T1 体育课

考察左上角在 $(i,j),(i+1,j),(i,j+1),(i+1,j+1)$ 的 $r\times c$ 的子矩形，计算 $(i,j)+(i+1,j+1)-(i+1,j)-(i,j+1)$，其中的运算指子矩形的对位相加相减，可得 $a_{i+r,j+c}=a_{i+r,j}+a_{i,j+c}-a_{i,j}$。

将下标从 0 开始标号，可以推出对于 $i\geq r,j\geq c$，$a_{i,j}=a_{i\%r,j}+a_{i,j\%c}-a_{i\%r,j\%c}$。

即确定了前 r 行和前 c 列的元素后，可以求出所有元素，但可能无解：

对于所有 $x\in[0,r),y\in[0,c)$，记 $b_{i,j}=a_{ir+x,jc+y}$，$a_{i,j}\in\{0,1\}$ 会带来限制：$\forall i,[b_{0,0}=b_{i,0}]$ 或 $\forall j,[b_{0,0}=b_{0,j}]$，即 b 要么每行一样，要么每列一样

枚举每个 (x,y) 是每行一样还是每列一样，考察原题限制，计算 $(x,j+1)-(x,j)$ 和 $(i+1,y)-(i,y)$

具体来说，分别用 col_j 和 row_i 表示第 j 列上 "列相同的位置" 有几个、第 i 行上 "行相同的位置" 有几个。

1. 先算 "列相同的情况"，即同一行的可以随便填，每一列枚举对应的位置（注意只考虑 "列相同" 的这 col_i 的位置）有几个填 1，接下来后面对应位置列 1 的个数都要和当前列相同，记 $num=\lfloor\frac{m-i+c-1}{c}\rfloor$ 为这些列的个数，方案相加后每列求积即可，即 $\prod_{i=0}^{c-1}\sum_{j=0}^{col_i}\binom{col_i}{j}^{num}$
2. 对于 "行相同的情况"，这里需要容斥，即在计算时要减去可以作为 "列相同" 的情况。用 $row_i=3$ 举例：类似的考虑前 m 列的每一行随便填，首先一行（对应位置上）的值不能为 0 或 3，不然所有位置都要填 $0/1$，和 "列相同" 一样，然后还要考虑某一位置一列下来都一样的情况，该位置也是 "列相同" 了，这边也要做个小容斥（3 个都 "列相同"），最后的权值即为 $2\times 3^{num}-6\times 2^{num}+6$

   其它 row_i 情况类似。

时间复杂度 $O(2^{rc}rc)$

T2 优化采购

先考虑所有非叶子节点构成一条链的情况，相当于从链底开始，维护一个变量 x ，每次经过一个点将 $x\leftarrow|x-a_i|+b_i$

我们离线所有询问统一处理，经过一个点时，不同的询问所被修改的 $(a_i,\ b_i)$ 可能不同，但所有相同的 $(a_i,\ b_i)$ 形成了时间上连续的一段，而每个点的段数之和为 $O(n+q)$ 。我们按时间建线段树，每个连续段将时间在一段区间的询问作用修改 $(a_{i,\ t},\ b_{i,\ t})$ ，即区间修改。若记区间内每个 x 的最小值为 $\min$ ，最大值为 $\max$，若 $a_i\leq\min$ 或 $\max\leq a_i$，可以直接打区间乘 -1、区间 $+$ 的 tag，否则直接向子树递归。

注意到每一次向子树递归都会导致区间内 x 的极差减少至少 1，而所有线段树节点对应区间的极差之和是 $O(nv\log n)$ 的，故复杂度为 $O((n+q)v\log n)$

对于普通的二叉树，仍然用线段树维护仅考虑点 u 子树时所有询问的答案，原先的 $(a_u,\ b_u)$ 操作对变成了 u 的两个儿子的线段树对应位置相减。考虑线段树合并，在合并两棵线段树的对应区间时，如果某棵线段树区间内所有权值相等，可以转化成 $(a,\ b)$ 操作对，类似链的情况区间操作即可。考虑复杂度证明，一个询问 $(u,\ x)$ 会给 u 对应的线段树的 $\log q$ 个节点增加至多 v 的势能；而合并两棵线段树时，每当第一棵树节点 $[l,r]$ 内所有权值相等，向第二棵线段树节点 $[l,r]$ 下递归时，第二棵线段树节点 $[l,r]$ 的势能至少会减少 1。

时间复杂度 $O((n+q)v\log q)$，空间复杂度 $O((n+q)\log q)$

T3 尝试了飞行

考虑性质 A，将所有点按拓扑顺序考虑，记 $f[i][j]$ 为第一条路径结尾在 i，第二条路径结尾在 j 的方案数。

记当前枚举的点是 k，前 $k-1$ 个点都被至少一条路径覆盖，考虑之前两条路径的结尾，有以下情况：

- 从 $k-1$ 对应路径延伸而来，$f[k][x] \leftarrow f[k-1][x]$，$f[x][k] \leftarrow f[x][k-1]$
- 从另一条路径延伸而来，$f[k][k-1] \leftarrow f[x][k-1]$，$f[k-1][k] \leftarrow f[k-1][x]$
- 从以上两条路径延伸而来，$f[k][k] \leftarrow f[k-1][x], f[x][k-1]$

可以 $O(n^2)$ dp。

考虑存在环的情况，由于每个强连通分量是一个简单环，沿用上述转移，即每次将两条路径结尾所属的环所属的路径移动到下一个环。

将 $f[i][j]$ 的状态改为，第一条路径结尾在点 i，第二条路径结尾在点 j，当前没有在环上移动过，且拓扑序前 $\min(c_i, c_j) - 1$ 个环上的点都合法的方案数。

记 i 转移到的下一个点为 k，若不在同一个环上则和和无环的转移类似，否则：

如果当前 j, k 只有一个点在环上，不妨设其为点 j。考虑这个点在环上走的部分的结尾 l。如果 l 不是 j 在环上的前一个点，不绕圈不能覆盖所有点，因此有 $k-1$ 种方案，否则正好有 k 种方案。

考虑两个点都在环上走的部分，记环长为 l，点的编号依次为 $0, 1, \cdots, l-1$。

假设起点为 a, b，终点分别为 c, d，将环分成 $[a, b-2], b-1, [b, a-2], a-1$ 四段，考虑 c, d 在环上的哪一段，可以发现若 c, d 所在段固定，则方案数是固定的，当 $k > 1$ 时，只可能是 $\frac{k(k-1)}{2}, \frac{k(k-1)}{2} - 1, \frac{k(k+1)}{2}, \frac{k(k+1)}{2} - 1$ 中的一种。

以上常数种转移可以二维前缀和优化，时间复杂度 $O(m^2)$。