NOI 2025 Day1 B. 序列变换

ForgotMe/Para

2025 年 7 月 14 日

问题 (序列变换)

- 给定 $B=[b_1,\ldots,b_n]$，$C=[c_1,\ldots,c_n]$。对于 $D=[d_1,\ldots,d_n]$，设 $S(D)$ 为所有满足 $d_i=0$ 的下标 i 的集合，定义 $f(D)=\sum_{i\in S(D)} b_i$，$g(D)=\prod_{i\in S(D)} c_i$。
- 对于一个给定的正整数序列 $A=[a_1,\ldots,a_n]$，做任意次如下修改：选择两个相邻的元素，同时减去两者的较小值。
- 记所有可以得到的序列形成的集合为 $T(A)$，求 $\max_{D\in T(A)} f(D)$ 以及 $\left(\sum_{D\in T(A)} g(D)\right) \bmod 10^9+7$。

数据范围：$n\le 5000$，$\sum n\le 4\times 10^4$。

- 声明：本题数据并不好造，可能存在部分解法冲过了更多的分数，出题人对能想到的 $O(n^3)$ 的做法以及剪枝进行了针对性构造，但是由于 $O(n^3)$ 做法的常数过小，所以只能通过开大数据范围来加以区分，这也是 $\sum n$ 开的比较大的原因。实际上出题人以及大部分验题人的程序只跑了 0.4s，但是有验题人的程序跑了 3.7s，所以最后为了避免卡常开了 6s。

$n \leq 8$

$n \leq 8$

- 注意到两个相邻的位置最多操作一次，写一个 $O(n!)$ 的全排列枚举即可。期望得分：10。

- 注意到一次操作后一定会有一个数变成 0，接下来若想继续进行操作，只能朝着不是 0 的数的方向进行操作。

- 注意到一次操作后一定会有一个数变成 0，接下来若想继续进行操作，只能朝着不是 0 的数的方向进行操作。
- 形式化地，所有序列可以通过如下方式操作出来：
  1. 把序列划分为若干段 $[l_i, r_i]$，操作只会在段内进行。
  2. 每个段 $[l_i, r_i]$ 选择一个位置 $k_i \in [l_i, r_i]$。
  3. 对于每段，依次操作 $(l, l+1), \ldots, (k-1, k)$ 和 $(r, r-1), \ldots, (k+1, k)$。
  4. 同时需要保证所有以上操作都是合法的。

- 注意到一次操作后一定会有一个数变成 0，接下来若想继续进行操作，只能朝着不是 0 的数的方向进行操作。
- 形式化地，所有序列可以通过如下方式操作出来：
  1. 把序列划分为若干段 $[l_i, r_i]$，操作只会在段内进行。
  2. 每个段 $[l_i, r_i]$ 选择一个位置 $k_i \in [l_i, r_i]$。
  3. 对于每段，依次操作 $(l, l+1), \ldots, (k-1, k)$ 和 $(r, r-1), \ldots, (k+1, k)$。
  4. 同时需要保证所有以上操作都是合法的。
- 显然地，最后每段只有 k_i 处是可能有值的。

- 考虑进一步观察，发现在合法的前提下，在同一段中，选择任意的 k_i 操作后剩下的值都是一样的，也就是 a_{k_i} 一定会是 $\left|\sum_{j=l_i}^{r_i} (-1)^j a_j\right|$。

- 考虑进一步观察，发现在合法的前提下，在同一段中，选择任意的 k_i 操作后剩下的值都是一样的，也就是 a_{k_i} 一定会是 $\left|\sum_{j=l_i}^{r_i}(-1)^j a_j\right|$。
- 以下记 $w(l,r)=\left|\sum_{j=l}^{r}(-1)^j a_j\right|$。

- 考虑进一步观察，发现在合法的前提下，在同一段中，选择任意的 k_i 操作后剩下的值都是一样的，也就是 a_{k_i} 一定会是 $\left|\sum_{j=l_i}^{r_i}(-1)^j a_j\right|$。
- 以下记 $w(l,r)=\left|\sum_{j=l}^{r}(-1)^j a_j\right|$。
- 对于第一问，设 dp_i 表示划分了区间 $[1,i]$ 所得到的 $f(D)$ 最大值。

