DP 好题乱讲

qinyubo

2077 年 7 月 18 日

目录

计数类 DP

计数的要点是不重不漏。
由于求概率、期望的题大多都是与计数有关的，所以放一块了。

[CF1743G] Anti-Fibonacci Cut

定义斐波那契字符串为:
$F_0 = \texttt{0}, F_1 = \texttt{1}, F_n = F_{n-1} + F_{n-2} (n \geq 2)$
其中 + 表示字符串的顺序拼接。例如，前几个斐波那契串如下:
$\texttt{0}, \texttt{1}, \texttt{10}, \texttt{101}, \texttt{10110}, \texttt{10110101}, \texttt{1011010110110}, \cdots$
再定义 $g(s)$ 表示把 s 分割成若干个子串，且没有任何子串是斐波那契串的方案数。
给出 n 个 01 串 $s_1, s_2, \cdots, s_n$，要求对于每个 $1 \leq i \leq n$，计算 $g(s_1 + s_2 + \cdots + s_i) \bmod 998244353$。
$1 \leq n \leq 3000$，$1 \leq |s_i| \leq 1000$，**空间限制 4M**

[CF1743G] Anti-Fibonacci Cut

先考虑没有空间限制怎么做。
假设我们要求 $g(s)$，那么一个简单易行的 DP 是设 f_i 表示前 i 个字符的方案数，那么有：

$$f_i = \sum_{0 \le j < i} [s_{j+1..i} \text{ is not a Fibonacci string}] f_j$$

由于斐波那契串的长度是指数级的，记一下前缀和，减一下不合法的转移就行了。
设 $\sum |s_i| = T$，则时间 $O(T \log T)$，空间 $O(T)$。

[CF1743G] Anti-Fibonacci Cut

接下来考虑卡空间。
我们发现，空间的瓶颈在于要找出所有不合法的转移并减掉，所以不得不记录整个 f 数组。
但是注意到我们只会往字符串后面加字符，而对于任意 $1 \le i \le j$ 都有 F_i 是 F_j 的前缀。
这启发我们用一个数组维护当前字符串有哪些后缀可以作为斐波那契串的前缀，这样的话每增加一个字符就排除不再是斐波那契串的前缀的后缀并把新的后缀放进去，更新答案只需要看每个后缀长度是不是斐波那契数就行了。
由于 F_i 的最长 Border 是 F_{i-2}，所以任意时刻数组里的后缀数量是 $O(\log T)$ 的，这样我们就成功地把空间复杂度降到了 $O(\log T + \max\{|s_i|\})$。

[P2106] Sam 数

一个数是好的，当且仅当其十进制表示下相邻两位数字之差不超过 2。
求 n 位好的数字的个数，对 10^9+7 取模。
$1 \le n \le 10^{18}$

[P2106] Sam 数

数位 DP。设 $f_{i,j}$ 表示第 i 位填 j，前 $i-1$ 位的方案数。
则 $f_{i,j} = \sum_{0\le k\le 9, |k-j|\le 2} f_{i-1,k}$。
然后发现可以矩阵加速，于是没了。

[CF1327F] AND Segments

给定 n，k，m 与 m 条限制 $\{(l_i, r_i, x_i) | i \in [1, m]\}$，求满足如下条件的长度为 n 的 a 数组个数对 998244353 取模的结果：

- 对于任意 $1 \le i \le n$ 有 $0 \le a_i < 2^k$；
- 对于任意 $1 \le i \le m$ 有 $\text{AND}_{l_i \le j \le r_i} a_j = x_i$。

其中 AND 表示二进制按位与运算。

$1 \le n \le 5 \times 10^5$，$1 \le k \le 30$，$0 \le m \le 5 \times 10^5$，$0 \le x_i < 2^k$

[CF1327F] AND Segments

看到位运算，想到拆位计算答案最后乘起来。
如果一段按位与是 1，那么这一段必须全都是 1；
如果一段按位与是 0，那么这一段必须有一个 0。
先把必须是 1 的位置都填上，再考虑 0。
可以 $O(n)$ 预处理出数组 g_i，表示最小的 j 使得 $[i,j]$ 这一段位置可以都填 1。
然后设 f_i 表示前 i 个位置且第 i 个位置填 0 的方案数，那么有：

$$f_i = \begin{cases} 0 & \text{if } a_i \text{ must be } 1 \\ \sum\limits_{j=g_{i-1}-1}^{i-1} f_j & \text{otherwise} \end{cases}$$

最终的答案就是 $\sum\limits_{i=g_n-1}^{n} f_i$。前缀和优化即可 $O(n)$。

[P3214] 卡农

对于 $S=\{1,2,\cdots,n\}$，试求

$$\sum_{T\subseteq P(S)-\{\varnothing\}}[|T|=m]\left[\forall i\in[1,n],\left(\sum_{H\in T}[i\in H]\right)\bmod 2=0\right]$$

其中 $P(S)$ 表示 S 的幂集，即 $\{T|T\subseteq S\}$。对 10^8+7 取模。
$1\leq n,m\leq 10^6$

[P3214] 卡农

容易发现可以求出当 T 是有序集时的答案，最后除以 $m!$ 即可。考虑一个集合一个集合地往 T 中加。于是设 f_i 表示当 $m = i$ 时的答案。

如果只考虑偶数限制的话，假设确定了前 $i-1$ 个（非空的）集合，那么第 i 个集合也就确定了，方案数为 $\binom{2^n-1}{i-1}(i-1)!$。

但是此时第 i 个集合可能为 $\varnothing$。容易发现此时前 $i-1$ 个集合凑成一个 $i-1$ 的合法方案，故这部分的方案数为 f_{i-1}。

然后去掉有重复的。假设第 i 个集合与第 j 个集合（$j < i$）重复，那么去掉 i 和 j 后会形成一个 $i-2$ 的合法方案。此时 j 有 $i-1$ 中选法，第 i 个集合有 $2^n - 1 - (i-2) = 2^n - i + 1$ 种选法。故方案数为 $(i-1)(2^n - i + 1)f_{i-2}$。

