动态规划

liuhengxi

February 3, 2023

约定：本文在有标注 *0-indexed* 下，从 0 开始编号；否则默认从 1 开始。

$a \leftarrow b$ 表示 C++ 中的 `a=b`

$a \xleftarrow{+} b$ 表示 C++ 中的 `a+=b`

$[\texttt{true}] = 1, [\texttt{false}] = 0$

题目后面的 (*2600) 表示 Codeforces 难度 2600。(~2600) 表示估计难度 2600。

这里默认大家都会以下内容：

- 背包 DP
- 区间 DP
- 树形 DP
- 状压 DP
- 数位 DP
- 基础的时间复杂度均摊证明

- 树上背包复杂度：$O(\sum_u \sum_{v_1<v_2\in ch_u} siz_{v_1} \cdot siz_{v_2}) = O(n^2)$

- 树上背包复杂度：$O(\sum_u \sum_{v_1<v_2\in ch_u} siz_{v_1} \cdot siz_{v_2}) = O(n^2)$
  - 树上背包最多选 k 个时：$O(\sum_u \sum_{v_1<v_2\in ch_u} \min\{siz_{v_1}, k\} \cdot \min\{siz_{v_2}, k\}) = O(nk)$

- 树上背包复杂度：$O(\sum_u \sum_{v_1 < v_2 \in ch_u} siz_{v_1} \cdot siz_{v_2}) = O(n^2)$
  - 树上背包最多选 k 个时：$O(\sum_u \sum_{v_1 < v_2 \in ch_u} \min\{siz_{v_1}, k\} \cdot \min\{siz_{v_2}, k\}) = O(nk)$
- $1 \sim n$ 的整除整数对：$\sum_{i=1}^n \lfloor \frac{n}{i} \rfloor = O\left(n \sum_{i=1}^n \frac{1}{i}\right) = O(n \log n)$

$T(n) = 2T(\frac{n}{2}) + O(n)$，则 $T(n) = O(n\log n)$。
$T(n) = 2T(\frac{n}{2}) + O(n\log n)$，则 $T(n) = O(n\log^2 n)$。
$T(n) = 2T(\frac{n}{2}) + O(n^2)$，则 $T(n) = O(n^2)$。

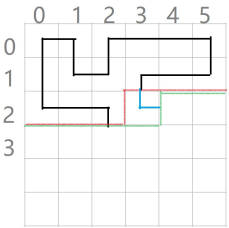
插头 DP 是一类用状压的方法处理网格线上的路径的 DP。

插头 DP 是一类用状压的方法处理网格线上的路径的 DP。

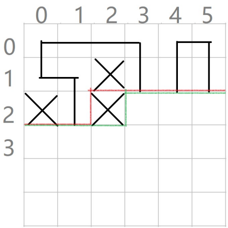
Example 1 (HDU 1693)

求用若干条回路覆盖 $n \times m$ 棋盘的方案数，给定的位置有障碍。（有障碍的不能放，没障碍的必须放）

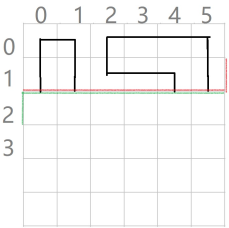
$$1 \leq n, m \leq 11$$



$f(2,3,0010100) \to f(2,4,0010100)$



$f(2,2,0100111) \to f(2,3,0100111)$



$f(1,6,1100110) \to f(2,0,0110011)$

Example 2 (Codeforces 100220F)

求用一条回路覆盖 $n \times m$ 棋盘的方案数。

$$1 \leq nm \leq 100$$

Example 3 (luogu P5056 / acm.timus.ru 1519)

求用一条回路覆盖 $n \times m$ 棋盘的方案数，给定的位置有障碍。

$$1 \leq n, m \leq 12$$

Example 4 (~1700)

有转移

$$f_i = \max_{j=\max\{1,i-d\}}^{i-1} \{f_j + a_i + b(i-j)\}$$

在 $O(n)$ 内进行 DP。

Example 4 (~1700)

有转移

$$f_i = \max_{j=\max\{1,i-d\}}^{i-1} \{f_j + a_i + b(i-j)\}$$

在 $O(n)$ 内进行 *DP*。

多重背包加入一个物品的转移，按照下标模 $cost$ 分类后满足上述形式，因此可用单调队列优化。

Example 5 (~1900)

