# 1001 Almost Acyclic

首先，无环或所有简单环的交集非空，等价于存在一个点，删掉它以后图无环，称一张图中这样的点为关键点。

引理：合法的图除了树，基环树，关键点数 $\leq 2$。

证明：除了树，基环树以外的联通图，边数都至少为点数 $+1$。取该连通图的任意生成树，以及考虑任意两条非树边 $(u_1, v_1), (u_2, v_2)$。那么 $u_1 \to v_1$ 树上简单路径所有点构成简单环，$u_2 \to v_2$ 同理。显然关键点至少得同时属于这两条简单路径，设这两条简单路径的交为 $p \to q$（我们知道树上两条简单路径的交要么为空，要么仍为一条路径）。我们声称关键点只可能是 $p, q$，为此需要说明 $p \to q$ 路径上其他点不可能为关键点，这只需要考虑删掉其他点，$u_1 \to u_2, v_1 \to v_2$ 的树上简单路径都得以保留，$u_1 \to v_1, u_2 \to v_2$ 这两条非树边也都得以保留，于是一定存在一个同时包含 $u_1, v_1, u_2, v_2$ 的简单环（这四个点可能有一些是相同的）。

引理的证明可以画图理解。

根据引理，原问题的答案可以写为 $ans_1 - ans_2 + (\binom{n}{2} - n + 1)tree + \sum\limits_{i=1}^{n}(\binom{i}{2} - i + 1)cirtree_i$。

其中 $ans_1$ 的含义是，对所有点 $u$，$u$ 是关键点的子图数量之和；$ans_2$ 的含义是，对所有无序点对 $(u, v), u \neq v$，$u, v$ 同时为关键点的子图数量之和；$tree$ 的含义是原图的生成树数量，$cirtree_i$ 的含义是原图的有恰 $i$ 个点在环上的生成基环树数量。

接下来考虑如何计算公式中的每一项，$tree$ 可以用矩阵树定理 $O(n^3)$ 计算。$cirtree_i$ 的计算是 2015 年的杭电多校题 HDU5304 ~~所以 2023 年多校又出这个题是冷饭热炒~~，在此简单介绍做法：枚举点集 $S$ 作为基环树的环，那么将把 $S$ 排列成环的方案数乘上把 $S$ 点集缩成一个点的生成树数量，就是环点集为 $S$ 的基环树数量。把 $S$ 缩成一个点的生成树数量指的是，认为原图中 $S$ 以外的点间的连边保持不变，$S$ 内部的边被删去，新建大点 $0$，$0$ 向所有不在 $S$ 的 $u$ 连若干条重边，边数是 $u$ 与 $S$ 内点连边的边数，对这张图生成树计数。把 $S$ 排列成环的方案数可以状压 DP 算，计算方式类似哈密顿路计数。$cirtree_i$ 的计算可以在 $O(2^n n^3)$ 时间内解决。

$ans_1, ans_2$ 的计算需要首先求出原图所有子集的生成树数量，这步预处理也可以 $O(2^n n^3)$ 计算。

具体求 $ans_1$ 的方法：按照定义枚举 $u$，考虑删掉 $u$ 后图变为森林，森林有很多连通块，这些连通块的点集两两无交，且并为全集去掉 $u$。每个连通块内部首先有生成树的方案数，还要乘上连通块内选至少一条边连向 $u$ 的方案数（不然最终图不连通），不同连通块的方案数可以直接相乘。那么预处理每个连通块的方案数后，问题转化为子集 `exp` 问题，即每个子集有一个权值，对所有全集的划分方案，求所有子集权值之积的和，这个问题有简单的 $O(3^n)$ 做法。这一步 $O(n3^n)$。

