欧拉？欧拉！

Elegia
李白天

2022 年 12 月 10 日

IIIS, THU

题意

题意

给定 n 和模数 M, 计算

$$\sum_{p \in S_n} x^{\#[p_i > i]} y^{\#[p_i > p_{i+1}]}.$$

保证 $n \le 60$.

算法 1

算法 1

- 枚举所有排列, 复杂度 $O(n!)$ 甚至 $O(n \cdot n!)$.

算法 1

- 枚举所有排列, 复杂度 $O(n!)$ 甚至 $O(n \cdot n!)$.
- 期望得分 10.

算法 2

算法 2

- 状压 DP, 记 $f(S,m,k,i)$ 表示前 $|S|$ 个数填了 S, 并且目前的超过数和降低数分别是 m 和 k, 且目前最后一个数是 i 的方案数.

算法 2

- 状压 DP, 记 $f(S,m,k,i)$ 表示前 $|S|$ 个数填了 S, 并且目前的超过数和降低数分别是 m 和 k, 且目前最后一个数是 i 的方案数.
- 复杂度 $O(2^n \operatorname{poly} n)$.

算法 2

- 状压 DP, 记 $f(S,m,k,i)$ 表示前 $|S|$ 个数填了 S, 并且目前的超过数和降低数分别是 m 和 k, 且目前最后一个数是 i 的方案数.
- 复杂度 $O(2^n \operatorname{poly} n)$.
- 期望得分 30.

算法 3

算法 3

事实上本题似乎存在用 0 次容斥, 1 次容斥, 2 次容斥, 3 次容斥的几种不同思路, 最后都能够通过本题.

- 一开始编这道题的时候是 3 次容斥, 后来减少到 2 次容斥
- Isonan 胡杨 验题的时候告诉我可以 1 次容斥, 后来又发现可以不容斥.

这里只择 2 次容斥的做法讲解.

算法 3

首先考虑如下问题: 有 k 条链, 第 i 条链的约束是形如 $<<>>>>>$ 的, 其中有 L_i 个小于号和 R_i 个大于号 $(\sum_i (L_i + R_i + 1) = n)$. 此外还要求第 i 条链的峰 $\le a_i$, 问有多少种合法排列.

算法 3

首先考虑如下问题: 有 k 条链, 第 i 条链的约束是形如 $<<>>>>>$ 的, 其中有 L_i 个小于号和 R_i 个大于号 $(\sum_i (L_i + R_i + 1) = n)$. 此外还要求第 i 条链的峰 $\le a_i$, 问有多少种合法排列.

不妨考虑从小到大插入, 设 a_i 单调不降. 记 $S_i = \sum_{j \le i} (L_j + R_j + 1)$, 那么此时我们还剩 $a_i - S_{i-1}$ 个数可以用, 答案就是

$$\prod_{i=1}^{k} \binom{a_i - S_{i-1}}{L_i + R_i + 1} \binom{L_i + R_i}{L_i}.$$

算法 3

我们考虑进行 2 次容斥.

外层容斥是对所有 $p_i > i$ 做二项式反演. 要么 $a_i = i$, 要么 $a_i = n$.

内层对相邻大小关系做容斥.

注意到如果我们要保证每个连通块的峰值限制要么是 i, 要么整个连通块不能有限制, 否则不能用前述公式计算.

算法 3

我们考虑进行 2 次容斥.

外层容斥是对所有 $p_i > i$ 做二项式反演. 要么 $a_i = i$, 要么 $a_i = n$.

内层对相邻大小关系做容斥.

注意到如果我们要保证每个连通块的峰值限制要么是 i, 要么整个连通块不能有限制, 否则不能用前述公式计算.

逐个确定容斥方式.

算法 3

不妨考虑如下场景:

- a_i 要么是 i 要么是 n, 已经确定.
- 从大到小确定 p_i 和 p_{i+1} 的关系 (不限制/$<$/$>$).

考虑 p_{i-1} 和 p_i 的大小关系的时候:

- 如果 p_i 所在这个连通块的峰值有限制, 那么 p_{i-1} 和 p_i 的大小关系的大小关系分 “$<$” 和 “不限制” 两种贡献拆开.
- 否则, p_{i-1} 和 p_i 的大小关系的大小关系分 “$>$” 和 “不限制” 两种贡献拆开.

算法 3

总共用一个四维的 DP 实现, $f(i,j,k,l)$ 分别记录当前下标, j 记录第一步容斥枚举了几个, k 记录第二步容斥枚举了几个, l 记录前面支撑点有限制的连通块的大小之和.

算法 3

总共用一个四维的 DP 实现, $f(i,j,k,l)$ 分别记录当前下标, j 记录第一步容斥枚举了几个, k 记录第二步容斥枚举了几个, l 记录前面支撑点有限制的连通块的大小之和.

转移比较抽象, 不写出式子了.

高复杂度的话可能会得到 60 分, 稍微优化一下转移则是一个常数很小的 $O(n^5)$, 期望得分 100.