袋鼠

算法 1（$n \leq 6$）

若 $n \neq m$，则答案为 0.

$K = t$ 与 $K = n - t$ 答案相同。

对矩阵的组成进行搜索，并对行和、列和剪枝。

算法 2（$n \leq 8$）

考虑将矩阵的每行按该 01 串的字典序排序。若两个矩阵排序后的矩阵相同，则它们属于同一个等价类。搜索这样的等价类，每个类对答案的贡献是一个组合数。实现细节参见代码。

算法 3（$n \leq 9$）

考虑类似 meet-in-middle 的方法，将矩阵分成上下两部分别搜索，在每一部中采用算法 2 的方法，同时只对行和进行限制，并将列和的状态进行压缩并记录。最后，枚举下部的列和，可以快速计算出上部的列和，查表并统计答案。

算法 4 （$n \leq 9$）

验题人表示记忆化搜索再随便剪枝就过了。

黎明卿

题意

构造图 G，满足点和边的限制，使最小顶点割最大。

$n-1 \le m \le 2n-1.$

算法（$n \le 50$）

此处直接给出全部数据范围下的解法。

记 $\kappa(G)$ 为图的最小顶点割的基数，$\delta(G)$ 为图中顶点的最小度数。

定理：

$$\kappa(G) \le \delta(G) \le \left\lfloor \frac{2m}{n} \right\rfloor.$$

该定理的正确性显然。

同时，我们将看到，对于给定的点数和边数，必存在 G，使上述不等式各处均取等。

以下将对 $\left\lfloor \frac{2m}{n} \right\rfloor$ 进行讨论。

1. $\left\lfloor \frac{2m}{n} \right\rfloor = 1.$

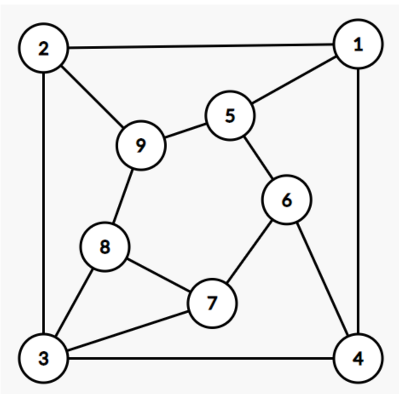
   此时 $m = n-1$. 单链为所求。

2. $\left\lfloor \frac{2m}{n} \right\rfloor = 2.$

   此时 $n \le m < 1.5n$. 当 $n = m$ 时，一个包含 n 个点的环满足条件；在该图上加若干边，$\kappa(G) = 2$ 不变。

3. $\left\lfloor \frac{2m}{n} \right\rfloor = 3.$

   此时 $1.5n \le m < 2n$. 当 $m = \lceil 1.5n \rceil$ 时：
   a. $n = 5$ 时可单独构造；
   b. $n > 5$ 时，与下图相似的构造满足条件。



在该类图上加若干边，$\kappa(G) = 3$ 不变。

事实上，另一种构造是 Harary 图 (标程使用了这一构造)。
这一构造方法适用于所有 $m \ge n-1$ 的情形。

三角形

算法 1（$n \le 10^3, K = 1$）

这一部分分来源于 Codeforces 220D.

1. 若三角形的三边都不与网格的边平行，则该三角形中存在唯一一个顶点，满足它是该三角形的一个最大与网格边平行的外接矩形的一个顶点。枚举这一顶点，只需对另外两点的各维坐标的奇偶性进行讨论。
2. 对三角形的一边与网格的边平行的情形，可以利用数学技巧快速计算。

时间复杂度 $O(n^2)$.

以下将对算法 1 的 n 与 K 的部分分别优化 (扩展)。

算法 2（$K \le 40$）

考虑算法 1 的情形 1.

仍旧枚举该顶点，记为 (i, j). 一个观察是，(i, j) 的答案与 $(i \bmod K, j \bmod K)$ 的答案相同。使用与算法 1 相似的方法：考虑三角形面积的叉积形式，问题转化为统计 $xy \equiv p \pmod K$ 的 (x, y) 的个数 $f(p)$，并统计 $\sum f^2(p)$.