求 $ans_2$ 的方法与 $ans_1$ 类似：枚举点对 $u, v$，同样考虑删掉 $u, v$ 后图的所有连通块，$u$ 不能同时向某个连通块内的两个点连边，否则删掉 $v$ 仍有环，$v$ 同理。那么一个连通块内应该选出不超过一个点与 $u$ 连边，不超过一个点与 $v$ 连边，并且不能都没有。类似地也能转化为子集 `exp` 问题。但有一个特殊 case：如果所有连通块都只向 $u$ 或 $v$ 连边，且 $u, v$ 间没有边，那么图不联通非法。这个特殊 case 可以发现图一定是两棵树构成的森林，于是也可以简单地容斥掉。这一步 $O(n^2 3^n)$。

综上所述，该问题可以在 $O(n^2 3^n)$ 内解决，事实上使用 $O(n^2 2^n)$ 的子集 `exp` 可以得到渐进意义下更优的 $O(n^4 2^n)$ 做法，但在本题数据范围下不如前者。

# 1002 Assignment

原题中的最大不同位限制 $k$ 在本文中是 $K$，设 $a$ 进行一系列操作后变为了 $c$。

首先 $a$ 的形态其实不是很重要，假设能对每个 $1 \le l \le r \le n, 0 \le k \le K$ 算出，对初始全为 $0$ 的子段 $[l,r]$，染成与 $b[l,r]$ 有至多 $k$ 个位置不同的最小染色代价和 $dp_{i,j,k}$。那最后考虑一下所有被染色过至少一次的位置构成的连续段的形态，设 $res_{i,k}$ 为把 $a[1,i]$ 染成与 $b[1,i]$ 有至多 $k$ 个位置不同的最小染色代价和，枚举 $i$ 位置是从来没有被染色过的情况，从 $res_{i-1}$ 直接转移；还是被染色过，并且染色连续段延申到 $j \le i$，用 $res_{j-1}$ 与 $dp_{j,i}$ 合并的转移即可。

那么这个 $dp$ 数组怎么算呢？首先要用到一个引理是，存在一组最优解，进行的所有操作的区间，两两不交或包含。证明只需考虑对于一对相交但不包含的区间 $[l_1,r_1],[l_2,r_2]$，其中 $l_1 \le l_2 \le r_1 \le r_2$，假设 $[l_1,r_1]$ 较早，那么将 $r_1$ 调整至 $l_2-1$ 对最终 $c$ 形态毫无影响。不断进行这样的调整，不会使答案变大，且所有操作区间的总长不断严格减小，于是这个调整一定会结束。最终得到了一组符合引理的操作方案。

另一个引理是对于一对包含关系的区间 $[l_2,r_2] \in [l_1,r_1]$，一定是 $[l_1,r_1]$ 对应操作先执行，否则直接删去 $[l_2,r_2]$ 的操作仍合法但答案减小。

最后一定存在一组最优解，任意两个操作不共左端点，否则选较早的操作右移左端点仍能调整。

对于 $[l,r]$ 这个子段，要么存在一次操作 $[l',r']=[l,r]$，要么一定存在一个 $p \le l < r$ 使得所有操作要么 $r \le p$ 要么 $l > p$。这一点根据引理可以直接得到。

对于第二种情况，可以直接枚举 $p$ 然后用 $dp_{l,p}, dp_{p+1,r}$ 对第二维进行 $\min,+$ 卷积合并得到。

对于第一种情况，一定存在一组最优解使得 $[l,r]$ 这一次操作对应的 $x$ 为 $b_l$，因为 $c_l$ 一定为 $x$，如果 $c_l \ne b_l$，那么 $[l,r]$ 这一次操作右移1 左端点仍合法且可以被第二种情况贡献到。

设 $mn_{l,r,k}$ 为在最开始进行一次 $[l,r]$ 全部变为 $b_l$ 操作的前提下，染成与 $a[l,r]$ 有至多 $k$ 个位置不同的 $c$ 的最小染色代价和，根据上述引理（主要是操作区间两两不交或包含）可以用 $mn, dp$ 转移 $mn$。具体转移参考代码。

时间复杂度 $O(n^3k^2)$。

# 1003 Many Topological Problems

假设我们已经确定了每个点 $u$ 的点权 $a_u$，怎么求出在这种情况下有多少合法的树使得 $a$ 是这颗树的 $k$ 合法拓扑序?

