模拟题 2021-3-13 题解

Company

算法一

费用流，参考 BZOJ1221: [HNOI2001] 软件开发

期望得分：70

算法二

三分强制购买的毛巾数量，然后用最优的方式去安排消毒方式。

确定了购买毛巾的数量后就可以贪心了，尽可能多得使用费用较小的消毒方式。

复杂度$O(n\log n)$　期望得分：100

Mirror

算法一

若β是α的子序列，那么对于$i<j$，一定满足$\alpha_{\beta_i}<\alpha_{\beta_j}$。

- 设$\alpha_u=\beta_i$，$\alpha_v=\beta_j$，首先有$u<v$。
- 注意$\alpha_{\beta_i}=u$，$\alpha_{\beta_j}=v$。

枚举$0\le k\le m$，强制对于$i\le k$满足$\alpha_{\beta_i}\le m$，且对于$i>k$满足$\alpha_{\beta_i}>m$。

Case #1:　考虑$i\le k$的这些α_{β_i}，这些$\alpha_{\beta_i}\le m$

- 由于对于$i>k$满足$\alpha_{\beta_i}>m$，故对于$i\le k$，这些$\alpha_{\beta_i}\le m$取值必须取在$\beta_j(j\le k)$这些值中。
- 由于对于$i<j$，一定满足$\alpha_{\beta_i}<\alpha_{\beta_j}$，所以填写方式是唯一的。（暴力判断是否合法）

Case #2:　考虑$i>k$的这些α_{β_i}，这些$\alpha_{\beta_i}>m$

- 由于对于$i<j$，一定满足$\alpha_{\beta_i}<\alpha_{\beta_j}$，所以填写的方案数为$\binom{n-m}{m-k}$。

Case #3: 考虑$i > m$且$\alpha_i > m$的那些位置，共有$n - m - k$个位置

- 设分配N个位置的方案数为f_N，考虑最后一位是否是自环，可以得到递推式

$$f_N = f_{N-1} + (N-1)f_{N-2}$$

至此，已经完成了所有计算，只需要枚举m判断后累加$\binom{n-m}{m-k}f_{n-m-k}$即可，复杂度$O(n + m^2)$。

Network

算法一

先考虑$Q = 1$的情况，在这个subtask中，只需要计算一个点上有多少信息。

这个问题比较别扭，先考虑最终有哪些点上有这个点的信息，并且求出该条信息到达每个点的时间。

可以从当前点开始DFS遍历，需要计算通过每条边的时间。预处理出每条边在哪些时间区间是开启的，在经过这条边时找到最早的一个能使用的。

将整棵树都遍历后，也就计算出了哪些点最终得到了这个点的信息，但是我们真正需要计算并不是这个问题，不过可以类比一下。

同样从当前点开始遍历，计算出每个点能到达目标点的最晚时间，类似地，需要找出的是最晚的一条能使用的边。

时间复杂度$O(N + QM)$，可以通过$Q = 1$的数据。

算法二

先来考虑树退化为一条链的情况，一个点上的信息集合一定是一段区间。对于每个点，分别计算出左侧和右侧分别有多少点的信息可以传递到它，下面介绍计算左侧点数的方法，右侧可以相同得到。

参考前一页的算法，从左往右遍历，维护到达这个点的时间为t的点数量，经过一条边时需要计算对应的变动。

一条边的开启时间是一些不相交的区间，容易发现，经过一条边时可以计算出每个时间区间有多少点经过，比如当前枚举到的时间区间为$[l, r]$，前一个时间区间为$[l', r']$，那么$[r'+1, r]$这段时间内的点是在这个区间通过的，对应的变动只需要计算区间$[r' + 1, l - 1]$区间内的点数，将他们移动到l。

这些修改都可以用线段树来完成。

时间复杂度$O((N + M)\log N)$，可以通过链的数据。

算法三

链的情况已经解决了，树的情况只需要类比一下。

考虑进行点分治，每次处理经过重心的所有传递的贡献。

先计算出所有点到达重心的时间，经过边时的变动和算法二相同，合并不同子树的信息时可以用线段树合并，这一部分的复杂度为$O((N+M)\log N)$。

所有点到达重心后，再用同样的方式从重心往下遍历，在遍历时涉及到了回溯操作，可以用可持久化来解决，复杂度也为$O((N+M)\log N)$。

注意到这样直接做是会算重的，对于每棵子树需要再减去来自自己这棵子树的贡献，这一部分的问题规模和前一步相同。

时间复杂度$O((N+M)\log^2 N)$。

算法四

还有另外一个截然不同的思路，考虑直接在连边和断边的同时在线维护每个点的集合大小。

同一个连通块中所有点的集合大小都是相同的，可以在选择每个连通块中深度最浅的点，定义为这个连通块的根，将每个连通块的答案存在对应的根上。

在断开一条边时，更新新分离出来的连通块的根上的信息，在断开的这条边上记录断边前这两个连通块的交集大小（即现在的答案），当下次连上这条边时，假如这两个连通块现在的集合大小为A和B，边上记录的交集大小为C，那么现在的并集大小为$A+B-C$。

随便选择一个动态树数据结构就可以完成维护，比如Link-Cut Tree。

时间复杂度$O((N+M)\log N)$，空间复杂度$O(N)$。