D. 花

算法一

直接暴力，复杂度 $O(2^n n^2)$，期望得分 20 分。

算法二

考虑把 2SAT 图建出来，缩点，此时缩完之后每一个点有一个与之对应的点，即为 inv_u，这两个点的取值必须不同。

传递闭包之后，如果有 $u \to inv_u$，或 $inv_u \to u$，那么这一对点的取值就确定了，高处的那个为 0，低处的那个为 1，就可以忽略这对点了。

然后考虑在经过上述处理的图中进行暴搜。若在限制的意义下不连通（即所有限制连边，u 与 inv_u 连边形成的图不连通），则答案显然是各个连通块的乘积。

否则，考虑选择某一个点 u，枚举它是 0 还是 1。若是 0，那么所有 $i \to u$ 都是 0，所有 $inv_u \to i$ 都是 1，若是 1 则反过来。

所以说，枚举了一个点的取值之后，会多确定能由它到达，或到达它的所有点。

我们试图选择这些点个数最大的那个，然后将其去掉递归，考虑分析复杂度，我们试图用度数来限制，肯定减少的总和要大于度数。

若所有点度数 $=0$，则都是孤立点，连通块分解后是 $O(1)$ 的。

若所有点度数 ≤ 1，则连通块大小不会超过 2，连通块分解后是 $O(1)$ 的。

若所有点度数 ≤ 2，则是一个个链和环，如果我们随机选择一个最大的点确定，则期望会被切分成两半，那么这么递归是 $O(\text{poly}(n))$ 的。

所以说，所有点度数至少是 3，也就是说复杂度是：

$T(n) = \text{poly}(n) + \max\{T(n-1)+T(n-4), T(n-2)+T(n-3)\} = O(1.3803^n)$

实际上，由于去掉了一些点导致其它点的度数进一步降低，加上随机化的部分，这个算法很难卡满，数据使用了一个每个点度数都为 3 的二分图以及一些类似的结构，大概能卡到 2×10^7 次左右的递归。如果有人知道如何造卡满的数据或者可以分析出更低的复杂度欢迎洛谷私信 dengyaotriangle 怒斥出题人。

如果没有随机化，则链和环的情况无法处理，这样的复杂度是：

$T(n) = \mathrm{poly}(n) + \max\{T(n-1) + T(n-3), 2T(n-2)\} = O(1.415^n)$
可能无法通过 $n \leq 60$。