中国象棋(chess)

限制等价于同一行或同一列不能有超过两个棋子。

直接设计状态 $f_{i,j,k}$ 表示处理了前 i 行，有 j 列有 1 个棋子，有 k 列有 2 个棋子的方案数。考虑转移：

- 这一行什么都不填：$f_{i+1,j,k} \leftarrow f_{i,j,k}$。
- 这一行在一个空列放一个棋子：$f_{i+1,j+1,k} \leftarrow f_{i,j,k} \times (n-j-k)$。
- 这一行在两个空列放一个棋子：$f_{i+1,j+2,k} \leftarrow f_{i,j,k} \times \binom{n-j-k}{2}$。
- 这一行在一个有 1 个棋子的列放一个棋子：$f_{i+1,j-1,k+1} \leftarrow f_{i,j,k} \times j$
- 这一行在两个有 1 个棋子的列放一个棋子：$f_{i+1,j-2,k+2} \leftarrow f_{i,j,k} \times \binom{j}{2}$。
- 这一行在一个空列和一个有 1 个棋子的列放一个棋子：$f_{i+1,j,k+1} \leftarrow f_{i,j,k} \times (n-j-k) \times j$。

时间复杂度 $O(n^3)$。

序列(array)

由于 $0 \le a_1 \le V = 10^5$，说明至少有一对 b_i, b_j 的差小于等于 $\dfrac{V}{k-1}$，那么序列的美丽值至多是 $\dfrac{V}{k-1}$。

不妨枚举这个美丽值 x，求解有多少长度为 k 的子序列的美丽值**至少**为 x。

题目所求均与和 a 的顺序无关，因此将 a_i 从小到大排序。维护状态 $f_{i,j}$ 表示处理了前 i 个数，选择的第 j 个数为 a_i 且相邻两项的差 $\ge x$ 的方案数。

转移有 $f_{i,j} = \sum\limits_{a_k \le a_i - x} f_{k,j-1}$，该转移可以使用前缀和优化。对单个 x 求解时间复杂度为 $O(nk)$。

总时间复杂度为 $O(nk) \times O(\dfrac{V}{k}) = O(nV)$。

无向图

显然，一张 n 个点的图的割边数量 $< n$。

考虑直接设计状态 $f_{i,j}$ 表示 i 个点，有 j 条割边的连通图数量。

枚举图中的一条割边，那么就可以分成称两个更小的子图。但是由于割边的数量有 j 条，因此需要对最终的结果乘 $\dfrac{1}{j}$ 进行去重。

也就是 $f_{i,j} = \dfrac{1}{j} \sum\limits_{1=k}^{i-1} \sum\limits_{l=0}^{j-1} f_{k,l} f_{i-k,j-l-1} \binom{i-1}{k-1} k(i-k)$。

但是这种转移方式对于 $j=0$ 并不适用。

但是对于 $f_{i,1} \sim f_{i,i-1}$ 的计算并不依赖 $f_{i,0}$，因此可以用 i 个点的连通图的数量 g_i，减去 $\sum\limits_{j=1}^{i-1} f_{i,j}$ 来得到 $f_{i,0}$。

现在的问题转化为求解 g_i，仍然考虑容斥，用所有图的数量减去非连通图的数量。

对于一个非连通图，枚举 1 号店所在的连通块大小，其余的部分是一个任意图。

也就是有 $g_n = 2^{\frac{n(n-1)}{2}} - \sum\limits_{i=1}^{n-1} g_i \binom{n-1}{i-1} 2^{\frac{i(i-1)}{2}}$。

总时间复杂度为 $O(n^4)$。存在更优的做法。

围棋 (go)

检验一个棋盘是否满足条件的方法可以是：一行一行扫描，如果找到满足条件的位置就直接统计，没有扫描到的行无论是什么都可以了。

而对于一行而言，我们关注的是其能和两行模板分别在哪些位置能够匹配，而对于所有 3^m 种串，可以直接一处理出两个 $m-c+1$ 为的二进制数分别表示能否和第一行匹配，能否和第二行匹配。记 $f_{S,T}$ 表示和第一行匹配的结果为 S，第二行匹配结果为 T 的串的数量，cnt_T 为第二行匹配结果为 T 的串的数量。

记 $dp_{i,j}$ 表示已经填了前 i 行，第 i 行和第一行的匹配情况为 j ，且前 i 行没有激活模板的方案数。

增加一行之后已经进入答案部分的转移：$ans = ans \times 3^m$。

$dp_{i,j}$ 对于答案的贡献，$ans \leftarrow dp_{i,j} \times \sum\limits_{j \cap T \neq \varnothing} cnt_T$，后面的和式可以用高位前缀和优化。

$dp_{i,j}$ 向后的转移：$dp_{i,S} \leftarrow dp_{i,j} \times \sum\limits_{j \cap T = \varnothing} f_{S,T}$。后面的和式可以用高位前缀和优化。

总时间复杂度为 $O(3^m(m+c) + 4^{m-c+1}n)$，存在更优的做法。