20pts

枚举所有修改的情况，$f[i]$ 表示 $[1,i]$ 的划分方案，DP转移，时间复杂度为 $O(n^3)$。

+20pts

处理所有前缀 $f[i]$ 表示 $[1,i]$ 的划分方案，$g[i]$ 表示 $[i,n]$ 的划分方案，因为 $a_i \le 20$，所以如果转移的长度不超过 20 暴力枚举，否则一定可以，根据 f,g 数组，枚举 x 最终所在区间的长度，如果超过 20 一定可行，利用前缀和优化，否则暴力枚举所有情况，时间复杂度为 $O(nV^2)$，$V=20$。

100pts

考虑不修改序列的情况怎么做。 相当于每个区间 $[l,r]$ 要满足 $r-l+1 \ge \max a_l \dots a_r$。 设 dp_i 表示 $a_1 \dots a_i$ 被划分为若干个区间的数量。

考虑分治，设当前区间为 $[l,r]$，我们需要处理所有跨过 mid 的转移。 设 $mxL_i = \max\{a_i \dots a_{mid}\}$，$mxR_i = \max\{a_{mid+1} \dots a_i\}$。 那么一个跨过 $[l,r]$ 的区间合法当且仅当 $r-l+1 \ge \max\{mxL_l, mxR_r\}$。 这是一个二维偏序的形式，使用树状数组维护即可做到，时间复杂度为 $O(n\log^2 n)$。

再考虑一般的情况怎么做，继续利用上述分治结构，类比部分分，先求出每个前后缀的答案 f,g。 如果我们将 a_x 改为 1，我们需要计算所有包含 x 的合法区间 $[l,r]$ 对应的 $f_{l-1} \times g_{r+1}$ 。 因此只需要考虑分治树上所有包含 x 的区间来计算贡献。

不妨设 $x \in [l,mid]$，那么我们需要修改 mxL 中的若干个值， 并重新计算这些被修改的值带来的贡献。 这依然是一个二维偏序问题，每个被修改的值会产生 $O(\log n)$ 的代价。 对于一个分治树上的区间 $[l,r]$，依然考虑它的左半部分 $[l,mid]$。可以发现一个 mxL_i 最多只在一个 x 处被修改，因此 mxL 变化量的总和是 $O(mid-l)$ 的。 因此直接二维偏序计算答案即可，时间复杂度 $O(n\log^2 n)$。