sol

time：2024/6/28

T1 石头门（door）

特别暴力的一道题，出题人感觉暴力实现的好应该都可过（吧?

Sub1

老规矩，放暴力过去

Sub2

只有一种字符，那合并操作和匹配操作都不考虑字符情况，暴力合并 $O(nm)$ + 计算长度查询即可，时间 $O(nm)$

Sub3

合并操作被变相 ban 了，直接硬搞 $O(mns)$ ，字符集很小，二进制压成一个 `int` 变成 $O(nm)$

Sub4

继续 Sub4 的压缩思路，考虑每个点开一个 `std::bitset` （空间有 $\frac{1}{4}$ 都被这玩意占了，剩下的被用成栈空间了）压缩自己和前 $|s|$ 个点的信息，每次询问直接对于每个点查，合并就暴力和，时间 $O(nm \mid s \mid 4\frac{1}{w})$ ，反正可过

T2 回路（fleury）

Subtask 1(5 pts)

随便爆搜即可。

Subtask 2(25 pts)

本身想放一个暴力的，但是没想到还要选起点的问题，就是 $O(nm^2)$ 了。

本 Subtask 大概猜一个结论：从每一个起点开始答案是一样的。这样就可以做到 $O(m^2)$ 的复杂度了。证明后面再给。

Subtask 3(10 pts)

如果只有简单环，那么答案显然是 1，因此直接输出 1 即可得分。

Subtask 4、5(60 pts)

已经给出所有点是偶度点，所以对于一个连通图，一定有欧拉回路。

考虑欧拉回路的过程，容易发现，如果我们不是最后一次删除的话，那么一定不是桥；否则我们如果是最后一次删除的话，那么就是桥。

证明也十分显然：如果不是最后一次删除却走的是桥，说明至少有两个桥，那么我随便走了一个桥之后肯定就不再能回到这个点走另外的一个桥了，说明最多只有一个桥。如果是最后一次却不是桥，这说明下一次走到这个点时（显然是可以走到的）必须停止。而这显然是根所必备的条件。

综上，一个连通块的答案就是边数 - 点数 + 1。随便 dfs 即可。时间复杂度 $O(n+m)$。

T3 炼金术（alchemy）

Sub1

暴力，随便打

Sub2

发现铜之炼金术占的位置可以直接去掉，其它位置随便乱搞就好了，反正 $|s|=1$ ，就每个位置 $\frac{51}{52}$ 的概率不出现 s ，全乘起来用 1 减就好

Sub3

不存在你的位置，就全是他的发挥空间，继续 Sub2 的思想考虑不存在 s 的概率，对 s 求出 kmp 的 $ne[]$ 数组后考虑 dp ，令 $d[i][j]$ 表示填到第 i 个位置 kmp 中的指针为 j （这个指针就是 kmp 中跳 ne 的指针）的方案数， $d[i][|s|]$ 就是出现了 s ，我们强制其为 0 其它的每个位置暴力转移，时间 $O(52n|s|)$ （当然我们发现填大写字母是一样的，可以优化为 $O(27n|s|)$ ，我本来想给这个部分分的，但 ~~就这题有 5 个 Sub 不好看~~ 已经快是正解了就不管了）

对于优化，不难发现转移其实和 i 等于多少无关，设 $d[i][j]=\sum d[i-1][k]*v_{j,k}$ （ $v_{j,k}=0$ 就是不可从此转移），这个 $v_{j,k}$ 与 i 无关，可以用 kmp 预处理得到，直接上矩阵优化，就是 $O(|s|^3\log n)$

最后不存在 s 的方案数就是 $\sum d[n][i], i=0\sim|s|-1$

upd：mike 找到了 $O(n\log n)$ 做这个 sub 的方法，就是 kmp + dp + 容斥（ + NTT），空间 $O(n\log n)$

Sub4

考虑加入你以后的影响，就是把原字符串分成几段，每一段都求出不存在 s 的方案数，乘起来用总数减，时间 $O(|s|^3\log n)$

T4 甜甜圈（donut）

首先，因为若第 i 维不稳定，排它的时候小于 i 的维数会被随机打乱，所以小于他的维数不重要了，也就是说，每次询问我们只关心最大的 c_i ，不妨即其为 mxc

Sub1

暴力，关于随机打乱，就是如果两个数第 $mxc\sim k$ 维都一样就意味着它们可以随机互换，出题人打的 $O(m(kn\log n+nk))$ 不排除有更优秀的暴力

Sub2

$k=1$ 方便了平衡树等数据结构，只要你愿意爆码可过 ~~其实出题人没打过，口嗨一把~~

Sub3

不存在修改，提示我们离线排序，可优化掉每次排序的 $\log$ ，变成 $O(kn\log n+m(n+kn))$

Sub4

继续离线排序的思想，把修改全都读入一起排，每次可以 $O(n)$ 得到稳定的情况下排得的序列，问题在得可以交换的数，不难发现可以互换的数一定相邻，不妨在离线排好的数列上建一棵树，叶子就是每个数字，其它节点代表 k 维，每个点的子树里面的点从第 k 维到这个点代表的维数都相同，那么两个点的 lca 就代表了它们相同的最多维数（可以想象成一个 Tire ）， ST 表可以做到 $O(1)$ 查询，每次询问对排好序的所有相邻数查一次，就是 $O(n)$ ，总复杂度 $O(kn\log n+n\log n+mn)$

最后

估计题目还是有一定难度，比较合理的得分大概是 $60(100)+100(15/40)+40+20$ 大概 200 分左右吧?