动态规划进阶

by 275307894a

吐槽：你对动态规划的看法

吐槽：你对动态规划的看法

- 状态设计不来

吐槽：你对动态规划的看法

- 状态设计不来
- 转移写不来

吐槽：你对动态规划的看法

- 状态设计不来
- 转移写不来
- 优化写不来

吐槽：你对动态规划的看法

- 状态设计不来
- 转移写不来
- 优化写不来
- 有些东西非常脑洞

吐槽：你对动态规划的看法

- 状态设计不来
- 转移写不来
- 优化写不来
- 有些东西非常脑洞
- 容易算重/漏

吐槽：你对动态规划的看法

- 状态设计不来
- 转移写不来
- 优化写不来
- 有些东西非常脑洞
- 容易算重/漏
- DP 的状态并不足以算出贡献

前言

前言

可能与想象的不同，这里并不会讲具体某一个 DP 算法，比如树形 DP、背包 DP，而是会用一些方法尝试去解决上面提到的各种问题。

前言

可能与想象的不同，这里并不会讲具体某一个 DP 算法，比如树形 DP、背包 DP，而是会用一些方法尝试去解决上面提到的各种问题。

因此所讲的内容可能并不入门：它涉及到某些有效的思想，而不是算法。但同时它又是最入门的：它会教你如何设计状态，如何避免后效性，这都是 DP 最基础的东西。

前言

可能与想象的不同，这里并不会讲具体某一个 DP 算法，比如树形 DP、背包 DP，而是会用一些方法尝试去解决上面提到的各种问题。

因此所讲的内容可能并不入门：它涉及到某些有效的思想，而不是算法。但同时它又是最入门的：它会教你如何设计状态，如何避免后效性，这都是 DP 最基础的东西。

学习 DP 的过程是一个潜移默化理解，而不是背套路的过程。可能今天所讲的方法并不确切，也可能会有更好的理论替代它，但是对于理解总是有帮助的。

一：如何去重

AGC013D

一：如何去重

AGC013D

一开始有 n 个颜色为黑白的球，但不知道黑白色分别有多少， m 次操作，每次先拿出一个球，再放入黑白球各一个，再拿出一个球，最后拿出的球按顺序排列会形成一个颜色序列，求颜色序列有多少种。答案对 10^9+7 取模。

一：如何去重

AGC013D

一开始有 n 个颜色为黑白的球，但不知道黑白色分别有多少， m 次操作，每次先拿出一个球，再放入黑白球各一个，再拿出一个球，最后拿出的球按顺序排列会形成一个颜色序列，求颜色序列有多少种。答案对 10^9+7 取模。

$n, m \leq 3000$。

一：如何去重

AGC013D

一开始有 n 个颜色为黑白的球，但不知道黑白色分别有多少， m 次操作，每次先拿出一个球，再放入黑白球各一个，再拿出一个球，最后拿出的球按顺序排列会形成一个颜色序列，求颜色序列有多少种。答案对 10^9+7 取模。

$n, m \leq 3000$。

我们考虑一个弱化的问题：假设我们知道黑色和白色有多少个，应该怎么做？

一：如何去重

AGC013D

一开始有 n 个颜色为黑白的球，但不知道黑白色分别有多少， m 次操作，每次先拿出一个球，再放入黑白球各一个，再拿出一个球，最后拿出的球按顺序排列会形成一个颜色序列，求颜色序列有多少种。答案对 10^9+7 取模。

$n, m \leq 3000$。

我们考虑一个弱化的问题：假设我们知道黑色和白色有多少个，应该怎么做?

我会 $O(nm)$ 的 dp！设 $dp_{i,j}$ 表示进行了第 i 次操作，这 n 个球里面有 j 个黑球的颜色序列数。转移只需要模拟这四种情况就行。

一：如何去重

AGC013D

一开始有 n 个颜色为黑白的球，但不知道黑白色分别有多少， m 次操作，每次先拿出一个球，再放入黑白球各一个，再拿出一个球，最后拿出的球按顺序排列会形成一个颜色序列，求颜色序列有多少种。答案对 10^9+7 取模。

$n,m \le 3000$。

我们考虑一个弱化的问题：假设我们知道黑色和白色有多少个，应该怎么做?

我会 $O(nm)$ 的 dp！设 $dp_{i,j}$ 表示进行了第 i 次操作，这 n 个球里面有 j 个黑球的颜色序列数。转移只需要模拟这四种情况就行。

那现在怎么做？是不是把所有可能的初始情况加起来?

一：如何去重

AGC013D

一开始有 n 个颜色为黑白的球，但不知道黑白色分别有多少， m 次操作，每次先拿出一个球，再放入黑白球各一个，再拿出一个球，最后拿出的球按顺序排列会形成一个颜色序列，求颜色序列有多少种。答案对 10^9+7 取模。

$n,m \le 3000$。

我们考虑一个弱化的问题：假设我们知道黑色和白色有多少个，应该怎么做?

我会 $O(nm)$ 的 dp！设 $dp_{i,j}$ 表示进行了第 i 次操作，这 n 个球里面有 j 个黑球的颜色序列数。转移只需要模拟这四种情况就行。

那现在怎么做？是不是把所有可能的初始情况加起来?

