PTZ winter 2021 Day5 C

假设众数 x 出现的次数为 t，那么我们尝试在 $n-k$ 个数中选出 m 对数，使得每对数两两不同，那么知道 $m \le \min(n-k-t, \lfloor \frac{n-k}{2} \rfloor)$，且可以取等号。

考虑把所有数放在一排，使在一对的数放在相邻的位置，现在有 $n-k+1$ 个前缀和，把其中某对的两个数交换就会多一种前缀和，现在有 $n-k+1+m$ 个不同的前缀和，如果存在两个前缀和模 n 相同，那么这个序列就不合法了，所以 $m<k$，考虑到 $k \le n/4$，所以 $m \ne \lfloor \frac{n-k}{2} \rfloor$，那么就有 $m=n-k-t<k$。所以 $t>\frac{n}{2}$。此时如果 $\gcd(n,x) \ne 1$，那么取 $n/\gcd(n,x)$ 个 x 就模 n 为 0 了，所以知道 $\gcd(n,x)=1$。

那么我们可以让所有数乘上 x^{-1}，此时我们有 t 个 1，所以我们全部数的总和必须小于 n，容易发现这也是必要的，那么方案数就是 $\binom{n-1}{n-k}$，选众数的方案数为 $\phi(n)$。所以答案就是 $\binom{n-1}{n-k}\phi(n)$。复杂度 $O(K)$。

CF1601D

考虑排序这些人，使得答案为原序列的一个子序列。

考虑如果 $s_j \ge a_i \wedge s_i \ge a_j$，那么先放其中一个不会影响另一个数的可加入性，所以他们的顺序无关紧要。

$s_j < a_i \wedge s_i < a_j$，只能加入其中一个，所以顺序也无关紧要。

$s_j < a_i \wedge s_i \ge a_j$，先加入 i 会使 j 放不了，先放 j 对 i 没有影响，所以 j 要在前面。

发现按照 $\frac{a_i}{s_j} < \frac{a_j}{s_i}$ 排序即可满足上述条件，即按 $a_i \times s_i$ 从小到大排序。

剩下的就是线段树优化 `DP`，复杂度 $O(n \log n)$。

AGC036D

考虑 `Johnson` 算法。发现没有负环等价于可以给每个点赋一个势 h_i，使得对于所有边 (u,v,w)，$h_v-h_u+w \ge 0$。

那么就是找到一组势 h_i，如果边 (u,v,w) 使得 $h_v-h_u+w<0$ 则需要删掉这条边，求最小代价。

由于存在 $(i,i+1,0)$ 的边，所以 $h_{i+1} \ge h_i$。

考虑 $i<j$，如果有 $h_j-h_i-1 \ge 0$，则不需要支付贡献。

考虑 $i>j$，如果 $h_j-h_i+1 \ge 0$，则不需要支付贡献。

那么考虑相当于把点划分成若干段，第 i 段势为 i，那么我们需要支付同一段里 -1 的边的代价，以及势相差 2 以上的段之间 1 的边的代价。

朴素的 $O(n^3)$ `dp` 即可。

PTZ summer 2021 Day1 H

每个豆子独立，所以考虑求出每个格子的 `sg` 值。

由于豆子只能走下或在行内左右，所以从下往上 `dp`，设 $f_{i,j}$ 为 (i,j) 的 `sg` 值。

假设第 i 行存在没有盘子的格子，那么设 R_j 表示现在在第 j 列，不能向左走的 `sg` 值，类似的，L_j 表示不能向右走的，转移显然，那么 $f_{i,j}=mex(L_{j-1},R_{j+1},f_{i+1,j})$。

假设这一行全都有盘子，那么上面的做法会发现 L_j, R_j 的转移会成环。发现 `sg` 最大为 3，我们可以考虑求出 $Lt_{j,k}$ 表示当 $L_1 = k$ 时，L_j 的值，同理有 $Rt_{j,k}$ 表示 $R_m = k$，R_j 的值。然后再求出 $Lf_{j,k}$ 表示 $L_j = k$ 时，L_m 的值，同理有 $Rf_{j,k}$ 表示 $R_j = k$ 时，R_1 的值。

由 $f_{i,1} = mex(L_m, R_2, f_{i+1,1})$，那么 $L_m = Lf_{2,mex(L_1,f_{i+1,2})}, R_2 = Rt_{2,mex(R_1,f_{i+1,m})}$。但是此刻我们所求的是 $f_{i,1}$，从 $(i,1)$ 出发时是没有 L_1, R_1 对应的决策的，我们可以把它看作 -1，所以
$f_{i,1} = mex(Lf_{2,mex(f_{i+1,2})}, Rt_{2,mex(f_{i+1,m})}, f_{i+1,1})$。类似的，
$f_{i,m} = mex(Rf_{m-1,mex(f_{i+1,m-1})}, Lt_{m-1,mex(f_{i+1,1})}, f_{i+1,m})$。

那么考虑 $f_{i,j}(1<j<m) = mex(L_{j-1}, R_{j+1}，f_{i+1,j})$，那么
$L_{j-1} = Lt_{j-1,mex(L_m,f_{i+1,1})}, L_m = Lf_{j+1,mex(L_j,f_{i+1,j+1})}$，同样的，$L_j$ 是 -1，则
$L_{j-1} = Lt_{j-1,mex(Lf_{j+1,mex(f_{i+1,j+1})},f_{i+1,1})}$，同理求出 R_{j+1} 即可。

