Dynamic Programming 2

Thaddeus

安徽师范大学附属中学

2017 年 9 月 20 日

DP

Dp 的本质就是将问题的状态压缩，将性质相同的状态放在一起考虑对于答案的贡献。Dp 的考察点，往往在于状态的设计和转移的优化。

提高组内的 dp 更多注重状态的设计及转移的优化。讲课内容不同于讲义，但是严重依赖讲义基础。

DP

提高组中，dp 主要考察的方向是 dp 状态的设置以及转移的优化。

因为提高组是普及组的进阶，所以默认大家会一些基础的 dp。提高组方向的 dp 主要以杂题选讲的形式呈现。

AGC2013 A Sorted Arrays

给定一个数组，要将其划分成若干连续子序列。使得每一段内都单调不升或者单调不降。

数据范围 $n \leq 10^5$

AGC2013 A Sorted Arrays

给定一个数组，要将其划分成若干连续子序列。使得每一段内都单调不升或者单调不降。

数据范围 $n \leq 10^5$

$dp1[i], dp2[i]$ 分别表示以 i 结尾的单调不上升和单调不下降子序列前最少分成多少段。

时间复杂度 $O(n)$

51nod1241 特殊的排序

要将一个长度为 n 序列排序，每次只能将一个元素放到序列的开头和结尾。求最少多少次能将序列排成有序。

数据范围 $n \leq 50000$

51nod1241 特殊的排序

要将一个长度为 n 序列排序，每次只能将一个元素放到序列的开头和结尾。求最少多少次能将序列排成有序。

数据范围 $n \leq 50000$

观察最少次数恰好为 $n-$ 序列的 lis

时间复杂度 $O(n\log n)$

COCI2015 Stanovi

不断平行或垂直切割 $n*m$ 矩形，要求切出的矩形必须都至少有一边是原矩形的边界。

给定 K，最后的代价是 $\sum(x*y-K)^2$，x,y 是切割完矩形的大小。求最小代价。

COCI2015 Stanovi

不断平行或垂直切割 $n * m$ 矩形，要求切出的矩形必须都至少有一边是原矩形的边界。

给定 K，最后的代价是 $\sum (x * y - K)^2$，x, y 是切割完矩形的大小。求最小代价。

数据范围 $n, m \leq 300$

COCI2015 Stanovi

不断平行或垂直切割 $n*m$ 矩形，要求切出的矩形必须都至少有一边是原矩形的边界。

给定 K，最后的代价是 $\sum(x*y-K)^2$，x, y 是切割完矩形的大小。求最小代价。

数据范围 $n, m \leq 300$

$dp[i][j][S]$ 表示 $i*j$ 的矩形，边界情况为 S 的最小代价。

暴力枚举，注意常数优化。

CF392E Deleting Substrings

给定一个序列，每次可以删除一段连续的序列。$w_l...w_r$，需要满足

- $|w_i - w_{i-1}| \le 1$. $i \in [l, r)$
- $2 * w_i \ge w_{i-1} + w_{i+1}$. $i \in (l, r)$

删除长度为 d 的序列，会有 w_d 的收益，可以任意时刻停止删除序列。

时间复杂度 $n \le 400$

CF392E Deleting Substrings

给定一个序列，每次可以删除一段连续的序列。$w_l ... w_r$，需要满足

- $|w_i - w_{i-1}| \leq 1.\ i \in [l, r)$
- $2 * w_i \geq w_{i-1} + w_{i+1}.\ i \in (l, r)$