显然不是，不同的初始状态可能会产生同样的颜色序列。

一：如何去重

AGC013D

一开始有 n 个颜色为黑白的球，但不知道黑白色分别有多少， m 次操作，每次先拿出一个球，再放入黑白球各一个，再拿出一个球，最后拿出的球按顺序排列会形成一个颜色序列，求颜色序列有多少种。答案对 10^9+7 取模。

$n, m \le 3000$。

一：如何去重

AGC013D

一开始有 n 个颜色为黑白的球，但不知道黑白色分别有多少， m 次操作，每次先拿出一个球，再放入黑白球各一个，再拿出一个球，最后拿出的球按顺序排列会形成一个颜色序列，求颜色序列有多少种。答案对 10^9+7 取模。

$n,m \le 3000$。

现在引出了去重的第一个方法：增加限制。也就是说，如果有多个状态同时对应了一种方案，而我们只想将其中一种算入答案，那么需要找到一个限制，使得对于任意合法法案，都恰有一种状态满足这个限制，那么在 dp 的过程中加入这个限制就可以计算出正确的答案。

一：如何去重

AGC013D

一开始有 n 个颜色为黑白的球，但不知道黑白色分别有多少， m 次操作，每次先拿出一个球，再放入黑白球各一个，再拿出一个球，最后拿出的球按顺序排列会形成一个颜色序列，求颜色序列有多少种。答案对 10^9+7 取模。

$n,m\leq 3000$。

现在引出了去重的第一个方法：增加限制。也就是说，如果有多个状态同时对应了一种方案，而我们只想将其中一种算入答案，那么需要找到一个限制，使得对于任意合法法案，都恰有一种状态满足这个限制，那么在 dp 的过程中加入这个限制就可以计算出正确的答案。

举个很简单的例子，$\{1,2,1\}$ 和 $\{1,1,2\}$ 显然是同一个集合，我们希望对集合计数，怎么办呢？很简单，一个集合中的所有排列方式一定恰有一个是升序排列的，只需要强制集合升序就行。

一：如何去重

AGC013D

一开始有 n 个颜色为黑白的球，但不知道黑白色分别有多少， m 次操作，每次先拿出一个球，再放入黑白球各一个，再拿出一个球，最后拿出的球按顺序排列会形成一个颜色序列，求颜色序列有多少种。答案对 10^9+7 取模。

$n,m\le 3000$。

一：如何去重

AGC013D

一开始有 n 个颜色为黑白的球，但不知道黑白色分别有多少， m 次操作，每次先拿出一个球，再放入黑白球各一个，再拿出一个球，最后拿出的球按顺序排列会形成一个颜色序列，求颜色序列有多少种。答案对 10^9+7 取模。

$n,m\le 3000$。

对于这道题而言，我们记 (i,p_i) 表示第 i 次操作以后黑球有 p_i 个。如果将这些点在平面上连接起来，就会形成一条折线，一条折线合法当且仅当它始终在 $[0,n]$ 范围内。对于所有同构的折线，只需要计算一次。

一：如何去重

AGC013D

一开始有 n 个颜色为黑白的球，但不知道黑白色分别有多少， m 次操作，每次先拿出一个球，再放入黑白球各一个，再拿出一个球，最后拿出的球按顺序排列会形成一个颜色序列，求颜色序列有多少种。答案对 10^9+7 取模。

$n, m \leq 3000$。

对于这道题而言，我们记 (i, p_i) 表示第 i 次操作以后黑球有 p_i 个。如果将这些点在平面上连接起来，就会形成一条折线，一条折线合法当且仅当它始终在 $[0, n]$ 范围内。对于所有同构的折线，只需要计算一次。

注意到初始情况是取遍所有 $[0, n]$ 中的正整数的，也就是说，在一种颜色序列对应的折线中，总有恰好一种 p_i 的最小值为 0！因此只需要在 dp 的时候增加一维 $0/1$ 表示是否最小值取到 0 即可。时间复杂度 $O(n^2)$。

一：如何去重

CF720D Slalom

一个 $n \times m$ 的网格，其中有 k 个矩形障碍，保证这些障碍不重叠。求从 $(1,1)$ 走到 (n,m)，每步只能往右或往上走，不经过任何障碍的方案数。

两种方案被视为不同，当且仅当存在一个障碍，它在第一种方案里被从右侧绕过，而在第二种方案里被从左侧绕过（第一种左，第二种右同理）。

$n,m \le 10^6$，$k \le 10^5$。在今天的讨论中，你可以看作 $n,m \le 10^3$。

一：如何去重

CF720D Slalom

一个 $n \times m$ 的网格，其中有 k 个矩形障碍，保证这些障碍不重叠。求从 $(1,1)$ 走到 (n,m)，每步只能往右或往上走，不经过任何障碍的方案数。

两种方案被视为不同，当且仅当存在一个障碍，它在第一种方案里被从右侧绕过，而在第二种方案里被从左侧绕过（第一种左，第二种右同理）。

$n,m \le 10^6$，$k \le 10^5$。在今天的讨论中，你可以看作 $n,m \le 10^3$。

如果我们考虑不同的方案为经过的格子不同，那么这是一个简单的 dp 问题，不过多展开了。

一：如何去重

CF720D Slalom

一个 $n \times m$ 的网格，其中有 k 个矩形障碍，保证这些障碍不重叠。求从 $(1,1)$ 走到 (n,m)，每步只能往右或往上走，不经过任何障碍的方案数。

两种方案被视为不同，当且仅当存在一个障碍，它在第一种方案里被从右侧绕过，而在第二种方案里被从左侧绕过（第一种左，第二种右同理）。

$n,m \le 10^6$，$k \le 10^5$。在今天的讨论中，你可以看作 $n,m \le 10^3$。

如果我们考虑不同的方案为经过的格子不同，那么这是一个简单的 dp 问题，不过多展开了。

首先我们修改定义为：第一种方案中从左侧绕过，第二种方案中从下面绕过，容易发现不影响判定。

一：如何去重

CF720D Slalom

一个 $n \times m$ 的网格，其中有 k 个矩形障碍，保证这些障碍不重叠。求从 $(1,1)$ 走到 (n,m)，每步只能往右或往上走，不经过任何障碍的方案数。