考虑按照点权从小到大插入点，那么对于点权为 $i(i>1)$ 的点，其可以接在点权为 $[i-k+1, i-1]$ 这些点的下面，选哪个点决定了它父亲的编号，并且对每个点这个决策是独立的，方案数可以直接相乘。

所以点权序列确定后，有 $\prod_{i=2}^{n} \min(i-1,k)$ 种方案。

注意到对于任意的点权序列都有这么多种确定树形态的方案，而点权序列有 $n!$ 种，故答案是 $n! \prod_{i=2}^{n} \min(i-1,k)$。

当 $k=n$ 时，该值为 $n!(n-1)!$，否则值为 $n!k!k^{n-k-1}$。

# 1004 Do You Like Interactive Problems?

~~sol0：暴力实现题目所述的操作，打表发现式子很简单，直接交即可。~~

先特判掉 $n=1$ 时答案为 $0$，然后分两类情况讨论（选中的数为 $y$）：

1. $y$ 在中间。

此时要同时问到相邻点或问到这个点。有以下三种状态：

- 同时问到了相邻点或问到了这个点，此时期望步数为 $0$；
- 只问到了相邻点中的一个，此时问到另一个相邻点或当前点都可以转移到上一种状态，期望步数为 $1/(2/n)=n/2$；
- 没问到相邻点或这个点，则有两种转移：
  - 问到这个点，此时直接转移到第一种状态，概率为 $1/n$；
  - 问到任意一个相邻点，此时转移到第二种状态，概率为 $2/n$；

  期望 $E=1/n\cdot 0+2/n\cdot n/2+(n-3)/n\cdot E+1$，解得 $E=2n/3$。

2. $y=1$ 或者 $y=n$。

此时情况等价于上面第二种状态，即期望 $n/2$ 步。

答案为 $2n/3\cdot(n-2)/n+n/2\cdot 2/n=(2n-1)/3$。

# 1005 Equivalence

考虑如何判定一组 $(T_1,T_2)$ 是否合法，首先 $T_2$ 的边权只有不超过 $1$ 组合法解，因为考虑所有 $T_2$ 上的边 $(u,v)$，其边权只能是 $dis_1(u,v)$，但如何判定所有点对都满足条件？

考虑 $dis_1(u,v)=dis_2(u,v)$ 意味着什么，设 $T_2$ 上 $u\to v$ 简单路径依次经过了 $a_1\sim a_k$ 这些点，那么 $dis_2(u,v)=\sum\limits_{i=1}^{k-1}dis_2(a_i,a_{i+1})=\sum\limits_{i=1}^{k-1}dis_1(a_i,a_{i+1})$，在我们赋边权的情况下该式一定成立，而 $\sum\limits_{i=1}^{k-1}dis_1(a_i,a_{i+1})$ 这个求和式对应了 $T_1$ 上一条依次从 $a_1$ 走到 $a_2$，$a_2$ 走到 $a_3$ ...... $a_{k-1}$ 走到 $a_k$ 的非简单路径的走过的边权之和（走过多次算多次）。

现在我们希望知道这条非简单路径走过的边权和是否与直接从 $a_1$ 走到 $a_k$ 这条简单路径的边权和一致，记在两条路径中边 $i$ 分别被经过了 $p_i,q_i$ 次，那么这充要于对于所有 $p_i\neq q_i$ 的 $i$，$w_i=0$。

接下来一个重要性质在于：$p_i=q_i\pmod 2$，再加上 $p_i\geq q_i$，可以发现 $p_i\neq q_i$ 充要于 $p_i\geq 2$，而 $p_i\geq 2$ 的边也很好刻画：它一定在一对简单路径 $i\neq j$，$path(a_i,a_{i+1})$ 与 $path(a_j,a_{j+1})$ 的交上面。