有转移

$$f_i = \max_{j=1}^{i-1} \{f_j + u_j + v_i - x_j k_i\}$$

其中 $\{x_i\}$ 单调递增，$\{k_i\}$ 单调。在 $O(n)$ 内进行 *DP*。

对于取 min 的 DP，把上凸壳换成下凸壳。

- $\{k_i\}$ 单调递增，用单调队列维护。
- $\{k_i\}$ 单调递减，用单调栈维护。

对于取 min 的 DP，把上凸壳换成下凸壳。

- $\{k_i\}$ 单调递增，用单调队列维护。
- $\{k_i\}$ 单调递减，用单调栈维护。

对于只有 $\{x_i\}$ 单调的 DP，仍用单调栈维护，但是查询改为在凸壳上二分，而不是取栈顶或队首。

对于 $\{k_i\}$ 也不单调的 DP，有两种较常用的做法。

1. 平衡树维护
2. CDQ 分治

平衡树可以用 `std::set` 实现，用 `lower_bound` 二分。
下面讲如何用 CDQ 分治做。

考虑分治区间 $[l,r)$ 的过程。

- 若 $r-l=1$，则区间内没有转移，直接退出。
- 若 $r-l>1$
  - 令 $mid=\lfloor (l+r)/2 \rfloor$。
  - 递归区间 $[l,mid)$。

考虑分治区间 $[l, r)$ 的过程。

- 若 $r - l = 1$，则区间内没有转移，直接退出。
- 若 $r - l > 1$
  - 令 $mid = \lfloor (l + r)/2 \rfloor$。
  - 递归区间 $[l, mid)$。
  - 解决区间 $[l, mid)$ 对 $[mid, r)$ 的贡献。

考虑分治区间 $[l, r)$ 的过程。

- 若 $r - l = 1$，则区间内没有转移，直接退出。
- 若 $r - l > 1$
  - 令 $mid = \lfloor (l + r)/2 \rfloor$。
  - 递归区间 $[l, mid)$。
  - 解决区间 $[l, mid)$ 对 $[mid, r)$ 的贡献。
    此时 $[l, mid), [mid, r)$ 区间内部的顺序并不影响。

考虑分治区间 $[l, r)$ 的过程。

- 若 $r - l = 1$，则区间内没有转移，直接退出。
- 若 $r - l > 1$
  - 令 $mid = \lfloor (l + r)/2 \rfloor$。
  - 递归区间 $[l, mid)$。
  - 解决区间 $[l, mid)$ 对 $[mid, r)$ 的贡献。
    此时 $[l, mid), [mid, r)$ 区间内部的顺序并不影响。
    对于 $[l, mid)$ 内的元素，按照 $\{x_i\}$ 排序。

考虑分治区间 $[l, r)$ 的过程。

- 若 $r - l = 1$，则区间内没有转移，直接退出。
- 若 $r - l > 1$
  - 令 $mid = \lfloor (l + r)/2 \rfloor$。
  - 递归区间 $[l, mid)$。
  - 解决区间 $[l, mid)$ 对 $[mid, r)$ 的贡献。
    此时 $[l, mid), [mid, r)$ 区间内部的顺序并不影响。
    对于 $[l, mid)$ 内的元素，按照 $\{x_i\}$ 排序。
    对于 $[mid, r)$ 内的元素，按照 $\{k_i\}$ 排序。

考虑分治区间 $[l, r)$ 的过程。

- 若 $r - l = 1$，则区间内没有转移，直接退出。
- 若 $r - l > 1$
  - 令 $mid = \lfloor (l + r)/2 \rfloor$。
  - 递归区间 $[l, mid)$。
  - 解决区间 $[l, mid)$ 对 $[mid, r)$ 的贡献。
    此时 $[l, mid), [mid, r)$ 区间内部的顺序并不影响。
    对于 $[l, mid)$ 内的元素，按照 $\{x_i\}$ 排序。
    对于 $[mid, r)$ 内的元素，按照 $\{k_i\}$ 排序。
    此时 DP 满足前述的形式，可以用单调栈或单调队列解决。

考虑分治区间 $[l,r)$ 的过程。

- 若 $r-l=1$，则区间内没有转移，直接退出。
- 若 $r-l>1$
  - 令 $mid=\lfloor (l+r)/2 \rfloor$。
  - 递归区间 $[l,mid)$。
  - 解决区间 $[l,mid)$ 对 $[mid,r)$ 的贡献。
    此时 $[l,mid),[mid,r)$ 区间内部的顺序并不影响。
    对于 $[l,mid)$ 内的元素，按照 $\{x_i\}$ 排序。
    对于 $[mid,r)$ 内的元素，按照 $\{k_i\}$ 排序。
    此时 DP 满足前述的形式，可以用单调栈或单调队列解决。
  - 递归区间 $[mid,r)$。

