

Deleting Substrings 解题报告

长郡中学 陈胤伯

1 试题来源

<http://codeforces.com/contest/392/problem/E>

2 试题大意

有一个正整数序列 $a_{1 \sim n}$ 。

每次你可以删除连续的一段 $a_{l \sim r}$ ，满足：

1. $|a_i - a_{i+1}| = 1$ 对于任意 $l \leq i < r$ 成立；
2. $2a_i - a_{i+1} - a_{i-1} \geq 0$ 对于任意 $l < i < r$ 成立。

删除长为 l 的连续一段可以获得 v_l 的收益。

你需要进行若干次删除，最大化总收益。

$n \leq 400$, $|v_i| \leq 2000$, $1 \leq a_i \leq 10^9$ 。

3 算法介绍

由于每次删除后序列会重新接起来，导致我们不太好设计DP状态。

我们需要有序地考虑这个问题。

3.1 计算 ans_i

我们定义 ans_i 表示序列 $a_{1 \sim i}$ 的答案。 ans_n 即为所求。

对于 $a_{1 \sim i}$ ，我们关注其最后一个元素 a_i 。

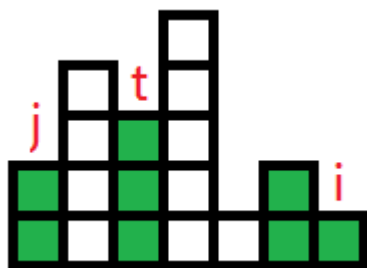
如果最终 a_i 未被删除，则直接转移到 ans_{i-1} 。

否则, a_i 一定可以作为最后一次删除的右端点, 那么我们枚举最后一次删除的左端点 j , 从 $ans_{j-1} + solve(j, i)$ 转移过来。

3.2 计算 $solve(j, i)$

$solve(j, i)$ 表示把 $a_{j \sim i}$ 这一段消光, 并且最后一次删除的左右端点分别为 j, i , 最大的总收益。

我们考虑最后一次删除, 根据题意, 删除的一段一定是一段凸的:



比如图中的绿色标记位置。

我们枚举这一段中的最高点 t , 于是最后一次删除的长度也确定了 (为 $1 + (a_t - a_l) + (a_t - a_r)$), 剩下的只需要确定 $j \sim t$ 、 $t \sim i$ 这两段的中间位置该如何选择。

我们先考虑 $j \sim t$ 这一段, 另一半同理。

由于最后一次删除的收益已被计算, 我们只需要考虑在凑出最后局面之前的最大总收益。换言之, 我们要决策 $j \sim t$ 这一段中间哪些位置放到最后消。

设所求为 $work(j, t)$, 显然有:

$$work(j, t) = \min_{j < j' \leq t \text{ and } a_{j'} = a_j + 1} \{work(j', t) + solve'(j + 1, j' - 1)\}$$

其中 $solve'(l, r)$ 表示把 $a_{l \sim r}$ 这段消完的最大总收益。

3.3 计算 $solve'(j, i)$

这个求法和 $solve(j, i)$ 类似。

依然考虑最后一次删除, 如果这次删除的左右端点恰好为 j, i , 那就是 $solve(j, i)$ 。

否则，我们一定能在这一段中找个位置断开，使得两边的删除互不影响。也就是枚举一个 $mid \in [j, i]$ ，转移到 $solve'(j, mid) + solve'(mid + 1, i)$ 。

3.4 总结

通过不断转化为递归子问题，我们最终用 DP 解决了这个问题。事实上这道题也可以看做是在用 DP 转移 DP。