test 题解

难点在于绝对值。

有一个 trick 是，要最大化绝对值，并且绝对值里面是两数相减的形式，可以钦定两者大小关系然后计算。这样即使算错了，也不会比最优解更大。

由于 n 很小，完全可以枚举一个集合 $S \subseteq \{1, \dots, n\}$，表示 S 中的数贡献为 $x - r$，不在 S 中的数的贡献为 $r - x$。然后我们把每个人的实际得分拆到每道题上，我们可以对每道题算出一个系数 $a_i = \sum\limits_{j=1}^{n} [s_{j,i} = 1](-1)^{[j \in S]}$，然后根据系数的大小来贪心的分配每道题的分数。

时间复杂度 $O(2^n nm)$。

go 题解

首先，不难 dfs 求出有哪些白棋是死棋。

黑棋翻转成白棋是简单的。分为两种情况。

1. 这颗黑棋周围有气，或者周围有活的白棋，那么这颗黑棋翻转后就是活的，并且会使与之相邻的、原先是死棋的白棋连通块也变成活棋。答案减去变成活棋的白棋连通块大小。注意一个白棋连通块可能有多个棋子与这颗黑棋相邻，注意判重。
2. 这颗黑棋周围没有气，并且周围没有活的白棋，那么这颗黑棋翻转后就是死棋。答案 $+1$。

白棋翻转成黑棋稍微复杂一些。也分为两种情况。

1. 这颗白棋本来是死棋，那么翻转后成了黑棋，就不计入答案了。答案 -1。
2. 这颗白棋本来是活棋，那么需要考虑有多少原先是活棋的白棋在翻转后成了死棋。发现这就是求割掉一个点后与空地所在连通块不连通，本质上是求割点。新建一个超级源点连向所有空地，在空地和白棋之间连边，跑 tarjan，如果一个点 u 的子节点 v 的 low_v 大于等于 dfn_u，那么 u 代表的白棋翻转成黑棋后，v 的子树里的点都会成为死棋，只要记录子树 sz 即可。

时间复杂度 $O(n^2)$。

offset 题解

观察数据范围和时间限制可知，复杂度大概是小常数线性对数，所以我们要维护的东西形式应该尽量简单。也就是说，我们需要将题目给的式子化简成尽量简单的形式。

推得最终结果的方法有很多，下面演示其中一种。

$$
\begin{aligned}
&\sum_{l'=l}^{r}\sum_{r'=l'}^{r}((\sum_{i=l'}^{r'} a_i)^2 + (r-l+2)(r'-l')a_{l'}a_{r'}) \\
&= \sum_{l'=l}^{r}\sum_{r'=l'}^{r}((\sum_{i=l'}^{r'} a_i^2) + (\sum_{l' \le i < j \le r'} 2a_ia_j) + (r-l+2)(r'-l')a_{l'}a_{r'}) \quad \text{将平方项拆开} \\
&= \sum_{i=l}^{r}(i-l+1)(r-i+1)a_i^2 + \sum_{l \le i < j \le r} 2a_ia_j(i-l+1)(r-j+1) + \sum_{l \le l' \le r' \le r}(r-l+2)(r'-l')a_{l'}a_{r'} \quad \text{对每一项计算贡献系数} \\
&= \sum_{i=l}^{r}(i-l+1)(r-i+1)a_i^2 + \sum_{l \le i < j \le r} 2a_ia_j(i-l+1)(r-j+1) + (r-l+2)(j-i)a_ia_j \quad \text{将枚举范围相同的式子合并} \\
&= \sum_{i=l}^{r} -i^2a_i^2 + (-l+1)(r+1)a_i^2 + (l+r)ia_i^2 + \sum_{l \le i < j \le r} -2ija_ia_j + 2a_ia_j(-l+1)(r+1) + 2ia_ia_j(r+1) - 2j(-l+1)a_ia_j + ja_ia_j(r-l+2) - ia_ia \\
&= \sum_{i=l}^{r} -i^2a_i^2 + (-l+1)(r+1)a_i^2 + (l+r)ia_i^2 + \sum_{l \le i < j \le r} -2ija_ia_j + 2a_ia_j(-l+1)(r+1) + (l+r)(i+j)a_ia_j \quad \text{互相抵消} \\
&= (\sum_{i=l}^{r} -i^2a_i^2 + \sum_{l \le i < j \le r} -2ija_ia_j) + (-l+1)(r+1)(\sum_{i=l}^{r} a_i^2 + \sum_{l \le i < j \le r} 2a_ia_j) + (l+r)(\sum_{i=l}^{r} ia_i^2 + \sum_{l \le i < j \le r}(i+j)a_ia_j) \quad \text{同类项分组} \\
&= -(\sum_{i=l}^{r} ia_i)^2 + (-l+1)(r+1)(\sum_{i=l}^{r} a_i)^2 + (l+r)(\sum_{i=l}^{r} ia_i)(\sum_{i=l}^{r} a_i)
\end{aligned}
$$

用树状数组维护 $\sum ia_i$ 和 $\sum a_i$ 即可。

复杂度 $O(q \log n)$。

string 题解

第二条限制提示我们要容斥。

设 f_i 表示恰好有 i 个元素不满足限制的方案数，则答案为 $\sum\limits_{i=0}^{n}(-1)^i f_i$。

枚举 i 表示有多少元素不满足限制，这样我们有一个系数 $\binom{n}{i}$。首先，剩下的 $n-i$ 个元素没有限制，所以它们构成的 2^{n-i} 个集合可以任意决定选不选，所以有一个 $2^{2^{n-i}}$ 的系数。

我们尝试枚举出现次数为 1 的元素有多少个，设为 j，这样就有 $\binom{i}{j}$ 的系数。我们要选若干个集合精确覆盖这 j 个元素，设集合数量为 k，也就是说将 j 个元素划分到 k 个无序集合里，也就是斯特林数 $\left\{{j \atop k}\right\}$。另外，这 k 个集合也可以任意包含没有限制的 $n-i$ 个元素，所以还会有 $2^{(n-i)k}$ 的系数。

这样，答案为 $\sum\limits_{i=0}^{n}(-1)^i\binom{n}{i}2^{2^{n-i}}\sum\limits_{j=0}^{i}\binom{i}{j}\sum\limits_{k=0}^{j}2^{(n-i)k}\left\{{j \atop k}\right\}$，容易 $O(n^3)$ 计算。

整理一下可得 $\sum\limits_{i=0}^{n}(-1)^i\binom{n}{i}2^{2^{n-i}}\sum\limits_{k=0}^{i}2^{(n-i)k}\sum\limits_{j=k}^{i}\binom{i}{j}\left\{{j \atop k}\right\}$，最内层的 $\sum\limits_{j=k}^{i}\binom{i}{j}\left\{{j \atop k}\right\}$ 根据组合意义/公式，等于 $\left\{{i+1 \atop k+1}\right\}$，于是可以做到 $O(n^2)$。

还有一种理解方法。你可以直接不枚举出现次数为 1 的元素有多少个，考虑将 i 个元素划分成若干个集合，其中一个集合是“垃圾桶”，扔进去的元素出现次数就为 0。由于这个集合可以是空的，所以需要新增一个元素 0，0 所在的集合就是“垃圾桶”。然后就得到 $\left\{{i+1 \atop k+1}\right\}$。