所以我们说 $dis_1(u,v)=dis_2(u,v)$ 充要于所有 $1\leq i<j<k,path(a_i,a_{i+1}),path(a_j,a_{j+1})$ 交的部分边权均为 $0$。

接下来我们声称，$(T_1,T_2)$ 合法充要于对所有 $T_2$ 上的边二元组 $i\neq j$，$path(u_i,v_i),path(u_j,v_j)$ 交的部分边权均为 $0$。

- 必要性：如果有一对边二元组 $i\neq j$ 满足 $path(u_i,v_i),path(u_j,v_j)$ 交的部分边权不均为 $0$，一定可以找到一条 $T_2$ 上的路径 $P$ 同时包含这两条边，那么这条路径对应端点一定不满足 $dis_1=dis_2$。
- 充分性：显然。

最后考虑如何快速判定这一结论，一条非 $0$ 边可以作为两条路径的交当且仅当其被至少两条路径覆盖。

所以我们最后的算法是：对 $T_2$ 的每条边 $(u,v)$，在 $T_1$ 上将 $(u,v)$ 简单路径上的所有边边权都加 $1$，最后如果 $T_1$ 上有边被覆盖至少两次，并且其边权不为 $0$，输出无解。如果所有边都合法，输出有解。

说了那么多，考虑原问题，发现答案就是被覆盖至少两次并且边权不为 $0$ 的 $T_1$ 上边数量。用树上差分计算所有边被覆盖次数，时间复杂度 $O(n\log n)$，考虑到只要判定一条边是否被覆盖至少两次的特殊性或使用 $O(n)-O(1)$ LCA 也可以做到 $O(n)$。

## 1006 Fences

先对整个点集做一个非严格凸包（即允许三线共点，根据约定一定存在一个不退化的凸包），如果 $k$ 在凸包上，就直接输出凸包周长。

显然要求的多边形除了点 $k$ 以外都是凸的。根据调整法易证。考虑凸性意味着什么，从 $k$ 开始逆时针转，多边形上其他点的极角是递增的。所以这个多边形形如：从 $k$ 延伸出一条射线，两边都是凸壳。

拿新的多边形和凸包对比，观察到凸包上的点一定都在要求的多边形上。所以如果 $k$ 不在凸壳上，将所有点以 $k$ 为中心做极角排序，顺时针和逆时针分别做一次，从凸包上一个点开始延伸的凸壳长度，然后对每个夹角求答案取 min 即可。

复杂度 $\mathcal{O}(n\log n)$。

## 1007 Make 2

首先观察每次操作序列的和会上升 $1$，由于最终和为 $2n$，所以初始和必然小于等于 $2n$。同时发现对于一个 $i$，$a_i+a_{i+1}$ 的值是不降的，由于这个值最终为 $4$，所以如果存在一个时刻，有任意一个 $a_i>3$ 就一定不合法了。

注意到如果 $a_i=3$，且有 $2$ 与它相邻，这么这组为 $5$ 必然不合法了；否则我们操作一次这个 $i$，这样这个序列就变成只有 $1,2$ 两个元素了。

如果序列 $a$ 给定，并先不考虑 $a_i>1$ 的条件，那么每个位置需要几次操作是可以被直接计算的：考虑递推，第一个位置不能操作，第二个位置的操作会贡献给第一个位置让它变成 $2$，以此类推。