得出转移方程：

$$f_i = \binom{2^n-1}{i-1}(i-1)! - f_{i-1} - (i-1)(2^n - i + 1)f_{i-2}$$

[P3600] 随机数生成器

给定 n，x，q 和 q 个区间 $\{(l_i, r_i) | i \in [1, q]\}$。
$a_1, a_2, \cdots, a_n$ 在 $[1, x]$ 内均匀随机，求 $\max\limits_{1 \le i \le q} \min\limits_{l_i \le j \le r_i} a_j$ 的期望。
对 666623333 取模。
$1 \le n, x, q \le 2000$

[P3600] 随机数生成器

值域很小，考虑差分。即
$\sum_i i \times P(X = i) = \sum_i i \times (P(X \le i) - P(X < i))$。
枚举一个 t，将小于等于 t 的数字赋 0，反之赋 1，那么题目变成：对于 $1 \le i \le n$，a_i 有 $\frac{t}{x}$ 的概率为 0，有 $1 - \frac{t}{x}$ 的概率为 1，求给定的 q 个区间都包含 0 的概率。
显然我们可以枚举 0 的个数，那么答案就是
$\sum_{i=0}^{n} \left(\frac{t}{x}\right)^i \left(1 - \frac{t}{x}\right)^{n-i} \frac{f_i}{\binom{n}{i}}$，其中 f_i 表示放 i 个 0 满足给定的 q 个区间都包含 0 的方案数。

[P3600] 随机数生成器

发现 f 数组与 t 无关，可以预处理。
具体地，设 g_i 为最小的 j 使得 $[i,j]$ 这一段位置可以都填 1，$h_{i,j}$ 为前 i 个位置放了 j 个 0 且第 i 个必须放 0 的方案数，那么：

$$h_{i,j} = \sum_{k=g_{i-1}-1}^{i-1} h_{k,j-1}$$

g 数组 $O(n)$ 预处理，h 数组前缀和即可 $O(n^2)$。

[P3239] 亚瑟王

有 n 张牌，第 i 张有发动概率 p_i，伤害 d_i。
游戏共 r 轮，每轮依次考虑每张未发动的牌，并以其发动概率发动它；如果发动了某张牌，则本轮立即停止，进入下一轮。
求 r 轮的总伤害期望。
共 T 组数据，$1 \leq T \leq 444$，$1 \leq n \leq 220$，$0 \leq r \leq 132$，$0 < p_i < 1$，$0 \leq d_i \leq 1000$

[P3239] 亚瑟王

由于期望线性性，可以计算每张牌的期望然后加起来。

设 f_i 为 r 轮中第 i 张牌的发动概率，那么答案就是 $\sum_{i=1}^{n} f_i d_i$。

考虑设 $g_{i,j}$ 表示 r 轮中前 i 张牌发动 j 张的概率，那么 $f_i = \sum_{j=0}^{i-1} g_{i-1,j}(1-(1-p_i)^{r-j})$。

然后问题转移到了 g 的求解上。可以讨论第 i 张牌发不发动来转移：

若第 i 张牌发动，则 $g_{i,j} \leftarrow g_{i-1,j-1}(1-(1-p_i)^{r-j+1})$；

若第 i 张牌不发动，则 $g_{i,j} \leftarrow g_{i-1,j}(1-p_i)^{r-j}$。

综上，$g_{i,j} = g_{i-1,j-1}(1-(1-p_i)^{r-j+1}) + g_{i-1,j}(1-p_i)^{r-j}$。

轮廓线 DP

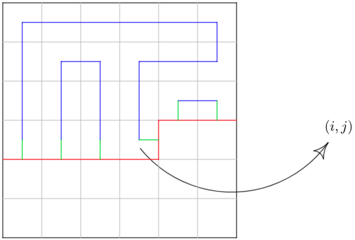
轮廓线 DP 本质上是状压维护轮廓线上插头状态的 DP。
让我们结合一下例题来理解。

[P5056]【模板】插头 DP

给出 $n \times m$ 的方格，有些格子有障碍物不能铺线，其它格子必须铺，形成一个闭合回路。求总方案数。
$2 \leq n, m \leq 12$

[P5056]【模板】插头 DP

设 $dp_{i,j,s}$ 表示当前 DP 到了第 i 行第 j 列，轮廓线状态为 s 的方案数。



如上图所示，红线即为轮廓线，绿线即为插头，蓝线即为一种可能的方案。

[P5056]【模板】插头 DP

由于我们要确保只形成一个闭合回路，所以我们要用三进制括号序列存储状态。
即，把插头再分为左插头和右插头，比如上图对应的状态为 $(1120212)_3$。
转移的话，如果当前格子有障碍物就不能铺线，否则分类讨论即可。

[P5074] Eat the Trees

给出 $n \times m$ 的方格，有些格子有障碍物不能铺线，其它格子必须铺，可以形成多个闭合回路。求总方案数。
$2 \leq n, m \leq 12$

[P5074] Eat the Trees

由于可以多个闭合回路，所以连三进制都不用，二进制状压表示有无插头即可。
~~好像比模板题还简单?~~

[UVA11270] Tiling Dominoes

给出 $n \times m$ 的方格，求用 1×2 的多米诺骨牌完全平铺的方案数。
若干组数据，$1 \leq n \times m \leq 100$

[UVA11270] Tiling Dominoes

发现 $\min(n, m) \leq 10$，考虑轮廓线 DP。
具体状态和转移方程手推不难。

[P2595] 多米诺骨牌

有一个 $n \times m$ 的矩形表格，其中有一些位置有障碍物。
现在要在这个表格内放一些 1×2 的多米诺骨牌，使得任何相邻两行之间都有至少一个骨牌横跨，任何相邻两列之间也都至少有一个骨牌横跨。
求有多少种不同的放置方法，注意你并不需要放满所有没有障碍的格子。对 19901013 取模。
$1 \le n, m \le 15$

