Colin 的模拟题

Solutions

一、矩阵染色

染色方案数：$f(i)=f(i-1)*(m^2-3*m+3), f(1)=m*(m-1)$

故所求答案为：$\sum_{i=1}^{n}(m^2-m)(m^2-3*m+3)^{i-1}*(2*i)^m$

令 $t=m^2-3*m+3$，则原式$=\dfrac{2^m(m^2-m)}{\mathrm{t}}\sum_{i=1}^{n}t^i*i^m$

令$S_{n,m}=\sum_{i=1}^{n}t^i*i^m$，则题目关键变成了求 $S_{n,m}$，而这也就决定了这道题所得的分数。

①．对于前面 20 分的数据 n<=10^5，我们可以直接暴力求解。

②．对于中间 40 分的数据 m<=100，我们考虑递推求 $S_{n,m}$。

$$
\begin{aligned}
S_{n+1,m}&=t+\sum_{i=1}^{n}t^{i+1}*(i+1)^m\\
&=t+t\sum_{i=1}^{n}t^i*\sum_{j=0}^{m}C(m,j)*i^j\\
&=t+t\sum_{j=0}^{m}C(m,j)*S_{n,j}
\end{aligned}
$$

故我们可以设计一个 m*m 的矩阵来加速递推。

③．对于接下来 20 分的数据，m<=10^5，但 p<=1000，我们可以发现：t^i*i^m 在模 p 的剩余系中一定存在一个长度不超过 p*p 的循环，故我们只要找到这个循环即可。

④．对于最后 20 分的数据，m<=10^3，p<=10^9，我们考虑另外一种推导：

由 $S_{n+1,m}=S_{n,m}+t^{n+1}*(n+1)^m$，

得 $S_{n,m}+t^{n+1}*(n+1)^m=t+t\sum_{j=0}^{m}C(m,j)*S_{n,j}$

我们注意到当得到 $S_{n,j}$（$j<m$）后，上面这个式子其实是关于 $S_{n,m}$ 的一个一元一次方程。故我们只需求得 $S_{n,0}$，便可在 $O(m^2)$ 的时间内推出 $S_{n,m}$，而 $S_{n,0}$ 是可以通过等比数列求和得到的。

二、Link-Cut-Tree

子任务一：

记树的大小为 $size_i$，第 i 棵树上的所有点到 j 点的距离和为 $dis_{i,j}$，点 u 与点 v 间的距离 $d(u,v)$。

对于所求答案，包括每棵树内部的和树与树之间的，对于前者我们可以直接 dp 求出。对于后者，若只有两棵树，应连接两棵树中 dis 值最大的。

若有三棵树，我们可以枚举三棵树的相对位置，设第一棵树与第二棵树通过 (x,y) 相连，第二棵树与第三棵树通过 (u,v) 相连，则树与树之间的答案为：

$$\begin{aligned}&size_2 dis_{1,x} + size_1 dis_{2,y} + size_1 * size_2\\&+ size_3 dis_{2,u} + size_2 dis_{3,v} + size_2 * size_3\\&+ size_3 dis_{1,x} + size_1 dis_{3,v} + (d(y,u)+2) size_1 * size_3\end{aligned}$$

其中 x, v 应为树 1 和树 3 中 dis 值最大的点。

对于树 2，当确定 y 后，我们只要求出 $size_3 * dis_{2,u} + size_1 * size_3 * d_{y,u}$ 的最大值，便可得到答案，而这个也是可以通过 dp 在 $O(n)$ 的时间内求出的。

子任务二：

经分析观察，我们可以得到结论：对于任意的连通块，只要连通块中黑点的个数为偶数，就一定可以通过删去若干条边使得连通块满足度数的限制，否则一定无解。

证明：

先证明后无解情况：如果连通块中黑点的个数为奇数，由于任意连通块的度数和一定为偶数，因此必然无解。

对于结论的前面部分，我们使用归纳法：

1. 若树只包含 2 个黑点，此时我们可以只保留他们之间的路径来满足度数要求。

2. 假设对于大小小于 k 的任意连通块结论成立，现在我们有一棵 k 个点的树，则有以下几种情况：

(a).如果树中不存在白点，此时 k 必然是偶数。任选一个点 u 作为根，设点 u 有 p 个儿子，其中有 q 个儿子的子树大小为偶数。

i. 若 q > 0，可以删去与一个子树大小为偶数的儿子的连边，得到两个大小< k 的且含有偶数个黑点的连通块。

ii. 若 q = 0，可知 p 为奇数，那么点 u 自身的度数限制已经满足。而每棵子树等价于，

删掉与 u 的连边之后，将根的颜色取反。这样每棵子树都变成了含有偶数个黑点且大小< k 的连通块。

(b) 如果树中存在至少 1 个白点。我们任选一个白点 u 作为根，设且 u 有 p 个儿子，其中有 q 个儿子的子树中含有偶数个黑点。

i. 若 q>0，可以删去与一个子树中含有偶数个黑点的儿子的连边，得到两个大小< k 的且含有偶数个黑点的连通块。

ii. 若 q=0，可知 p 为偶数，那么点 u 自身的度数限制已经满足。而每棵子树等价于，删掉与 u 的连边之后，将根的颜色取反。这样每棵子树都变成了含有偶数个黑点且大小< k 的连通块。

这样我们就可以得到一个算法：首先判断是否有解，然后枚举每条边，判断其是否可以被删去，即去掉该边后各个连通块中黑点个数的奇偶性不发生改变。由于边与边之间互不影响，故只需去掉所有可以被删去的边，便满足了字典序的要求。

三、**跨越国度**

我们将题目所给的平面图转成对偶图后，题目的实质变为了：给你一颗树，边带权，点有多种颜色，需要支持修改点的颜色和询问最近同色点对的操作。对于这个问题，我们可以采用点分治的方法来解决。

对于每一个重心，我们维护 $x+1$ 个堆，x 为其所管辖范围内出现过的所有颜色种数，其中维护着属于该种颜色的各点到他的距离，则对于该重心该颜色，答案就是堆中的前两个元素。而对于后面一个堆，我们维护各种颜色的答案，则对于该重心，答案就为堆首元素的值。

因为有修改操作，所以每一个重心的答案为时间轴上的若干线段，我们得到所有重心的所有线段后，通过线段覆盖，便可得到整棵树在所有时刻答案。

考虑一下复杂度，堆的总个数和堆中元素总个数都是 $O(n\log n)$ 的，而对于每一个修改操作，会涉及到 $O(\log n)$ 个重心，每一个重心需要 $O(\log n)$ 的时间进行修改，故总的复杂度为 $O(n\log^2 n)$。