《木桶效应》解题报告

山东省临沂第四中学 周泽坤

2024 年 8 月 28 日

目录

1 题目大意 **2**

2 数据范围 **2**

3 解题过程 **2**

3.1 算法一 2
3.2 算法二 2
3.3 算法三 2
3.4 算法四 3
3.5 算法五 3
3.6 算法六 3
3.7 算法七 4

4 参考资料 **4**

1 题目大意

有 m 个长度为 n 的排列，其中共有 q 个位置的值已经确定，其余位置未确定。求所有本质不同的排列组对应的 $\prod_{i=1}^{n}(\min_{j=1}^{m} p_{j,i})$ 之和。对 998244353 取模。两组排列 P, Q 本质不同，当且仅当存在 i, j 使得 $P_{i,j} \neq Q_{i,j}$。保证至少存在一种合法方案。

2 数据范围

对于所有的数据，满足 $1 \leq n \leq 50, 1 \leq m < 998244353, 0 \leq q \leq 10, 1 \leq x \leq m, 1 \leq y, w \leq n$。

- Subtask 1(4pts)：$n \leq 5, m \leq 3, q \leq 5$。
- Subtask 2(8pts)：$n \leq 7, m \leq 3, q \leq 5$。
- Subtask 3(8pts)：$m \leq 2, q = 0$。
- Subtask 4(12pts)：$q = 0$。
- Subtask 5(16pts)：$n \leq 20, q \leq 5$。
- Subtask 6(12pts)：$q \leq 5$。
- Subtask 7(20pts)：$q \leq 7$。
- Subtask 8(12pts)：$q \leq 9$。
- Subtask 9(8pts)：无特殊限制。

3 解题过程

3.1 算法一

爆搜或状压 dp，期望得分 $4 \sim 12$ 分。

3.2 算法二

首先考虑 $m = 2, q = 0$ 的情况。按照值域从小到大扫描，每次把当前扫到的数填到序列里面，在这个过程中 dp 计算答案。

设 $dp_{i,j}$ 表示考虑了值域 $1 \sim i$，当前有 j 个位置是两个排列都已经填过了对应的答案，这样就可以知道两个排列分别有多少个位置只有自己填过。

转移枚举当前 i 这个数填在另一边已经填过的位置或者新位置，i 所贡献的答案容易在过程中计算。时间复杂度 $O(n^2)$，结合算法一期望得分 20 分。

3.3 算法三

考虑 $q = 0$ 的情况。我们记所有排列的同一个位置叫做一列，一个排列叫做一行。这时无法再记录每个位置详细的状态了，但我们发现，我们只需要记录有多少列被填进去过数即可。因为每一列只会在第一次被填入数的时候对答案产生影响。

记录 $dp_{i,j}$ 表示填完了 $1\sim i$ 的数，有 j 列被填过数的排列的答案。转移的时候需要枚举一个 k 表示当前新填进去了 k 列，考虑 $dp_{i-1,j}$ 转移向 $dp_{i,j+k}$ 的系数。

首先，每一行的空位数量是容易确定的，这里就是 $j+k-i+1$。所以，如果没有当前选择的这 k 列必须填进去数的限制，那么这里方案数就是 $(j+k-i+1)^m$。对于有这个限制的情况，考虑类似斯特林数通项公式的方式进行容斥。容斥这 k 列有多少列没有任何一个排列填在这里，记为 p，那么此时方案数就是 $(j+k-i+1-p)^m$。所以总的转移系数就是 $\binom{n-j}{k}\sum_{p=0}^{k}(-1)^p\binom{k}{p}(j+k-i+1-p)^m)$。复杂度 $O(n^4)$。

可以进一步注意到容斥部分的转移系数只和 $j-i$ 有关，所以可以对于每个 $j-i,k$ 预处理转移系数，复杂度 $O(n^3)$。上述做法均能通过 Subtask4，结合算法一期望得分 32 分。

3.4 算法四

对于 $q\neq 0$ 的情况，我们记录存在 $p_{x,y}=w$ 限制的 y 为特殊列，存在上述限制的 x 记为特殊行。显然特殊列个数是 $O(q)$ 的。考虑直接将特殊列是否填过压入状态进行转移，记录 $dp_{i,j,s}$ 表示当前填完了 $1\sim i$ 的数，一共有 j 列填过数，其中 s 内的特殊列填过数的答案。转移枚举新加入的列数 k 和特殊列集合 t 进行转移。

考虑这时的转移系数，仍然用类似的方式进行计算，但这里需要考虑到特殊位置对方案数带来的影响。这种影响分为两部分，第一部分是对于当前 i 的特殊位置，他们的方案是唯一的，只能是给定的数。第二部分是对于一行中值比 i 大的特殊位置，对应的位置是不能填数的，需要空出来留给对应的数去填。