时间复杂度 $T(n)=2T(\frac{n}{2})+O(n\log n)$，$T(n)=O(n\log^2 n)$。

Example 6 (Codeforces 643C (*2400))

有一个长度为 n 的序列 t，你需要把它划分为 k 个非空连续段。

每一段 $[l, r]$ 的代价为 $\sum_{i=l}^{r} \frac{t_i}{\sum_{j=l}^{i} t_j}$。考虑最小化总代价。

$1 \le n \le 2 \cdot 10^5, 1 \le k \le \min\{n, 50\}, 1 \le t_i \le 10^5$。

Example 7 (codeforces 319c (*2100))

你要砍 n 棵树，高度分别为 $1 = a_1 \le a_2 \le \cdots \le a_n$。

每棵树有权值 $b_1 \ge b_2 \ge \cdots \ge b_n = 0$。

每砍 1 单位高度，花费之前砍完的树的权值的最小值，（若没砍完过，花费 $+\infty$）。

第 1 个高度不需要花费。$1 \le n \le 10^5, a_i, b_i \le 10^9$

Example 8 (Codeforces 311B (*2400))

直线上有 m 个位置，n 只猫分布在这些位置上，第 i 个位置与第 $i-1$ 个相距 d_i。

第 i 只猫会在 t_i 时刻出现在第 h_i 个位置。

现在安排 p 个人接这些猫，每个人会从位置 1 出发，以 1 的速度向位置 n 前进。

到达一个位置时，人会接走已经出现的所有猫，不花费时间。

最小化猫的等待时间之和。

$1 \le n \le 10^5, 2 \le m \le 10^5, 1 \le p \le 100, 1 \le d_i < 10^4$

$1 \le h_i \le m, 0 \le t_i \le 10^9$

为了方便，n, m 与原题相反。

Definition (区间包含单调性)

若函数 $w(l,r)$（通常表示花费）满足：

$$\forall a \le b \le c \le d, w(b,c) \le w(a,d)$$

称 $w(l,r)$ 满足区间包含单调性。

Definition (区间包含单调性)

若函数 $w(l, r)$（通常表示花费）满足：

$$\forall a \le b \le c \le d, w(b, c) \le w(a, d)$$

称 $w(l, r)$ 满足区间包含单调性。

Definition (四边形不等式)

若函数 $w(l, r)$ 满足：

$$\forall a \le b \le c \le d, w(a, c) + w(b, d) \le w(a, d) + w(b, c)$$

称 $w(l, r)$ 满足四边形不等式。

常见的四边形不等式优化 DP 有：

常见的四边形不等式优化 DP 有：

- 1D1D（状态 $O(n)$，每个状态转移 $O(n)$）

常见的四边形不等式优化 DP 有：

- 1D1D（状态 $O(n)$，每个状态转移 $O(n)$）
- 2D1D（状态 $O(n^2)$，每个状态转移 $O(n)$）

常见的四边形不等式优化 DP 有：

- 1D1D（状态 $O(n)$，每个状态转移 $O(n)$）
- 2D1D（状态 $O(n^2)$，每个状态转移 $O(n)$）
  - 区间
  - 分层（$f_{i,j}$ 从 $f_{i-1,k}$ 转移得到）

Theorem

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

$k_i = \min \{j \mid f_i = f_j + w(j, i), j < i\}$

若 w 满足四边形不等式，则 k（称为决策点）单调递增。

Theorem

$f_i = \min_{j<i} \{f_j + w(j, i)\}$
$k_i = \min \{j \mid f_i = f_j + w(j, i), j < i\}$
若 w 满足四边形不等式，则 k（称为决策点）单调递增。

Corollary

$f_i = \min_{j<i, j\le t} \{f_j + w(j, i)\}$
$k_i = \min \{j \mid f_i = f_j + w(j, i), j < i\}$
若 w 满足四边形不等式，则 k 单调递增。

Theorem

$f_i = \min_{j<i} \{f_j + w(j, i)\}$
$k_i = \min \{j \mid f_i = f_j + w(j, i), j < i\}$
若 w 满足四边形不等式，则 k（称为决策点）单调递增。

Corollary

$f_i = \min_{j<i, j\le t} \{f_j + w(j, i)\}$
$k_i = \min \{j \mid f_i = f_j + w(j, i), j < i\}$
若 w 满足四边形不等式，则 k 单调递增。
令 $w'(l, r) = \begin{cases} w(l, r) & l \le t \\ w(l, r) + M & l > t \end{cases}$。其中 M 是一个很大的数。
对 w' 运用定理即得证。

