都出栈，不是的话x入栈，直到所有字母都判断完了，输出栈内元素数量即可。

C

根据以下构造方法，可以证明只要原图连通，则能保证有一定有合法解。

首先我们对原图求任意的生成树，dfs 树或用并查集都可以，然后将 1 号点赋权为任意，接着开始 dfs。

遍历到一个点时，若它的父亲的点权与它连接父亲的那条边权相同，这也就意味着这个点不能再染成连父亲的边权了，否则这条边会出现两个端点点权都为边权的情况，边断开，图不连通，此时我们将这个点赋值为除了父亲点权的任意值。

若不同，则将这个点点权赋值为那条边的边权，保证这条边有且仅有一个端点权值与边权相同，可以保证连通。

这样构造无论哪个点都会在其中一种情况中被讨论，于是保证有解。

D

分类讨论题。

先考虑第一次插入，只能是AAB或者ABB（和c\_{AB}的值有关），那么就相当于是在原串左边加了个A，或者原串右边加了个B。因此我们只需要考虑c\_{AA}或者c\_{BB}的值。

不妨设c\_{AB}=A（另一种情况是对称的，结果只会相同），则我们只需要考虑c\_{AA}的值。

如果c\_{AA}=A，结果只有一种，就是前面n-1个A，最后接一个B。

否则，c\_{AA}=B，此时字符串的第一个字母确定是A，最后两个字母确定是AB。再考虑c\_{BA}的值。如果c\_{BA}=A，说明B之间两两不相邻，此时答案是斐波那契数列的第n-3项。否则，c\_{BA}=B，说明B可以相邻，则其他位置的AB取值不受限制，答案是2^(n-3)。

E

如果考虑以时间轴 DP，会发现即使是最终方案相同，标记顺序不同会导致概率不同。所以需要换一种方式 DP。

注意到若a[l],a[r]都被选定，那么对于l≤i≤r,a[l]≤a[i]≤a[r]的 a[i] 来讲，单独考虑 [l,r] 内和整体考虑两种方式下，a[i] 成为 [l,r] 内下一个被选中的概率是一样的。

因此考虑区间DP，令dp[l][r]表示当前已经选定a[l]和a[r]之后，在[l,r]中选出数的个数的期望，g[l][r]表示满足l≤i≤r,a[l]≤a[i]≤a[r]的i的个数。

枚举中间点x，则转移为：dp[l][r]=sum(dp[l][x]+dp[x][r])/g[l][r]+1，x满足l≤x≤r,a[l]≤a[x]≤a[r]。

时间复杂度O(n^3)，考虑优化。

由于转移的两个求和都属于带偏序的区间求和，所以可以将dp[l][x]和dp[x][r]的区间和分别用一个权值树状数组维护，每次右移r时插入dp[x][r]和dp[l][r]，顺便维护g[l][r]，即可将复杂度降为O(n^2 logn)。

F

首先，对于普通的树，到一个点最远的点一定是直径的端点之一。记 S表示直径长度。

先求出一条直径，若直径的两个端点颜色相同，则最长距离一定为直径，此时其他结点颜色可以任意分配，对答案的贡献为dis(x,y)\*2^(n-2)。否则，令两个端点分别为 x,y，并钦定 x,y 不同色。枚举答案 d，所有到 x 距离 >d 的点颜色必须与 y 一样，所有到 y 距离 >d 的点颜色必须和 x 一样。由于 x,y 是直径的两个端点，可以发现，若一个点 z 到 x, y 的距离都不超过 d，则其到任何一个点的距离不超过 d，所以 z 的颜色并不会对答案产生影响。

所以，定义 cnt[i] 表示到直径两端的距离不超过 i 的点数。定义 f[d] 表示答案不超过 d 的树的形态数，g[d] 表示答案为 d 的树的形态数，dis1,dis2 表示从直径的两端点出发到其他点的距离。定义 L=max(min(dis1[i],dis2[i]))。此处 L 的意义为，在所有形态的树中，最小的答案（同色点对最大距离）。对于每个点取到直径两端点近的那个颜色即可。

最终的总权值为 sum(g[i]\*i, i=L,...,S)。

容易得到 f[d]=2^{cnt[d]}。但是我们想要答案等于 d 的树的形态数 g[i]。很明显，只需要容斥减去 f[d−1] 即可，也就是 g[d]=f[d]-f[d-1]。

注意 x,y 共有 2 种颜色分配方案，最后要乘2。