两种方案被视为不同，当且仅当存在一个障碍，它在第一种方案里被从右侧绕过，而在第二种方案里被从左侧绕过（第一种左，第二种右同理）。

$n,m \le 10^6$，$k \le 10^5$。在今天的讨论中，你可以看作 $n,m \le 10^3$。

如果我们考虑不同的方案为经过的格子不同，那么这是一个简单的 dp 问题，不过多展开了。

首先我们修改定义为：第一种方案中从左侧绕过，第二种方案中从下面绕过，容易发现不影响判定。

假设我们已经确定每个矩形是从左边过还是从下面过，那么我们强制一条路径一直向左走，直到不能走了才往上走。一个从下面过的矩形对路径没有限制，一个从左边过的矩形强制在走到这个矩形前就要高于这个矩形，可以在 dp 时简单处理，时间复杂度 $O(n^2)$。

一：如何去重

CF720D Slalom

一个 $n \times m$ 的网格，其中有 k 个矩形障碍，保证这些障碍不重叠。求从 $(1,1)$ 走到 (n,m)，每步只能往右或往上走，不经过任何障碍的方案数。

两种方案被视为不同，当且仅当存在一个障碍，它在第一种方案里被从右侧绕过，而在第二种方案里被从左侧绕过（第一种左，第二种右同理）。

$n,m \le 10^6$，$k \le 10^5$。在今天的讨论中，你可以看作 $n,m \le 10^3$。

如果我们考虑不同的方案为经过的格子不同，那么这是一个简单的 dp 问题，不过多展开了。

首先我们修改定义为：第一种方案中从左侧绕过，第二种方案中从下面绕过，容易发现不影响判定。

假设我们已经确定每个矩形是从左边过还是从下面过，那么我们强制一条路径一直向左走，直到不能走了才往上走。一个从下面过的矩形对路径没有限制，一个从左边过的矩形强制在走到这个矩形前就要高于这个矩形，可以在 dp 时简单处理，时间复杂度 $O(n^2)$。

容易用线段树优化到 $O(k \log n)$，但是这不在今天讨论的范围内。

一：如何去重

一些题目：

ARC101F Robots and Exits

NOI2007 生成树计数

二：转移方向

二：转移方向

DP 的本质是 DAG 上的转移，所谓转移方向·，就是需要找到一个合理的顺序，使得在某个状态被 DP 到的时候其前驱都被 DP 到了，通常被称作”没有后效性“。

二：转移方向

DP 的本质是 DAG 上的转移，所谓转移方向·，就是需要找到一个合理的顺序，使得在某个状态被 DP 到的时候其前驱都被 DP 到了，通常被称作”没有后效性“。

这在图上的 DP 中尤为重要。

二：转移方向

CF1155F

给定一张图，保证其是边双连通分量。求一个最小的边集，使得这个边集是这张图边集的子集，且只保留这个边集中的边后这张图是边双连通分量。

$n \leq 14$

二：转移方向

CF1155F

给定一张图，保证其是边双连通分量。求一个最小的边集，使得这个边集是这张图边集的子集，且只保留这个边集中的边后这张图是边双连通分量。

$n \leq 14$

考虑对边双连通分量的某种拆解：其基础是一个环，然后每次加入一条链，满足链的两端点在当前边双连通分量中，剩下的点不在。容易发现加入后仍然是一个边双连通分量。

二：转移方向

CF1155F

给定一张图，保证其是边双连通分量。求一个最小的边集，使得这个边集是这张图边集的子集，且只保留这个边集中的边后这张图是边双连通分量。

$n \le 14$

考虑对边双连通分量的某种拆解：其基础是一个环，然后每次加入一条链，满足链的两端点在当前边双连通分量中，剩下的点不在。容易发现加入后仍然是一个边双连通分量。

则设 dp_S 表示 S 点集连接成边双连通分量的最小边数，然后枚举一条链加入即可。时间复杂度 $O(3^n n^2)$。

二：转移方向

UOJ #22 外星人

给定一个长度为 n 的序列 a 和一个数 x，x 的最终值 y 定义令 i 从 1 到 n，使 $x \to x \bmod a_i$。你需要重排 a，使 y 最大，输出最大值及方案数。

$n \leq 10^3, x \leq 5 \times 10^3$。

二：转移方向

UOJ #22 外星人

给定一个长度为 n 的序列 a 和一个数 x，x 的最终值 y 定义令 i 从 1 到 n，使 $x \to x \bmod a_i$。你需要重排 a，使 y 最大，输出最大值及方案数。

$n \le 10^3, x \le 5 \times 10^3$。

一个显著的观察是，对 x 真正有效的 a_i 是递减的，这就启发我们，按照 a 从大到小进行 dp。

二：转移方向

UOJ #22 外星人

给定一个长度为 n 的序列 a 和一个数 x，x 的最终值 y 定义令 i 从 1 到 n，使 $x \to x \bmod a_i$。你需要重排 a，使 y 最大，输出最大值及方案数。