Example 9 (~2200)

给定 a, b, c, d，满足

$a_0 > a_1 > \cdots > a_n > 0, 0 < b_0 < b_1 < \cdots < b_n$

$c_0 > c_1 > \cdots > c_n > 0, 0 < d_0 < d_1 < \cdots < d_n$

$f_0 = 0, f_i = \min_{j<i}\{f_j + a_j b_i + c_j d_i\}$

在 $O(n \log n)$ 内进行 *DP*。

Theorem

$f_i = \min_{j<i}\{w(j,i)\}$
$k_i = \min\{j \mid f_i = w(j,i), j < i\}$
若 w 满足四边形不等式，则 k 单调递增。

Theorem

$f_i = \min_{j<i}\{w(j,i)\}$
$k_i = \min\{j \mid f_i = w(j,i), j < i\}$
若 w 满足四边形不等式，则 k 单调递增。

此时可用分治，时间复杂度 $O(n\log n)$。

考虑分治区间 $[l, r]$ 的过程。

此时已知 $kl \le k_l \le \cdots \le k_r \le kr$，将此过程记为 $\text{solve}(l, r, kl, kr)$

若 $r < l$ 退出。

- 令 $mid = \lfloor (l + r)/2 \rfloor$。
- $O(r - l)$ 枚举求出 k_{mid} 和 f_{mid}。
- $\text{solve}(l, mid - 1, kl, k_{mid})$。

考虑分治区间 $[l, r]$ 的过程。

此时已知 $kl \leq k_l \leq \cdots \leq k_r \leq kr$，将此过程记为 solve$(l, r, kl, kr)$

若 $r < l$ 退出。

- 令 $mid = \lfloor (l + r)/2 \rfloor$。
- $O(r - l)$ 枚举求出 k_{mid} 和 f_{mid}。
- solve$(l, mid - 1, kl, k_{mid})$。
- solve$(mid + 1, r, k_{mid}, kr)$。

考虑分治区间 $[l, r]$ 的过程。

此时已知 $kl \le k_l \le \cdots \le k_r \le kr$，将此过程记为 solve$(l, r, kl, kr)$

若 $r < l$ 退出。

- 令 $mid = \lfloor (l + r)/2 \rfloor$。
- $O(r - l)$ 枚举求出 k_{mid} 和 f_{mid}。
- solve$(l, mid - 1, kl, k_{mid})$。
- solve$(mid + 1, r, k_{mid}, kr)$。

初始时调用 solve$(0, n, 0, n)$。

由于递归 $O(\log n)$ 层，每层枚举的长度之和为 $O(n)$，所以时间复杂度为 $O(n \log n)$。

Theorem

$f_{u,i} = \min_{j<i} \{f_{u-1,j} + w(j, i)\}$
$k_{u,i} = \min \{j \mid f_{u,i} = f_{u-1,j} + w(j, i), j < i\}$
若 w 满足四边形不等式，则 $\forall u, i < j, k_{u,i} \le k_{u,j}$。

Theorem

$f_{u,i} = \min_{j<i} \{f_{u-1,j} + w(j,i)\}$
$k_{u,i} = \min \{j \mid f_{u,i} = f_{u-1,j} + w(j,i), j < i\}$
若 w 满足四边形不等式，则 $\forall u, i < j, k_{u,i} \leq k_{u,j}$。

对每层用分治，时间复杂度 $O(mn\log n) \quad (u \leq m, i \leq n)$。

Theorem

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

$k_{u,i} = \min \{j \mid f_{u,i} = f_{u-1,j} + w(j,i), j < i\}$

若 w 满足四边形不等式，则 $\forall u, i, k_{u,i} \le k_{u+1,i}$。

Theorem

$f_{u,i} = \min_{j<i} \{f_{u-1,j} + w(j, i)\}$
$k_{u,i} = \min \{j \mid f_{u,i} = f_{u-1,j} + w(j, i), j < i\}$
若 w 满足四边形不等式，则 $\forall u, i, k_{u,i} \le k_{u+1,i}$。

结合前述定理，有 $k_{u-1,i} \le k_{u,i} \le k_{u,i+1}$
按照 u 从小到大，i 从大到小的顺序 DP，可以 $O((m+n)n)$。

Example 10 (Codeforces 321E (*2600))