用类似的方式容斥计算转移系数，容斥掉一些普通列和一些特殊列，此时的方案数就是在剩余的若干普通列和若干特殊列中填的方案数。对于普通行，方案数仍然是 $j+k-i+1-p$，其中 p 为容斥掉的总列数。对于特殊行 x，如果存在 $p_{x,y}=i$，那么这一行的方案数就是 1，否则方案数应当减去 $p_{x,y}>i$ 的限制个数。暴力每次计算转移系数，复杂度为 $O(n^4 4^q q)$，期望得分 48 分。

3.5 算法五

算法三中通过合并本质相同的转移将复杂度降到了 $O(n^3)$，考虑使用类似的方法优化算法四。但是，我们发现，算法四的转移系数不只与 $j-i,k,s,t$ 有关，还与 i 有关。因为对于特殊行我们需要计算 $p_{x,y}>i$ 的限制个数。

然而，本质不同的转移个数级别仍然是可以缩小的。考虑 i 的影响分别是 $p_{x,y}=i$ 的限制是否存在，以及 $p_{x,y}>i$ 的限制个数。我们考虑将只存在 $p_{x,y}<i$ 的特殊列在这里视为普通列，因为它在这里与普通列没有区别。那么，我们记录集合 s',t' 分别表示 s,t 中的列内有哪些 $p_{x,y}\geq i$ 的限制，并同时记录在全局上是否存在 $p_{x,y}=i$ 的限制。

那么每个本质不同的转移都可以用 $(j-i,k,s',t',0/1)$ 来表示。通过 s',t' 我们可以知道 s,t 和 $p_{x,y}>i$ 的限制数量，通过一个 0/1 变量我们可以知道 s 中值最小的限制对应的值是否为 i，也就知道哪些限制的值为 i。通过这种方式，我们就以 $O(n^2 3^q)$ 的状态表示了所有本质不同的转移，对于每个转移分别计算系数，复杂度为 $O(n^3 4^q q)$，期望得分 60 分。

3.6 算法六

上述做法很难有优化空间了，原因在于容斥特殊列的部分占用了大量复杂度，考虑进行转化来去掉容斥过程。

考虑统计对于所有安排 $p_{i,j}$ 的方案，其对应存在满足 $x_i \le \min_{j=1}^{m} p_{j,i}$ 的 x_i 序列个数，不难发现其与原问题等价。进一步的，转化为统计对于所有 x_i 序列，对应的满足条件的 p 排列个数之和。

仍然沿用扫值域的思想，设 $dp_{i,j,s}$ 表示当前 x_i 序列填完了 $1 \sim i$ 的数，填了 j 个位置，其中特殊列对应的位置填完了 s 集合。仍然枚举 k 和 t 进行转移，我们注意到，这里不需要进行容斥了，只需要统计对应满足条件的 p 序列个数。

容易发现，对于每个排列的方案数是独立的，所以对于每个排列分开考虑。对于非特殊行，那么方案数就是将 x_i 从小到大排序后 $\prod_{i=1}^{n}(i-x_i+1)$。这个系数的计算方式考虑从大到小依次填 p_i。所以对于普通行，转移系数直接拆到每一步里面进行计算即可。

对于特殊行，我们记录 $num_{s,p,i}$ 表示考虑 s 集合内的特殊列，p 这一行值 $\ge i$ 的限制个数。那么转移系数会发生一点变化，如果这一个位置存在限制，那么这里方案数唯一，否则方案数为 $i-x_i+1-num_i$。为了方便计算,我们将那个从大到小填数的过程逆序考虑,所以这时我们每次填数是不能占用后面需要用的位置的，所以需要减去 num_i。这些系数都可以在转移的过程中快速计算，具体来说，转移方程为 $dp_{i,j,s} = \sum_{k=0}^{j} \sum_{t \subset s} dp_{i-1,k,t} \binom{n-k}{j-k-(|s|-|t|)} (\frac{(j-i+1)!}{(k-i+1)!})^{m-q} \prod_{p=0}^{q-1} \frac{(j-i-num_{s,p,i}+1)!}{(k-i-num_{t,p,i}+1)!}$。复杂度 $O(n^3 3^q q)$，期望得分 80 分。

3.7 算法七

观察上面的转移式，对于 s,t 有关的系数部分基本上都只和他们本身有关，和 $s-t$ 有关的唯一一项是组合数里面的 $|s|-|t|$。那么我们可以按照 $|t|$ 对 t 进行分类，每一类将只和 t 有关的部分进行一个高维前缀和，转移到 s 的位置。转移的时候枚举 s 计算只和 s 相关的系数部分和带有 $|s|-|t|$ 的组合数即可。分组显然只会分 $O(q)$ 组，算上高维前缀和的复杂度为 $O(n^3 2^q q^2)$，再进行一些常数优化即可通过。期望得分 $92 \sim 100$ 分。

4 参考资料

与集训队成员焦思源同学，孙培轩同学的讨论。