QOJ2570 Maximal Subsequence 解题报告

题目大意

给定一个长度为 n 的序列 $a_1, \ldots, a_n$，你可以删除一些数，并保证未删除的数的相对位置不变。

求至多保留多少个数，使得最终序列的 LIS 长度小于最开始序列的 LIS 长度。

数据范围

$1 \le n \le 5 \times 10^5$，$1 \le a_i \le 10^9$。

解题过程

算法一

令 f_i 为以第 i 个位置结尾的 LIS 长度，g_i 为以第 i 个位置开头的 LIS 长度，m 为原序列的 LIS 长度。

首先，只有 $f_i + g_i = m + 1$ 的元素 a_i 会产生影响，称这些 i 为 “关键点”。

考虑对于两个关键点 i, j 满足 $i < j$，$f_j = f_i + 1$，$a_j > a_i$，则连边 $i \to j$。那么限制等价于每条从 $f_i = 1$ 的点到 $f_i = m$ 的点的路径上至少有一个点被选择删除。

容易构造出最小割模型：每个关键点拆成两点 i, i'，其间有边 $i \to i'$。从源点 s 连向所有 $f_i = 1$ 的关键点 i，从所有 $f_i = k$ 的关键点 i' 连向汇点 t。对两个关键点 i, j 满足 $i < j$，$f_j = f_i + 1$，$a_j > a_i$，连边 $i' \to j$。网络图中所有边的容量均为 1。

由于该网络图的最大流 $\le n$，因此粗略估计，暴力建图跑网络流的时间复杂度为 $O(n^3)$，使用线段树优化建图后时间复杂度为 $O(n^2 \log n)$。

算法二

考虑算法一中网络图的最大流的组合意义：求最多选择多少个不相交，长度为 m 的上升子序列。

现在完全的抛弃掉之前的网络图，直接考虑该转换后的问题如何做。我们在算法一最开始所描述的图上处理（两个关键点 i, j 满足 $i < j$，$f_j = f_i + 1$，$a_j > a_i$，则连边 $i \to j$）。目标是求至多选择多少条点不相交的从第 1 层到第 m 层的路径。

令所有 $f_i = k$ 的关键点 i 为 “第 k 层”，则所有 $i \to j$ 的边一定满足 i 在第 k 层，j 在第 $k + 1$ 层。

> 性质 1：把每层的点按照下标从小到大依次排列，则每层的点的 a 是单调不增的。
>
> 证明：若有两个同层关键点 i, j，满足 $i < j, a_i < a_j$，则 $f_j \ge f_i + 1$，与 $f_i = f_j$ 矛盾。

将每层的点按下标从小到大排列。

> 性质 2：对于第 k 层的点 i，连到的所有第 $k + 1$ 层的点 j 构成第 $k + 1$ 层点序列的一段区间。
>
> 证明：第 k 层的点 i 连向第 $k + 1$ 层的点 j 的要求是 $i < j, a_j > a_i$，而 $i < j$ 构成第 $k + 1$ 层点序列的一段后缀，$a_j > a_i$ 构成第 $k + 1$ 层点序列的一段前缀，前缀与后缀的交是区间。

对所有点 i 处理出其连向的所有 j 在下一层的点序列上构成的区间 $[l_i, r_i]$。

> 性质 3：对于第 k 层的点 i, j，满足 $i < j$，则 $l_i \le l_j, r_i \le r_j$。
>
> 证明：下标限制的后缀，j 比 i 要求更高，因此 j 对应后缀比 i 更靠后，即：$l_i \le l_j$。权值限制的前缀，j 比 i 要求更低，因此 j 对应前缀比 i 更靠后，即：$r_i \le r_j$。

接下来分析最优解的形式。

> 性质 4：存在一组最优解，使得任意两条选择的路径都不会出现 “交叉”：即对于任意两条选择的路径和层 $k(1 \le k < m)$，令两路径在第 k 层的点分别是 a_k, b_k，第 $k+1$ 层的点分别是 a_{k+1}, b_{k+1}，则不会有 $a_k < b_k, a_{k+1} > b_{k+1}$。
>
> 证明：只要有两个选择的路径和层 k，满足对应的 $a_k < b_k, a_{k+1} > b_{k+1}$，则可以交换 a_{k+1}, b_{k+1}，根据性质 3，对应路径仍然合法，且会减少 “交叉点” 数量。不断调整可使得 “交叉点” 数量变为 0，即任意两条路径都不出现 “交叉”。

现在所有选择的路径都拥有 “严格顺序”，因此选择的第一个路径是选每层的第一个，因为这对后续路径选取的影响最小。

于是得到了一个贪心算法：每次取尽可能靠前的，与之前已选择的不交的 LIS。

该过程可以变为选取若干轮，每轮执行：

- 选取每层未被删除的最小的 i。
- 将所有这一轮选择的点全部删除，并重新计算所有点的 f_i 和 g_i。
- 将所有 $f_i + g_i < ans + 1$ 的点 i 删除。

暴力实现该过程，时间复杂度：$O(n^2 \log n)$。

算法三

我们想要快速维护每一轮过后点的删除情况。

由上面的分析可得，一个点连向下一层的一段区间，也是由上一层的一段区间连向其。

容易归纳得出，任意轮结束后，被删除的点一定是每层的一个前缀。

一个点被删除的原因可以归为三种：

- 他在该轮中被选择了。
- 上一层删掉了某点（删掉了对应前驱/后继区间的右端点后，整个区间就全部被删空了，导致该点没有合法前驱了）。
- 下一层删掉了某点。

我们只需要存储每一层有哪些点，不需要对层之间的连边建图，而实际建立的图是当 “删除 x 会导致 y 被删除时”，加一条 $x \to y$ 的边。每轮只需要删除所有该轮选择的点和这些点能“到达”的所有点。直接 dfs，不重复经过已被删除的点。于是分完层后求答案的部分的时间复杂度为均摊 $O(n)$。

时间复杂度：$O(n \log n)$，瓶颈在求 f_i, g_i。

参考资料

QOJ2570 题面：https://qoj.ac/problem/2570

QOJ2570 题解：https://qoj.ac/download.php?type=attachments&id=821&r=2