将长度为 n 的序列划分为 k 个非空连续段。
若 $i<j$ 在同一段，产生花费 $c_{i,j}$。
最小化总花费。
$1 \le n \le 4000, 1 \le k \le \min\{n, 800\}$

Theorem

$f_{i,j} = \min_{i \le k < j} \{f_{i,k} + f_{k+1,j}\} + w(i,j)$。
$k_{i,j} = \min\{k \mid f_{i,j} = f_{i,k} + f_{k+1,j} + w(i,j), i \le k < j\}$。
若 w 满足区间包含单调性和四边形不等式，则 $\forall i,j, k_{i,j-1} \le k_{i,j} \le k_{i+1,j}$。

按照区间长度从小到大的顺序 DP，每个长度最多枚举 $O(n)$ 个转移，所以时间复杂度为 $O(n^2)$。

有凸函数 f，需要求 $f(m)$，可以通过二分 m 处斜率求 $f(m)$。

有凸函数 f，需要求 $f(m)$，可以通过二分 m 处斜率求 $f(m)$。

二分斜率是否大于 k 时，改为最大化 $g(x) = f(x) - kx$，通过判断取到最大值的点与 m 的大小，判断 k 与斜率的大小。

有凸函数 f，需要求 $f(m)$，可以通过二分 m 处斜率求 $f(m)$。

二分斜率是否大于 k 时，改为最大化 $g(x)=f(x)-kx$，通过判断取到最大值的点与 m 的大小，判断 k 与斜率的大小。

例如，要求在 n 个元素中选 m 个，可以把选 x 个的答案看成函数 $f(x)$，求 $f(m)$。

通过给选一个元素加上 k 的代价，改为最大化 $g(x)=f(x)-kx$。

有凸函数 f，需要求 $f(m)$，可以通过二分 m 处斜率求 $f(m)$。

二分斜率是否大于 k 时，改为最大化 $g(x) = f(x) - kx$，通过判断取到最大值的点与 m 的大小，判断 k 与斜率的大小。

例如，要求在 n 个元素中选 m 个，可以把选 x 个的答案看成函数 $f(x)$，求 $f(m)$。

通过给选一个元素加上 k 的代价，改为最大化 $g(x) = f(x) - kx$。

注意由于可能有三点共线，最终求出的斜率对应的函数最大值最大值不一定在 m 取到。

Example 11 (Codeforces 739E (*3000))

有 n 个 *Pokemon*，a 个 *Poke Ball*，b 个 *Ultra Ball*。

对于每个 *Pokemon*，你可以使用 *Poke Ball* 和 *Ultra Ball*，但是最多分别用一个。

对第 i 个 *Pokemon*，使用 *Poke Ball* 有 p_i 的概率成功获得它，使用 *Ultra Ball* 有 u_i 的概率成功获得，一起使用有 $p_i + u_i - p_i u_i$ 的概率成功获得。

最大化期望获得的 *Pokemon*。你的方案必须在开始前确定（即后续过程不依赖你前面是否成功获得某个 *Pokemon*）。

$2 \le n \le 2000, 0 \le a, b \le n, 0 \le p_i, u_i \le 1$，允许误差 10^{-4}。

Theorem

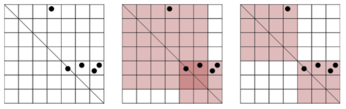
将一个长为 n 的序列划分为 k 个连续段，每段 $[l,r)$ 的花费为 $w(l,r)$，最小化总花费，记答案为 $f(k)$。

若 w 满足四边形不等式，则 $f(k)$ 是凸函数。

Example 12 (UOJ 240. (IOI2016) aliens (~3100))

给定一个 $n \times n$ 的网格上的 m 个点，每个点位于一个格子内部。

用最多 k 个对角线与 $n \times n$ 网格的主对角线相同的正方形覆盖这些点，最小化这些正方形的面积并。



注意：尽管小方格 $(4,6)$ 上包含 2 个点，但该小方格只需要被覆盖一次就足够。样例覆盖方法如上图所示：左边的图表示这个样例中对应的方格，中间的图表示一个次优解，它总共覆盖了 41 个小方格。而右边的图则表示最优解，它总共覆盖了 25 个小方格。

wqs 二分在国外叫 Alien Trick。

Example 6 可以用 wqs 二分做到 $O(n\log V)$。
Example 10 可以用 wqs 二分做到 $O(n\log n\log V)$。

适用范围：在 n 个元素中选 k 个元素，元素顺序没有影响，且 DP 时需要记录已选元素个数。

将 n 个元素 shuffle，有很大概率最优解在前 i 个元素中选的个数与 $\frac{ik}{n}$ 之差不超过 $O(\sqrt{n})$。

Example 11，$O(n \times \sqrt{n} \times \sqrt{n}) = O(n^2)$。

Example 13 (Codeforces 626F (*2400))