[P2595] 多米诺骨牌

首先轮廓线 DP 预处理出数组 $f_{a,b,c,d}$ 表示左上角为 (a,b)，右下角为 (c,d) 的子矩阵没有行列限制的方案数。
这一部分可以在 $O(n^2m^32^m)$ 的复杂度内完成，常数小一点是可以过的去的。
然后可以 $O(2^m)$ 把行限制容斥掉，列限制用 DP：
设 $g_{i,j}$ 表示第 i 列至第 j 列没有列限制的方案数，这一部分可以用若干个子矩阵的答案乘起来。
再设 h_i 表示前 i 列满足列限制的方案数，那么可以枚举第一次不满足的位置：

$$h_i = g_{1,i} - \sum_{j=1}^{i-1} h_j \times g_{j+1,i}$$

最终时间复杂度 $O(n^2m^32^m)$，瓶颈在轮廓线 DP。

斜率优化 DP

斜率优化 DP 是一种非常重要的优化手段。

[P4360] 锯木厂选址

从山顶上到山底下沿着一条直线有 n 棵树要砍，树被砍倒后要运送到锯木厂。
山脚下有一个锯木厂。另外两个锯木厂将新修建在山路上。
每棵树有重量 w_i，第 i 棵树与第 $i+1$ 棵树距离为 d_i。
特别地，当 $i=n$ 时，d_i 为第 i 棵树到山脚的距离。
假定运输每公斤木材每米需要一分钱。
试拟定两个锯木厂的位置，使得花费最小，输出花费。
$2 \le n \le 2 \times 10^4$，$1 \le w_i \le 10^4$，$0 \le d_i \le 10^4$

[P4360] 锯木厂选址

板子题，略。

[P3628] 特别行动队

给你一个长为 n 的序列 x 以及三个参数 a、b、c。求把这个序列分成若干段的最大价值和。

一段序列 $x_l, x_{l+1}, \cdots, x_r$ 的价值为 $a(\sum_{i=l}^{r} x_i)^2 + b(\sum_{i=l}^{r} x_i) + c$。

$1 \le n \le 10^6$，$1 \le x_i \le 100$，$-5 \le a \le -1$，$|b|, |c| \le 10^7$

[P3628] 特别行动队

很套路的斜率优化题。
设 f_i 表示前 i 个元素分段的最大价值和，则：

$$f_i = \max_{0 \le j < i} f_j + a(s_i - s_j)^2 + b(s_i - s_j) + c$$

其中 s 数组是 x 的前缀和。
然后根据套路，把它变成一次函数的形式：

$$as_j^2 + f_j - bs_j = 2as_is_j - as_i^2 - bs_i - c + f_i$$

把 $as_j^2 + f_j - bs_j$ 看作 y 坐标，$2as_j$ 看作 x 坐标，s_i 看作斜率，$f_i - as_i^2 - bs_i - c$ 看作截距。
那么转移的过程可以看作是一条斜率为 s_i 的直线从 y 坐标负半轴无穷远的地方向上移动，第一个碰到的点就是最优转移。
单调队列维护凸包即可做到 $O(n)$。

[P1721] 国王饮水记

给你一个长为 n 的数组 a，你最多可以使用 k 次操作，每次操作选定一些位置，并将这些位置上的数赋值为它们的平均值。
求 k 次操作后 a_1 的最大值。
$1 \leq n \leq 8000$，$1 \leq k \leq 10^9$，$1 \leq a_i \leq 10^5$

[P1721] 国王饮水记

首先，低于 a_1 的数是完全没用的。所以，把这些数删掉。
其次，一个数最多只用一次。
然后，可以发现最优策略是从小往大合并。当 $k \geq n$ 时我们可以每个数单独合并，否则就只能一段一段合并了。所以下文默认 k 与 n 同阶。
设 $f_{i,j}$ 表示排序后合并了前 i 个共 j 次的 a_1 最大值。那么：

$$f_{i,j} = \max_{0 \leq p < i} \frac{s_i - s_p + f_{p,j-1}}{i - p + 1}$$

其中 s 是 a 的前缀和数组。发现这个式子是点 (i, s_i) 到 $(p-1, s_p - f_{p,j-1})$ 的斜率，而前者差分递增，所以单调队列维护后者即可 $O(nk)$。

[P6302] 回家路线

铁路系统中有 n 个站点，编号 $1 \sim n$。现在是 0 时刻，你在 1 号站点，要到 n 号站点。共 m 班列车，对于 i 号列车，它将在时刻 p_i 从站点 x_i 出发，在时刻 q_i 直达站点 y_i。你只能在时刻 p_i 上车，也只能在时刻 q_i 下车。

你可以通过多次换乘到达 n 号站点。一次换乘是指对于两班列车，假设分别为 u 号与 v 号列车，若 $y_u = x_v$ 并且 $q_u \le p_v$，那么你可以乘坐完 u 号列车后在 y_u 号站点等待 $p_v - q_u$ 个时刻，并在时刻 p_v 乘坐 v 号列车。

[P6302] 回家路线

你对于一种乘车方案有烦躁值。

- 你在站点等待时将增加烦躁值，对于一次 $t(t \geq 0)$ 个时刻的等待，烦躁值将增加 $At^2 + Bt + C$，其中 A, B, C 是给定的常数。注意：你登上第一班列车前，即从 0 时刻起停留在 1 号站点的那些时刻也算作一次等待。
- 若你最终在时刻 z 到达 n 号站点，则烦躁值将再增加 z。

求你到达 n 号站点的最小烦躁值。

$2 \leq n \leq 10^5$，$1 \leq m \leq 10^6$，$0 \leq A \leq 10$，$0 \leq B, C \leq 10^7$，$1 \leq x_i, y_i \leq n$，$x_i \neq y_i$，$0 \leq p_i < q_i \leq 4 \times 10^4$

[P6302] 回家路线

先不考虑地点的限制，设 f_i 表示第 i 个时刻的最小烦躁值。
那么 $f_i = \min\limits_{1\le j\le m, q_j\le i} f_{p_j} + A(i-q_j)^2 + B(i-q_j) + C$。
这个式子可以用斜率优化转移。
接下来考虑地点的限制。其实也很好考虑：每个城市开一个单调队列，每趟列车拆成两个事件：
在 p_i 时刻用 x_i 的单调队列计算答案，并在 q_i 的时刻插入 y_i 的单调队列里。
把事件以时间排序，就可以做到 $O(n+m\log m)$。最后答案随便统计一下就行了。

WQS 二分

有一类 DP，其显著特征是「恰好选 m 个」，如果普通 DP 的话必然要设一维状态表示选了多少个。
如果最终的答案是一个关于 m 的凸壳的话，我们可以通过 WQS 二分把它碾成一个 log。