删除长度为 d 的序列，会有 w_d 的收益，可以任意时刻停止删除序列。

时间复杂度 $n \leq 400$

区间 dp，转移时考虑枚举最后一次删除的序列，判断序列是否合法只需要记录最后两个被删除的元素。

时间复杂度 $O(n^3)$

SHOI2017 组合数问题

求 $\sum_{i=0}^{\infty} \binom{nk}{ik+r} \ mod \ p$

$1 \leq n \leq 10^9, 0 \leq r < k \leq 50, 2 \leq p \leq 2^{30}-1$

SHOI2017 组合数问题

求 $\sum_{i=0}^{\infty} \binom{nk}{ik+r} \ mod \ p$

$1 \le n \le 10^9, 0 \le r < k \le 50, 2 \le p \le 2^{30}-1$

考虑组合数的本质含义，就是求 nk 个球中选取 x 个球的方案数，其中 $x \ mod \ k = r$

SHOI2017 组合数问题

求 $\sum_{i=0}^{\infty} \binom{nk}{ik+r} \ mod \ p$

$1 \le n \le 10^9, 0 \le r < k \le 50, 2 \le p \le 2^{30}-1$

考虑组合数的本质含义，就是求 nk 个球中选取 x 个球的方案数，其中 $x \ mod \ k = r$

$dp[i][j]$ 表示 i 个球选取了 $mod \ k = j$ 个球的方案数。

SHOI2017 组合数问题

求 $\sum_{i=0}^{\infty}\binom{nk}{ik+r} \ mod \ p$

$1 \le n \le 10^9, 0 \le r < k \le 50, 2 \le p \le 2^{30}-1$

考虑组合数的本质含义，就是求 nk 个球中选取 x 个球的方案数，其中 $x \ mod \ k = r$

$dp[i][j]$ 表示 i 个球选取了 $mod \ k = j$ 个球的方案数。

每个球都是等价的，可以使用矩阵快速幂优化

时间复杂度 $O(\log n * k^2)$

SDOI2016 征途

给你一个长为 n 的序列，要求将这个序列分成 m 段，使得每段内数字之和构成的方差最小。输出这个最小方差与 m^2 的乘积。

数据范围 $m \leq n \leq 3000$

SDOI2016 征途

给你一个长为 n 的序列，要求将这个序列分成 m 段，使得每段内数字之和构成的方差最小。输出这个最小方差与 m^2 的乘积。

数据范围 $m \leq n \leq 3000$

考虑答案就是 $m * \sum_{i=1}^{m} s_i^2 - sum^2$

SDOI2016 征途

给你一个长为 n 的序列，要求将这个序列分成 m 段，使得每段内数字之和构成的方差最小。输出这个最小方差与 m^2 的乘积。

数据范围 $m \leq n \leq 3000$

考虑答案就是 $m * \sum_{i=1}^{m} s_i^2 - sum^2$

$dp[i][j]$ 前 i 个数分成了 j 段的最小 $\sum s_i^2$

SDOI2016 征途

给你一个长为 n 的序列，要求将这个序列分成 m 段，使得每段内数字之和构成的方差最小。输出这个最小方差与 m^2 的乘积。

数据范围 $m \le n \le 3000$

考虑答案就是 $m * \sum_{i=1}^{m} s_i^2 - sum^2$

$dp[i][j]$ 前 i 个数分成了 j 段的最小 $\sum s_i^2$

$dp[i][j] = min\{dp[k][j-1] + (S_i - S_k)^2\}$

可以使用斜率优化。时间复杂度 $O(n^2)$

gym 100886A

给定 n 个数 x_i，将每个数放入 $\{A\}, \{B\}, \{C\}$ 其中的一个。求

$$minimize\{max\{sumA, sumB, sumC\} - min\{sumA, sumB, sumC\}\}$$

数据范围 $n \leq 400, 0 \leq x_i \leq 30$

gym 100886A

给定 n 个数 x_i，将每个数放入 $\{A\},\{B\},\{C\}$ 其中的一个。求 $minimize\{max\{sumA, sumB, sumC\} - min\{sumA, sumB, sumC\}\}$

数据范围 $n \leq 400, 0 \leq x_i \leq 30$

直接背包 $dp[i][j][k]$ 前 i 个物品，$sumA = j, sumB = k$ 是否可行。复杂度 $O(n^3 * max\{x_i\}^2)$ 的，显然无法通过所有测试数据。

gym 100886A

发现 $x_i \le 30$，这样不同的物品只有 30 种，问题变成了多重背包。对同一种物品进行转移的时候，记 $g[j][k]$ 最多往 $\{A\},\{B\}$ 加入多少个 i 可以合法。

gym 100886A

发现 $x_i \le 30$，这样不同的物品只有 30 种，问题变成了多重背包。对同一种物品进行转移的时候，记 $g[j][k]$ 最多往 $\{A\},\{B\}$ 加入多少个 i 可以合法。

初始化 $g[j][k] = dp[i-1][j][k]?0 : inf$

转移 $g[j][k] = min\{g[j-i][k]+1, g[j][k-i]+1\}$

最后 $dp[i][j][k] = [g[i][j][k] \le cnt_i]$

gym 100886A

发现 $x_i \leq 30$，这样不同的物品只有 30 种，问题变成了多重背包。对同一种物品进行转移的时候，记 $g[j][k]$ 最多往 $\{A\},\{B\}$ 加入多少个 i 可以合法。

