Sky 不想拼图

首先 $n \leq 10$ 的部分是留给爆搜的。

在本机搜一下前面几项，会得到这么一个序列：1, 1, 2, 3, 9, 16, 35, 65, 143, 281, 590, 1174, 2440, 4925, 10142, 20563, 42178, 85819, 175632, 357875, 731536, 1491966, 3047879, 6218844, 12699982, 25919176, 52922491, 108022099, 220541999

设 $f[i]$ 表示拼完前 $4 \times i$ 的木板的方案数。这时考虑OEIS。可以得到递推式：
$f[n] = f[n-1] + f[n-2] + 5 * f[n-4] - f[n-5] + f[n-6] - f[n-8]$

关于这个式子的证明可以参考此论文。

然后构造一个行向量与一个转移矩阵，考虑矩阵乘法的过程，可以得到下面的式子：

$$\begin{bmatrix} f_8 \\ f_7 \\ f_6 \\ f_5 \\ f_4 \\ f_3 \\ f_2 \\ f_1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 5 & -1 & 1 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} f_9 \\ f_8 \\ f_7 \\ f_6 \\ f_5 \\ f_4 \\ f_3 \\ f_2 \end{bmatrix}$$

所以中间这个矩阵就是我们所需的转移矩阵了。直接上矩阵快速幂即可。

Sky 不想做图论

$n \leq 8$ 暴力即可。

首先，期望割点个数 $\times n^n$ 就是所有图的割点的个数和（因为恰好有 n^n 种图且出现概率相等），又等于每个点成为割点的次数和。

显然 $1 \cdots n$ 的每个点是完全平等的，于是每个点成为割点的次数相等。我们只需要求出 1 是割点的次数乘上 n 即可。

考虑计算结点1 不是割点的方案数，分两种情况：

第一种，所有 $A_{2\dots n}$ 都不等于 1 。因为这样的话 1 的度数是 1，不可能是割点。这样的方案数是 $n(n-1)^{n-1}$ 。

其他情况下，考虑一个图的一个连通块必定是一棵基环树，而如果 1 不是割点且 1 不是叶节点，则 1 一定在环上，而且 $A_{2\dots n}$ 里面恰有一个等于 1 （想象如果有多个 A_i 等于 1， 则其中一个是 1 在基环上的相邻点，其他的则是基环之外的子树、这样的话 1 必定是割点；如果基环里只有 1 一个结点 ($A_1=1$) 的话也类似讨论）。

考虑枚举 1 所在连通块的大小为 $i+1$ ，枚举哪个点 j 的 $A_j=1$ ；把 1 这个点（及 $(1,j),(1,A_1)$ 两条边）删掉之后这个基环树必定会变成一棵树，其中 A_i 即为 i 的父亲。那么有根树计数可得这样的树有 i^{i-2} 种。

而 j 的取值有 i 种， A_1 的取值有 $i+1$ 种，整个图的剩下的 $n-i-1$ 个点有 $(n-i-1)^{n-i-1}$ 种连法；从 $n-1$ 个点中选出 i 个点有 $\binom{n-1}{i}$ 种方案。全部乘起来对i求和，这样的方案数就有

$\sum_{i=1}^{n-1}\binom{n-1}{i}i^{i-1}(i+1)(n-i-1)^{n-i-1}$

直接计算即可，也可以用多项式继续优化。

Sky 不想旋转

这是一个比较简单的题目，容易发现：对整个矩阵的操作其实只是 4 个部分不停交换位置，所以这个大矩阵的变换可以等价为一个 2×2 的小矩阵进行变换， 最后在通过小矩阵此时每个元素的顺序来还原之前的大矩阵。

Sky 不想做背包

20分

二进制枚举，枚举集合 a、集合 b，把 a 当作选择的集合，把 b 当作要检验的子集，复杂度 $O(qn2^{2n})$

40分

开一个数组 $f[]$，$f[s]$ 表示 s 这个原材料集合的特征值是否异或起来等于 0，另外再用 $g[s]$ 表示这个原材料集合的强度和

对 $f[]$ 做高维前缀和之后，$f[s]$ 就表示集合 s 是否存在一个子集异或起来等于 0。

不同的询问数量总共只有 $\frac{n(n+1)}{2}$ 种

直接暴力预处理的话，复杂度是 $O\left(n2^n+\sum_{l=1}^{n}2^l(n-l+1)\right)=O(n2^n)$

60分

考虑如何处理单次询问，可以贪心做这个事请

把所有原材料按照强度值递减排序，然后维护异或运算下的基向量组，初始时向量组为空，依次加入向量，如果这个新加入的向量能用已有的向量线性表示，则不加入它，否则加入它并且将它的强度累加到答案中。

正确性：

假设按照上述贪心策略选出来的是如图左侧的方案，而另一种策略选出了如图右侧的方案（不妨给两个方案去掉交集）。

由于向量组的秩不变，所以任何一种方法选出来的向量个数一定都是相等的，都等于向量组的秩。

有一点可以肯定： B 方案中，向量 b_i 的前面（序号意义上的前），在 A 方案中至少选择了 i 个向量。因为之所以 A 方案中没选择 b_i 就是因为 b_i 能用已选的向量线性表示，而用一个线性无关的向量组生成的另一个线性无关的向量组中向量的个数肯定不大于前者。这也就导出另一个结论，b_1 肯定在 a_1 的后面，b_2 肯定在 b_2 的后面，以此类推，b_i 肯定在 a_i 的后面。那么 B 方案的强度之和也就自然而然不大于 A 方案的强度之和。

那么 60 分的做法也就很显然了，直接每次排序，然后按照贪心的方式选择，算出答案就好了。

100分

使用线段树来维护区间上子问题的答案，并且存下每个区间的方案。

对于一次查询，我提取出 $O(logn)$ 个子区间，然后利用之前存下的方案进行合并。

为什么我用子问题最优解的方案直接合并就可以得到整个问题的最优解呢?

考虑两个向量组合并，假设第一个向量组中选择哪些向量当作答案已经确定了，那我就把这些向量加到第二个向量组中去，并且一开始这些向量处于已被选择的状态，这个时候就相当于要在第二个向量组中再选一些向量使得选出来的向量线性无关且强度和最大，这不就回到了前面所讲的问题了吗，直接采取相同的贪心策略即可得到正确答案。

设 $m = max(a_i)$，时间复杂度 $O\left(nlognlogm + qlognlog^2m\right)$