[P1484] 种树

给定 n 个整数 $a_1, a_2, \cdots, a_n$，选至多 m 个求和，要求不能选相邻的两个整数。求最大总和。

$1 \le n \le 5 \times 10^5$，$1 \le m \le \dfrac{n}{2}$，$|a_i| \le 10^6$

[P1484] 种树

设 $f_{i,0/1}$ 表示前 i 个数，第 i 个选或不选的最大值，则

$$f_{i,0} = \max(f_{i-1,0}, f_{i-1,1})$$

$$f_{i,1} = a_i + cost + f_{i-1,0}$$

先跑一遍如果选的个数不超过 m 就直接输出，否则 WQS 二分。

[P4983] 忘情

给定长度为 n 的序列 a 和 m，求把 a 分成 m 段的最小代价。
一段的代价为这段里元素的总和加一的平方。
$1 \leq m \leq n \leq 10^5$，$1 \leq a_i \leq 1000$

[P4983] 忘情

分成 m 段，考虑 WQS 二分。
如果没有 m 段限制，那么可以设 f_i 表示前 i 个的最小代价。
那么 $f_i=\min_{j=0}^{i-1} f_j+(s_i-s_j+1)^2+cost$，其中 s 数组是 a 的前缀和。
斜率优化就可以单次 DP $O(n)$，再套上 WQS 二分就 $O(n\log n)$。

[P2619] Tree I

给定无向带权连通图 (V, E)，每条边是黑色或白色。让你求一棵最小权的恰好有 k 条白色边的生成树。
$|V| \leq 5 \times 10^4$，$|E| \leq 10^5$

[P2619] Tree I

WQS 二分，甚至连 DP 都不用。
给每条白边附加一个代价 $cost$，在最小生成树外面套个 WQS 二分就行了。

[P4383] 林克卡特树

给你一棵 n 个点的树，边有边权，求在其中选出 k 条不交路径的最大边权和，路径可以为空（即单个节点）。
$1 \le k \le n \le 3 \times 10^5$

[P4383] 林克卡特树

很容易想到的是树形 DP。
设 $f_{i,j,0/1/2}$ 表示以 i 节点的子树，子树内部有 j 条路径，其中 i 节点没在路径上/是路径的端点/在路径中间。
转移方程的话考虑如果 i 多了一棵 j 的子树，那么：
如果 (i,j) 不选的话，
$f_{i,a+b,0} \leftarrow f_{i,a,0} + \max(f_{j,b,0}, f_{j,b,1}, f_{j,b,2})$；
$f_{i,a+b,1} \leftarrow f_{i,a,1} + \max(f_{j,b,0}, f_{j,b,1}, f_{j,b,2})$；
$f_{i,a+b,2} \leftarrow f_{i,a,2} + \max(f_{j,b,0}, f_{j,b,1}, f_{j,b,2})$。
如果 (i,j) 选的话，
$f_{i,a+b,1} \leftarrow f_{i,a,0} + \max(f_{j,b,0}, f_{j,b,1}) + val(i,j)$；
$f_{i,a+b,2} \leftarrow f_{i,a,1} + \max(f_{j,b,0}, f_{j,b,1}) + val(i,j)$。
再套个 WQS 二分把第二维碾掉即可。

数据结构优化 DP

有一些 DP，可以用数据结构来优化转移。

[P7302] 免费的馅饼

有一个长度为 h 的地图，有 n 个普通的宝箱，第 i 个宝箱会在第 t_i 时刻出现在坐标 x_i，其中有 w_i 个金币。
如果你在 t_i 时刻在坐标 x_i 上，那么你就可以获得这个宝箱里的金币。
在 0 时刻，你可以传送至地图的任意位置。
你的移动速度为 2，求你能获得的最多金币个数。
$1 \le h \le 10^8$，$1 \le n \le 10^5$，$1 \le t_i \le 10^8$，$1 \le w_i \le 1000$，$1 \le x_i \le h$

[P7302] 免费的馅饼

将宝箱按出现时间排序。设 f_i 表示获取 i 号宝箱的前提下前 i 个宝箱获取的最大金币数，那么有

$$f_i = \max_{|x_i - x_j| \le 2(t_i - t_j)} f_j + w_i$$

把限制拆开发现是个二位偏序，线段树优化即可。

[ARC073F] Many Moves

在一行中有 N 个格子，从左往右编号为 1 到 N。
有 2 颗棋子，一开始分别位于位置 A 和 B 。按顺序给出 Q 个要求，每个要求给出一个位置 x_i，要求将两个棋子中任意一个移动到位置 x_i。
将一颗棋子移动一格需要花费 1 秒。
为了回答要求，你只能移动棋子，并且同一时刻只能移动一颗棋子。要求的顺序是不可更改的。在同一时间允许两颗棋子在同一个格子内。
求最小需要多少秒回答全部要求。
$1 \leq N, Q \leq 2 \times 10^5$，$1 \leq A, B \leq N$，$1 \leq x_i \leq N$

[ARC073F] Many Moves

设 $f_{i,j,k}$ 表示处理了前 i 个请求，两颗棋子分别在 j 和 k 的最小时间。
发现处理了前 i 个请求后，一定会有一颗棋子在 x_i 上。
于是可以少一维，设 $f_{i,j}$ 为处理了前 i 个请求，另外一颗棋子在 j 的最小时间。
有 $f_{i,j} \leftarrow f_{i-1,j} + |x_i - x_{i-1}|$，$f_{i,x_{i-1}} \leftarrow \min_k f_{i-1,k} + |k - a_i|$。
发现从 f_{i-1} 到 f_i 就是一个全局加和一个单点取 min。线段树优化即可。

[CF833B] The Bakery

将一个长度为 n 的序列 a 分为 k 段，使得总价值最大。
一段区间的价值表示为区间内不同数字的个数。
$1 \le n \le 35000$，$1 \le k \le 50$

[CF833B] The Bakery

设 $f_{i,j}$ 表示分 i 段，前 j 个数的最大价值。
那么 $f_{i,j} = \max_{p=0}^{j-1} f_{i-1,p} + val(p+1, j)$。
问题变成了如何快速地求 $val(i, j)$。
一个简单易行的方式是求出 pre_i，即 a_i 上一次出现的位置。
那么扫到一个 i 就把 $(pre_i, i]$ 在线段树上区间加 1，这样求 $val(p, i)$ 就只需要线段树上单点查询 p 了。
当然差分一下用树状数组搞也是可以的。