有一个长为 n 的序列 a，要求把 a 分为若干组，使得每组的 $\max - \min$ 之和不超过 k。

求分组方案数 $\bmod\ (10^9 + 7)$。

$1 \le n \le 200, 1 \le k \le 1000, 1 \le a_i \le 500$

Example 14 (Codeforces 367E (*2700))

有 n 个区间，你需要为每个区间分配左右端点，端点属于 $[1, m] \cap \mathbb{N}$。

你需要保证区间两两互不包含且至少存在一个区间的左端点等于 x，输出方案数对 $10^9 + 7$ 取模的结果，区间有标号。

$1 \le nm \le 10^5, 1 \le x \le m$。

Example 15 (LOJ 2743. (JOI Open 2016) Skyscraper(~2400))

将互不相同的 n 个整数 $a_1, a_2, \cdots, a_n$ 按照一定顺序排列。

假设排列为 $p_1, p_2, \cdots, p_n$，要求：

$|p_1 - p_2| + |p_2 - p_3| + \cdots + |p_{n-1} - p_n| \le L$。

求满足题意的排列的方案数 $\bmod (10^9 + 7)$。

$1 \le n \le 100, 1 \le L, a_i \le 1000$

AtCoder Regular Contest 148 Task E 可以用“连通块” DP 的技巧解决。

官方题解很神奇

一般子集和问题

给定 n 个正整数 $a_1, \cdots, a_n$，令 $m = \sum a_i$。

对于 $0 \le k \le m$，求是否存在 $S \subseteq [1, n] \cap \mathbb{N}$，使得 $\sum_{i \in S} a_i = k$。

一般子集和问题

给定 n 个正整数 $a_1, \cdots, a_n$，令 $m = \sum a_i$。

对于 $0 \le k \le m$，求是否存在 $S \subseteq [1, n] \cap \mathbb{N}$，使得 $\sum_{i \in S} a_i = k$。

元素值域较小的子集和问题

给定 n 个正整数 $a_1, \cdots, a_n$，$a_i \le d$，令 $m = \sum a_i$。

给定一个 $0 \le c \le m$，求是否存在 $S \subseteq [1, n] \cap \mathbb{N}$，使得 $\sum_{i \in S} a_i = c$。

元素值域较小的多重子集和问题

给定 n 个正整数 $a_1, \cdots, a_n$，$a_i \leq d$。
给定一个 $c \geq 0$，求是否存在 $\lambda_i \geq 0$，使得 $\sum_{i=1}^{n} \lambda_i a_i = c$。

可以 $O(d^2 \log c/w)$。
$O(d^2 \log c)$ 可以推广到带权的背包。

mod m 意义下的子集和问题

- $O(m\log m)$ 做法，常数大，~~$m=10^6$ 还跑不过 $O(m\log^2 m)$~~

mod m 意义下的子集和问题

- $O(m\log m)$ 做法，常数大，~~$m=10^6$ 还跑不过 $O(m\log^2 m)$~~
- $O(m\log^2 m)$ 做法：树状数组维护 `01` 串的哈希。

mod m 意义下的子集和问题

- $O(m\log m)$ 做法，常数大，~~$m = 10^6$ 还跑不过 $O(m\log^2 m)$~~
- $O(m\log^2 m)$ 做法：树状数组维护 01 串的哈希。

  UOJ 771

  SOJ 1643

Problem 1 (Codeforces 713C 加强)

给定长为 n 的序列 a，求一个严格递增的序列 b，最小化 $\sum_{i=1}^{n} \mid a_i - b_i \mid$。

$1 \le n \le 5 \cdot 10^5, 1 \le a_i \le 10^9$

Problem 2 (LOJ 2542. (PKUWC2018) 随机游走)

给定一棵 n 个结点的树，你从点 x 出发，每次等概率随机选择一条与所在点相邻的边走过去。

有 Q 次询问，每次询问给定一个集合 S，求如果从 x 出发一直随机游走，直到点集 S 中所有点都至少经过一次的话，期望游走几步。

特别地，点 x（即起点）视为一开始就被经过了一次。

答案对 998244353 取模。

Problem 3 (Codeforces 930E)