- 考虑进一步观察，发现在合法的前提下，在同一段中，选择任意的 k_i 操作后剩下的值都是一样的，也就是 a_{k_i} 一定会是 $\left|\sum_{j=l_i}^{r_i}(-1)^j a_j\right|$。
- 以下记 $w(l,r)=\left|\sum_{j=l}^{r}(-1)^j a_j\right|$。
- 对于第一问，设 dp_i 表示划分了区间 $[1,i]$ 所得到的 $f(D)$ 最大值。
- 转移时枚举下一个划分点以及 k_i，就可以做到 $O(n^3)$，期望得分：30。

- 考虑如何省去枚举 k_i 这一步。设 L_i 表示从 i 开始一直向左操作且操作合法的最远位置，R_i 表示从 i 开始一直向右操作且操作合法的最远位置，那么合法的 k_i 一定在 $[L_{r_i}, R_{l_i}]$ 内，且奇偶性确定。

- 考虑如何省去枚举 k_i 这一步。设 L_i 表示从 i 开始一直向左操作且操作合法的最远位置，R_i 表示从 i 开始一直向右操作且操作合法的最远位置，那么合法的 k_i 一定在 $[L_{r_i}, R_{l_i}]$ 内，且奇偶性确定。
- 预处理一个区间最值即可做到 $O(n^2)$。

- 考虑如何省去枚举 k_i 这一步。设 L_i 表示从 i 开始一直向左操作且操作合法的最远位置，R_i 表示从 i 开始一直向右操作且操作合法的最远位置，那么合法的 k_i 一定在 $[L_{r_i}, R_{l_i}]$ 内，且奇偶性确定。
- 预处理一个区间最值即可做到 $O(n^2)$。
- 结合 $n \le 8$ 与特殊性质 A 后，期望得分：52。

- 对于第二问，直接按照上面的方式动态规划只能通过特殊性质 B。

- 对于第二问，直接按照上面的方式动态规划只能通过特殊性质 B。
- 这是因为特殊性质 B 保证了 $w(l_i, r_i)$ 一定不会是 0。

- 对于第二问，直接按照上面的方式动态规划只能通过特殊性质 B。
- 这是因为特殊性质 B 保证了 $w(l_i, r_i)$ 一定不会是 0。
- 也即，没有特殊性质 B 的情况会因为最后操作出了 0 而算重：

- 对于第二问，直接按照上面的方式动态规划只能通过特殊性质 B。
- 这是因为特殊性质 B 保证了 $w(l_i, r_i)$ 一定不会是 0。
- 也即，没有特殊性质 B 的情况会因为最后操作出了 0 而算重：
  1. 对于一段 $[l_i, r_i]$，若 $w(l_i, r_i) = 0$，则此时选择不同 k_i 会得到相同的序列；

- 对于第二问，直接按照上面的方式动态规划只能通过特殊性质 B。
- 这是因为特殊性质 B 保证了 $w(l_i, r_i)$ 一定不会是 0。
- 也即，没有特殊性质 B 的情况会因为最后操作出了 0 而算重：
  1. 对于一段 $[l_i, r_i]$，若 $w(l_i, r_i) = 0$，则此时选择不同 k_i 会得到相同的序列；
  2. 对于相邻的两段 $[l_i, r_i], [l_{i+1}, r_{i+1}]$，若 $w(l_i, r_i) = w(l_{i+1}, r_{i+1}) = 0$，则此时分别操作 $[l_i, r_i], [l_{i+1}, r_{i+1}]$ 与直接操作 $[l_i, r_{i+1}]$ 会得到相同的序列。

- 考虑对 L_i, R_i, k_i 做一些更紧的限制：

- 考虑对 L_i, R_i, k_i 做一些更紧的限制：
  - 设 L_i 表示从 i 开始一直向左操作 $(x, x-1)$ 直到操作不合法或出现了 $a_x = 0$ 时的 x，R_i 同理；

- 考虑对 L_i, R_i, k_i 做一些更紧的限制：
  - 设 L_i 表示从 i 开始一直向左操作 $(x, x-1)$ 直到操作不合法或出现了 $a_x = 0$ 时的 x，R_i 同理；
  - 对于 k_i，若 $w(l_i, r_i) = 0$，则 $k_i = \max(l_i, L_{r_i})$。

- 考虑对 L_i, R_i, k_i 做一些更紧的限制：
  - 设 L_i 表示从 i 开始一直向左操作 $(x, x-1)$ 直到操作不合法或出现了 $a_x = 0$ 时的 x，R_i 同理；
  - 对于 k_i，若 $w(l_i, r_i) = 0$，则 $k_i = \max(l_i, L_{r_i})$。
- 引理：在加入上述限制后，可操作出的序列对应的 $\{(l_i, r_i, k_i)\}$ 是唯一的。