利用上述性质结合一些分类讨论，我们发现合法的序列是有一些合法的段拼接而成的，合法的段形如以下四种：

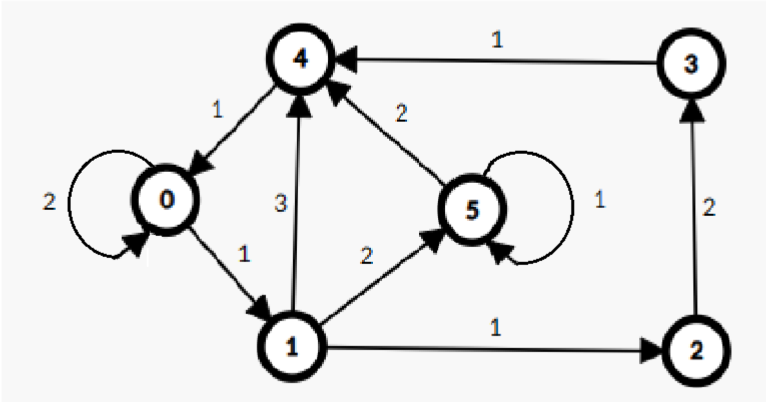
- $2$
- $1,3,1$
- $1,1,2,1,1$
- $1,2,1,1,\dots,1,2,1$ （这种情况两个 $2$ 之间的 $1$ 的个数可以为任意非负整数）

假设第 $i$ 个位置被操作的次数是 $f_i$。

部分证明：

- 开头是 $1,1,1,a,b,c$ 的段不合法，我们直接递推可以推出 $f_2=1,f_3=2$，此时第四个位置值为 $a+2$，若 $a\geq 2$ 则此时 $(a+2)+b\geq 5$，不符合，所以 $a$ 只能为 $1$，继续递推可得到 $a=b=c=1$，并且 $c$ 需要操作 $0$ 次，那么这个序列是 $1,1,1,1,1,1$，但初始不能进行操作，所以不符合。
- 对于开头是 $1,1,2,a,b$ 的序列：可以推出 $f_2=1,f_3=2$，此时仍然可以得到 $a=1$，然后推得 $f_4=1$，此时第五个位置为 $b+1$ 且 $f_5=0$，那么 $b=1$，且有 $f_6=0$，那么就得到了一组合法情况 $1,1,2,1,1$。
- 其他情况同上述情况递推可得，不再赘述。

由于有 $m$ 个限制，考虑对这四种情况构建自动机，满足一个序列是好的当且仅当从起点出发第 $i$ 步在自动机上走边权为 $a_i$ 的出边，最终走回起点。构建结果如下：




于是我们只要 $O(m)$ 次矩阵快速幂和矩阵乘法即可。由于整个测试点只需要跑 $1000$ 次左右矩阵快速幂，所以如果矩阵的大小较大也可以通过。也可以在算快速幂的时候预处理一些幂次的结果。

# 1008 XOR Subsequence

假设初始序列是 $a_0, \cdots, a_{n-1}$，生成得到的序列是 $b_1, \cdots, b_{2^n-1}$，满足 $b_i$ 是 $i$ 这个集合内的 $a_j$ 的 XOR 和。但是由于 $b_i$ 被打乱了，所以我们设打乱后得到序列为 $c_1, \cdots, c_{2^n-1}$，并且令 $d_i$ 为 $c_i$ 在初始生成的序列中的下标。

做法：维护可重集合 $S_1, S_2$，初始时将每个 $c_i$ 加入 $S_1$，接下来进行 $n$ 轮：

- 取出 $S_1$ 中最小值，设为 $x$；
- 将 $x$ 和 $S_2$ 中所有子集（包括空集）的 XOR 和从 $S_1$ 中删去（如果多个只删一个）；
- 将 $x$ 加入 $S_2$。

$n$ 轮结束后 $S_2$ 即为答案集合。以下是简要证明。

考虑我们最后要找的是 $c$ 中的 $n$ 个数，如果这 $n$ 个数满足对应的 $d$ 线性无关，那么就是合法的。（如果线性相关也不一定不对，但可以找到一组值相等的线性无关的解）

接下来考虑由于需要字典序最小，那么考虑贪心。我们需要时刻满足 $d$ 的集合线性无关，但需要注意我们并不知道 $d$。考虑从小到大选择 $c$，每次选择最小的 $c_i$ 加入答案集合。注意到此时加入了一个 $d_i$，假设在此之前已加入 $cnt$ 个数，考虑这个 $d$ 和之前已加入的任意一个子集（可以为空）的 XOR 的结果如果再被加入那么就会导致线性相关，所以我们需要删除这些数。这样就可以保证每一步都线性无关且字典序最小。