有长为 k 的 01 串，有 $n+m$ 个限制条件。

- 前 n 个要求区间 $[l_i, r_i]$ 上至少有一个 1。
- 后 m 个要求区间 $[l_i, r_i]$ 上至少有一个 0。

Problem 4 (Codeforces 1580D)

给定一个长为 n 的序列 a，求一个长为 m 的子序列 $a_{b_1}, a_{b_2}, \cdots, a_{b_m}$ $(1 \le b_1 < b_2 < \cdots < b_m)$。

最大化 $m \cdot \sum_{i=1}^{m} a_{b_i} - \sum_{i=1}^{m} \sum_{j=1}^{m} f(\min(b_i, b_j), \max(b_i, b_j))$。

$1 \le m \le n \le 4000, 1 \le a_i < 2^{31}$

Problem 5 (Codeforces 809D)

给定 n 个区间 $[l_i, r_i]$，有序列 $a_i \in [l_i, r_i]$，求 a_i 最长严格上升子序列最长能是多少。

$1 \leq n \leq 3 \cdot 10^5, 1 \leq l_i \leq r_i \leq 10^9$

Problem 6 (Codeforces 500F)

有 n 种商品，第 i 种商品的价格是 c_i，购买后可以增加 h_i 的快乐指数，将于第 t_i 天上市。商品的保质期为 p 天，过期后不能再购买，即第 i 种商品只能在第 t_i 天到第 t_i+p-1 天之间购买，每种商品只能购买一次。

有 q 个询问，每次给定两个整数 a, b，求在第 a 天去购物，最多使用 b 元的情况下可以得到的最大快乐指数。询问之间互不干扰。

$1 \le n, c_i, h_i, b \le 4000, 1 \le p, t_i \le 10000, 1 \le q \le 20000, 1 \le a \le 20000$

Problem 7 (Codeforces 1421E)

有一个长度为 n 的序列，你要进行恰好 $n-1$ 次操作，每次操作你需要选择连续的两个数 a_i, a_{i+1}，然后从序列中删掉这两个数，并将 $-(a_i+a_{i+1})$ 插回到原位置。你需要让所有操作结束后序列剩下的唯一一个数最大，输出这个值。

$1 \le n \le 2 \cdot 10^5, -10^9 \le a_i \le 10^9$

Problem 8 (Codeforces 1146G)

你要在一条线上建 n 个房子，每个房子高度 $\in [0,h]$。

建一个高度为 a 的房子，能带来 a^2 的收益。

有 m 个限制，第 i 个限制规定 $[l_i,r_i]$ 中的房子高度不能超过 x_i，若违反，需要罚款 c_i（一个限制中，有多个房子违反只需要罚款一次）。

求最大收益。

Problem 9 (Codeforces 578D)

给定一个字符串 S，求有多少个与 S 等长字符串 T，满足 $\mathrm{LCS}(S,T) = |S| - 1$，其中 S, T 只包含前 m 个小写字母。

LCS 表示最长公共子序列。

$1 \le n = | S | \le 10^5, 2 \le m \le 26$

Problem 10 (Codeforces 494C)

有一个长为 n 的数列，初始时为 $a_1, a_2, \cdots, a_n$。

给你 q 个操作，第 i 个操作将 $[l_i, r_i]$ 内的数全部加一，有 p_i 的概率被执行。保证区间不会交错，即：
$\forall i, j \in [1, q], l_i \le r_i < l_j \le r_j$ 或 $l_i \le l_j \le r_j \le r_i$ 或 $l_j \le r_j < l_i \le r_i$ 或 $l_j \le l_i \le r_i \le r_j$ 。

求操作完成后数列的最大值的期望。

$1 \le n \le 10^5, 1 \le q \le 5000, 0 \le a_i \le 10^9, 1 \le l_i \le r_i \le n, 0 \le p_i \le 1$

Problem 11 (Codeforces 627D)

给出一棵 n 个点的无根树，每个节点有一个权值 a_i。

你可以指定树根和访问顺序，使得 *dfs* 序中前 k 个结点的权值最小值最大，求出这个值。

$2 \le n \le 2 \cdot 10^5, 1 \le k \le n, 1 \le a_i \le 10^6$

Problem 12 (Codeforces 1179D)

给一棵 n 个点的树，求加入一条边（可以形成重边）后，最多有多少条无向简单路径。

简单路径定义为任意一个点最多经过一次的路径。两条路径相同当且仅当经过的边集相同。

$2 \leq n \leq 5 \cdot 10^5$

Problem 13 (Codeforces 1750F)