- 证明：考虑反证。假设存在两个不同的 (l_i, r_i, k_i) 形成的操作序列，使得操作出来的序列相同。那么只需要证明它们的第一个三元组 (l, r, k) 相同，剩余部分对长度归纳即可。

- 证明：考虑反证。假设存在两个不同的 (l_i, r_i, k_i) 形成的操作序列，使得操作出来的序列相同。那么只需要证明它们的第一个三元组 (l, r, k) 相同，剩余部分对长度归纳即可。
- 设两个操作序列的对应第一个三元组分别为 $(l_1, r_1, k_1), (l_2, r_2, k_2)$。

- 证明：考虑反证。假设存在两个不同的 (l_i, r_i, k_i) 形成的操作序列，使得操作出来的序列相同。那么只需要证明它们的第一个三元组 (l, r, k) 相同，剩余部分对长度归纳即可。
- 设两个操作序列的对应第一个三元组分别为 $(l_1, r_1, k_1), (l_2, r_2, k_2)$。
- 首先证明 $l_1 = l_2$。

- 证明：考虑反证。假设存在两个不同的 (l_i, r_i, k_i) 形成的操作序列，使得操作出来的序列相同。那么只需要证明它们的第一个三元组 (l, r, k) 相同，剩余部分对长度归纳即可。
- 设两个操作序列的对应第一个三元组分别为 $(l_1, r_1, k_1), (l_2, r_2, k_2)$。
- 首先证明 $l_1 = l_2$。
    - 如果 $l_1 < l_2$，那么 k_1 必须是 l_1，否则在最初为正整数序列的前提下，l_1 这个位置一定不会相同。此外必须有 $l_2 = l_1 + 1$，否则 $l_1 + 1$ 这个位置将不合法。

- 证明：考虑反证。假设存在两个不同的 (l_i, r_i, k_i) 形成的操作序列，使得操作出来的序列相同。那么只需要证明它们的第一个三元组 (l, r, k) 相同，剩余部分对长度归纳即可。
- 设两个操作序列的对应第一个三元组分别为 $(l_1, r_1, k_1), (l_2, r_2, k_2)$。
- 首先证明 $l_1 = l_2$。
  - 如果 $l_1 < l_2$，那么 k_1 必须是 l_1，否则在最初为正整数序列的前提下，l_1 这个位置一定不会相同。此外必须有 $l_2 = l_1 + 1$，否则 $l_1 + 1$ 这个位置将不合法。
  - 由于 (l_1, r_1, l_1) 操作后 a_{l_1} 不能改变，那么 $w(l_1 + 1, r_1) = 0$，因此 $L_{r_i} \geq l_i + 1$，那么 k_1 就不能是 l_1，矛盾。$l_1 > l_2$ 同理。故 $l_1 = l_2$。

- 证明：考虑反证。假设存在两个不同的 (l_i, r_i, k_i) 形成的操作序列，使得操作出来的序列相同。那么只需要证明它们的第一个三元组 (l, r, k) 相同，剩余部分对长度归纳即可。
- 设两个操作序列的对应第一个三元组分别为 $(l_1, r_1, k_1), (l_2, r_2, k_2)$。
- 首先证明 $l_1 = l_2$。
  - 如果 $l_1 < l_2$，那么 k_1 必须是 l_1，否则在最初为正整数序列的前提下，l_1 这个位置一定不会相同。此外必须有 $l_2 = l_1 + 1$，否则 $l_1 + 1$ 这个位置将不合法。
  - 由于 (l_1, r_1, l_1) 操作后 a_{l_1} 不能改变，那么 $w(l_1 + 1, r_1) = 0$，因此 $L_{r_i} \geq l_i + 1$，那么 k_1 就不能是 l_1，矛盾。$l_1 > l_2$ 同理。故 $l_1 = l_2$。
- 其次证明 $r_1 = r_2$。