DP 套 DP

DP 套 DP 是很有用的 DP 技巧，核心思想是把小 DP 的状态与答案看做大 *DP* 的状态，一般需要暴搜 DP 自动机。

[P4590] 游园会

给定 n，k 以及一个长度为 k 的由 `N`、`O`、`I` 构成的字符串 s。试对于 $0 \le i \le k$ 求出在所有长度为 n 的字符串中有多少个满足：

- 该字符串不包含子串 `NOI`；
- 该字符串与 s 的最长公共子序列长度为 i。

$1 \le n \le 1000$，$1 \le k \le 15$

[P4590] 游园会

发现单纯把 LCS 长度记到状态里不好转移，考虑把整个 LCS DP 数组都压进状态里。

考虑 LCS DP 数组：$f_{i,j} = \max(f_{i,j-1}, f_{i-1,j}, f_{i-1,j-1} + [s_i = t_j])$

由于我们的 t_j 每次只增加一个字符，于是只需要记录关于 i 的那一维就好。

具体的，设 f_i 为当前字符串和 s 的前 i 位的 LCS 长度。假设增加一位字符后的 DP 数组为 g，那么：

$$g_i = \max(g_{i-1}, f_i, f_{i-1} + [s_i == ch])$$

其中 ch 是增加的那个字符。

[P4590] 游园会

然后我们又可以发现 f_i 至多比 f_{i-1} 增加 1。那么它的差分数组就是一个 01 数组，可以 2^{15} 二进制状压。
于是我们设 $g_{i,j,f}$ 表示长度为 i 的字符串，其末尾包含 `NOI` 的前 j 位，其与 s 的 DP 数组差分为 f 的方案数。
转移方程枚举一下新加的字符就好了。

[P8497] 移除石子

有 n 堆石子排成一行，第 i 堆有 a_i 枚，你的任务是通过如下的操作将所有石子移除：

- 操作一：选择一堆石子，将其中的至少 2 枚石子移除；
- 操作二：选择一个连续的编号区间 $[l, r]$（$1 \leq l \leq r \leq n$）并满足 $r - l \geq 2$，将其中的每一堆石子都恰好移除 1 枚。

你可以采用任意顺序执行任意多次上述两种操作，直到无法再执行操作为止。若最后你能将所有石子全部移除则胜利。

[P8497] 移除石子

为了使这个游戏更有意思，你在开始操作前可以且必须将 m 个石子放入某一堆或某几堆石子中，然后开始操作。
现在，你可以自由选择初始局面。具体的，对于每个 a_i 你可以自由选择 $[l_i, r_i]$ 内的任意整数。
试计算有多少种初始局面使得你至少存在一种获胜方案，对 10^9+7 取模。
注意，这里的初始局面指的是放 m 个石子之前的局面。
$3 \le n \le 1000$，$0 \le m \le 100$，$0 \le l_i \le r_i \le 10^9$

[P8497] 移除石子

考虑确定了一个局面，如何判定它是否可以获胜。
容易发现，**在大多数情况下**，存在一个最小的下界 m_0，使得当 $m \geq m_0$ 时，一定有获胜方案。
唯二的特殊局面为：$\forall 1 \leq i \leq n, a_i = 0$ 与 $n = 3, a_1 = a_2 = a_3 = 1$，此时当 $m = 0$ 时有获胜方案，但当 $m = 1$ 时没有。
于是设 $f_{i,j,k}$ 表示将 $1 \sim i$ 堆石子移成 0，且左端点落在 $1 \sim i-1$、右端点至少为 $i+1$ 的操作二数量为 j，左端点为 i 的操作二数量为 k 的 m 的下界。
那么有：

$$f_{i,j,k} = \min(m+1, \min_{0 \leq p \leq j} \min_{p+q \geq j} f_{i-1,q,p} + w(a_i - k - p - q))$$

其中，$w(x) = \begin{cases} -x & x \leq 0 \\ 1 & x = 1 \\ 0 & x > 1 \end{cases}$

[P8497] 移除石子

乍一看十分不可做，但是发现重复的操作二是没有意义的，所以后两维度应该不会很大。
不难分析得到后两维都可以不超过 2，这样我们对于每个 i 就只要记录 9 个数字了。
但是这样不同的状态数是 102^9 级别的，没关系，写个暴搜发现只有至多 8765 个有用状态。
然后没了。时间复杂度 $O(8765n)$。

动态 DP

DP 大多都是静态的，但有些题不得不要你的 DP 支持修改。
于是，动态 DP 应运而生。
它的思想是由于大多数 DP 方程都是线性的，所以可以用一个矩阵来描述 DP 的转移。
下面通过一个例子来说明矩阵的妙用。

[P7739] 密码箱

有一个序列 a，初始是 $[0,1]$，进行若干次操作：

- `W` 类型：给数列的最后一项加 1。
- `E` 类型：若数列的最后一项为 1，则给倒数第二项加 1；否则先给数列的最后一项减 1，接着在数列尾再加两项，两项的值都是 1。

定义一个序列的权值为：

$$f(a_1, a_2, \cdots, a_k) = \begin{cases} a_1 & k = 1 \\ f(a_1, \cdots, a_{k-2}, a_{k-1} + \frac{1}{a_k}) & k \geq 2 \end{cases}$$

[P7739] 密码箱

你需要维护一个初始长度为 n 的由 `W`、`E` 构成的操作序列。支持 q 次操作：

- `APPEND` ch，在现有操作序列后追加一次 ch 类型操作，其中 ch 为字符 `W` 或 `E`。
- `FLIP` l r，反转现有操作序列中第 l 个至第 r 个操作，即所有 `W` 变为 `E`，所有 `E` 变为 `W`。
- `REVERSE` l r，翻转现有操作序列中第 l 个至第 r 个操作，也就是将这个区间中的操作逆序。

你需要在操作前以及每次操作后求出初始序列 $[0,1]$ 在依次执行当前操作序列后的权值，输出分子和分母模 998244353 后的余数。

$1 \leq n, q \leq 10^5$

[P7739] 密码箱

容易发现，最后的答案就是连分数 $a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots + \cfrac{1}{a_n}}}$。