对一个 01 串 s，你可以对该串进行以下操作若干次：

选择一个子串 $[l,r]$，满足 $s_l=s_r=1$，$\sum_{i=l}^{r}[s_i=1]\geq\sum_{i=l}^{r}[s_i=0]$，将 $[l,r]$ 内的所有字符均替换为 1。

定义一个串是好的，当且仅当该串可以通过若干次操作，使得 s 全为 1。

求长度为 n 的好的串 s 的个数，对 m 取模。

$1\leq n\leq 5000, 10\leq m\leq 10^9$，不保证 m 是素数。

Problem 14 (SOJ 1697)

在一条直线上有 n 个粒子，初始分别在 $1, 2, \cdots, n$ 处，在 1 和 n 处各有一堵墙。每个粒子初始时向左或向右（给定）以 1 的速度运动。

粒子碰到墙时会立即改为向相反方向运动。两个粒子相遇时，有且只有一个粒子会消失，有 p 的概率左边的粒子会消失，有 $1-p$ 的概率右边的粒子会消失。

试求最后 n 号粒子存活下来的概率模 998244353。

$2 \leq n \leq 10^3$

Problem 15 (SOJ 1669)

你有一个长度为 n 的数组 a_i，你还有 q 个区间 $[l_i, r_i]$，还有一个整数 m。

你可以将至多 m 个 a_i 变为 0，你想要最小化 $\sum_{i=1}^{q} \max_{j=l_i}^{r_i} a_j$，输出这个最小值。

$1 \le n, q \le 50, 0 \le m \le n, 1 \le a_i \le 10^9, 1 \le l_i \le r_i \le n$

Problem 16 (SOJ 1618)

给出 n 个点，以及任意两个点 i, j 之间存在一条无向边的概率 $p_{i,j}$，求图中联通块个数的期望。

不要使用除法。

$1 \leq n \leq 18$

Problem 17 (UOJ 770. (UER #11) 切割冰片)

为了切割冰片，跳蚤国王放置了 $n+m$ 个激光发射器：

- 其中有 m 个竖直向上的激光发射器，第 i 个竖直向上的激光发射器位于 $(i,0)$，所发出的激光的强度为 10^{100}；
- 其中有 n 个水平向右的激光发射器，第 i 个水平向右的激光发射器位于 $(0,i)$，所发出的激光的强度为 l_i。

激光发射器所发射的激光在空气中几乎无法传播，但在冰中可以正常穿行。

最开始，所有光束发射器都是关闭的。操作员需要按照某种顺序将它们依次激活。当一个发射器被激活时，它会瞬间向它所对应的方向发出一束激光。该激光会不断沿该射线方向延伸，直到触碰到另一束激光已经融化过的点（即接触到空气），或延伸距离到达了这个激光发射器所发出的的激光的强度。该激光所经过的路径上的冰会同时融化。

两种切割结果是不同的，当且仅当冰片上被融化的点所组成的集合不相同。计算出本质不同的方案数。答案对 998244353 取模。

$1 \le n \le 500, 1 \le m, l_i \le 10^9$

Problem 18 (UOJ 757. (IOI2022) 鲶鱼塘)

0-indexed

有一个 $n \times n$ 的网格，有 m 个特殊格。

定义 (i,j) 格在第 i 列。

对于第 i 列，选择一个前缀 $(i,0),\cdots,(i,p_i-1)$ 覆盖（可以为空）。

若一个特殊格未被覆盖，且其左右有一个被覆盖，则获得 w_i 的收益。换句话说，对于一个特殊格 (x_i,y_i)，若 $y_i \ge p_{x_i}$ 且 $y_i < \max\{p_{x_i-1},p_{x_i+1}\}$，则获得 w_i 的收益。

最大化收益。

$2 \le n \le 10^5, 1 \le m \le 3 \cdot 10^5, 0 \le x_i, y_i \le n-1, 1 \le w_i \le 10^9, \forall i < j, (x_i,y_i) \neq (x_j,y_j)$

Problem 19 (QOJ 4812. Counting Sequence)

给定 n 和 c。

一个序列 $a_1, a_2, \cdots, a_m$ 是好的当且仅当：

- $a_i > 0$。
- $\mid a_{i+1} - a_i \mid = 1$。
- $\sum_{i=1}^{m} a_i = n$。

对于一个序列定义 $f(a) = \sum_{i=1}^{m-1} [a_i > a_{i+1}]$

定义其权值为 $c^{f(a)}$。

求所有好的序列的权值和。对 998244353 取模。

$1 \leq n \leq 3 \cdot 10^5, 0 \leq c < 998244353$