2 Beautiful Sequence

2.1 题目来源

44th Petrozavodsk Programming Camp (2023 Winter), Day 2. GP of ainta, Problem F[1].

2.2 题目大意

给定一个长度为 N 的数列 $a_1, a_2, \cdots, a_N$，可以任意重新排列，输出 $\sum_{i=1}^{N}[a_i \geq \max(a_{i-1}, a_{i+1})]$ 的最大可能值，其中 $a_0 = a_{N+1} = -\infty$。

2.3 数据范围

本题单个测试内有多组测试数据。保证 $1 \leq N \leq 3 \times 10^5$ 且 $\sum N \leq 5 \times 10^6$。

2.4 解题方法

记所有元素构成的可重集为 U，称可重集 S 是**好的**当且仅当存在一个重排方式使得 $U \setminus S$ 均满足其大于等于两边的元素，考虑如何判定一个可重集 S 是否是**好的**。

先将可重集 S 按照任意顺序排序，记相邻两个数（含开头末尾的 $-\infty$）的最大值排序后依次为 $w_1 \leq w_2 \leq \cdots \leq w_m$，其中 $m = |S| + 1$，则现在我们需要将 $U \setminus S$ 中的元素插入到这 m 个空位里。

由于空隙内的元素需要满足大于等于两端的数值的限制。因此，在同一个连续段内，要么没有 $U \setminus S$ 中的元素，要么均相等。故可以假设 $U \setminus S$ 中的元素去重并排序后的结果为 $q_1 \leq q_2 \leq \cdots \leq q_k$，则我们需要为每一个 q_i 找到 w_i 使得 $q_i \geq w_i$ 并且每一个 q_i 对应不同的 w_i。

显然，这等价于 $k \leq m$ 且 $q_i \geq w_i (1 \leq i \leq k)$，而 q 是固定的。因此我们只需"最小化" w_i，首先必然有 $w_m = \max(S)$。

假设 S 中的元素排序过后从小到大依次为 $p_1 \leq p_2 \leq \cdots \leq p_{|S|}$，则按顺序排列即可做到 $w_i = p_i (1 \leq i \leq |S|)$。

定理 2.4.1. $w_i \geq p_i (1 \leq i \leq |S|)$。

证明. 假设 $w_i < p_i$，则 w 中有至少 i 个数小于 p_i，而 p 中只有至多 $i-1$ 个元素小于 p_i。

再考虑上左右两端的 $-\infty$，假设这至多 $i-1$ 个元素与两个 $-\infty$ 之间被大于等于 p_i 的元素划分为 t 个连续段，其中每一段的长度为 $s_1, s_2, \cdots, s_t$，则必然有 $t \geq 2$ 且 $\sum_{j=1}^{t} s_j \leq i+1$。

[1]QOJ 链接：QOJ 6540

另一方面，有恰好 $\sum_{j=1}^{t}(s_j-1)$ 个 w 中的元素小于 p_i，而有 $\left(\sum_{j=1}^{t} s_j\right)-t \le i+1-t \le i-1$，与假设中 w 中有至少 i 个数小于 p_i 矛盾。

综上所述，原命题得证。

因此，我们取 $w_i = p_i(1 \le i \le |S|)$ 且 $w_{|S|+1} = p_{|S|}$ 为所采用的 w。

我们梳理一下我们现在所得到的判定 S 的合法性的方法：

- 记 S 中的元素从小到大排序后为 $p_1 \le p_2 \le \cdots \le p_{|S|}$；
- 记 $U \setminus S$ 中的元素去重后从小到大排序为后为 $q_1 \le q_2 \le \cdots \le q_{|q|}$；
- 那么，S 是**好的**当且仅当 $|q| \le |S|+1$ 且 $p_i \le q_i(1 \le i \le |q|)$。

记数列 a 中的元素去重后排序依次为 $b_1 < b_2 < \cdots < b_m$，对应出现次数分别为 $c_1, c_2, \cdots, c_m$。显然存在某一最优策略使得 c_m 个 b_m 均不在 S 中，因为它们不能提供任何匹配。因此可以不考虑这 c_m 个元素，此时求解 $|S|$ 的最小可能值。

值得注意的是，此时条件中 $|q| \le |S|+1$ 变成了 $|q| \le |S|$，这也方便了我们进行下一步。

此时，相当于要将元素划分为两类 A 和 B，需要为每**种** B 中的数字 x 分配一个 A 中小于等于 x 的元素，在此基础上最小化 $|A|$。

我们按照 $i = 1, 2, \cdots, m-1$ 的顺序依次考虑每一种数字 b_i，同时维护一个初始为空小根堆，储存每个暂时划分到 B 中的数字的出现次数。同时维护变量 r，表示当前可被用于匹配的元素的数量，初始时 $r = 0$。由于 b 单调递增，因此在考虑到第 i 种数字时，之前的元素均可以被用于匹配。

使用贪心算法，当循环遍历到 i 时，先暂时将所有元素划分入 B，即将 c_i 插入小根堆中：

- 若 $r > 0$，则将 r 减去 1，表示将这种数字匹配上先前剩下的任意一个待匹配元素；
- 否则，$r = 0$，将小根堆的最小值 x 取出并将其减去 1 后插入回去（若 x 不为 1），表示将 B_i 这种数与 x 进行匹配。注意当 $x = 1$ 时需要将 r 加上 1，因为 x 对应的一种数不再需要匹配另外一个元素了，因此原先被匹配的元素可以被用于其它种类数字的匹配。

可以使用调整的方法证明上述贪心过程可以得到正确的 $|A|$ 的最小值（即为 $r = 0$ 对应操作被执行的次数）。

使用堆模拟上述过程即可做到 $\mathcal{O}(N \log N)$ 的时间复杂度。

2.5 参考资料

[1] Sunghyeon Jo (ainta). Editorial of Grand of ainta, *44th Petrozavodsk Programming Camp (2023 Winter), Day 2.*