# DP 选讲

Alan_Zhao

2025 年 7 月 24 日

# 目录

# AGC061C. First Come First Serve

有 $n$ 个人，其中第 $i$ 个人会在 $a_i$ 和 $b_i$ 两个数中选一个，问这 $n$ 个人的数字有多少种可能的**顺序**，对 $998244353$ 取模。

$n \leq 5 \times 10^5$，$a_i, b_i$ 形成一个 $1, 2, \ldots, 2n$ 的排列，$a_i < a_{i+1}$，$b_i < b_{i+1}$。

# Solution (I)

对于一种顺序，考虑两个对应这个顺序的选择方案。考虑这两个方案中任意一个不同的位置 $i$，一定不会有其他人在区间 $(a_i, b_i)$ 内选择。此时 $i$ 选择 $a_i$ 还是 $b_i$ 都不会影响最终的顺序。

所以，可以认为每个顺序对应的选择方案是，对于每个选了 $b_i$ 的 $i$，一定需要 $(a_i, b_i)$ 中有被选择的元素。显然这样的方案是唯一的。

## Solution (II)

考虑容斥掉不满足这个条件的方案。对于每个 $i$，有三种选择：

- 选择 $a_i$ 或 $b_i$，系数是 $1$。
- 选择 $b_i$ 并且钦定 $(a_i, b_i)$ 中所有元素都不被选择，系数是 $-1$。每次选择这种情况，都可以确定一个区间 $[L_i, R_i]$ 内的选择情况，其中 $L_i \le i \le R_i$。注意到这些区间有交的时候，贡献是 $0$，所以只需考虑无交的情况。

于是，直接 DP 决定每个 $i$ 选择哪种情况即可。如果一个 $i$ 选择了第三种情况，直接从 $f_{L_i-1}$ 转移到 $f_{R_i}$。时间复杂度 $O(n)$。

# P11427. 绝顶之战

有一个长度为 $m$ 的空间和 $n$ 个物品，第 $i$ 个物品的长度为 $a_i$。

按顺序将物品放入空间，若有长度 $\geq a_i$ 的空位则必须任选一个放入，否则跳过。

求所有可能的空间占用长度，空间占用长度定义为被放入物品的长度之和。

$n \leq 14$，$m, a_i \leq 10^{16}$。

## Solution

先枚举物品集合 $S$，判断是否能放入恰好 $S$ 中的物品。

考虑如果有两个集合 $S, T$ 表示能放进去和不能放进去的物品集合，只需要 DP 出最大的满足条件的空间长度 $L$，那么所有的在 $[\sum_{x\in S} a_x, L]$ 间的长度都是可行的。

于是设 $f_{S,T}$ 表示这个最大的空间长度。按顺序考虑每个物品，放入一个物品相当于将整个空间分成两部分，而无法放入物品相当于将这个 DP 值对 $a_i - \varepsilon$ 取 $\min$。两种情况都是容易转移的。

时间复杂度 $O(4^n)$。

# P5972. Desant

给定一个 $1, 2, \ldots, n$ 的排列 $a$。

对于每个 $k \in [1, n]$，求出：

- $a$ 中长度为 $k$ 的子序列的最小逆序对数。
- 取得该最小值的子序列数量。

$n \leq 40$。

# Solution (I)

暴力的做法是从左往右枚举每个数选不选，然后对于每个选的数，逆序对个数需要加上这个数前面比它大的数的个数。

发现，当确定完 $[1, i]$ 的选择情况后，我们只关心对于每个 $j \in [i+1, n]$，$a_1, a_2, \ldots, a_i$ 中有多少个被选且比 $a_j$ 大的数，而不关心 $[1, i]$ 的具体选择情况。

于是，设 $a_{i+1}, a_{i+2}, \ldots, a_n$ 从小到大排序后是 $x_1, x_2, \ldots, x_{n-i}$，只需要在 DP 状态中记录，对于每个 $j \in [1, n-i-1]$，有多少个 $a_1, \ldots, a_i$ 中被选的数，值在 $(x_j, x_{j+1})$ 之间。转移是简单的。

# Solution (II)

考虑时间复杂度。对于固定的 $i$，显然状态个数不会超过 $\prod_{j=1}^{n-i-1}(x_{j+1}-x_j)$，显然在每段长度相等时取到最大值，并且在相等的情况下，每段长度都是 $3$ 时取到最大值。

转移时还要带一个额外的 $O(n)$，于是时间复杂度就是 $O(n^2 3^{n/3})$。

# QOJ 7597. Alexey the Sage of The Six Paths

有一张初始为空的二分图，左部和右部各有 $n$ 个点，编号为 $1, 2, \ldots, 2n$。你需要添加恰好 $m$ 条边，如果添加后第 $i$ 个点的度数为 $d_i$，那么需要花费 $p_{i,d_i}$ 的代价。

