

《Links》题解

tl;dr 直接跳到「算法五」

由于 portal 这个名字实在过于中二，下简称为「点」。

为什么结束时满足条件并不意味着整个过程满足条件

看一个🍎

$N = 4, K = 3, G = 1$

第一步（选择 1，一个团）：●○○○ （其实是个环）

第二步（选择 3，两个团）：●○●○ ← 违反限制

第三步（选择 2，一个团）：●●●○ ← 满足限制

大概需要读完题才能着手的算法一

12% 的数据满足 $G = 1$ 。

这意味着第一个占领的点可以任选，之后每一个点必须与之前点形成的团相邻，即左边或右边。于是答案为 $N \cdot 2^{K-1}$ 。

似乎过了样例就能信心满满的算法二

32% 的数据满足 $N \leq 8$ 。

枚举所有至多 $N!$ 种占领顺序并时刻检查是否违反了团数不超过 G 的限制。单组数据时间复杂度 $O(N \cdot N!)$ 。

样例的前三组数据是不是很良心吖~~

可能程序也挺容易写的算法三

60% 的数据满足 $N \leq 20$ 。

考虑状态压缩 DP。用 $f[s]$ 表示「选择二进制数 s 所表示集合中的所有点」且「中途不违反团数限制」的方案数（取模）。从 0 到 $2^N - 2$ 枚举 s ，对于每个 s ，枚举一个不在集合 s 中的元素 u 尝试加入，并检查是否超过了 G 个团；若没有超过，则令 $f[s \cup u]$ 增加 $f[s]$ 。

【为什么枚举到 $2^N - 2$ ？因为 $2^N - 1$ 时已经全部安排上啦 (╯▽╰)】

最后找出二进制下 1 的位数不超过 K 的所有 s ，累加 $f[s]$ 即为答案。

状态数 $O(2^N)$ ，转移 $O(N)$ ，总时间复杂度 $O(N \cdot 2^N)$ 。

好像终于不能继续暴力下去的算法四

另外 16% 的数据满足 $G \leq 20$ 。

一脸 DP 的样子，考虑 DP 吧！

然后 Isq 在一个纸箱上躺了半天也没想出针对这些数据的算法（'-`）【忽略那些奇怪的细节吧】

或许是超级无敌旋转棒棒的算法五

如前所述，DP！

考虑点的编号的话，状态会不那么容易表示。不妨将「已经选择的点数目」与「目前的组数」放入状态中。用 $f[i][j]$ 表示选择了 i 个点，形成 j 个组的方案数。

一个问题是，点以编号区分，但状态并不加以记录。状态中究竟认为点在哪些方面相同，将直接影响状态转移方程的设计，因此我们需要提前考虑。

略去心路历程，这里给出我们最终发现可行的定义：对于 $f[i][j]$ 所表示的状态，认为所有选择的点依选择次序编为 1 至 n 号，它们之间的排列顺序不同则认为方案不同；这个编号与实际编号无关，我们只是考虑了一个虚构的点集，最后再尝试将这个点集对应到实际编号上。此外，状态中两个团之间间隔的点数是不确定的。

这样一来状态转移便很明晰了，对于 $f[i][j]$ ，考虑第 i 个加入的点：

- (1) 若它紧贴之前的某一个团，则当前状态有 $f[i-1][j] \cdot 2j$ 的增量； $2j$ 的因子是由于之前也有 j 个团，贴在其中一个逆时针或顺时针方向（或者说左边/右边）都是一种可能的情况。
- (2) 若它将之前的两个团连接起来，则当前状态有 $f[i-1][j+1] \cdot (j+1)$ 的增量； $j+1$ 的因子是由于之前 $j+1$ 个团中任意相邻两个都可以被连接，且排列成一个环。
- (3) 若它自成一个新的团，则当前状态有 $f[i-1][j-1] \cdot (j-1)$ 的增量； $j-1$ 的因子是由于之前 $j-1$ 个团中任意相邻两个之间都可以插入一个新的团。

状态数 $O(GK)$ ，状态转移 $O(1)$ ，于是时间复杂度为 $O(GK)$ 。

这一部分结束后，考察所有 $f[K][i]$ ，其中 $1 \leq i \leq G$ 。对于这样的 $f[K][i]$ ，需要计算将 K 个点对应到原本的 N 个点中 K 个的方案数。不妨认为状态中的 1 号点可以任取为实际任意一个点；此后状态中 2 至 K 号点的顺序，以及它们之间是否相邻（同一个团的相邻，不同团的不相邻；尽管没有记录在状态中，但是体现于方案的计算过程），均已确定。也即：需要将剩余 $N-K$ 个点划分为 i 个连续的段（亦即「团」），作为团之间的分隔。根据插板法，这个值为 $\binom{n-k-1}{i-1}$ 。【 $\binom{n}{m}$ 也写作 C_n^m 】

因此，遍历 $i \leq G$ ，累加 $f[K][i] \cdot n \cdot \binom{n-k-1}{i-1}$ 即为答案。

时间复杂度为 $O(GK)$ 。【好像数据范围 N 可以出得更大的样子？？】

废话几句

感谢大家能读到这里啦。

这道题目应该还是比较精妙的一道 DP 问题，部分分似乎好像可能大概..... 也给的挺足吧？希望 NOIP 前能给大家带来一次不那么坑的比赛 > <

嘛要说为什么想到最后这个状态表示..... 可能核心还是在于「先计算后对应」的思路吧【原谅伦无语次的 Isq QAQ】其实好多 DP 问题都是多试试几种状态表示法说不定哪个就走通了呢（笑）总之程序还是超好写的.....

对于这个 idea 有兴趣的话，欢迎来尝试这个题~ 「JOI Open 2016」摩天大楼

三道题都是从以前的比赛中搬的... 命题人是岛娘 (xiaodao)，在此表示感谢！

还有金老师两年前对于校内模拟赛的支持，以及给 Isq 提供机会重新审视这道埋藏多年的题目；以及 chy (@watermelly) 同学近日抽出时间给前两题补足了题解，也一并致谢啦。

以上。祝一切安好！

Isq (@cyand1317)

2018.10