mines

【做法1】

直接2^n枚举所有情况验证。

时间复杂度$O(2^n \times n^2)$，期望得分20分。

【做法2】

如果x爆炸后能使y爆炸，即$p_y \in [p_x - r_x, p_x + r_x]$，则连一条$x$到$y$的有向边。

对于同一个强连通分量中的点，只要有一个爆炸，所有都会爆炸，故每一个强连通分量的花费即为其中节点花费的最小值。

$O(n^2)$的枚举两个点连边，然后缩强连通分量得到一个拓扑图。

答案即为拓扑图中所有入度为0的强连通分量的花费和。

对于修改操作，可以对每个强连通分量开一个set来记其中所有节点的花费，每次取set中最小的即可。

时间复杂度$O(n^2 + Q \log n)$，期望得分50分。

【做法3】

考虑优化做法2，用线段树优化连边。

时间复杂度$O(n \log n)$，期望得分100分。

permutation

【做法1】

暴力枚举所有的排列。

时间复杂度$O(N! \times N)$，期望得分20分。

【做法2】

显然每次询问的答案只与y有关。

考虑枚举P_y，则$1 \le P_x \le [\frac{P_y-1}{2}]$。

$ans_y = \sum_{i=y}^{n} [\frac{i-1}{2}] * \frac{(i-2)!}{(i-y)!} * (n-y)!$

对于每组询问可以$O(n)$计算答案

时间复杂度$O(NQ)$.期望得分60.

【做法3】

考虑对这个式子进行化简

$ans_y = (n-y)! * \sum_{i=y}^{n} [\frac{i-1}{2}] * \frac{(i-2)!}{(i-y)!}$

$ans_y = (n-y)! * \sum_{i=0}^{n-y} [\frac{i+y-1}{2}] * \frac{(i+y-2)!}{i!}$

设$f_i = [\frac{i-1}{2}] * (i-2)!, g_{n-i} = \frac{1}{i!}$

$ans_y = (n-y)! * \sum_{i+j=n+y} f_i * g_j$，这是一个卷积，可以用$NTT$优化。

时间复杂度$O(N \log N)$，期望得分80分。

【做法4】

考虑所有满足p_y是前y项值中最大值全排列，共$\frac{n!}{y}$个，对于p_x的值，可以分

为3类：

(1) $2\cdot p_x\le p_y$，

(2) $2\cdot p_x=p_y$，

(3) $2\cdot p_x\ge p_y$。

根据对称性，第(1)类和第(3)类的全排列个数是相等的，所以我们计算第二类全排列的个数f_y，答案即为$\frac{\frac{n!}{y}-f_y}{2}$

考虑$p_y=2k,p_x=k$的情况，y之前除x外剩余$y-2$个位置需要在$2k-2$个数中选取$y-2$个，方案数$=C_{2k-2}^{y-2}\cdot(y-2)!\cdot(n-y)!$

第二类全排列的总个数$=\sum_{k=1}^{n/2}C_{2k-2}^{y-2}\cdot(y-2)!\cdot(n-y)!$

现在只需对于每个y，求解$\sum_{k=1}^{n/2}C_{2k-2}^{y-2}$

设$m=[n/2],g_x=\sum_{k=0}^{m-1}C_{2k}^{x}$，根据组合数学，可以推出$g_{x-1}+2g_x=C(2m,x+1)$，故可以按顺序求解$g_i$ 的值。

最后，$f_y=g_{y-2}\cdot(y-2)!\cdot(n-y)!$

时间复杂度$O(n)$，期望得分100分。

【题目链接】

$https://csacademy.com/contest/archive/task/permutations/statement/$

square

【做法1】

从初始状态开始bfs

时间复杂度$O(3^{NM} * NM)$，期望得分20分。

【做法2】

我们可以发现，每行或者每列肯定至多染一次。不妨枚举染色的顺序，一共有$(N + M)!$种。

但是如果要枚举颜色就太慢了，我们倒着来看。如果这一行是最后一个被染，那么这一行的颜色肯定都是一样的。如果是倒数第二个，那么除了被最后一个覆盖的部分之外其他部分也应该是相同的。

我们只要模拟这个过程就行了。如果操作到某一步所有格子都被访问了，那么就结束了。

时间复杂度$O((N + M)! * (N + M))$.期望得分30.

【做法3】

从做法2中受到启发，我们可以得到一种判断是否存在解的方法:每次删去完全相同的一行或一列，直到整个矩阵变白。

那么如果有两行同时出现这种情况，先删哪一个呢?在判断可行性时，其实这对我们是无所谓的。

具体做法：首先将所有已经可以删的加入队列，然后开始bfs，每次删去之后将新产生的可以删去的加入队列，最后判断一下队列是不是满的就行了。

时间复杂度$O(NM)$,期望得分20.

【做法4】

现在问题已经转化成了求最小的染色数。

首先有这样一个结论:一个合法矩阵的任何一个子矩阵都是可染色的。

另外行染色和列染色一定有一种是用满了的。

不妨假设每行都被染了一遍色，现在的问题是列上最多能省下多少。

如果一列上没有染过色，那么这些列都是一样的，那么问题其实就是最多能有多少列是一样的。

这个问题可以用$trie$树来处理。

设最多有x行完全一样，y列完全一样。

所以答案就是$N + M - max(x, y)$

时间复杂度$O(NM * max(N, M))$或$O(NM)$,期望得分$70 - 100$.

【题目链接】

$https://csacademy.com/contest/archive/task/matrix\_coloring/$