NOIp 模拟赛

1

Sep 2023

1 a

直接枚举一个答案，判断它的所有倍数第一次和最后一次出现的间隔是否不小于 k。

时间复杂度 $O(n \ln n)$。

2 b

考虑把问题转换为每次删一个字符，有多少种不同的过程序列。

我们钦定每次删掉一个极长连通段的最后一个字符，也即删掉一个 $s_i \neq s_{i+1}$ 的位置 i。不难发现这样删除序列和过程序列一一对应，可以直接计数这个删除序列。

考虑区间 dp，枚举一个区间最后一个删除的位置。这样之前的删除中，左右两边就独立了，可以分别计数，最后合并时乘上一个组合数。

考虑限制是什么。如果这个区间最后删除的是 k，那么在删除它时，它后面的字符一定是 s_{r+1}，那么直接判断 s_k 是否等于 s_{r+1} 就行了。

状态数是 $O(n^2)$，转移的时间复杂度是 $O(n)$，总时间复杂度就是 $O(n^3)$。

3 c

先离散化，在后面的分析中将 n 和 m 视为同阶。

考虑怎么计算两点间的最短路，发现有最短路等于曼哈顿距离减去在不绕路的情况下最多能走到多少额外边。曼哈顿距离之和是好算的，考虑最多额外边个数之和怎么算。

将点对分为两类：左上到右下与左下到右上。这两类分别计算，下面以第一类为例。

我们枚举每一个起始点，考虑它到每个它右下点的点，最多能经过多少不绕路的额外边。

我们可以先 dp 求出从额外边 x 的起点到额外边 y 的终点，能最多经过多少条额外边。

在枚举一个起始点之后，用这个 dp 值去更新，对于每一个作为某条额外边终点的点，到它最多经过多少条额外边，记其为 $g_{i,j}$。

然后我们考虑对于每一个 k，有哪些点经过的额外边数不小于 k：我们找到所有 $g_{i,j} \ge k$ 的点 (i,j)，将以它们为左上角、(n,n) 为右下角的矩形并起来，这个并集中的所有点就是经过的额外边数不小于 k 的所有点。

更进一步地，我们发现以 $g_{i,j} = k$ 的 (i,j) 为左上角的矩形并，肯定包含了以 $g_{i,j} > k$ 的 (i,j) 为左上角的矩形并。那么上一步中对于所有 k，我们只要将 $g_{i,j} = k$ 的 (i,j) 代表的矩形求个面积并，并对于所有 k 将这个面积并求和就可以了。

枚举 k 并计算面积并的过程可以先把所有终点排序，然后对于每个枚举到的起始点，在 $O(n+m)$ 的时间复杂度内解决。现在的唯一问题是怎么快速更新所有 $g_{i,j}$。

我们可以先枚举起始点所在的行，然后倒着枚举其所在的列，并用这一列中新加的起点去更新每个终点的 $g_{i,j}$。每个起点会被加入 $O(n)$ 次，每次更新是 $O(m)$ 的，所以这一部分的时间复杂度为 $O(nm^2)$。

总时间复杂度就是 $O(n^3)$。

4 d

考虑一条链的情况，可以直接用线段树维护每条边的颜色，查询时线段树二分找到 x 前面最后一个合法的和后面最后一个合法的。

拓展到树，考虑轻重链剖分。对于每个点，记录下它能到轻子树中的多少点，记为 f_i。查询时先跳到深度最浅的合法点，然后就和链相同了，只不过将计数变为对 f_i 求和。

修改可以拆成两条点到根的链的反转。修改时会改变 $O(\log n)$ 条重链的边的情况，以及 $O(\log n)$ 个点的 f 值，这都是能快速维护的。

代码实现有一定细节。

时间复杂度 $O(n\log^2 n)$。