注意到，f 的计算依赖于 $p(x')$ 与 $q(y')$，即模意义下与 $x' < K$ 同余的 x 个数和与 $y' < K$ 同余的 y 个数，于是 $f(z) = \sum_{xy \equiv z \bmod K} p(x)q(y)$.

时间复杂度 $O(K^4)$.

算法 2（$K \le 100$）

考虑对算法 2 进行优化。

注意到，对于一个固定的位置，$p(x)$ 只有 2 种可能的值，且相同的值的下标在模意义下连续，$q(y)$ 亦如是。同时，对于相邻两个格点，p, q 只会左右移动一位，因此相邻两个格点的 p, q 至多仅有 2 位不同。在枚举 (i, j) 时，考察 p, q 的差异，每次只需对 f 进行 $O(K)$ 次修改。

时间复杂度 $O(K^3)$.

算法 3（$n \le 10^6$）

对于算法 1 的情形 2，容易观察到这部分公式相当于一系列 gcd 的和。进行一级或二级分块可以做到 $O(n)$；也可使用杜教筛进行进一步优化。

历史

By slz

算法 1（$n \le 10$）

显然断边的方式只有 $(n-1)!$ 种，所以直接枚举即可。

算法 2（$n \le 20$）

转压 dp 即可。

算法 3（$n \le 300$）

指数级算法是没有前途的。

原题相当于启发式分裂的过程，那么把这个过程倒过来，就变成启发式合并。

记 $F[i][j]$ 表示区间 $[i,j]$ 合并成了一段的期望值。

那么对于 $F[i][j]$，其中每一个点变成断点的概率都是相等的，那么直接枚举断点，从 $F[i][k]$ 和 $F[k+1][j]$ 转移即可。

时间复杂度 $O(n^3)$。

算法 4（ $s_i = 1$ ）

注意到所有的 s_i 都相等，那么不拿发现，对于算法三中的区间，只要长度是相等的，那么 dp 值就是相等的。所有对于这部分，dp 的状态可以化简成一维的。

即 $F[i]$ 表示长度为 i 的区间的权值和（期望值最后再除于 $(n-1)!$ ），那么

$$F[i] = \sum_{j=1}^{i-1} F[j] \cdot (i-j-1)! \cdot C_{i-2}^{j-1} + F[i-j] \cdot (j-1)! \cdot C_{i-2}^{j-1} + min(j, i-j)$$

这样子已经可以做到 $O(n^2)$ 了。

擅长多项式的同学这个式子可以做到 $O(n \log n)$ 或$O(n \log^2 n)$ 。

不过，似乎这个式子可以变成这样：

$$\frac{F[i]}{(i-2)!} = \sum_{j=1}^{i-1} \frac{F[j]}{(j-1)!} + \frac{F[i-j]}{(i-j-1)!} + \frac{\min(j, i-j)}{(i-2)!}$$

这样就可以前缀和优化了。

算法 5（全部子任务）

动态规划也是没有前途的，直接计数最好。

我们记 $1-2$ 为 1 号断点，$(n-1) - n$ 为 $n-1$ 号断点，其他的以此类推。

考虑对于一段区间 $[l,r]$，假设它作为左半段贡献，那么满足 r 号断点晚于 $l \sim r-1$ 号断点。

如果枚举右区间的终点 x，再让 $r+1 \sim x-1$ 断点晚于 r 号断点，在让 r 号断点晚于 $l-1, x$ 号断点。这样做是 $O(n^3)$ 的。

但是我们发现，我们并不需要枚举 x，我们只需要保证 $r-l+r+2$ 号断点以后存在一个点早于 r，而 $r+1 \sim r-l+r+1$ 都晚于 r 即可。而存在一个点早于 r 事实上并不是一个限制，因为所有情况都满足这个条件。

所以这个做法被优化到 $O(n^2)$ 了。

如果实现上述做法，你会发现，对于一段定长 len ，即 $r-l+1=len$ ，它们的系数都是一样的，所以只需要再前缀和优化即可。