给定 $l, r$，求最大匹配大小在 $[l, r]$ 间的最小代价，或者输出无解。

$n, m \le 30$。

# Solution (I)

在确定每个点度数的情况下，只需要求出最大匹配的最大值和最小值，那么在最小值到最大值区间内的每个大小都是可以取到的。

根据 Hall 定理，最大匹配大小等于 $\min_{S\subseteq L}\{|L|-|S|+|N(S)|\}$，其中 $L$ 是左部点集。所以，最大匹配的最小值小于等于 $r$，等价于存在一个集合 $S$ 使得 $|L|-|S|+|N(S)|\leq r$。最大匹配的最大值大于等于 $l$，只需要左右部都有至少 $l$ 个度数非零的点。

# Solution (II)

于是，需要在 DP 过程中钦定 $S$ 以及 $N(S)$。注意到这两部分可以分开 DP，因为只需要 $S$ 中点的度数之和大于等于 $|N(S)|$，并且 $N(S)$ 中点的度数之和大于等于 $|S|$，就一定可以连出一组合法的边集。

那么对左右部点分别 DP，以左部为例，设 $f_{i,a,b,c,d}$ 表示 $[1,i]$ 的点中，目前 $|S|=a$ 且 $S$ 中点的度数之和等于 $b$，度数大于零的点有 $c$ 个，所有点的度数之和是 $d$ 的最小代价。

对左部和右部分别 DP 后，枚举两边的点集大小，点集度数和，以及度数非零的点个数，即可计算答案。

时间复杂度 $O(n^3m^3)$。

# P10004. Permutation Counting 2

给定 $n$，求有多少个 $1 \sim n$ 的排列 $p$ 满足：

- $p$ 有 $x$ 个相邻升序对（$p_i < p_{i+1}$）。
- $p$ 的逆排列 $p^{-1}$ 有 $y$ 个相邻升序对。逆排列定义为满足 $p^{-1}_{p_i} = i$ 的排列 $p^{-1}$。

对于所有 $x, y \in [0, n)$ 求答案，对给定的模数 $MOD$ 取模。

$n \leq 500$。

## Solution (I)

对这个问题做二维的二项式反演，则只需要对每对 $(x,y)$，计算钦定 $p$ 的 $x$ 个连续上升段，并且钦定 $p^{-1}$ 的 $y$ 个连续上升段的方案数 $g_{x,y}$。

考虑逐个确定这 $x$ 个上升段，一个长为 $k$ 的上升段对 $p^{-1}$ 的影响是，往 $p^{-1}$ 的各个上升段后面填入总共 $k$ 个数。

# Solution (II)

于是设 $a_{i,j}$ 表示第 $i$ 个 $p$ 的上升段，往第 $j$ 个 $p^{-1}$ 的上升段后面填了多少数。可以建立 $p$ 到 $a$ 的一一对应，其中 $a$ 需要满足：

- $\sum_{i,j} a_{i,j} = n$。
- $a$ 的每行每列之和都大于 $0$。

对第二条限制做一次二维的二项式反演，即可算出 $g_{x,y}$，然后通过 $g_{x,y}$ 可以求出原问题的答案。

二维的二项式反演可以对两维分别做反演，所以时间复杂度 $O(n^3)$。

# ABC 310 Ex. Negative Cost

你有 $n$ 种技能，初始 $0$ 魔力，打一个 $H$ 血的怪。

第 $i$ 种技能消耗 $C_i$ 魔力，造成 $D_i$ 伤害。魔力值需要时刻非负。$D_i$ 是正的，但是 $C_i$ 可以为负数。

技能可以无限次使用，求杀死怪物的最少技能发动次数。

$n, C_i \leq 300$，$D_i \leq 10^9$，$H \leq 10^{18}$，至少有一个 $C_i$ 是负的。

# Solution (I)

记 $C=300$，把所有 $C_i$ 取反。一个合法的操作序列 $x_1,\dots,x_L$ 需要满足 $\forall i,\sum_{j=1}^{i} C_{x_j}\ge 0$。

考虑把一个合法操作序列拆成若干个合法且最大前缀和不超过 $2C$ 的序列，这总能通过适当地重排操作来做到。

对于每个合法且和不超过 $2C$ 的序列，可以进一步拆成若干个长度以及最大前缀和都不超过 $2C$ 的序列，这是因为，对于长度 $>2C$ 的序列，总可以拆出一段和为 $0$ 且长度不超过 $2C$ 的子段，将这个子段拆成单独一个序列即可。