# 1009 Far Away from Home

对每种商品，其对答案的贡献可以写成分段函数，转折点个数是不超过 $2\sum t_i$ 的。

对于每一段，极值只会在端点上取到，直接将转折点排序后维护斜率等信息即可。

值域不超过 $10^9 \cdot 5 \cdot 10^5 = 5 \cdot 10^{14} < 2^{53}$，所以直接用 double 维护即可。

复杂度 $O(T\sum t_i \log \sum t_i)$。

# 1010 Border Queries

我们定义字符串 $S'$ 为 $S$ 删去开头和结尾字符后得到的字符串。

首先我们发现一个字符串 $s$ 是好的当且仅当 $s$ 是 $S'$ 的子串且存在一个 $S$ 的 border 长度 $l$ 满足 $|s| + l = |S|$。

> 证明：根据题意，当一种将 $S$ 划分成三个非空子串 $(s_1, s_2, s_3)$ 的方案是好的，此时 $s_1$ 是 $s_1 + s_2$ 的 border 且 $s_3$ 是 $s_2 + s_3$ 的 border。因此 $s_1$ 是 $s_1 + s_2$ 的后缀，所以 $s_1 + s_3$ 是 $S$ 的后缀，同理 $s_1 + s_3$ 是 $S$ 的前缀，所以 $s_1 + s_3$ 是 $S$ 的 border。且当 $s_1 + s_3$ 是 $S$ 的 border 时，容易证明满足 $s_1$ 是 $s_1 + s_2$ 的 border 且 $s_3$ 是 $s_2 + s_3$ 的 border。
> 同时，若存在一个 $S$ 的 border 长度 $l$ 满足 $|s_1| + |s_3| = l$，此时必然满足 $s_1 + s_3$ 是 $S$ 的 border，读者自证不难。又因为要满足 $s_1, s_3$ 非空，所以需要满足 $s$ 是 $S'$ 的子串。

接下来就是求区间有多少个 $S'$ 的子串且长度属于某个集合。考虑先用 SAM 求出对于每个位置 $r$，最小的位置 $l$ 满足 $[l, r]$ 是 $S'$ 的子串，记作 $pos_r$。我们注意到随着 $r$ 增加，$pos_r$ 是不降的。

那么对于一个询问区间 $[l, r]$，我们观察 $pos_l \sim pos_r$，要么 $pos_r < l$，否则一定有一个位置 $i$ 满足 $i$ 及之前的 $pos$ 都小于 $l$，$i$ 之后的 $pos$ 都不小于 $l$。注意到如果 $pos$ 都不小于 $l$，那么这一区间的贡献是可以预处理出来的；如果都小于 $l$，那么其实就是我们要对每个 $[l, j](j \leq i)$ 求答案，这也是一个前缀和的形式，可以预处理。因此每次询问我们只要二分出这个 $i$ 之后就可以 $O(1)$ 算答案。

也可以不用二分，用桶记录 $pos$ 的值。

复杂度 $O(|\Sigma|n + m + q)$，其中 $|\Sigma| = 26$ 表示字符集。$q$ 上带 $\log$ 可以过，$(n + m)$ 上带 $\log$（使用 SA 替换 SAM）也能过。

# 1011 Werewolves

考虑将 $n$ 个人尽可能平均地分成 $m$ 组，使得每组都在一定条件下能猜对，任意两组能猜对的条件需要满足并为全集且互不相交。观察发现将 $n$ 个人按照编号对 $m$ 取模的值分组，并且使得第 $x$ 组能猜对当且仅当所有人的编号之和 $\equiv x \pmod m$ 即满足条件。

# 1012 Equalize the Array

结论：答案为 YES 当且仅当序列最小值是众数。

证明：当序列最小值是众数时，一直操作这个数就行；当序列最小值不是众数时，这个数永远无法被操作，而其他数只能变大，导致序列无法全部相等。