$n \leq 10^3, x \leq 5 \times 10^3$。

一个显著的观察是，对 x 真正有效的 a_i 是递减的，这就启发我们，按照 a 从大到小进行 dp。

设 $dp_{i,j}$ 表示从大到小安排好第 i 个 a_i，现在 x 为 j 的方案数。若 $j < a_i$，则无事发生，a_i 可以被排到后面任意一个位置。否则有两种情况，若 a_i 对 x 有影响，那么其必须恰好在这个位置。否则其不能在最开头，可以被插入到后面的位置里面去。

二：转移方向

UOJ #22 外星人

给定一个长度为 n 的序列 a 和一个数 x，x 的最终值 y 定义令 i 从 1 到 n，使 $x \to x \bmod a_i$。你需要重排 a，使 y 最大，输出最大值及方案数。

$n \le 10^3, x \le 5 \times 10^3$。

一个显著的观察是，对 x 真正有效的 a_i 是递减的，这就启发我们，按照 a 从大到小进行 dp。

设 $dp_{i,j}$ 表示从大到小安排好第 i 个 a_i，现在 x 为 j 的方案数。若 $j < a_i$，则无事发生，a_i 可以被排到后面任意一个位置。否则有两种情况，若 a_i 对 x 有影响，那么其必须恰好在这个位置。否则其不能在最开头，可以被插入到后面的位置里面去。

时间复杂度 $O(nx)$。

二：转移方向

一些题目：

清华集训2014 主旋律

三：贡献形式

即通常所说的拆贡献。拆贡献实际上是分段计算贡献，主要思想是众所周知的加法原理和乘法原理，使贡献能被 DP 状态所表示。

三：贡献形式

即通常所说的拆贡献。拆贡献实际上是分段计算贡献，主要思想是众所周知的加法原理和乘法原理，使贡献能被 DP 状态所表示。

AGC002F

三：贡献形式

即通常所说的拆贡献。拆贡献实际上是分段计算贡献，主要思想是众所周知的加法原理和乘法原理，使贡献能被 DP 状态所表示。

AGC002F

给你 n 种颜色的球，每种颜色的球有 k 个，把这 $n \times k$ 个球排成一排，把每一种颜色的最左边出现的球涂成白色(初始球不包含白色)，求有多少种不同的颜色序列，答案对 10^9+7 取模。

三：贡献形式

即通常所说的拆贡献。拆贡献实际上是分段计算贡献，主要思想是众所周知的加法原理和乘法原理，使贡献能被 DP 状态所表示。

AGC002F

给你 n 种颜色的球，每种颜色的球有 k 个，把这 $n \times k$ 个球排成一排，把每一种颜色的最左边出现的球涂成白色(初始球不包含白色)，求有多少种不同的颜色序列，答案对 10^9+7 取模。

$1 \le n,k \le 2000$。

三：贡献形式

即通常所说的拆贡献。拆贡献实际上是分段计算贡献，主要思想是众所周知的加法原理和乘法原理，使贡献能被 DP 状态所表示。

AGC002F

给你 n 种颜色的球，每种颜色的球有 k 个，把这 $n \times k$ 个球排成一排，把每一种颜色的最左边出现的球涂成白色(初始球不包含白色)，求有多少种不同的颜色序列，答案对 $10^9 + 7$ 取模。

$1 \leq n, k \leq 2000$。

我们把这个题看成这样一个问题：给 $n \times k$ 个球染色，染 n 种颜色，每个 $k - 1$ 个，另外染 n 个白球，要求前缀的其它颜色种类数不多于白球个数。

三：贡献形式

即通常所说的拆贡献。拆贡献实际上是分段计算贡献，主要思想是众所周知的加法原理和乘法原理，使贡献能被 DP 状态所表示。

AGC002F

给你 n 种颜色的球，每种颜色的球有 k 个，把这 $n \times k$ 个球排成一排，把每一种颜色的最左边出现的球涂成白色(初始球不包含白色)，求有多少种不同的颜色序列，答案对 10^9+7 取模。

$1 \le n, k \le 2000$。

我们把这个题看成这样一个问题：给 $n \times k$ 个球染色，染 n 种颜色，每个 $k-1$ 个，另外染 n 个白球，要求前缀的其它颜色种类数不多于白球个数。

设 $f_{i,j}$ 表示已经放了 i 个白球，另外放了有 j 种颜色的方案数。转移可以考虑除去已经填掉了的 $i+j(k-1)$ 个位置中，第一个放的是什么。如果是白球，那么 $f_{i,j} \to f_{i+1,j}$，如果是其它颜色的球，那么提前选掉这些球的位置，是 $\binom{nk-i-j(k-1)-1}{k-2}$，只用选 $k-2$ 个的原因是要先填一个在最开头。

三：贡献形式

即通常所说的拆贡献。拆贡献实际上是分段计算贡献，主要思想是众所周知的加法原理和乘法原理，使贡献能被 DP 状态所表示。

AGC002F

给你 n 种颜色的球，每种颜色的球有 k 个，把这 $n \times k$ 个球排成一排，把每一种颜色的最左边出现的球涂成白色(初始球不包含白色)，求有多少种不同的颜色序列，答案对 10^9+7 取模。

$1 \le n, k \le 2000$。

我们把这个题看成这样一个问题：给 $n \times k$ 个球染色，染 n 种颜色，每个 $k-1$ 个，另外染 n 个白球，要求前缀的其它颜色种类数不多于白球个数。

设 $f_{i,j}$ 表示已经放了 i 个白球，另外放了有 j 种颜色的方案数。转移可以考虑除去已经填掉了的 $i+j(k-1)$ 个位置中，第一个放的是什么。如果是白球，那么 $f_{i,j} \to f_{i+1,j}$，如果是其它颜色的球，那么提前选掉这些球的位置，是 $\binom{nk-i-j(k-1)-1}{k-2}$，只用选 $k-2$ 个的原因是要先填一个在最开头。

