注：若评测机器性能有所差别可以适当更改时限. 下为组题人本机下运行时限：

- floral 0.5s
- corolla 0.5s
- bloom 0.5s

花卉港湾

首先我们求出直径. 令其两个端点为 a, b. 那么如果 u 不可到达 a, b 中的任意一个点（即 $dis(u, a) \leq K$ 且 $dis(u, b) \leq K$）那么就 -1 了. v 同理. 否则 u, a, b, v 四个点就是联通的，于是答案 ≤ 3. 答案是否 $= 1$ 是容易判断的，所以我们聚焦于如何判断答案是否 $= 2$.

这就意味着我们要看能不能找到一个点 w 使得可以 $u \to w \to v$. 我们把直径 $p_1, \ldots, p_m$ 拉出来，那么树上的每个节点 s 应该能唯一对应到一个 c_s，使得如果把直径边都删掉的话那么 s 在 p_{c_s} 为根树中. 形象地说就是 s 挂在 p_{c_s} 上. 令 g_s 表示 s 到 p_{c_s} 的距离. 那么同时每个 p_i 也能算出一个 h_i 表示 $c_s = i$ 中的 g_s 的最大值. 那么我们的问题其实就等价于是否存在一个 $k \neq c_u, c_v$ 并且 $|k - x_u| + g_u + h_k \geq K$ 且 $|k - x_v| + g_v + h_k \geq K$. 注意到它是大于号，所以绝对值可以直接拆掉，然后变成四个分类讨论. 问题实际上可以化简成如下形式：限定 $k + h_k \geq W_0$ 时，$h_k - k$ 的最大值；$h + h_k$ 的最大值；$h_k - k$ 的最小值. 后两个很简单，第一个需要上一个线段树.

礎石花冠

首先一个绕不开的问题是求该图的强连通分量. 考虑使用 Kasaraju 法求强连通分量，那么我们每次要做的就是找一条未访问过的出边. 考虑分别用线段树维护区间内 a_i 及 b_i 最大/最小的还未访问过的点即可. 这一步所需要的复杂度是 $O((n + m) \log n)$ 的.

求出强连通分量之后每个强连通分量用一个环来代替一定是最优的. 剩下的问题是如何对于一个 DAG 求 $f(G)$.

对于 G 求 $f(G)$ 其实没有什么最优性可言 —— 对于一条边 $u \to v$，其能被删除当且仅当存在一条 > 1 条边的 $u \to v$ 的路径.

对于一般图这个问题是 $O(|V||E|)$ 的. 我们叙述一种容易在我们这个问题上优化的做法. 假设点的编号按拓扑序正序排列，那么我们从后往前做.

$[i + 1, n]$ 不会指向前面，所以我们对 $[i + 1, n]$ 求出的子图 G' 满足局部的最优性. 我们现在考虑往 G' 中加入 i 之后会发生什么. 我们把 G' 给 copy 一份拉出来，令其为 G''. 我们先考虑所有在 G'' 中的零入度的点 x，如果存在边 $i \to x$，那么显然在 G' 上应该连上 $i \to x$，那么此时所有能被 x 指向的点的可达性就也解决了. 如果不存在边 $i \to x$，那么我们就在 G'' 中把 x 这个点给删除了，然后这样 x 指向的点的入度就会 -1. 不断这样找零入度的点，直到没有需要继续处理的零入度的点了. 那么此时现在显然 G'' 中的点在原图上和在 G' 上都是被 i 可达的. 于是我们就完成了加入 i 的操作.

但这题我们是从一个竞赛图中删去了若干条边. 实际上，如果我们仔细分析一下，上述操作的复杂度（如果实现地好的话，比如你没必要真的去 copy 成 G'' 而是只需要维护一下每个点的 deg）是 $O((n + m)\sqrt{m})$ 的.

首先我们发现加边的操作只会发生 $O(n+m)$ 次（题目中已经说过了），那么我们看除了加边我们还会做什么操作. 实际上唯一影响我们复杂度的操作就是 “删除 x 并减小其指向的点的 deg” 的操作. 那么一个这样的 x 能有多少出边呢？我们发现对于 x，其在 G' 上指向的点集 S_x 应该满足两两不可达，这意味着任意 $u,v\in S_x$ 应当满足 $(u,v)\in E'$，其中 E' 为题目中删的边集. 那么于是 $|S_x|\le\sqrt{|E'|}=\sqrt{m}$.

所以我们理论上要做的操作只有 $O((n+m)\sqrt{m})$ 次. 而一个小细节是，我们查询两个点是否有边的操作其实是查询两个强连通分量是否有边. 查询这个实际上只需要提前求出两个强连通分量之间有多少条边 $\in E'$，然后看一下是否等于 $sz_x\times sz_y$ 就行了.

磷磷开花

说不定可能比 T2 简单. 谁知道呢.