考虑连分数多嵌套一层后的变化：$\frac{a'}{b'} = x + \frac{1}{\frac{a}{b}} = \frac{b + xa}{a}$

发现 $\gcd(a', b') = gcd(b + xa, a) = gcd(a, b)$，而初始 $a = 1$，$b = 0$，所以不用担心约分的事。

我们可以写成矩阵的形式：$\begin{bmatrix} a' \\ b' \end{bmatrix} = \begin{bmatrix} x & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}$。

所以最后的答案就是 $\begin{bmatrix} a_1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a_2 & 1 \\ 1 & 0 \end{bmatrix} \cdots \begin{bmatrix} a_n & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$。

[P7739] 密码箱

再来考虑 `W` 操作和 `E` 操作对这个矩阵序列来说有什么影响。

先考虑 `W` 操作，可以构造出 $\begin{bmatrix} a_i & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ k & 1 \end{bmatrix} = \begin{bmatrix} a_i + k & 1 \\ 1 & 0 \end{bmatrix}$。故 `W` 操作就是在这个矩阵序列最后乘上 $\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$。

[P7739] 密码箱

再考虑 E 操作，如果最后一项不为 1，那么先把最后一项减 1，再新添两个 1，那么就等价于在矩阵序列后面乘上 $\begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ -1 & 0 \end{bmatrix}$。

如果最后一项为 1 的话那么给倒数第二项加 1，相当于在矩阵序列倒数第二项后插入 $\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$。

由于此时最后一项是 $\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$，这启发我们去发现：

$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 2 & 1 \\ -1 & 0 \end{bmatrix}$。

所以 E 操作根本不需要分类讨论，就简单地在最后面乘上 $\begin{bmatrix} 2 & 1 \\ -1 & 0 \end{bmatrix}$ 就好了。

[P7739] 密码箱

初始序列是 $\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}\begin{bmatrix}1 & 1\\1 & 0\end{bmatrix}=\begin{bmatrix}1 & 0\\1 & 1\end{bmatrix}$，所以令 $W=\begin{bmatrix}1 & 0\\1 & 1\end{bmatrix}$，$E=\begin{bmatrix}2 & 1\\-1 & 0\end{bmatrix}$，那么最终答案就是

$$\begin{bmatrix}1 & 0\\1 & 1\end{bmatrix}\underbrace{WEEWE\cdots EW}_{\text{操作序列}}\begin{bmatrix}1\\0\end{bmatrix}$$

平衡树维护即可。

[P4719]【模板】动态 DP

给定一棵 n 个点的树，点带点权。有 m 次操作，每次操作给定 x, y，表示修改点 x 的权值为 y。

你需要在每次操作之后求出这棵树的最大权独立集的权值大小。

$1 \leq n, m \leq 10^5$

[P4719]【模板】动态 DP

$f_i, 0/1$ 表示子树 i 内 i 选或不选的最大独立集。朴素的 DP 方程不难列出：
$$\begin{cases} f_{i,0} = \sum\limits_{j \in \text{son}(i)} \max(f_{j,0}, f_{j,1}) \\ f_{i,1} = a_i + \sum\limits_{j \in \text{son}(i)} f_{j,0} \end{cases}$$
要支持修改的话，可以重链剖分来做。
重链剖分的话，要分轻重儿子来讨论。于是设 $g_{i,0/1}$ 表示 i 的轻儿子都不选或可选可不选的最大独立集。那么：
$$\begin{cases} f_{i,0} = g_{i,1} + max(f_{\text{hson}(i),0}, f_{\text{hson}(i),1}) \\ f_{i,1} = a_i + g_{i,0} + f_{\text{hson}(i),0} \end{cases}$$

[P4719]【模板】动态 DP

我们可以写成矩阵 $(\max, +)$ 乘法的形式：

$$\begin{bmatrix} f_{i,0} \\ f_{i,1} \end{bmatrix} = \begin{bmatrix} g_{i,1} & g_{i,1} \\ a_i + g_{i,0} & -\infty \end{bmatrix} \begin{bmatrix} f_{\mathrm{hson}(i),0} \\ f_{\mathrm{hson}(i),1} \end{bmatrix}$$

那么我们就可以用线段树维护矩阵，每次修改 $O(\log n)$ 个矩阵就行了。

[P5024] 保卫王国

给你一棵 n 个节点的树，点有点权，m 次询问，每次要求你两个节点强制选或不选，求最小覆盖集。
$1 \le n, m \le 10^5$

[P5024] 保卫王国

最小覆盖集 = 全集 − 最大独立集

然后选节点可以将节点权值赋 $-\infty$，不选就赋 ∞

然后就是上一题了。

[CF1286D] LCC

数轴上有 n 个点 $x_1, x_2, \cdots, x_n$，第 i 个点有速度 v_i，有 p_i 的概率向右移动，$1 - p_i$ 的概率向左移动。
现在所有点开始移动。求最早的两个点碰在一起的时间的期望模 998244353。特别地，如果没有点会碰撞，这个时间为 0。
$1 \le n \le 10^5$

[CF1286D] LCC

首先，最先碰到的点一定是相邻的两个点。
然后，我们就可以枚举相邻的两个点 t 和 $t+1$，算出它们的相遇时间，强制它们相遇，然后算 $1 \sim t$ 和 $t+1 \sim n$ 的点相遇时间都不少于 t 和 $t+1$ 的相遇时间的概率就行了。
设 $f_{i,0/1}$ 表示符合要求的前 i 个点且第 i 个点向左或者右的概率。
转移可以用矩阵描述，于是把相邻两点的相遇时间从小到大排序，动态修改转移矩阵，可以用线段树维护。$O(n \log n)$

决策单调性优化 DP

决策单调性优化 DP 用于解决形如下列形式的 DP 转移方程：

$$f_i = \max_{j<i} f_j + w(i, j)$$

$$f_{i,j} = \max_{k<j} f_{i-1,k} + w(k, j)$$

其中，如果 w 对于任意 $a < b \le c < d$ 满足四边形不等式 $w(a, c) + w(b, d) \ge w(a, d) + w(b, c)$，那么此 DP 具有决策单调性。
当 DP 方程里是 min 亦然，不等号反向即可（即，交叉优于包含）。

P1912 诗人小 G

给你 n 句诗，行标准长度 L，参数 P，要你把 n 句诗分成若干行，每行的诗句间用一个空格隔开。
定义一行的不协调度为 $|S-L|^P$，其中 S 为这行的长度，包含空格。
整篇诗的不协调度为所有行的不协调度之和，试使之最小化。
T 组数据，$1 \le T \le 10$，$1 \le n \le 10^5$