直接 dp 即可，时间复杂度 $O(nk)$。

三：贡献形式

在有关大小的排列计数中非常有用。

三：贡献形式

在有关大小的排列计数中非常有用。

CF704B Ant Man

三：贡献形式

在有关大小的排列计数中非常有用。

CF704B Ant Man

有 n 个元素，第 i 个元素有五个参数 x_i, a_i, b_i, c_i, d_i。

三：贡献形式

在有关大小的排列计数中非常有用。

CF704B Ant Man

有 n 个元素，第 i 个元素有五个参数 x_i, a_i, b_i, c_i, d_i。

你需要求出一个 $1 \sim n$ 的排列 p，满足 $p_1 = s, p_n = e$，同时最小化这个排列的权值。

三：贡献形式

在有关大小的排列计数中非常有用。

CF704B Ant Man

有 n 个元素，第 i 个元素有五个参数 x_i, a_i, b_i, c_i, d_i。

你需要求出一个 $1 \sim n$ 的排列 p，满足 $p_1 = s, p_n = e$，同时最小化这个排列的权值。

一个排列的权值为 $\sum_{i=1}^{n-1} f(p_i, p_{i+1})$，其中 $f(i, j)$ 的值有两种情况：

三：贡献形式

在有关大小的排列计数中非常有用。

CF704B Ant Man

有 n 个元素，第 i 个元素有五个参数 x_i, a_i, b_i, c_i, d_i。

你需要求出一个 $1 \sim n$ 的排列 p，满足 $p_1 = s, p_n = e$，同时最小化这个排列的权值。

一个排列的权值为 $\sum_{i=1}^{n-1} f(p_i, p_{i+1})$，其中 $f(i, j)$ 的值有两种情况：

- 若 $i > j$，则 $f(i,j) = x_i - x_j + c_i + b_j$。
- 若 $i < j$，则 $f(i,j) = x_j - x_i + d_i + a_j$。

三：贡献形式

在有关大小的排列计数中非常有用。

CF704B Ant Man

有 n 个元素，第 i 个元素有五个参数 x_i, a_i, b_i, c_i, d_i。

你需要求出一个 $1 \sim n$ 的排列 p，满足 $p_1 = s, p_n = e$，同时最小化这个排列的权值。

一个排列的权值为 $\sum_{i=1}^{n-1} f(p_i, p_{i+1})$，其中 $f(i, j)$ 的值有两种情况：

- 若 $i > j$，则 $f(i, j) = x_i - x_j + c_i + b_j$。
- 若 $i < j$，则 $f(i, j) = x_j - x_i + d_i + a_j$。

$n \le 5 \times 10^3$，$s \ne e$，$1 \le x_1 < x_2 < \cdots < x_n \le 10^9$，$1 \le a_i, b_i, c_i, d_i \le 10^9$。

三：贡献形式

在有关大小的排列计数中非常有用。

CF704B Ant Man

三：贡献形式

在有关大小的排列计数中非常有用。

CF704B Ant Man

考虑从小到大加入每个数，如果两个数的大小关系是确定的，那么两个数对答案的贡献也是确定的。

三：贡献形式

在有关大小的排列计数中非常有用。

CF704B Ant Man

考虑从小到大加入每个数，如果两个数的大小关系是确定的，那么两个数对答案的贡献也是确定的。

在加入数的过程中，已加入的数会形成若干个连续段，当加入一个点的时候，若其左边没有连续段，则说明其小于左边的数，否则大于左边的数，对于右边同理。

三：贡献形式

在有关大小的排列计数中非常有用。

CF704B Ant Man

考虑从小到大加入每个数，如果两个数的大小关系是确定的，那么两个数对答案的贡献也是确定的。

在加入数的过程中，已加入的数会形成若干个连续段，当加入一个点的时候，若其左边没有连续段，则说明其小于左边的数，否则大于左边的数，对于右边同理。

则设 $f_{i,j}$ 表示加入了 i 个数，有 j 个连续段。每加入一个数，会产生三种情况：合并两个连续段，作为一个连续段的某个端点，以及单独成为一个连续段，转移是平凡的。时间复杂度 $O(n^2)$。

三：贡献形式

在有关大小的排列计数中非常有用。

CF704B Ant Man

考虑从小到大加入每个数，如果两个数的大小关系是确定的，那么两个数对答案的贡献也是确定的。

在加入数的过程中，已加入的数会形成若干个连续段，当加入一个点的时候，若其左边没有连续段，则说明其小于左边的数，否则大于左边的数，对于右边同理。

则设 $f_{i,j}$ 表示加入了 i 个数，有 j 个连续段。每加入一个数，会产生三种情况：合并两个连续段，作为一个连续段的某个端点，以及单独成为一个连续段，转移是平凡的。时间复杂度 $O(n^2)$。

类似的题还有 ZJOI2012 波浪。

三：贡献形式

CF1824G Tenzing and Random Operations

有一长度为 n 的序列 a 和一整数 v，对该序列进行 m 次操作。

每次操作中，等概率地随机选择一整数 $i \in [1, n]$，对所有的 $j \in [i, n]$，赋值 $a_j \leftarrow a_j + v$，求出 m 次操作后 $\prod_{i=1}^{n} a_i$ 的期望值，对 $10^9 + 7$ 取模。

数据范围：$0 \le n \le 5000$，$1 \le m, v \le 10^9$，$a_i \in [1, 10^9] \cap \mathbb{Z}$。