复杂度 $O(nm)$。

CF1383C

考虑删去已经相同的位置，然后每有 $a_i = x, b_i = y$，那么看作 x 向 y 连了一条有向边，接下来对于每个弱连通块讨论。

对于一个有向无环图而言，我们可以用点数减一次操作解决，显然这也是最小的。

对于有环的图，我们可以花两步减少一个点，所以答案的一个上界就是 $2n-1-|LDAG|$，其中 n 为点数，$LDAG$ 为最大的点集，使得它的导出子图是 DAG。

> 结论：当一个弱连通块可以用 k 步解决时，那么一定存在一个它的子集 S，使得 $2n-1-k \le |S|$ 且 S 的导出子图是 DAG，即 $2n-1-|LDAG|$ 就是答案。
>
> 证明：考虑维护每个弱连通块的 $LDAG$ 的大小的和 P，一开始单个点就是一个连通块，当操作一条边 (u,v)。
>
> - 当 u,v 所属连通块不同，那么 P 不变，连通块数减一。
> - 当 u,v 在同一个连通块，把 v 从 $LDAG$ 删去，那么 P 减一，连通块数不变。
>
> 那么用 k 步解决，其中 $n-1$ 步用于删去连通块数，至多 $k-(n-1)$ 步把 P 减小，那么 $n-(k-(n-1)) \le |S|$ 即 $2n-1-k \le |S|$。

那么就找到最大的点集 S，使得这个点集导出子图是 DAG，那么答案就是 $40-|S|-c$，其中 c 是连通块数。

复杂度 $O(n+2^{20}\times 20)$。

PTZ summer 2021 Day3 I

考虑容斥，答案是 $\binom{n}{3}-$ 不合法的对数。考虑不合法的三元组只有三种：两个有交且都和另一个无交，一个和两个都有交且另两个无交，三个两两有交。这三种的个数分别记为 c_1, c_2, c_3。

我们考虑 d_i 为与第 i 个矩形有交的矩形个数，那么考虑 $\sum d_i(n-2) = 2c_1+4c_2+6c_3$，$\sum \binom{d_i}{2} = c_2+3c_3$。那么我们就得到了 c_1+c_2，接下来我们只需要算出 c_3。

考虑枚举哪个矩形的 u_i 是三个矩形里最小的，那么剩下的矩形需要和这个矩形有交，即保证 $d_j < u_i < u_j$，我们可以看作在 u_j 的位置插入了 $[l_j, r_j]$，然后在 d_j 处删去，那么我们就相当于每次查询现在的区间里，有哪些 $[l_j,r_j],[l_k,r_k]$ 使得 $[l_i,r_i],[l_j,r_j],[l_k,r_k]$ 两两有交。那么再次容斥，就变成了选两个区间和 $[l_i,r_i]$ 有交的方案减去选两个区间和 $[l_i,r_i]$ 有交且这两个区间无交的方案，前者可以直接算，后者考虑相当于满足 $l_i < r_j < l_k < r_i$，那么用线段树维护每个区间里有多少个区间的左端点和有多少个线段的右端点即可。

复杂度 $O(n\log n)$。

CF1246F

考虑求出 $[L_{i,k}, R_{i,k}]$ 表示那些至多 k 步可以走到 i 的点的形成的区间，那么答案就是 $\sum_{k=0}(n-1)-(R_{i,k}-L_{i,k})$。

考虑倍增求和，但是 L, R 计算并不独立，但是发现 $L_{i,k}=\min_{j\in[L_{i,k-1},T]} L_{j,0}$，其中 T 是最小的满足 $[L_{i,k-1}, T], [L_{i,k-1}, R_{i,k-1}]$ 字符集相等的位置，所以假如字符集没有变化，L, R 就是独立的。字符集变化只有 $O(\Sigma)$ 次。

复杂度 $O(n\Sigma\log n)$。

World Tour Finals 2019 Open Contest D

考虑我们二分答案 K。

我们考虑给两维分别赋一个权值，设为 (p,q)，然后我们找到使得 $(x_i,y_i)\cdot(p,q)$ 最小的那 K 个 (x_i,y_i)，把 $(\sum x_i, \sum y_i)$ 画在平面上。显然这些点会形成一个下凸壳，那么 K 可行当且仅当 (R,B) 在凸壳右上方。

考虑我们直接求出整个凸壳非常不现实，那么我们可以考虑二分 (p,q) 的斜率 x，假如按现在二分的 (p,q) 算出来 $(\sum x_i, \sum y_i)$ 在 (R,B) 左下方，那么已经知道 K 可行了，否则如果在 (R,B) 右上方则 K 一定不可行。在另外两个区域就分别移动二分指针即可。

把斜率当作实数二分很蠢，我们可以假装 $p+q=10^9+7$，然后二分 p，由于两数和足够大，所以有用的斜率一定能二分到。

现在我们要求使得 px_i+qy_i 最小那些 (x_i,y_i)，那么二分 z，求出 $px+qy\le z$ 的点数以及他们的两维和（由于 p,q 够大，所以你可以认为 $px+qy$ 两两不同），可以使用类欧，或者注意到你只会用 $\min(x,y)\le 2000$ 的那些点，直接枚举其中一维。

复杂度 $O(\log^4 W)$ 或 $O(W^{\frac{1}{3}}\log^3 W)$。