# Solution (II)

于是可以 DP 出 $f_i$，表示长度为 $i$ 且最大前缀和不超过 $2C$ 的序列，造成的最大伤害。这部分的时间复杂度是 $O(n^3)$。

然后需要取若干个 $f_i$ 拼成最终的操作序列，注意到大部分时候取的都是 $f_x/x$ 最大的那个 $x$，这是因为，设 $l_i$ 表示除了 $x$ 之外第 $i$ 次取的长度，如果取了 $> 2C$ 个其他的 $f_i$，那么一定存在 $l_i \equiv l_j \pmod{x}$，此时把 $[i,j]$ 这一段换成若干个 $x$ 一定是更优的。

所以对除了 $x$ 以外的 $f_i$，做一次多重背包即可。时间复杂度 $O(n^3)$。

# ARC 138 F. KD Tree

给定点列 $(i, p_i)$，$p$ 是一个 $1, 2, \ldots, n$ 的排列。

每次操作可以选平行于 $x$ 轴或者平行于 $y$ 轴的一条直线，该直线将点集分成非空的两部分，然后对两部分分别做这个过程。递归到点只剩一个时加入答案序列。(描述的其实就是 KD 树的建树过程。)

求能生成多少种不同的答案序列，对 $10^9 + 7$ 取模。

$n \le 30$。

# Solution (I)

如果是对操作序列计数，那么可以直接区间 DP：设 $f_{i,j,k,l}$ 表示 $x \in [i,j], y \in [k,l]$ 的点的操作序列个数，枚举最后一次操作是什么即可 $O(n^2)$ 转移。

沿用这个 DP 状态，但是现在是对可能被生成的排列计数。对于一个状态 $f_{i,j,k,l}$，将这个范围内的点的 $x, y$ 坐标分别离散化，设有 $M$ 个点，给可能的 $2M$ 种操作定一个顺序：$x_1 < y_1 < x_2 < y_2 < \cdots < x_K < y_K$，我们认为一个排列是被字典序最小的操作序列生成的。

# Solution (II)

设 $g_p, h_p$ 分别表示字典序最小的第一次操作是 $x_p, y_p$ 的可能被生成的排列个数，按照刚才定的顺序计算这些值，每次用不限制字典序最小的方案数减去钦定第一次操作字典序更小的方案数。

例如，计算 $g_p$ 时，要减去 $g_q$ （$q < p$）的贡献，设 $x$ 坐标不超过 $p$ 的点集是 $S_p$，此时转移是 $g_p \leftarrow g_p - \frac{g_q}{f_{S_M \oplus S_q}} \cdot f_{S_p \oplus S_q} \cdot f_{S_M \oplus S_p}$。

时间复杂度 $O(n^6)$。

# P8350. 进制转换

记 $a(i), b(i)$ 分别为 $i$ 的二进制和三进制下的数位和。

求 $\sum_{i=1}^{n} x^i y^{a(i)} z^{b(i)} \pmod{998244353}$。

$n \leq 10^{13}$。

# Solution

在三进制下从高位到低位确定 $x$ 的值。考虑最低的 $[0, i-1]$ 位对二进制表示的影响不会超过 $3^i - 1$，在不进位的情况下，这只会影响二进制表示的最低 $l_i = \lceil i \log_2 3 \rceil$ 位。

于是，在 DP 过程中直接钦定每个位是否进位 $j = 0/1$，并记录填完 $\geq i$ 的三进制位时，二进制下 $[0, i-1]$ 这部分的形态 $k$，以及当前是否仍然卡 $r$ 的上界 $l = 0/1$。

但是这样做仍然是 $O(n)$ 的。考虑根号分治，设一个 $B$ 使得 $3^B = \Theta(\sqrt{n})$，对于低 $B$ 位做上述 DP，高 $B$ 位直接暴力枚举三进制下的填法。

时间复杂度 $O(\sqrt{n})$。

# UOJ 84. 水题走四方

给定一棵树，开始时有两个人在根节点处。每秒钟，两个人都可以各自向下走一条边。任意时刻，一个人都可以不花费时间地传送到另外一个人所在的节点上。求至少到达树上每个节点至少一次，花费的最短时间。

$n \le 5 \times 10^6$。

# Solution (I)

两个人在同一个点时，是没有必要区分他们的。所以，设这两个人叫 A 和 B，总可以将每次传送都调整成 B 向 A 传送，而 A 只会停在某个点等 B，或者往下走。

并且，由于传送不花费时间，如果出现了 B 在某个点等 A 的情况，可以调整成 B 先传送到 A 所在的位置，然后与 A 一起往下走。这样的话，B 在任意时刻都一定往下走。