三：贡献形式

CF1824G Tenzing and Random Operations

有一长度为 n 的序列 a 和一整数 v，对该序列进行 m 次操作。

每次操作中，等概率地随机选择一整数 $i \in [1, n]$，对所有的 $j \in [i, n]$，赋值 $a_j \leftarrow a_j + v$，求出 m 次操作后 $\prod_{i=1}^{n} a_i$ 的期望值，对 $10^9 + 7$ 取模。

数据范围：$0 \le n \le 5000$，$1 \le m, v \le 10^9$，$a_i \in [1, 10^9] \cap \mathbb{Z}$。

先差分，变成 b 序列的单点加法，然后利用乘法分配律将拆成：第 i 个数只能在前 i 个里面选，将 n 个数乘起来，然后对所有这样的求和。

三：贡献形式

CF1824G Tenzing and Random Operations

有一长度为 n 的序列 a 和一整数 v，对该序列进行 m 次操作。

每次操作中，等概率地随机选择一整数 $i \in [1, n]$，对所有的 $j \in [i, n]$，赋值 $a_j \leftarrow a_j + v$，求出 m 次操作后 $\prod_{i=1}^{n} a_i$ 的期望值，对 $10^9 + 7$ 取模。

数据范围：$0 \le n \le 5000$，$1 \le m, v \le 10^9$，$a_i \in [1, 10^9] \cap \mathbb{Z}$。

先差分，变成 b 序列的单点加法，然后利用乘法分配律将拆成：第 i 个数只能在前 i 个里面选，将 n 个数乘起来，然后对所有这样的求和。

但是这还不够，需要进一步拆贡献。对于每个 b，可以表示成 a 加上若干个 v 的形式。这也可以用乘法分配律拆成若干个 a 相乘与若干个 v 相乘。

三：贡献形式

CF1824G Tenzing and Random Operations

有一长度为 n 的序列 a 和一整数 v，对该序列进行 m 次操作。

每次操作中，等概率地随机选择一整数 $i \in [1, n]$，对所有的 $j \in [i, n]$，赋值 $a_j \leftarrow a_j + v$，求出 m 次操作后 $\prod_{i=1}^{n} a_i$ 的期望值，对 $10^9 + 7$ 取模。

数据范围：$0 \le n \le 5000$，$1 \le m, v \le 10^9$，$a_i \in [1, 10^9] \cap \mathbb{Z}$。

先差分，变成 b 序列的单点加法，然后利用乘法分配律将拆成：第 i 个数只能在前 i 个里面选，将 n 个数乘起来，然后对所有这样的求和。

但是这还不够，需要进一步拆贡献。对于每个 b，可以表示成 a 加上若干个 v 的形式。这也可以用乘法分配律拆成若干个 a 相乘与若干个 v 相乘。

设 $f_{i,j}$ 表示到了第 i 个数，已经选中的 v 有 j 个。对于第 i 个位置的点，要么将 a_i 乘入答案，要么从以前的 v 中选一个乘入答案，要么新开一个 v 乘入答案，注意这个新开的 v 可以放在 $[1, i]$ 中的任意一个位置，这样就可以 $O(n^2)$ 转移。

三：贡献形式

CF1268E Happy Cactus

给定一张仙人掌图，第 i 条边连接 u, v，边权为 i

定义路径为 "Happy Path" 当且仅当其满足沿途边权递增。

定义点对 (u, v) Happy 当且仅当存在一条 Happy Path 以 u 为起点，v 为终点。

对于 $u = 1, 2 ... n$，求满足 (u, v) Happy 的 v 的数量。

$n, m \leq 5 \cdot 10^5$，可以先考虑 $m = n - 1$ 的情况。

三：贡献形式

CF1268E Happy Cactus

给定一张仙人掌图，第 i 条边连接 u, v，边权为 i

定义路径为 "Happy Path" 当且仅当其满足沿途边权递增。

定义点对 (u, v) Happy 当且仅当存在一条 Happy Path 以 u 为起点，v 为终点。

对于 $u = 1, 2 \ldots n$，求满足 (u, v) Happy 的 v 的数量。

$n, m \le 5 \cdot 10^5$，可以先考虑 $m = n - 1$ 的情况。

考虑 $m = n - 1$ 的情况，将边按照从大到小的顺序加入。设 f_i 表示以 i 开头的 Happy Path 条数，则对于一条加入的边 x, y，有 $f_x + f_y \to f_x, f_y$。

三：贡献形式

CF1268E Happy Cactus

给定一张仙人掌图，第 i 条边连接 u,v，边权为 i

定义路径为 "Happy Path" 当且仅当其满足沿途边权递增。

定义点对 (u,v) Happy 当且仅当存在一条 Happy Path 以 u 为起点，v 为终点。

对于 $u=1,2...n$，求满足 (u,v) Happy 的 v 的数量。

$n,m\le 5\cdot 10^5$，可以先考虑 $m=n-1$ 的情况。

考虑 $m=n-1$ 的情况，将边按照从大到小的顺序加入。设 f_i 表示以 i 开头的 Happy Path 条数，则对于一条加入的边 x,y，有 $f_x+f_y\to f_x,f_y$。

但是有环的情况下就不能这么做，因为一个环的最小边如果这样计算可能会算出两种走到最大边的方式。因此我们需要其贡献 -1。所以对于每个环，如果最小边能走到最大边，那么记录加入最大边的时候两端点的权值，然后在计算最小边的时候减去即可。