初始化 $g[j][k] = dp[i-1][j][k]?0 : inf$

转移 $g[j][k] = min\{g[j-i][k]+1, g[j][k-i]+1\}$

最后 $dp[i][j][k] = [g[i][j][k] \leq cnt_i]$

注意到 $j,k \leq \frac{sum\{x_i\}}{3}$，所以复杂度将为 $O(30 * sum\{x_i\}^2/9)$

51nod1201 整数划分

求 n 的整数划分方案数。

$n \leq 50000$

51nod1201 整数划分

求 n 的整数划分方案数。

$n \leq 50000$

由于每个数字都不同，所以最多划分成 $\sqrt{2n}$ 个数。

51nod1201 整数划分

求 n 的整数划分方案数。

$n \leq 50000$

由于每个数字都不同，所以最多划分成 $\sqrt{2n}$ 个数。

考虑不断给序列中每个元素 $+1$，$dp[i][j]$ 表示序列中有 i 个元素，且剩下的和为 j。

51nod1201 整数划分

求 n 的整数划分方案数。

$n \leq 50000$

由于每个数字都不同，所以最多划分成 $\sqrt{2n}$ 个数。

考虑不断给序列中每个元素 $+1$，$dp[i][j]$ 表示序列中有 i 个元素，且剩下的和为 j。

$dp[i][j] = dp[i-1][j-i] + dp[i][j-i]$

51nod1201 整数划分

求 n 的整数划分方案数。

$n \le 50000$

由于每个数字都不同，所以最多划分成 $\sqrt{2n}$ 个数。

考虑不断给序列中每个元素 $+1$，$dp[i][j]$ 表示序列中有 i 个元素，且剩下的和为 j。

$dp[i][j] = dp[i-1][j-i] + dp[i][j-i]$

最终的答案就是 $\sum_{i=1}^{350} dp[i][n]$

51nod1201 整数划分

求 n 的整数划分方案数。

$n \le 50000$

由于每个数字都不同，所以最多划分成 $\sqrt{2n}$ 个数。

考虑不断给序列中每个元素 $+1$，$dp[i][j]$ 表示序列中有 i 个元素，且剩下的和为 j。

$dp[i][j] = dp[i-1][j-i] + dp[i][j-i]$

最终的答案就是 $\sum_{i=1}^{350} dp[i][n]$

时间复杂度 $O(n^{1.5})$

NOIP2017 联考 count

给定 n，求合法的 $(x_1, x_2, x_3, ..., x_{2m})$ 组数。一组 x 是合法的，当且仅当

- $\forall i \in [1, 2m]$，$x_i \in Z^+$，$x_i \mid n$.
- $\prod_{i=1}^{2m} x_i \ \le \ n^m$.

$n \le 10^9$，$m \le 100$。

NOIP2017 联考 count

令 $F(x)=\prod_{i=1}^{2m} x_i$。发现 $F(x)<n^m$ 与 $F(x)>n^m$ 方案数是对称的，只要计算 $F(x)=n^m$ 组数。

NOIP2017 联考 count

令 $F(x)=\prod_{i=1}^{2m}x_i$。发现 $F(x)<n^m$ 与 $F(x)>n^m$ 方案数是对称的，只要计算 $F(x)=n^m$ 组数。

对每个质因子 p 单独考虑，假设 n 中 p 的质因子有 w 个。

NOIP2017 联考 count

令 $F(x) = \prod_{i=1}^{2m} x_i$。发现 $F(x) < n^m$ 与 $F(x) > n^m$ 方案数是对称的，只要计算 $F(x) = n^m$ 组数。

对每个质因子 p 单独考虑，假设 n 中 p 的质因子有 w 个。

相当于求 $\sum_{i=1}^{2m} a_j = w * m, 0 \le a_j \le w$ 的方案数。

NOIP2017 联考 count

令 $F(x)=\prod_{i=1}^{2m}x_i$。发现 $F(x)<n^m$ 与 $F(x)>n^m$ 方案数是对称的，只要计算 $F(x)=n^m$ 组数。