- 证明：考虑反证。假设存在两个不同的 (l_i, r_i, k_i) 形成的操作序列，使得操作出来的序列相同。那么只需要证明它们的第一个三元组 (l, r, k) 相同，剩余部分对长度归纳即可。
- 设两个操作序列的对应第一个三元组分别为 $(l_1, r_1, k_1), (l_2, r_2, k_2)$。
- 首先证明 $l_1 = l_2$。
  - 如果 $l_1 < l_2$，那么 k_1 必须是 l_1，否则在最初为正整数序列的前提下，l_1 这个位置一定不会相同。此外必须有 $l_2 = l_1 + 1$，否则 $l_1 + 1$ 这个位置将不合法。
  - 由于 (l_1, r_1, l_1) 操作后 a_{l_1} 不能改变，那么 $w(l_1 + 1, r_1) = 0$，因此 $L_{r_i} \geq l_i + 1$，那么 k_1 就不能是 l_1，矛盾。$l_1 > l_2$ 同理。故 $l_1 = l_2$。
- 其次证明 $r_1 = r_2$。
  - 如果 $r_1 < r_2$，那么一定有 $w(r_1 + 1, r_2) = 0$，则 $k_2 \geq L_{r_2} \geq r_1 + 1$。

- 证明：考虑反证。假设存在两个不同的 (l_i, r_i, k_i) 形成的操作序列，使得操作出来的序列相同。那么只需要证明它们的第一个三元组 (l, r, k) 相同，剩余部分对长度归纳即可。
- 设两个操作序列的对应第一个三元组分别为 $(l_1, r_1, k_1), (l_2, r_2, k_2)$。
- 首先证明 $l_1 = l_2$。
  - 如果 $l_1 < l_2$，那么 k_1 必须是 l_1，否则在最初为正整数序列的前提下，l_1 这个位置一定不会相同。此外必须有 $l_2 = l_1 + 1$，否则 $l_1 + 1$ 这个位置将不合法。
  - 由于 (l_1, r_1, l_1) 操作后 a_{l_1} 不能改变，那么 $w(l_1 + 1, r_1) = 0$，因此 $L_{r_i} \geq l_i + 1$，那么 k_1 就不能是 l_1，矛盾。$l_1 > l_2$ 同理。故 $l_1 = l_2$。
- 其次证明 $r_1 = r_2$。
  - 如果 $r_1 < r_2$，那么一定有 $w(r_1 + 1, r_2) = 0$，则 $k_2 \geq L_{r_2} \geq r_1 + 1$。
    - 若 $w(l_1, r_1) = 0$，那么 $k_2 \leq R_{l_2} = R_{l_1} \leq r_1$；

- 证明：考虑反证。假设存在两个不同的 (l_i, r_i, k_i) 形成的操作序列，使得操作出来的序列相同。那么只需要证明它们的第一个三元组 (l, r, k) 相同，剩余部分对长度归纳即可。
- 设两个操作序列的对应第一个三元组分别为 $(l_1, r_1, k_1), (l_2, r_2, k_2)$。
- 首先证明 $l_1 = l_2$。
  - 如果 $l_1 < l_2$，那么 k_1 必须是 l_1，否则在最初为正整数序列的前提下，l_1 这个位置一定不会相同。此外必须有 $l_2 = l_1 + 1$，否则 $l_1 + 1$ 这个位置将不合法。
  - 由于 (l_1, r_1, l_1) 操作后 a_{l_1} 不能改变，那么 $w(l_1 + 1, r_1) = 0$，因此 $L_{r_i} \geq l_i + 1$，那么 k_1 就不能是 l_1，矛盾。$l_1 > l_2$ 同理。故 $l_1 = l_2$。
- 其次证明 $r_1 = r_2$。
  - 如果 $r_1 < r_2$，那么一定有 $w(r_1 + 1, r_2) = 0$，则 $k_2 \geq L_{r_2} \geq r_1 + 1$。
    - 若 $w(l_1, r_1) = 0$，那么 $k_2 \leq R_{l_2} = R_{l_1} \leq r_1$；
    - 若 $w(l_1, r_1) \neq 0$，那么 $k_2 = k_1 \leq r_1$。

