Astral Birth 解题报告

华东师范大学第二附属中学 万成章

October 2022

目录

1 题目内容

1.1 题目描述

给定一个长度为 n 的只由 $0,1$ 组成的序列 $a_{1\ldots n}$。

请对于每个整数 $m \in [1,n]$ 求出：将序列划分为 m 个非空连续段并将连续段以任意顺序重新拼接，得到一个新序列 b，其最长不下降子序列长度（Longest Non-Decreasing Subsequence，在本文中将其记作 LNDS）的最大值是多少。

1.2 数据范围

对于全部数据：$1 \le n \le 3 \times 10^5$。

Subtask 1 (15 pts)：$n \le 8$。

Subtask 2 (15 pts)：$n \le 20$。

Subtask 3 (15 pts)：$n \le 200$。

Subtask 4 (20 pts)：$n \le 2000$。

Subtask 5 (20 pts)：$n \le 80000$。

Subtask 6 (15 pts)：无特殊限制。

1.3 题目来源

本题是一道原创题。

2 解题过程

2.1 Subtask 1

对于每个 m 直接枚举所有把序列划分为 m 个连续段的方案，然后再枚举所有排列连续段的方案，取其中最大的 LNDS。

将序列划分为 m 个连续段的方案数为 $\binom{n-1}{m-1}$，即从 $n-1$ 个空隙中选出 $m-1$ 个，枚举全排列方案数为 $m!$，求 LNDS 可以直接 DP，复杂度为 $O(n)$。

总的复杂度为 $O(n! \times \mathrm{Poly}(n))$。

2.2 Subtask 2

我们只枚举划分，不枚举排列。考虑如何求给定的 m 个序列任意拼接得到的 LNDS 的最大值。

由于是 01 序列，所以最终不下降子序列的形态一定是一段前缀为 0，一段后缀为 1（前缀后缀可以为空）。那么对于原来的 m 个子段，每个就有三种可能的情形：完全在前缀 0 部分中；完全在后缀 1 部分中；处在 0,1 分界位置上。其中最后一种情形只能发生在一个子段上。

对于第一种情形，其对答案的贡献为 0 的个数；对于第二种情形，其对答案的贡献为 1 的个数；对于第三种情形，其对答案的贡献为它的 LNDS。

枚举 0,1 分界位置的序列是第 i 个序列，令 C_x, D_x, L_x 分别为第 x 个序列中 0 的个数，1 的个数，LNDS 长度，则答案为：

$$\max_i (L_i + \sum_{j \neq i} \max(C_j, D_j))$$

在求得 C_x, D_x, L_x 后可以容易地 $O(m)$ 计算。

总复杂度为 $O(2^n \times n)$。

2.3 Subtask 3

考虑对划分进行 DP，几个关键的要素是当前划分的段数以及是否包含了最终在 0,1 交界处的子段。

设 $f(i,j,k)$ 表示将前 $a_1, \ldots, a_i$ 划分为 j 段并重排的最大 LNDS，其中 $k=0$ 表示不允许某个子段中同时选择 0 和 1 进入最长不降子序列，$k=1$ 则表示允许。

$$f(l-1,j,k) + \max(C[l,r], D[l,r]) \longrightarrow f(r,j+1,k)$$

$$f(l-1,j,0) + L[l,r] \longrightarrow f(r,j+1,1)$$

其中 $C[l,r], D[l,r], L[l,r]$ 分别表示 $[l,r]$ 中 0 的个数，1 的个数和 LNDS。复杂度为 $O(n^3)$。

2.4 Subtask 4

我们想要省去转移时带的一个 $O(n)$，为此我们不是在划分端点处才记录状态，而是对于每个位置都记录当前划分的状态。

设 $f(i,j,k,l)$ 表示当前将 $a_1, \ldots, a_i$ 划分为 j 段，最后一段还可以往后扩展，k 的含义与 Subtask 3 解法中相同，$l=0$ 表示当前这一段我们期望选 0 进入 LNDS，$l=1$ 表示期望选 1。

大部分转移是简单的，比较特殊的转移有：

$$f(i,j,0,0) + [a_{i+1} = 1] \longrightarrow f(i+1,j,1,1)$$

表示当前这一段在 0 之后又选了 1，也就是把它作为分界处的子段。

时间复杂度为 $O(n^2)$。

2.5 正解

首先有一个直觉的猜想，就是对于一段值相同的连续段，一定是尽量不将它拆到两个子段中比较好。为此，我们首先修改一下题目要求，令划分出的 m 个子段中允许包含空序列，容易发现这不影响答案。

引理：存在一种最优方案，每个值相同的连续段被划分在同一子段中。

证明：假设存在一个连续段被拆到两个或更多子段中，如果其中每个子段均没有选择这个连续段的元素进入 LNDS，则我们随意将这个连续段全部分配给其中一个子段即可；否则，我们将这个连续段分配给选了这一段的那个子段，这样答案一定不会更小。

有了这一结论之后，我们就可以将原序列通过将值相同的连续段（简称连续段）缩成一个数，转化为一个 01 相间的序列，只不过每个元素实际都代表了一段长度。

下面我们讨论的是比较一般的情况——连续段数 ≥ 3 的序列。

考虑一个总长为 t 的序列，有 k 个连续段，那么答案为 t 当且仅当划分段数 $m \geq k-1$。这是因为几乎每一段都要单独划分出来，除了某一对相邻的 0 和 1 连续段可以合并。

这引导我们将问题转化为如下形式：**选出一个尽量长的子序列，使得其连续段数不超过 $m+1$，这子序列的长度就是划分 m 段时的答案。**

在原有序列中删除一个位于中间的连续段，不仅会少这一段，还会促使它两边的段合并为一段，所以连续段数会减少 2。而删除最前或最后的连续段会使得连续段数减少 1。同时我们会发现删掉两个相邻的连续段一定是不优的，因为这两个连续段一定一个是 0，一个是 1，考虑删除之后它们所在位置的左侧（或右侧，都一样）是什么连续段，将这两段中与之同种的连续段直接拼上去，段数不增加而长度增加，一定更优。

所以现在问题进一步转化：**对于每个 x，选择一些价值之和为 x 的连续段（准备将它们删除），要求它们两两不相邻，且长度之和最小。其中靠边的连续段价值为 1，中间的连续段价值为 2。**

靠边的连续段很容易处理，只要枚举一下两侧的连续段是否要选，然后将同一过程做四次即可，现在我们要在中间选择一些两两不相邻的连续段删去，使得总和最小。

这是一个经典的反悔贪心问题。考虑最短的连续段，我们要么将它选上，要么就要将它两侧的段都选上（否则可以调整成选它），因此我们用小根堆维护

所有连续段长度，每次选出最小值 x，设它两侧的段的长度分别为 y, z，再将 $y+z-x$ 加入堆中。进行 k 次上述操作后得到的就是选择 k 个段的最小总长度。

注意我们上面讨论的是一般情况，对于连续段数 ≤ 2 的情况，由于可能找不到形如 01 的子序列，所以答案的计算方式可能有变，需要特殊处理一下，但这是平凡的。

时间复杂度为 $O(n \log n)$。

3 参考资料

本题出题过程中有过与集训队员郭羽冲的讨论。