对每个质因子 p 单独考虑，假设 n 中 p 的质因子有 w 个。

相当于求 $\sum_{i=1}^{2m}a_j=w*m, 0\le a_j\le w$ 的方案数。

令 $dp[i][j]$ 表示前 i 个数 sum 是 j 的方案数。

NOIP2017 联考 count

令 $F(x)=\prod_{i=1}^{2m}x_i$。发现 $F(x)<n^m$ 与 $F(x)>n^m$ 方案数是对称的，只要计算 $F(x)=n^m$ 组数。

对每个质因子 p 单独考虑，假设 n 中 p 的质因子有 w 个。

相当于求 $\sum_{i=1}^{2m}a_j=w*m, 0\le a_j\le w$ 的方案数。

令 $dp[i][j]$ 表示前 i 个数 sum 是 j 的方案数。

$dp[i][j]=\sum_{k=0}^{w}dp[i-1][j-k]$

NOIP2017 联考 count

令 $F(x)=\prod_{i=1}^{2m} x_i$。发现 $F(x)<n^m$ 与 $F(x)>n^m$ 方案数是对称的，只要计算 $F(x)=n^m$ 组数。

对每个质因子 p 单独考虑，假设 n 中 p 的质因子有 w 个。

相当于求 $\sum_{i=1}^{2m} a_j = w*m, 0 \le a_j \le w$ 的方案数。

令 $dp[i][j]$ 表示前 i 个数 sum 是 j 的方案数。

$dp[i][j]=\sum_{k=0}^{w} dp[i-1][j-k]$

最后利用乘法原理即可。

时间复杂度 $O(\sqrt{n}+\log n*m^2)$

JSOI2016 招募团体

n 个招募人之间形成一棵树形依赖关系，每个人有招募费用 S_i，战斗力 P_i。要选择不超过 K 个人，$maximize\frac{\sum S_i}{\sum P_i}$

数据范围 $n, K \le 2500, P_i, S_i \le 1000$

JSOI2016 招募团体

n 个招募人之间形成一棵树形依赖关系，每个人有招募费用 S_i，战斗力 P_i。要选择不超过 K 个人，$maximize \frac{\sum S_i}{\sum P_i}$

数据范围 $n, K \leq 2500, P_i, S_i \leq 1000$

二分答案 ans，$\sum S_i \geq ans * \sum P_i$

JSOI2016 招募团体

n 个招募人之间形成一棵树形依赖关系，每个人有招募费用 S_i，战斗力 P_i。要选择不超过 K 个人，$maximize\frac{\sum S_i}{\sum P_i}$

数据范围 $n, K \le 2500, P_i, S_i \le 1000$

二分答案 ans，$\sum S_i \ge ans * \sum P_i$

即 $maximize \quad \sum (S_i - P_i * ans)$

JSOI2016 招募团体

n 个招募人之间形成一棵树形依赖关系，每个人有招募费用 S_i，战斗力 P_i。要选择不超过 K 个人，$maximize \frac{\sum S_i}{\sum P_i}$

数据范围 $n, K \leq 2500, P_i, S_i \leq 1000$

二分答案 ans，$\sum S_i \geq ans * \sum P_i$

即 $maximize \quad \sum (S_i - P_i * ans)$

$dp[i][j]$ 表示 i 子树内招募 j 个人的最大值。

时间复杂度 $O(n^2)$

51nod1705 七星剑

剑上要铸造 7 颗星。假设现在有 $i-1$ 颗星，要铸造第 i 颗星。共有 n 种宝石，如果选择宝石 j 铸造，需要花费 $c(i,j)$ 的代价，那么有 $prob(i,j)$ 的概率成功，如果失败了，会损失 $lose(i,j)$ 颗星。

$n \leq 100$

51nod1705 七星剑

剑上要铸造 7 颗星。假设现在有 $i-1$ 颗星，要铸造第 i 颗星。共有 n 种宝石，如果选择宝石 j 铸造，需要花费 $c(i,j)$ 的代价，那么有 $prob(i,j)$ 的概率成功，如果失败了，会损失 $lose(i,j)$ 颗星。

$n \leq 100$

$dp[i]$ 表示铸造了 i 颗星的最小期望代价。

51nod1705 七星剑

剑上要铸造 7 颗星。假设现在有 $i-1$ 颗星，要铸造第 i 颗星。共有 n 种宝石，如果选择宝石 j 铸造，需要花费 $c(i,j)$ 的代价，那么有 $prob(i,j)$ 的概率成功，如果失败了，会损失 $lose(i,j)$ 颗星。

