Solution

1 circle

显然，如果钦定的 k 个点如果不是一个 DAG ，那么答案必定是 `impossible` ，也就是说，如果答案不是 `impossible` ，那么钦定的 k 个点也一定是一个 DAG 。那么题意就可以转化为这样:

给定一个 n 个点的竞赛图，每个点有黑白两种颜色，保证黑点和白点的生成子图是一个 DAG ，现在要删除尽量少的黑点使得整个图都变成一个 DAG 。

首先先证明一个众所周知的引理。

Lemma 1 对于一个 n 个点的强连通竞赛图，一定存在长度为 $3 \le i \le n$ 的环。

Proof 1 显然可以用归纳法来证明。首先在 $n = 3$ 时显然成立，考虑一个 n 个点的强连通竞赛图，随便选一个点 x 然后将其删掉，剩下的图假设分为了 $A_1, A_2, A_3, ..., A_m$ 这些强连通分量，并且对于 $\forall i < j, a \in A_i, b \in A_j$ ，存在 $a \to b$ 的边。

由于这是一个强连通图，则一定 $\exists\, a \in A_m$ ，存在 $a \to x$ 的边，同理一定 $\exists\, a \in A_1$ ，存在 $x \to a$ 的边。那么我们就一定可以构造出一条长度为 n 的环形如 $x \to A_1 \to A_2 \to ... \to A_m \to x$ 。又根据归纳法，每一个子强连通分量都满足引理介绍的性质，且每一个子强连通分量我都可以跳过，那么显然对于长度为 $3 \le i \le n$ 的环我都可以构造出来。

有了这个引理之后，我们的目标就从删除所有的环变为了删除所有的三元环，因为任意一个长度大于 3 的环都是一个强连通分量，一定包含了长度为 3 的环。

不过图中本质不同的三元环只有两种，一种是由两个黑点和一个白点构成的，另一种则是由一个黑点和两个白点构成的。显然后者对应的黑点是必须要删去的，前者对应的两个黑点必须得删掉其中一个，然而删掉哪一个却很难确定下来。

不妨考虑将白点和黑点的拓扑序全部求出来，设白点的拓扑序列为 X ，黑点的拓扑序列为 Y ，然后考虑一个特定的黑点 x ，显然 x 与任意的一个白点之间都有边相连。我们记黑点 $\to$ 白点的边为 `A` 类边，白点 $\to$ 黑点的边为 `B` 类边。

然后我们可以观察到一个显然的性质： x 必须被删去当且仅当 $\exists\, i, j \in X, pos_i < pos_j$[1] ，存在 $x \to i$ 的边且存在 $j \to x$ 的边。也就是说将 x 到序列 X 的每一个点的边的类型写成一个序列，那序列必将存在这样的一段 `AAA...AAABBB...BBB` 。换句话说，如果出现了 `A` ，那么之后如果一旦出现了一个 `B` ，x 将必定被删掉。

那么我们首先把必定被删掉的点先删完，那么剩下的点对应的边类型的序列将形如 `BBB...BBBAAA...AAA`（也可能没有 `B` 或没有 `A` ） 。之后不妨我们考虑两个黑点怎样能共存，不难发现，如果我们将每个黑点对应的边类型的序列中第一个 `A` 的位置记下来，设为 f_i ，那么对于 $\forall i, j \in Y, pos_i < pos_j$[2]，那么 i, j 能共存当且仅当 $f_i \le f_j$ 。于是对 f 数组求一个 `lis` ，和 k 比较一下，这道题就做完了。时间复杂度 $O(n^2)$。

[1] pos_i 表示 i 在对应序列中的位置，即下标。

[2] 这里同上

2 color

首先考虑每一种合法的染色方案，假设第一列的颜色种类数为 a ，那么剩余的部分也只能有 a 种颜色。

然后考虑前两列的颜色种类数，记为 x ，那么显然有 $x \geq a$ ，然后剩余的部分的颜色种类数记为 y ，也显然有 $y \leq a$ 。又因为这是一个合法方案，那么显然有 $x = a, y = a$ 。前两列的颜色种类数一定等于第一列的颜色种类数，那么说明第二列的颜色一定在第一列中都出现过。同理可以证明：对于 $\forall i \in [2, m-1]$ ，第 i 列出现的所有颜色一定在第一列中出现过，同时也在最后一列出现过。那么也就是说，第一列和最后一列的颜色是独立的，而中间部分能够使用的颜色则是这两列颜色的交集。

考虑首先枚举 a 确定第一列和最后一列的颜色种类数，然后再枚举 b 表示两列颜色的交集大小，那么中间部分的方案数显然为 $b^{n(m-2)}$ ，考虑第一列和最后一列的颜色方案数如何来计算，显然第一列和最后一列本质相同，因此方案数也相同，其次方案数等价于 n 个格子用恰好 a 种颜色来染色的方案数，这个显然等于 $S(n,a)a!$ ，其中 S 表示第二类斯特林数。于是最后答案即

$$\sum_{a,b} \binom{k}{a} \binom{a}{b} \binom{k-a}{a-b} (S(n,a)a!)^2 b^{n(m-2)}$$

考虑枚举 a, b 直接计算即可。时间复杂度为 $O(T(n^2 + n\log nm))$ 。注意 $m = 1$ 和 $m = 2$ 的特殊情况。

3 simulate

首先不要对轮去考虑，不要将问题变得有阶段性而变得不好做，题目的本意就是不断选择一个元素，将其减 2 并将两旁的元素加 1 ，使得操作之后所有的元素都非负，直到不能继续操作为止。那么可以看出，操作本质上是没有先后顺序的，我们就可以按照从 1 到 n 的顺序去操作。

考虑我们已经操作到第 i 个位置， 1 到 $i-1$ 都已经变成了 0 或 1 ，但第 i 个位置上的数可能很大。考虑用一个栈去维护 0 的位置，然后仅仅考虑栈顶元素到 i 的这么一段，假设它是长成这样的一个东西： `0 1 1 1 1 1 4 x` ，当前的 $a[i]$ 是 4 ，假设 $a[i+1]$ 是 x 。考虑我们对 $a[i]$ 进行操作： `0 1 1 1 1 1 4 x` $\rightarrow$ `0 1 1 1 1 2 2 x+1` $\rightarrow$ `0 1 1 1 2 0 3 x+1` $\rightarrow$ `0 1 1 2 0 1 3 x+1` $\rightarrow \cdots \rightarrow$ `1 0 1 1 1 1 3 x+1` 。也就是说，我们操作一次会把 $a[i]$ 减 1 ，同时会把 $a[i+1]$ 加 1 ，以及会把离 i 最近的一个 0 向后移动一位。然后如果当 0 移动到 $i-1$ 时，下一步就会把 $a[i]$ 减 2 ，把 $a[i+1]$ 加 1 ，然后把栈顶的 0 删掉。然后直接模拟这个过程就好了。时间复杂度 $O(n)$ 。