20231114题解

1. 路径覆盖

原题洛谷P3441，来点T1紫题震撼。

一个不太好想的贪心题。

10pts：l=1，直接找树的直径即可。这个可以O(n)处理出来，直接DFS即可。

50pts：跟预想有偏差，本来想着O(n2l)的爆搜有可能可以搞搞。

100pts：

首先显然路径都是叶子到叶子（把这棵树看成无根树）。

那么显然叶子结点最多被覆盖2\*l个，且每往里推一层，里面覆盖的结点个数最多也是2\*l个。但是一个结点可能可以处于多个层，所以我们的处理方法应该是：标记那些“只能位于第x层的结点”。也就是说，我们把第x-1层的结点删掉之后，剩下的叶子结点就必须位于第x层。因此我们可以不断进行删叶子操作来计算出层数。

所以我们可以从叶子出发，利用BFS搜索出每个结点对应的层数，设第x层有a[x]个点，则答案为sum{min{2\*l,a[x]}}。

时间复杂度O(n)。

1. 连通块计数

原题AGC015C。

20pts：注意数据范围极小，对每个询问，直接对子矩形内部做一次BFS即可，时间复杂度O(nmq)。

40pts：n=1，直接可以前缀和维护[1..m]的连通块数量，查询的时候注意对应格子是0和1的时候最终结果有差别，判断一下就可以。时间复杂度O(m+q)。

100pts：

以上做法都没用到本题的特殊性质：任意两个1之间都最多有一条横竖连通的道路（可能是0条）。

首先，一棵有n个结点的树有n-1条边，所以一个有m棵树，共n个结点的森林有n-m条边。

其次，连边只会在相邻两个1之间连边。

那么问题就变成了统计一个子矩形内1的个数以及相邻1的对数。

1的个数可以二维前缀和，横竖相邻1的对数可以分开二维前缀和，或者对某个点i，用三个值a,b,c分别记录它的值，它和它上边的结点是否连边，它和它左边的结点是否连边，这样就可以统一用一个二维前缀和查询了。时间复杂度O(nm+q)。

1. 种树

原题NC23051（牛客网）。

40pts：每次操作暴力更新查询即可，加点直接加，开花就递归修改，查询也是直接查询，时间复杂度O(n2)。

50pts：链的情况直接用前缀和维护，这样修改和查询复杂度都降至O(1)，总复杂度O(n)。

100pts：

显然可以DFS序维护线段树，但是本题的树是动态添加的，怎么办？

可以先把操作离线下来建好树，然后修改使用DFS序线段树来修改，加点时记录当前被加的点的值（这些值是多加上去的），然后查询的时候用当前值减去之前记录的多加上去的值即可。复杂度O(nlogn)。

1. 完全不回文串

原题NC221793。

首先本题需要发现一个性质：只要串内不存在长为2或3的回文串，整个串就是完全不回文的。

10pts：只有一种字母，答案是n\*(2/3)向下取整。

40pts：可以设置状态为dp[i]表示[1..i]修改为完全不回文串的最小代价。

状态转移可以枚举第i个字母被改成什么，但是i改完之后会影响前面的串是不是回文。利用上面的性质，我们需要考虑i-1和i-2号字母分别是什么，i只需要和他俩不相同即可。所以需要考虑dp[i][a][b][c]，表示i号是a，i-1号是b，i-2号是c时的最小修改次数。时间复杂度O(n\*263)。

60pts：稍微压一下状态。可以尝试把i号字母的那一维状态压掉，复杂度变成O(n\*262)。

100pts：

注意到另一个性质：每个字母最多和前后各两个字母不同，所以每个字母最多需要修改的距离是4。所以状态可以这么定义：dp[i][a][b][c]为[1..i]里面，i到si距离为a，i-1到si-1距离为b，i-2到si-2距离为c的最小修改次数。时间复杂度O(n\*125)，可以通过本题。

也可以尝试压掉i或者i-2那一维的状态，进一步优化时间。