$n \leq 100$

$dp[i]$ 表示铸造了 i 颗星的最小期望代价。

$dp[i] = min\{dp[i-1]+c(i,j)+(1-prob(i,j))*(dp[i]-dp[i-1-lose(i,j)])\}$

51nod1705 七星剑

剑上要铸造 7 颗星。假设现在有 $i-1$ 颗星，要铸造第 i 颗星。共有 n 种宝石，如果选择宝石 j 铸造，需要花费 $c(i,j)$ 的代价，那么有 $prob(i,j)$ 的概率成功，如果失败了，会损失 $lose(i,j)$ 颗星。

$n \leq 100$

$dp[i]$ 表示铸造了 i 颗星的最小期望代价。

$dp[i] = min\{dp[i-1]+c(i,j)+(1-prob(i,j))*(dp[i]-dp[i-1-lose(i,j)])\}$

可以解出 $dp[i]$ 的值，时间复杂度 $O(7n)$

NOIP2016 换教室

v 个教室，e 条道路，n 次课程，每次上课从上次上课的教室赶到这次上课的教室，需要花费一定体力。每次上课的位置在 c_i，你可以尝试以 k_i 的概率更换在成 d_i 教室上课。需要你在不超过 m 次尝试内，使得期望耗费体力最小。尝试更换必须在一开始决定。

$1 \le n \le 2000, 0 \le m \le 2000, 0 \le v \le 300, 0 \le e \le 90000$

NOIP2016 换教室

先 floyd 求出两两之间的最短路。

NOIP2016 换教室

先 floyd 求出两两之间的最短路。

$dp[i][j][0/1]$ 表示第 i 次结束时，尝试了 j 次更换，且最后一次有没有尝试更换。那么 $dp[i][j][0]$ 就等于：

NOIP2016 换教室

先 floyd 求出两两之间的最短路。

$dp[i][j][0/1]$ 表示第 i 次结束时，尝试了 j 次更换，且最后一次有没有尝试更换。那么 $dp[i][j][0]$ 就等于：

$$max\begin{cases} dp[i-1][j][0] + dist(c_{i-1}, c_i) \\ dp[i-1][j][1] + dist(c_{i-1}, c_i) * (1 - k_{i-1}) + dist(d_{i-1}, c_i) * k_{i-1} \end{cases}$$

NOIP2016 换教室

先 floyd 求出两两之间的最短路。

$dp[i][j][0/1]$ 表示第 i 次结束时，尝试了 j 次更换，且最后一次有没有尝试更换。那么 $dp[i][j][0]$ 就等于：

$$max\begin{cases} dp[i-1][j][0] + dist(c_{i-1}, c_i) \\ dp[i-1][j][1] + dist(c_{i-1}, c_i) * (1 - k_{i-1}) + dist(d_{i-1}, c_i) * k_{i-1} \end{cases}$$

这里利用到了期望的线性性，计算 $dp[i][j][1]$ 也是类似的。有关期望的问题，往往某些步骤是独立的，可以直接统计。

NOIP2016 换教室

先 floyd 求出两两之间的最短路。

$dp[i][j][0/1]$ 表示第 i 次结束时，尝试了 j 次更换，且最后一次有没有尝试更换。那么 $dp[i][j][0]$ 就等于：

$$max\begin{cases} dp[i-1][j][0]+dist(c_{i-1},c_i) \\ dp[i-1][j][1]+dist(c_{i-1},c_i)*(1-k_{i-1})+dist(d_{i-1},c_i)*k_{i-1} \end{cases}$$