三：贡献形式

CF1268E Happy Cactus

给定一张仙人掌图，第 i 条边连接 u, v，边权为 i

定义路径为 "Happy Path" 当且仅当其满足沿途边权递增。

定义点对 (u, v) Happy 当且仅当存在一条 Happy Path 以 u 为起点，v 为终点。

对于 $u = 1, 2...n$，求满足 (u, v) Happy 的 v 的数量。

$n, m \le 5 \cdot 10^5$，可以先考虑 $m = n - 1$ 的情况。

考虑 $m = n - 1$ 的情况，将边按照从大到小的顺序加入。设 f_i 表示以 i 开头的 Happy Path 条数，则对于一条加入的边 x, y，有 $f_x + f_y \to f_x, f_y$。

但是有环的情况下就不能这么做，因为一个环的最小边如果这样计算可能会算出两种走到最大边的方式。因此我们需要其贡献 -1。所以对于每个环，如果最小边能走到最大边，那么记录加入最大边的时候两端点的权值，然后在计算最小边的时候减去即可。

时间复杂度 $O(n + m)$。

三：贡献形式

这一部分有比较多的练习：

CF513G Inversions problem

CF1175G Yet Another Partiton Problem

PKUWC2018 随机算法

NOI2019 斗主地

九省联考2018 秘密袭击

ZJOI2016 线段树

八省联考2018 林克卡特树

四：保留有效状态

这是属于 DP 优化范畴的，基础的有两种方法：去除多余状态和合并有效状态。

CF1188C Array Beauty

定义一个序列 $b_1, \ldots, b_n$ 的美丽值为 $\min_{1 \le i < j \le n}\{|b_i - b_j|\}$。给定一个长度为 n 的序列 a，求 a 的所有长度为 k 的子序列的美丽值之和对 998244353 取模的结果。

$2 \le k \le n \le 1000$，$0 \le a_i \le 10^5$。

四：保留有效状态

这是属于 DP 优化范畴的，基础的有两种方法：去除多余状态和合并有效状态。

CF1188C Array Beauty

定义一个序列 $b_1, \ldots, b_n$ 的美丽值为 $\min_{1\le i<j\le n}\{|b_i - b_j|\}$。给定一个长度为 n 的序列 a，求 a 的所有长度为 k 的子序列的美丽值之和对 998244353 取模的结果。

$2 \le k \le n \le 1000$，$0 \le a_i \le 10^5$。

差分答案，枚举 p 表示美丽值至少是 p，则要求任意两个 b 的间隔都大于等于 p，最后的答案即为所有 p 对应方案数之和。

四：保留有效状态

这是属于 DP 优化范畴的，基础的有两种方法：去除多余状态和合并有效状态。

CF1188C Array Beauty

定义一个序列 $b_1, \ldots, b_n$ 的美丽值为 $\min_{1\le i<j\le n}\{|b_i - b_j|\}$。给定一个长度为 n 的序列 a，求 a 的所有长度为 k 的子序列的美丽值之和对 998244353 取模的结果。

$2 \le k \le n \le 1000$，$0 \le a_i \le 10^5$。

差分答案，枚举 p 表示美丽值至少是 p，则要求任意两个 b 的间隔都大于等于 p，最后的答案即为所有 p 对应方案数之和。

设 $f_{i,j}$ 表示选到第 i 个，选了 j 个，容易 two-pointers+前缀和优化到 $O(nk)$，这样复杂度是 $O(nkV)$ 的，其中 V 是值域。

四：保留有效状态

这是属于 DP 优化范畴的，基础的有两种方法：去除多余状态和合并有效状态。

CF1188C Array Beauty

定义一个序列 $b_1, \ldots, b_n$ 的美丽值为 $\min_{1 \le i < j \le n}\{|b_i - b_j|\}$。给定一个长度为 n 的序列 a，求 a 的所有长度为 k 的子序列的美丽值之和对 998244353 取模的结果。

$2 \le k \le n \le 1000$，$0 \le a_i \le 10^5$。

差分答案，枚举 p 表示美丽值至少是 p，则要求任意两个 b 的间隔都大于等于 p，最后的答案即为所有 p 对应方案数之和。

设 $f_{i,j}$ 表示选到第 i 个，选了 j 个，容易 two-pointers+前缀和优化到 $O(nk)$，这样复杂度是 $O(nkV)$ 的，其中 V 是值域。

