

懒癌 解题报告

杭州学军中学 吉如一

1 试题来源

原创。题目可以在这里找到：<http://uoj.ac/problem/76>

2 试题大意

在一个遥远的地方，一共有 $2^n - 1$ 个村庄，每个村庄住着 n 户人家，门牌号分别为 $1, 2, \dots, n$ ，每户人家家里养着一条狗。恰逢无药可治的懒癌流行，人人自危。每个村庄都有至少一只狗得了懒癌。一个村庄中，门牌号为 i 的人家的狗要么得懒癌，要么不得懒癌，一共 2^n 种情况，再去掉都没得懒癌的情况，一共 $2^n - 1$ 种。这每种情况都会发生在恰好一个村庄中。

这天来了个善良的人来到每个村庄中，告诉所有人一个爆炸性的新闻：“你们村里至少有一只狗得了懒癌！”

每个村庄中每户人家都不知道自己的狗到底是懒癌还是可爱，但是他能一眼看出某些人家的狗有没有得懒癌。由于这个社会里人与人之间的信任已经崩塌，一个人即使看出别人的狗是否得懒癌也不愿告诉他。

可以用一个 n 个结点的有向图来描述可见性， v 到 u 有一条有向边表示门牌号为 v 的人家能看出门牌号为 u 的家里的狗是否得了懒癌，没边则表示看不出。村庄中的每个人都知道这张有向图。

于是一个残酷的逻辑链条开始启动。对于每个村庄：

1. 第一天，早上每户人家的主人会出门看看别人家的狗，如果一个人能推断出自己家的狗得了懒癌，下午6点整，他就会掏出手枪一枪把自己家的狗毙了。
2. 如果有多个人都在同一天推断出了，那么他们会在下午6点整同时开枪。
3. 每个人都听得到这个村庄里的枪声。如果没有听到枪声，这个村里的人第二天会继续早上出门看狗，推断出自己家狗得了懒癌下午就杀狗。如果还没

有听到枪声，第三天也会如此，依次类推。（所以如果一个人听到了枪声那么就不会再开枪杀狗）

作为一个想帮助当地居民调节矛盾的你想要向当地居民展示灾难性的后果，请计算出对于所有前 233^n 天内有过枪声的村庄：

1. 开枪时间之和。如一个村庄在第 k 天下午响起枪声，则开枪时间为 k 。（多个人同时开枪只算一次）

2. 死亡的狗的总数。

你只用输出对998244353（ $7 \times 17 \times 2^{23} + 1$ ，一个质数）取模后的结果。

测试点编号	n 的规模	其他约定
1	$n \leq 8$	无
2		
3	$n \leq 20$	
4		
5	$n \leq 100$	
6		
7	$n \leq 3000$	每户人家都能看出其他每户人家的狗有没有得懒癌
8		无
9		
10		

3 算法介绍

3.1 算法一

首先考虑完全图的十分。假设小R是狗主人之一，我们以他的视角来模拟整个过程：

如果第一天，小R走出家门一看，发现别人的狗都没有生病，那么一定就是他自己的狗生病了，“砰”。

如果第二天，小R走出家门一看：“咦，我只看到了一只生病的狗？”然后他就可以开始分析：如果我的狗没有生病，那么昨天这只病狗的狗主人应该没有看到任何一只生病的狗，那昨天就应该有枪声！好，一定是我的狗病了，“砰”。

以此类推。

所以我们知道，如果有 k 只生病的狗，一定会在第 k 天传来第一次的枪声。而且因为这 k 个狗主人的状况是完全等价的，所以他们一定会同时开枪。所以有一个有 k 只生病的狗的生病状况，对总开枪时间和总枪声数的贡献都是 k 。所以两问的答案都是 $\sum_{i=1}^n \binom{n}{i} \times i$

期望得分：10分

3.2 算法二

上一个算法也还是挺有启发性的，我们发现我们在分析自己的狗有没有生病的时候的方法是：假设我没有生病，然后看前一天是否应该发出枪声。显然，这个方法对任意图都适用的。

考虑状压DP，令 $f[U]$ 表示 U 集合中的狗生病时的开枪时间。然后我们接着以小R的视角进行分析：如果我的狗没病，那么什么时候应该听到枪声呢？于是小R就在大脑中枚举了所有可能的生病情况 V ，那么如果他的狗没有生病，他最迟应该在 $\max f[V]$ 的时间听到枪声。所以他会第 $\max f[V] + 1$ 天开枪。

所以我们就知道怎么DP了！对于一个状态，我们枚举这个状态中每只生病的狗的狗主人，那么这个状态的开枪时间应该是这些生病的狗主人中开枪时间的最小值，从中我们也可以知道有多少人同时开枪。

那么一个狗主人会枚举哪些可能的生病情况呢：首先对于所有他看到的狗，他枚举的生病情况一定是和当前情况相同的，其次他自己一定是没有生病的，而至于哪些他看不到的狗，他就只好枚举这些狗的生病状况了。

最后，如果有限时间内不会开枪怎么办？事实上是存在这种情况的，但是没关系，如果我们发现转移出现了环，所以这个环中的所有状态都是在有限时间内不会开枪的(因为这个这个转移过程可以无穷迭代下去，DP的值会达到无穷大)。

时间复杂度大致是 $O(4^n n)$ 的，可以通过前两个测试点。

3.3 算法三

有了算法二，我们可以把表打出来看看，然后就可以发现一个小小的规律：对于生病状态 U 和 V ，如果 $U \in V$ ，那么必然有 $dp_U \leq dp_V$ 。至于这一个规律的证明我放在后面。