P1912 诗人小 G

设 f_i 为前 i 句诗的最小不协调度，容易列出 DP 方程

$$f_i = \min_{j=0}^{i-1} f_j + |s_i - s_j + 1|^P$$

其中 s 是诗句长度加 1 的前缀和数组。

发现 $|s_i - s_j + 1|^P$ 满足四边形不等式，决策单调性优化即可。

[P5617] 不可视境界线

给你 n 个半径为 r 的圆，第 i 个圆在 $(x_i, 0)$，要在这 n 个圆中选出 m 个，求它们的面积并的最大值。
$1 \le m \le n \le 10^5$，$1 \le r \le 10^4$，$0 \le x_i \le 10^9$

[P5617] 不可视境界线

看到选 m 个，可以想到 WQS 二分，把原问题分成 $O(\log n)$ 个子问题，即选的数量不限，每选一个圆就会付出 $cost$ 的代价，求面积并最大值。
可以设 f_i 表示前 i 个圆的最大面积并，那么有
$f_i = \max\limits_{0 \le j < i} f_j + w(i,j) + cost$，其中 $w(i,j)$ 是第 i 个圆减去第 j 个圆的面积大小。
发现 $w(i,j)$ 满足四边形不等式，决策单调性优化即可。

[P5574] 任务分配问题

给定长为 n 的序列，分成 m 段，让每段的顺序对个数的总和最小化。
$1 \le n \le 25000$，$1 \le m \le 25$

[P5574] 任务分配问题

设 $f_{i,j}$ 表示前 i 个任务分成 j 段的最小答案，那么

$$f_{i,j} = \min_{k=0}^{i-1} f_{k,j-1} + w(k, i)$$

，其中 $w(i, j)$ 是 $(i, j]$ 的顺序对个数。
容易发现 $w(i, j)$ 满足四边形不等式，决策单调性优化即可。

[CF868F] Yet Another Minimization Problem

给定长为 n 的序列，分成 m 段，每段的费用是其中相同元素的对数。求费用之和的最小值。

$2 \le n \le 25000$，$1 \le m \le 20$

[CF868F] Yet Another Minimization Problem

与上一题类似，故不在此赘述了。

杂项

一些没有明显分类的题，就放一块了。

[P7914] 括号序列

定义超级括号序列是由字符（、）、* 组成的字符串。
对于一个常数 k，定义符合规范的超级括号序列如下：

- ()、(S) 均是符合规范的超级括号序列，其中 S 表示任意一个仅由不超过 k 个字符 * 组成的非空字符串（以下两条规则中的 S 均为此含义)；
- 如果字符串 A 和 B 均为符合规范的超级括号序列，那么字符串 AB、ASB 均为符合规范的超级括号序列，其中 AB 表示把字符串 A 和字符串 B 拼接在一起形成的字符串；
- 如果字符串 A 为符合规范的超级括号序列，那么字符串 (A)、(SA)、(AS) 均为符合规范的超级括号序列。

给定长度为 n 的超级括号序列，有若干个字符尚未确定（用 ? 表示），求有多少种将所有尚未确定的字符一一确定的方法，使得得到的字符串是一个符合规范的超级括号序列？对 $10^9 + 7$ 取模。
$1 \le k \le n \le 500$

[P7961] 数列

给定整数 n, m, k，和一个长度为 $m+1$ 的正整数数组 $v_0, v_1, \cdots, v_m$。
对于一个长度为 n，下标从 1 开始且每个元素均不超过 m 的非负整数序列 $\{a_i\}$，我们定义它的权值为 $v_{a_1} \times v_{a_2} \times \cdots \times v_{a_n}$。
当这样的序列 $\{a_i\}$ 满足 $2^{a_1} + 2^{a_2} + \cdots + 2^{a_n}$ 的二进制表示中 1 的个数不超过 k 时，我们认为 $\{a_i\}$ 是一个合法序列。
计算所有合法序列 $\{a_i\}$ 的权值和对 998244353 取模的结果。
$1 \leq k \leq n \leq 30$，$0 \leq m \leq 100$，$1 \leq v_i < 998244353$

[P7962] 方差

给定长度为 n 的非严格递增正整数数列 $\{a_i\}$，其中 $\forall i \in [1, n), a_i \leq a_{i+1}$。
每次可以进行的操作是：任意选择一个正整数 $i \in (1, n)$，将 a_i 变为 $a_{i-1} + a_{i+1} - a_i$。
求在若干次操作之后，该数列的方差最小值是多少。请输出最小值乘以 n^2 的结果。
$1 \leq n \leq 10^4$，$1 \leq a_i \leq 600$

[P9169] 过河卒

有一个 n 行 m 列的棋盘。我们用 (i, j) 表示第 i 行第 j 列的位置。棋盘上有一些障碍，还有一个黑棋子和两个红棋子。

游戏的规则是这样的：红方先走，黑方后走，双方轮流走棋。红方每次可以选择一个红棋子，向棋盘的相邻一格走一步。具体而言，假设红方选择的这个棋子位置在 (i, j)，那么它可以走到 $(i-1, j), (i+1, j), (i, j-1), (i, j+1)$ 中的一个，只要这个目的地在棋盘内且没有障碍且没有红方的另一个棋子。

黑方每次可以将自己的棋子向三个方向之一移动一格。具体地，假设这个黑棋子位置在 (i, j)，那么它可以走到 $(i-1, j), (i, j-1), (i, j+1)$ 这三个格子中的一个，只要这个目的地在棋盘内且没有障碍。

[P9169] 过河卒

在一方行动之前，如果发生以下情况之一，则立即结束游戏，按照如下的规则判断胜负（列在前面的优先）：

- 黑棋子位于第一行。此时黑方胜。
- 黑棋子和其中一个红棋子在同一个位置上。此时进行上一步移动的玩家胜。
- 当前玩家不能进行任何合法操作。此时对方胜。

[P9169] 过河卒

现在假设双方采用最优策略，不会进行不利于自己的移动，即:

- 若存在必胜策略，则会选择所有必胜策略中，不论对方如何操作，本方后续获胜所需步数最大值最少的操作。
- 若不存在必胜策略，但存在不论对方如何行动，自己都不会落败的策略，则会选择任意一种不败策略。
- 若不存在不败策略，则会选择在所有策略中，不论对方如何操作，对方后续获胜所需步数最小值最大的操作。