这里利用到了期望的线性性，计算 $dp[i][j][1]$ 也是类似的。有关期望的问题，往往某些步骤是独立的，可以直接统计。

时间复杂度 $O(e+v^3+n*m)$

NOIP2017 联考 number

给定 n, m，求满足下列条件的正整数个数

- 不含前导 0
- 重排列后可以得到 n
- 是 m 的倍数

$n < 10^{20}, m \leq 100$

NOIP2017 联考 number

给定 n, m，求满足下列条件的正整数个数

- 不含前导 0
- 重排列后可以得到 n
- 是 m 的倍数

$n < 10^{20}, m \leq 100$

令 $f[i][j]$ 表示已经使用的数状态是 i，形成的数 $mod\ m = j$ 的方案数。

NOIP2017 联考 number

给定 n, m，求满足下列条件的正整数个数

- 不含前导 0
- 重排列后可以得到 n
- 是 m 的倍数

$n < 10^{20}, m \leq 100$

令 $f[i][j]$ 表示已经使用的数状态是 i，形成的数 $mod\ m = j$ 的方案数。

合法的状态不会超过 3^{10}，每个数暴力枚举下一位的转移。

NOIP2017 联考 number

给定 n, m，求满足下列条件的正整数个数

- 不含前导 0
- 重排列后可以得到 n
- 是 m 的倍数

$n < 10^{20}, m \leq 100$

令 $f[i][j]$ 表示已经使用的数状态是 i，形成的数 $mod\ m = j$ 的方案数。

合法的状态不会超过 3^{10}，每个数暴力枚举下一位的转移。

时间复杂度 $O(3^{10} * 10 * m)$

吃特色菜

求面积为正的合法六边形方案数。每条边长度是 $[1, n]$ 间的正整数，且不同边之间没有顺序。相同的边最多选 K 条，最长边长度至少为 L，其他边的长度不超过 X。

$2 \leq n \leq 10^9, 2 \leq L \leq n, n - L \leq 100, 1 \leq X$

吃特色菜

求面积为正的合法六边形方案数。每条边长度是 $[1, n]$ 间的正整数，且不同边之间没有顺序。相同的边最多选 K 条，最长边长度至少为 L，其他边的长度不超过 X。

$2 \leq n \leq 10^9, 2 \leq L \leq n, n - L \leq 100, 1 \leq X$

限制为 $x_1 \leq x_2 \leq x_3 \leq x_4 \leq x_5 \leq X, x_6$，$L \leq x_6 \leq N$，同种长度不能使用超过 K 次，$x_1 + x_2 + x_3 + x_4 + x_5 > x_6$。

吃特色菜

求面积为正的合法六边形方案数。每条边长度是 $[1, n]$ 间的正整数，且不同边之间没有顺序。相同的边最多选 K 条，最长边长度至少为 L，其他边的长度不超过 X。

$2 \le n \le 10^9, 2 \le L \le n, n - L \le 100, 1 \le X$

限制为 $x_1 \le x_2 \le x_3 \le x_4 \le x_5 \le X, x_6$，$L \le x_6 \le N$，同种长度不能使用超过 K 次，$x_1 + x_2 + x_3 + x_4 + x_5 > x_6$。

使用数位 dp，将每个数都看成 2 进制。

吃特色菜

求面积为正的合法六边形方案数。每条边长度是 $[1, n]$ 间的正整数，且不同边之间没有顺序。相同的边最多选 K 条，最长边长度至少为 L，其他边的长度不超过 X。

$2 \le n \le 10^9, 2 \le L \le n, n - L \le 100, 1 \le X$

限制为 $x_1 \le x_2 \le x_3 \le x_4 \le x_5 \le X, x_6$，$L \le x_6 \le N$，同种长度不能使用超过 K 次，$x_1 + x_2 + x_3 + x_4 + x_5 > x_6$。

使用数位 dp，将每个数都看成 2 进制。

就记 $dp[i][S][carry]$ 表示考虑后 i 位，数与数之间的关系为 S，同时记录 $x_1 + ... + x_5$ 的进位。

吃特色菜

求面积为正的合法六边形方案数。每条边长度是 $[1, n]$ 间的正整数，且不同边之间没有顺序。相同的边最多选 K 条，最长边长度至少为 L，其他边的长度不超过 X。

$2 \le n \le 10^9, 2 \le L \le n, n - L \le 100, 1 \le X$

限制为 $x_1 \le x_2 \le x_3 \le x_4 \le x_5 \le X, x_6$，$L \le x_6 \le N$，同种长度不能使用超过 K 次，$x_1 + x_2 + x_3 + x_4 + x_5 > x_6$。