容易观察到如下事实：选择 k 个数的美丽值一定不大于 $\frac{V}{k}$，因此 $p \le \frac{V}{k}$，所以复杂度变成 $O(nV)$。

四：保留有效状态

CF506E Mr. Kitayuta's Gift

给定一个小写字符串 s 和一个正整数 n。

要求在 s 中插入恰好 n 个小写字符使其回文的方案数，两个方案不同当且仅当它们得到的串不同，与插入顺序和位置无关。

$|s| \leq 200$，$n \leq 10^9$，答案对 $10^4 + 7$ 取模。

四：保留有效状态

CF506E Mr. Kitayuta’s Gift

给定一个小写字符串 s 和一个正整数 n。

要求在 s 中插入恰好 n 个小写字符使其回文的方案数，两个方案不同当且仅当它们得到的串不同，与插入顺序和位置无关。

$|s| \le 200$，$n \le 10^9$，答案对 10^4+7 取模。

DP 可以看成在 DFA 上转移。如果两个 DFA 上的节点所有的后继节点都相同，那么这两个点可以看作是相同的，可以合并成一个点。这是合并状态的本质思想。

四：保留有效状态

CF506E Mr. Kitayuta’s Gift

四：保留有效状态

CF506E Mr. Kitayuta’s Gift

对于这题，显然可以设 $dp_{k,i,j}$ 表示最终串从左往右最多匹配到 i，从右往左最多匹配到 j，已经考虑了长度为 k 的前缀和后缀的方案数，

四：保留有效状态

CF506E Mr. Kitayuta’s Gift

对于这题，显然可以设 $dp_{k,i,j}$ 表示最终串从左往右最多匹配到 i，从右往左最多匹配到 j，已经考虑了长度为 k 的前缀和后缀的方案数，

不妨假设 $2 \mid n+|s|$。对于对于一对 i,j，当 $k \to k+1$ 的时候，分类讨论：

四：保留有效状态

CF506E Mr. Kitayuta’s Gift

对于这题，显然可以设 $dp_{k,i,j}$ 表示最终串从左往右最多匹配到 i，从右往左最多匹配到 j，已经考虑了长度为 k 的前缀和后缀的方案数，

不妨假设 $2 \mid n + |s|$。对于对于一对 i, j，当 $k \to k + 1$ 的时候，分类讨论：

- $s_i = s_j$
  - $f_{i,j} \to f_{i+1,j-1}$
  - $f_{i,j} \times 25 \to f_{i,j}$
- $s_i \neq s_j$
  - $f_{i,j} \times 24 \to f_{i,j}$
  - $f_{i,j} \to f_{i+1,j}$，对于 j 同理。

四：保留有效状态

CF506E Mr. Kitayuta’s Gift

对于这题，显然可以设 $dp_{k,i,j}$ 表示最终串从左往右最多匹配到 i，从右往左最多匹配到 j，已经考虑了长度为 k 的前缀和后缀的方案数，

不妨假设 $2 \mid n+|s|$。对于对于一对 i,j，当 $k \to k+1$ 的时候，分类讨论：

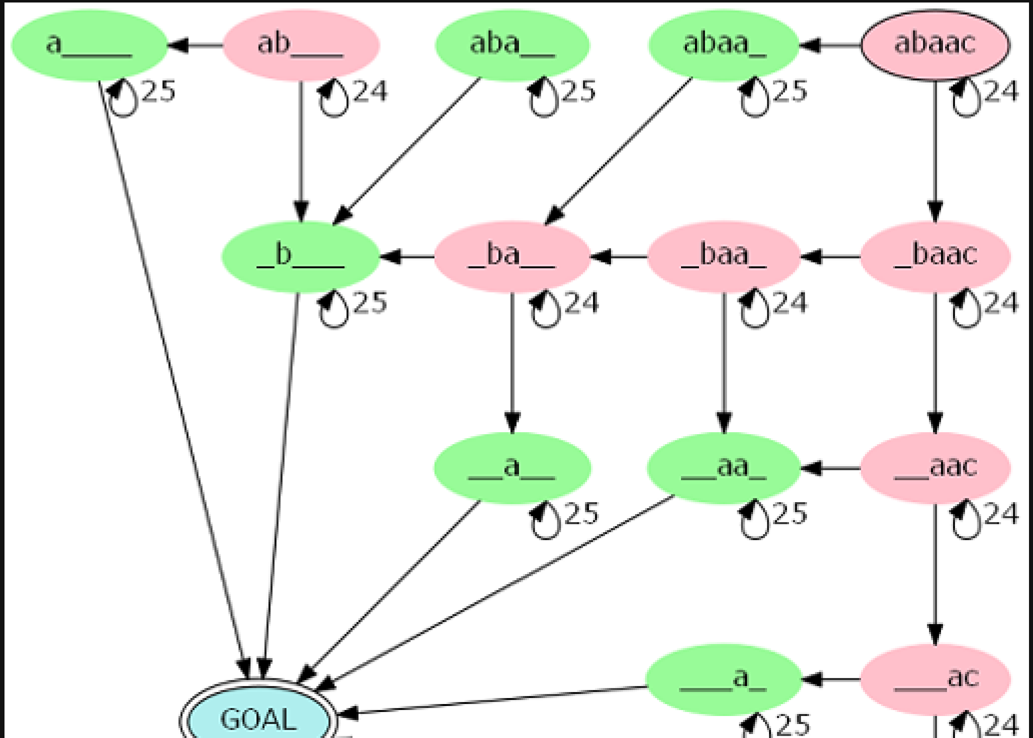
- $s_i = s_j$
  - $f_{i,j} \to f_{i+1,j-1}$
  - $f_{i,j} \times 25 \to f_{i,j}$
- $s_i \neq s_j$
  - $f_{i,j} \times 24 \to f_{i,j}$
  - $f_{i,j} \to f_{i+1,j}$，对于 j 同理。

显然可以矩乘，但是是 $O(|s|^6 \log n)$ 的。

四：保留有效状态

CF506E Mr. Kitayuta’s Gift

把自动机画出来长这样：



四：保留有效状态

CF506E Mr. Kitayuta’s Gift

四：保留有效状态

CF506E Mr. Kitayuta’s Gift

对于一个 $f_{1,n}$ 到最终 $f_{i+1,i}$ 的转移，最终方案数的贡献只与其经过了 x 个 $s_i = s_j$ 的点，$|s| - 2x$ 个 $s_i \neq s_j$ 的点有关。因此可以先记忆化搜索出 f_x 表示经过 x 个 $s_i = s_j$ 的点的方案数。然后剩下的可以设 $f_{i,j,0/1}$ 表示到了走了 i 个自环，走到第 j 个白点/黑点的方案数，可以矩乘，时间复杂度 $O(|s|^3 \log n)$。

四：保留有效状态

CF506E Mr. Kitayuta’s Gift

压缩后的自动机长这样：