如果在 $BB(27)$ 个回合之后仍不能分出胜负，则认为游戏平局。
请求出游戏结束时双方一共移动了多少步，或者判断游戏平局。
$1 \leq n \leq 10$，$2 \leq m \leq 10$

[CF773F] Test Data Generation

给定 n，m，p，求满足下列条件的序列 a 对 p 取模的结果：

- 序列长度 k 不超过 n 且为奇数；
- 序列中的值 a_i 不超过 m，且单调递增；
- 设 $g = \gcd(a_1, a_2, \cdots, a_k)$ 那么序列的最后一项 a_k 是偶数，且 $\frac{a_k}{g}$ 是奇数。

$1 \le n \le 30000$，$1 \le m \le 10^9$，$10^4 \le p \le 10^5 + 129$

[CF755G] PolandBall and Many Other Balls

有一排 n 个球，定义一个组可以只包含一个球或者包含两个相邻的球。现在一个球只能分到一个组中，求从这些球中取出 k 组的方案数。

$1 \le n \le 10^9$，$1 \le k \le 2^{15}$

[P2569] 股票交易

你知道了未来 T 天内某只股票的走势，第 i 天的股票买入价为每股 AP_i，卖出价为每股 BP_i（数据保证对于每个 i，都有 $AP_i \geq BP_i$），但是每天不能无限制地交易，于是股票交易所规定第 i 天的一次买入至多只能购买 AS_i 股，一次卖出至多只能卖出 BS_i 股。同时，在两次交易（某一天的买入或者卖出均算是一次交易）之间，至少要间隔 W 天，且一个人的手里的股票数不能超过 P。
假设你初始有无限的钱，求 T 天之后你最多能赚多少钱。
$1 \leq T, P \leq 2000$

[CF1713F] Lost Array

给出一个从 1 到 n 的数组 a。有一个从 0 开始标号的大小为 $(n+1)\times(n+1)$ 的矩阵 b，通过以下方式生成：

- 对于 $0 \le i \le n$，$b_{i,0} = 0$。
- 对于 $1 \le i \le n$，$b_{0,i} = a_i$。
- 对于 $1 \le i, j \le n$，$b_{i,j} = b_{i,j-1} \oplus b_{i-1,j}$。

现在，给出 $b_{1,n}, b_{2,n}, \cdots, b_{n,n}$，试还原出一个满足条件的 a 数组，保证有解。

$1 \le n \le 5 \times 10^5$

[P5911] PRZ

一支 n 个人的队伍要尽快过桥，第 i 个人有过桥时间 t_i 与重量 v_i，而桥有载重 W，所以他们决定分组过桥，一组的重量为组内所有人的重量之和，过桥时间为组内最慢的人的过桥时间。求最小总时间。

$1 \le n \le 16$

[P4438] 道路

一个国家有 $n-1$ 个城市与 n 个乡村，1 号城市是首都。对于每一个城市，恰有一条公路和一条铁路通向这座城市，没有道路通向任何乡村。

现在你要翻修 n 条道路，假设从 i 号乡村走到首都一共需要经过 x 条未翻修的公路与 y 条未翻修的铁路，那么该乡村的不便利值为 $c_i \times (a_i + x) \times (b_i + y)$。试最小化总不便利值。

$1 \leq n \leq 20000$，任意乡村可以通过不超过 40 条道路到达首都

[P6280] Exercise G

给定 n 和质数 m，求所有 k 的和对 m 取模的结果，使得存在一个长为 n 的排列使其最小循环节为 k。
$1 \le n \le 10^4$

[CF150D] Mission Impassable

给定一个字符串 s，你可以删掉长度为 i 的回文子串，获得 v_i 的积分，$v_i = -1$ 代表不能删除长度为 i 的串，删完之后前后拼接起来，求最多能获得多少积分。
$1 \leq |s| \leq 150$

[AGC020F] Arcs on a Circle

你有一个长度为 C 的圆与 N 个弧。弧 i 有长度 l_i。
现在这些弧随机地放在圆上，求圆被完全覆盖的概率。
$1 \le N \le 6$，$1 \le C \le 50$，C、l_i 为整数

[P3643] 划艇

有 n 个变量，第 i 个变量有范围 l_i 至 r_i。你可以在其中依次选出若干个变量，不能不选。你要为它们在各自的范围内赋值，使得赋的值单调递增。求方案数模 10^9+7 的结果。

$1 \leq n \leq 500$

[CF1761F1] Anti-median (Easy Version)

一个排列是好的，当且仅当对于所有正整数 m 都满足所有长度为 $2m+1$ 的子串的中位数不在第 $m+1$ 个。给定一个一些数被替换成 -1 的排列，你需要将 -1 填入所有可能的值后，统计好的排列数量。对 10^9+7 取模。
多测，$\sum n^2 \leq 10^6$

[P4164] 挖宝藏

有 n 处宝藏，第 i 个宝藏有价值 p_i，位置是 (x_i, y_i)。
如果网格 (x, y) 满足下面两个条件之一，则它是可挖掘的：

- $y = -1$。
- $(x-1, y+1), (x, y+1), (x+1, y+1)$ 这三个方格都已经被挖掘了。

挖掘一个方格的代价为 1。求最大价值。$1 \le n \le 10^3$

[CF713C] Sonya and Problem Wihtout a Legend

给定一个有 n 个正整数的数组，一次操作中，可以把任意一个元素加一或减一（元素可被减至负数或 0）。求使得原序列严格递增的求最小操作次数。

$1 \leq n \leq 3000$

[ARC104E] Random LIS

给出一个长度为 n 的序列 a，按照下列方式随机生成一个长度为 n 的序列 x：对于任意 $i \in [1, n]$，x_i 在 $[1, a_i]$ 中的整数均匀随机。求其最长上升子序列长度的期望，对 $10^9 + 7$ 取模。

$1 \leq n \leq 6$

[P4597] 序列 sequence

给定一个有 n 个正整数的数组，一次操作中，可以把任意一个元素加一或减一（元素可被减至负数或 0）。求使得原序列严格递增的求最小操作次数。

$1 \leq n \leq 5 \times 10^5$

最后的话

谢谢大家。