使用数位 dp，将每个数都看成 2 进制。

就记 $dp[i][S][carry]$ 表示考虑后 i 位，数与数之间的关系为 S，同时记录 $x_1 + ... + x_5$ 的进位。

时间复杂度 $O(30 * 3^6 * 5 * 2^5)$。

SDOI2017 序列计数

给定 n, m, p，求下列满足条件的序列个数 $mod\ 20170408$

- 每个数都是 $[1, m]$ 之间的正整数.
- n 个数字的和是 p 的倍数.
- 至少有一个数是质数

$1 \leq n \leq 10^9, 1 \leq m \leq 2 * 10^7, 1 \leq p \leq 100$

SDOI2017 序列计数

给定 n, m, p，求下列满足条件的序列个数 mod 20170408

- 每个数都是 $[1, m]$ 之间的正整数.
- n 个数字的和是 p 的倍数.
- 至少有一个数是质数

$1 \le n \le 10^9, 1 \le m \le 2 * 10^7, 1 \le p \le 100$

可以预处理出 cnt_i 表示用 $mod\ p = i$ 的数字个数。

SDOI2017 序列计数

给定 n, m, p，求下列满足条件的序列个数 $mod\ 20170408$

- 每个数都是 $[1, m]$ 之间的正整数.
- n 个数字的和是 p 的倍数.
- 至少有一个数是质数

$1 \le n \le 10^9, 1 \le m \le 2 * 10^7, 1 \le p \le 100$

可以预处理出 cnt_i 表示用 $mod\ p = i$ 的数字个数。

之后可以用矩阵快速幂求出 n 个数的和 $mod\ p = 0$ 的方案数。

SDOI2017 序列计数

给定 n, m, p，求下列满足条件的序列个数 $mod\ 20170408$

- 每个数都是 $[1, m]$ 之间的正整数.
- n 个数字的和是 p 的倍数.
- 至少有一个数是质数

$1 \le n \le 10^9, 1 \le m \le 2 * 10^7, 1 \le p \le 100$

可以预处理出 cnt_i 表示用 $mod\ p = i$ 的数字个数。

之后可以用矩阵快速幂求出 n 个数的和 $mod\ p = 0$ 的方案数。

用所有数字的方案减去只有合数的方案，就是最终的答案。

SDOI2017 序列计数

给定 n, m, p，求下列满足条件的序列个数 mod 20170408

- 每个数都是 $[1, m]$ 之间的正整数.
- n 个数字的和是 p 的倍数.
- 至少有一个数是质数

$1 \le n \le 10^9, 1 \le m \le 2 * 10^7, 1 \le p \le 100$

可以预处理出 cnt_i 表示用 $mod\ p = i$ 的数字个数。

之后可以用矩阵快速幂求出 n 个数的和 $mod\ p = 0$ 的方案数。

用所有数字的方案减去只有合数的方案，就是最终的答案。

时间复杂度 $O(m + p^2 * \log n)$

AGC005 K Perm Counting

给定 n, K，求满足 $|a_i - i| \neq K$ 的排列个数。

数据范围 $n \leq 4000$

AGC005 K Perm Counting

给定 n, K，求满足 $|a_i - i| \neq K$ 的排列个数。

数据范围 $n \leq 4000$

考虑计算 f_i 表示有 i 个位置 $a_x - x = K$，其他位置不确定的方案数。

AGC005 K Perm Counting

给定 n, K，求满足 $|a_i - i| \neq K$ 的排列个数。

数据范围 $n \leq 4000$

考虑计算 f_i 表示有 i 个位置 $a_x - x = K$，其他位置不确定的方案数。

答案就等于 $\sum_{i=0}^{n} (n-i)! * (-1)^i * f[i]$

AGC005 K Perm Counting

给定 n, K，求满足 $|a_i - i| \neq K$ 的排列个数。

数据范围 $n \leq 4000$

考虑计算 f_i 表示有 i 个位置 $a_x - x = K$，其他位置不确定的方案数。

答案就等于 $\sum_{i=0}^{n} (n-i)! * (-1)^i * f[i]$

如果将 i 与满足 $a_i - i = K$ 的 a_i 配对，发现构成了若干条链 $x, x+K, x+2K, x+3K..$，每一条链上可以 dp 出有 i 个位置合法的方案数，再将所有链合并。

AGC005 K Perm Counting

给定 n, K，求满足 $|a_i - i| \neq K$ 的排列个数。

数据范围 $n \leq 4000$

考虑计算 f_i 表示有 i 个位置 $a_x - x = K$，其他位置不确定的方案数。

答案就等于 $\sum_{i=0}^{n} (n-i)! * (-1)^i * f[i]$

如果将 i 与满足 $a_i - i = K$ 的 a_i 配对，发现构成了若干条链 $x, x+K, x+2K, x+3K..$，每一条链上可以 dp 出有 i 个位置合法的方案数，再将所有链合并。

时间复杂度 $O(n^2)$

谢谢大家