我们先看所有 a 都一样的情况. 这也就意味着每个元素都是没有区别的. 那么我们不妨假设钦定第 i 个培养皿是恰好第 i 个装入细胞的，然后最后再乘上 $n!$ 就是原本的答案.

那么在时刻 t，编号为 $[1,i]$ 的培养皿有细胞，并且尚未选取的二元组集合为 S，其中 $t=\frac{n(n-1)}{2}-|S|$，那么我们就可以有个简单的暴力 DP：$f(i,S)$ 表示达成这个状态的方案数，然后转移直接枚举选择了哪个二元组即可.

考虑优化. 我们把 S 拆成 $A+B+C$，其中 $A\subset[1,i]\times[1,i]$，$B\subset[1,i]\times[i+1,n]$，$C\subset[i+1,n]\times[i+1,n]$. 那么我们首先发现我们其实根本不关心 A 是什么，只关心 $|A|$，令其为 a. 然后我们还能发现只要两个 $|C_1|=|C_2|$ 那么 $f(i,A,B,C_1)=f(i,A,B,C_2)$，也就是说 f 也只与 $|C|$ 有关. 于是 A,C 两个维度都被优化掉了，我们研究一下最棘手的 B.

一个优化方向是还是研究什么样的 B_1,B_2 会满足 $f(i,A,B_1,C)=f(i,A,B_2,C)$. 考虑对于任意 $x\in[1,i]$，令 $b(x)$ 表示有多少 $(x,*)\in B$，那么我们可以类似地发现只要两个 B 的 b 序列一样，那么其 DP 值就一样. 更进一步地，我们实际上只关心所有 $\{b_x\}$ 形成的集合而不是序列，所以只要两个 B 的 b 集合相同，那么其 DP 值就是一样的.

于是我们考虑我们的 DP 如何转移. 首先 $a\times f(i,a,b,c)\to f(i,a+1,b,c)$，$c\times f(i,a,b,c)\to f(i,a,b,c+1)$ 两个最基本的转移只是操演定义，不再赘述. 然后考虑会使得 $i+1$ 上有细胞的转移，即使得 b 变动的转移（注意我们钦定了装入细胞的顺序所以只能是 $i+1$ 上被复制上细胞）.那么这个时候我们就需要枚举一个集合 $S\subset[1,i]$ 表示钦定 B 中有且仅有 $x\in S$ 的元素满足 $(x,i+1)\in B$，这个事情的概率是 $\prod_{x\in S}\frac{b_x}{n-i}\prod_{x\notin S}\frac{n-i-b_x}{n-i}$，然后再枚举一个数 k 表示 C 中出有多少个 $(x,*)$，这个事情的概率是 $P(i,c,k)=\frac{\binom{n-i-1}{k}\binom{(n-i-1)(n-i-2)/2}{c-k}}{\binom{(n-i)(n-i-1)/2}{c}}$. 对于给定的 k,S，我们会将概率乘上 $|S|\times f(i,a,b,c)$ 贡献给 $f(i+1,a+|S|-1,\{b_x-[x\in S]\}\cup\{k\},c-k)$.

这个转移复杂度有点略大了. 首先我们发现我们可以将枚举 S 和枚举 k 的这一步给拆开来.

然后注意到我们可以用一个 DP 来将上面那个枚举 S 给优化掉. 我们令 $g(k,a,b,c,0/1)$ 表示仅考虑前 k 个元素的贡献和. 那么就有如下的转移（其中 b' 表示 b_k-1 之后的 b）：

- $\frac{b_k}{n-i}g(k-1,a,b,c,0)\to g(k,a+1,b',c,1)$ （$(k,i+1)\in B$ 且这一轮恰好选择的是这个 pair)

- $\frac{b_k}{n-i}g(k-1,a,b,c,z) \to g(k,a+1,b',c,z)$ ($(k,i+1) \in B$ 且这一轮没选这个 pair)
- $(1-\frac{b_k}{n-i})g(k-1,a,b,c,z) \to g(k,a,b,c,z)$ ($(k,i+1) \notin B$)

然后最后我们 DP 完 g 之后要贡献回 f 上. 具体来说，就是枚举 k 然后将 $P(i,c,k)g_{a,b,c,i,1} \to f_{i,a-1,b\cup\{k\},c-k}$.

剩下的就是如何去实现这个代码了. 有若干细节需要注意，比如 g 直接开是开不下的但是 c 在转移中不起到作用所以可以省去一维，等等. 最终复杂度 $O(n^5M)$，其中 M 为所有 b 的状态的可能数总和，为 $O(2^n)$. 常数比较小，但是也和具体实现有关，所以时限略开大了一些.

看上去比较复杂，但想到优化掉 B 这一维度之后，剩下的优化都是自然的. 上面只是非常细致具体地写下了应该如何做转移而已.