- 证明：考虑反证。假设存在两个不同的 (l_i, r_i, k_i) 形成的操作序列，使得操作出来的序列相同。那么只需要证明它们的第一个三元组 (l, r, k) 相同，剩余部分对长度归纳即可。
- 设两个操作序列的对应第一个三元组分别为 $(l_1, r_1, k_1), (l_2, r_2, k_2)$。
- 首先证明 $l_1 = l_2$。
  - 如果 $l_1 < l_2$，那么 k_1 必须是 l_1，否则在最初为正整数序列的前提下，l_1 这个位置一定不会相同。此外必须有 $l_2 = l_1 + 1$，否则 $l_1 + 1$ 这个位置将不合法。
  - 由于 (l_1, r_1, l_1) 操作后 a_{l_1} 不能改变，那么 $w(l_1 + 1, r_1) = 0$，因此 $L_{r_i} \geq l_i + 1$，那么 k_1 就不能是 l_1，矛盾。$l_1 > l_2$ 同理。故 $l_1 = l_2$。
- 其次证明 $r_1 = r_2$。
  - 如果 $r_1 < r_2$，那么一定有 $w(r_1 + 1, r_2) = 0$，则 $k_2 \geq L_{r_2} \geq r_1 + 1$。
    - 若 $w(l_1, r_1) = 0$，那么 $k_2 \leq R_{l_2} = R_{l_1} \leq r_1$；
    - 若 $w(l_1, r_1) \neq 0$，那么 $k_2 = k_1 \leq r_1$。
  - 因此 $k_2 \leq r_1$，与 $k_2 \geq r_1 + 1$ 矛盾。$r_1 > r_2$ 同理。故 $r_1 = r_2$。

- 证明：考虑反证。假设存在两个不同的 (l_i, r_i, k_i) 形成的操作序列，使得操作出来的序列相同。那么只需要证明它们的第一个三元组 (l, r, k) 相同，剩余部分对长度归纳即可。
- 设两个操作序列的对应第一个三元组分别为 $(l_1, r_1, k_1), (l_2, r_2, k_2)$。
- 首先证明 $l_1 = l_2$。
  - 如果 $l_1 < l_2$，那么 k_1 必须是 l_1，否则在最初为正整数序列的前提下，l_1 这个位置一定不会相同。此外必须有 $l_2 = l_1 + 1$，否则 $l_1 + 1$ 这个位置将不合法。
  - 由于 (l_1, r_1, l_1) 操作后 a_{l_1} 不能改变，那么 $w(l_1 + 1, r_1) = 0$，因此 $L_{r_i} \geq l_i + 1$，那么 k_1 就不能是 l_1，矛盾。$l_1 > l_2$ 同理。故 $l_1 = l_2$。
- 其次证明 $r_1 = r_2$。
  - 如果 $r_1 < r_2$，那么一定有 $w(r_1 + 1, r_2) = 0$，则 $k_2 \geq L_{r_2} \geq r_1 + 1$。
    - 若 $w(l_1, r_1) = 0$，那么 $k_2 \leq R_{l_2} = R_{l_1} \leq r_1$；
    - 若 $w(l_1, r_1) \neq 0$，那么 $k_2 = k_1 \leq r_1$。
  - 因此 $k_2 \leq r_1$，与 $k_2 \geq r_1 + 1$ 矛盾。$r_1 > r_2$ 同理。故 $r_1 = r_2$。
- 最后证明 $k_1 = k_2$。这是显然的。

- 同样设 dp_i 表示划分了区间 $[1,i]$ 所得到的 $f(D)$ 最大值以及 $g(D)$ 之和。

- 同样设 dp_i 表示划分了区间 $[1,i]$ 所得到的 $f(D)$ 最大值以及 $g(D)$ 之和。
- 转移时按照上述限制划分区间 $[l_i,r_i]$ 以及确定 k_i 即可。

- 同样设 dp_i 表示划分了区间 $[1, i]$ 所得到的 $f(D)$ 最大值以及 $g(D)$ 之和。
- 转移时按照上述限制划分区间 $[l_i, r_i]$ 以及确定 k_i 即可。
- 时间复杂度 $O(n^2)$，期望得分：100。

- 观察这个转移式，从 $dp_j \to dp_i$ 时有一个枚举中转点 k_i 的过程。

- 观察这个转移式，从 $dp_j \to dp_i$ 时有一个枚举中转点 k_i 的过程。
- 考虑先将 dp_j 转移至所有合法的 k_i，然后 dp_i 进行区间查询。

- 观察这个转移式，从 $dp_j \to dp_i$ 时有一个枚举中转点 k_i 的过程。
- 考虑先将 dp_j 转移至所有合法的 k_i，然后 dp_i 进行区间查询。
- 限制类似于二维数点，使用分块与线段树可以做到 $O(n\sqrt{n}\log n)$。

- 观察这个转移式，从 $dp_j \to dp_i$ 时有一个枚举中转点 k_i 的过程。
- 考虑先将 dp_j 转移至所有合法的 k_i，然后 dp_i 进行区间查询。
- 限制类似于二维数点，使用分块与线段树可以做到 $O(n\sqrt{n}\log n)$。
- 欢迎各位选手提出更优秀的解法。