考虑 A 最终会从根走到某个叶子，而 B 会在这个过程中负责到达其他所有节点。考虑 A 走出的这条链上所有 B 传送到的点，设其中相邻的两个是 $x, y$，A 会停在 $x$ 等 B，而 B 会不断地访问 $x \rightsquigarrow y$ 这条链上的所有其他子树，每访问完一个子树都会传回 A，直到访问最后一个子树时，A 才会往下走到 $y$。

# Solution (II)

设 $S_x$ 表示以 $x$ 为根的子树，显然 B 会对每个 $S_x \setminus S_y$ 中的每个叶子节点，都做一次“访问并传送”。并且，将其中最深的一个叶子留到最后，来给 A 留出最多的向下走的时间，是最优的策略。

设 $\operatorname{sum}(S)$ 表示 $S$ 中叶子的深度和，$\max(S)$ 表示 $S$ 中叶子的最大深度，从 $x$ 能转移到 $y$ 当且仅当 $\max(S_u \setminus S_v) \geq \operatorname{dep}_y$，转移是

$$f_x + \operatorname{sum}(S_x \setminus S_y) - |S_x \setminus S_y| \cdot \operatorname{dep}_x \to f_y.$$

# Solution (III)

对于一个 $y$，能够转移到它的 $x$ 是根到 $y$ 上的一段前缀。并且，找到其中深度最大的合法点作为 $x$ 一定是最优的。那么暴力做法就是重链剖分后，二分找到深度最大的 $x$，时间复杂度 $O(n\log n)$。

注意到 $\max(S)$ 的定义与长链剖分很有关系，具体地，长链剖分后，如果 $x$ 和 $y$ 不在同一条长链上，那么切换的位置一定可以多进行一次转移，从而变得更优。于是只需要考虑与 $y$ 在同一长链上的 $x$ 的转移，直接使用单调栈维护即可。

时间复杂度 $O(n)$。

# UOJ 823. 铁轨回收

给定两个长为 $n$ 的非负整数序列 $A, B$，对于 $i = 1, 2, \ldots, (n-1)$，依次进行如下操作：

- 随机选一个整数 $j \in [i+1, n]$，执行 $A_j \leftarrow \min(A_i + A_j, B_j)$。

对于每个 $i = 0, 1, \ldots, B_n$，求所有操作后 $A_n = i$ 的概率，对 998244353 取模。

$n \le 50$，$A_i, B_i \le 30$。

# Solution (I)

认为初始的 $A_i$ 是 $A'_i$。

将每次操作的 $i, j$ 从 $i$ 向 $j$ 连边，形成一棵以 $n$ 为根的内向树。

考虑统计执行完 $1 \sim i-1$ 的操作后，$A_i = j$ 的方案数：

- 对于 $j < B_i$，只要统计 $[A'_i + \sum_{t \in son_j} A_t = j]$ 的方案数即可。
- 对于 $j = B_i$，做一个容斥，用总方案数减去 $j < B_i$ 的方案数。

# Solution (II)

考虑从后往前依次确定每个 $A_i$，如果 $A_i < B_i$ 或者选择了“减去贡献”，则需要前面和为某个固定值的一些 $A_j$ 连向它。如果 $A_i = B_i$ 且选择了“统计总方案数”，则对 $i$ 子树的连边情况没有任何限制。

于是考虑 DP，设 $f_{i,S,j}$ 表示考虑了 $[i,n]$，需要被连固定值的状态是 $S$，有 $j$ 个可以任意连的接口的方案数。

转移时，考虑 $A_i$ 的值，以及 $A_i$ 的父亲的类型：

- $fa_i$ 为那 $k$ 个点中的一个，此时 $k \leftarrow k+1$。
- 统计 $A_i = B_i$，并选择”统计总方案数 ”：此时 $k \leftarrow k+1$，并选择 $S$ 中的一个元素减去 $B_i$。

## Solution (III)

- 统计 $A_i = B_i$，并选择"减去 $j < B_i$ 的方案"，即做 $-1$ 的贡献。此时选择 $S$ 中的一个元素减去 $B_i$，然后在 $S$ 中加入一个 $t < B_i$ 的 $t - A'_i$。
- 统计 $A_i < B_i$ 的贡献。此时选择 $S$ 中的一个元素减去 $A_i$，然后在 $S$ 中加入 $A_i - A'_i$。

发现在这个过程中，$S$ 中元素之和是不会变大的，所以 $S$ 只有 $\pi(B_n)$ 种方案。于是总的时间复杂度是 $O(n^2\pi(B_n)B_n^{1.5})$。