所以我们在转移的时候就可以只枚举一种生病情况：首先对于所有他看到的狗，他枚举的生病情况一定是和当前情况相同的，其次他自己一定没有生病的，而至于那些他看不到的狗，由刚才的规律，可以全部当成生病的。

所以时间复杂度就可以降到 $O(2^n n)$ ，可以通过前四个测试点。

3.4 算法四

为了取得更多的分，我们需要一个多项式算法。我们重新来考虑有限时间内开枪的条件：转移无环。实际上这是一个很强的条件。

考虑根据转移建一个新图，如果 i 不能看到 j ，那么就在 i 到 j 连一条有向边。这个新图中可能有若干个强连通分量，如果一个生病状态中有生病的狗的主人在一个多于一个点的强连通分量中，那么这一个状态一定无法在有限时间内开枪（转移有环），否则一定能在有限时间内开枪（转移无环）。

那么我们先把所有可以到达大小大于1的强连通分量的点连同和他们相关的边一起删掉，于是我们得到了一个DAG！接着我们继续站在这张图的立场上来想怎么求一个状态的开枪时间：

首先这张图上有一些点被染黑了，接着每一时刻，我们可以把一个黑点染白，然后把这个点连向的点的集合的一个子集染黑。若干轮之后，所有点都变白了。假设有 k 个点一度被染黑，那么开枪时间就是所有操作方案中最大的 k 。

看起来这个转化比较意识流，但是如果我们考虑怎么DP所有操作方案中最大的 k ，我们会发现这个DP方法和算法二是等价的。如果再要解释的话，首先枚举狗主人相当于枚举反转的黑点，算狗主人开枪时间中的max体现在求最大的一度被染黑的点数，算状态的开枪时间中对狗主人开枪时间取min体现在多次被染黑的点只被计算一次。

而在这个模型下，算法三的小规律就很显然辣！增加黑点对答案的贡献一定非负。所以问题又变成了：

一个DAG上有一些点被染黑了，接着每一时刻，我们可以把一个黑点染白，然后把这个点连向的点全部染黑。若干轮之后，所有点都变白了。假设有 k 个点一度被染黑，求所有操作方案中最大的 k 。

这不就是求DAG上一个点集能直接或者间接到达的点数吗？

所以我们把判断一个生病状态在第几天开枪转化为了求DAG上一个集合能直接或间接到达的点数。这样第一问已经非常简单了。考虑算每一个点的贡献，

如果一个点能被 i 个点到达（包括自己），而DAG上一共有 N 个点，那么显然这个点的贡献就是 $(2^i - 1) \times 2^{N-i}$

接下来考虑第二问，一个人要在DAG上满足什么样的条件才会在第一天开枪呢？在原来的DP中，我们是在取min的过程中得到同时开枪的人数的，而我们知道，在现在的模型中，算状态的开枪时间中对狗主人开枪时间取min体现在多次被染黑的点只被计算一次。

所以我们脑补一下可以得到：如果一个点第一个反转，且之后得到的状态无论怎么操作都无法再把这个点染黑，那么这个点对应着一个最早开枪的狗主人。

所以第二问也可以很简单的解决了，还是考虑算每一个点的贡献，如果一个点能被 i 个点到达（包括自己），而DAG上一共有 N 个点，那么显然这个点的贡献就是 2^{N-i} 。

因为计算DAG中每一个点可以被多少点到达是 $O(nm)$ 的，所以时间复杂度是 $O(n^3)$ ，可以通过前六个点。

3.5 算法五

不难发现，这个问题是可以直接压位的，所以我们可以直接使用bitset大法。

时间复杂度 $O(\frac{n^3}{32})$ ，这个算法的常数很小，可以通过全部测试点。

3.6 数据构造方法

数据中需要让给定图的补图中能到达大小大于1的强连通分量中的点数尽可能的少，同时又需要存在这样的点。所以可以先随机出一张满的DAG，然后删掉其中一部分的边，再选出若干个拓扑序非常小的点连成一些强连通分量。

这样的随机数据再配合上一些构造的小范围的数据（因为不相关的联通块之间是独立的，所以可以直接加上去），基本就足以卡掉暴力和乱搞了。

3.7 总结

这是一道很难的题，同时也是一道非常有趣的题。这道题的灵感来自于一道某城市的小升初数学题，那题的题意中给出的认识图是 $n = 20$ 的完全图，然

后告诉你在第二天有人开枪，问有多少只狗生病。当时看到这题，就觉得它非常有趣，于是就拿去和吕凯风学长讨论并尝试加强成一道OI题，在经过了一个多礼拜的讨论之后这题终于诞生啦。

因为给出的信息十分复杂（认识关系图），而且含有大量的推理成分，所以很难像普通的算法题一样一步到位分析出优秀的多项式算法——人脑分析是很难找到这题的切入点的。所以需要利用一些OI中的思维——利用计算机来模拟人脑在最优情况下的分析——这样就可以把问题转化成DP的问题。

接着，需要观察DP的过程，发现一些特殊的性质并转化成一个图论模型，最后就可以用一个非常简洁并且常规的算法来解决这个问题了——实际上这题在UR乃至UER的所有C题中都是代码长度最短的。

这道题在UR赛场上仅有一人获得30分，两人获得20分，其他得分的选手都仅获得了完全图那个点的分数。

总的来说，这是一道思维含量远大于代码含量的题，而且这题思维难度产生在一步一步的分析与对题目性质的剖析，并没有涉及任何偏僻的知识。所以我觉得这是一道难得的好题目。