IOI2026 中国国家集训队集中培训 第三试 A. 无题

IOI 2026 中国国家集训队工作组

2025 年 12 月 4 日

问题（无题（nameless））

- 给定一个长为 n 的非负序列 $\{a_n\}$，要将序列划分为若干段。
- 定义一个划分的权值是每一段的 $\max$ 和最后一个数之积的总和。
- q 组询问，每次询问给定一个段数 k，求分成 k 段的方案中权值的最大值。

数据范围：$n \leq 5 \times 10^5$，$q \leq 10^5$。

$n \leq 300$

$n \leq 300$

- 直接 DP，状态记录当前的位置和段数，枚举下一个位置转移。

$n \leq 300$

- 直接 DP，状态记录当前的位置和段数，枚举下一个位置转移。
- 时间复杂度 $O(n^3)$，期望得分：10。

我们可以观察到如下的性质：

我们可以观察到如下的性质：

定理

对于最优解，除最后一段外，每一段都以最大值结尾。

我们可以观察到如下的性质：

定理

对于最优解，除最后一段外，每一段都以最大值结尾。

证明.

若某一段（不是最后一段）不是最大值结尾，可以将末尾向前移动到最大值处，扔掉的部分给下一段。
当前段最大值不变，末尾变大；下一段最大值不会变小，末尾不变。故答案不会变劣。 □

特殊性质：$a_n = 0$

特殊性质：$a_n = 0$

- 即最后一段权值固定为零，只需最大化前 $k-1$ 段的权值。

特殊性质：$a_n = 0$

- 即最后一段权值固定为零，只需最大化前 $k-1$ 段的权值。
- 由上述性质，答案上限为前 $n-1$ 个数中前 $k-1$ 大数的平方和。

特殊性质：$a_n = 0$

- 即最后一段权值固定为零，只需最大化前 $k-1$ 段的权值。
- 由上述性质，答案上限为前 $n-1$ 个数中前 $k-1$ 大数的平方和。
- 这也是可以取到的，只需令前 $k-1$ 段的结尾依次取前 $n-1$ 个数的前 $k-1$ 大，最后一段自动补齐即可。

特殊性质：$a_n = 0$

- 即最后一段权值固定为零，只需最大化前 $k-1$ 段的权值。
- 由上述性质，答案上限为前 $n-1$ 个数中前 $k-1$ 大数的平方和。
- 这也是可以取到的，只需令前 $k-1$ 段的结尾依次取前 $n-1$ 个数的前 $k-1$ 大，最后一段自动补齐即可。
- 使用你喜欢的方法实现。时间复杂度 $O(n \log n)$，期望得分：10。

- 考虑扩展特殊性质的做法。

- 考虑扩展特殊性质的做法。
- 枚举最后一段的开头，在前面找前若干大的值，然后加上最后一段的权值即可。

- 考虑扩展特殊性质的做法。
- 枚举最后一段的开头，在前面找前若干大的值，然后加上最后一段的权值即可。
- 显然最优解一定会被枚举到。

- 考虑扩展特殊性质的做法。
- 枚举最后一段的开头，在前面找前若干大的值，然后加上最后一段的权值即可。
- 显然最优解一定会被枚举到。
- 使用 set 或双堆维护前 k 大，时间复杂度 $O(nq \log n)$，期望得分：40。

设 $A_{i,j}$ 为最后一段开头在 i，选 j 段的权值最大值。我们可以进一步观察到如下的性质：

设 $A_{i,j}$ 为最后一段开头在 i，选 j 段的权值最大值。我们可以进一步观察到如下的性质：

定理

矩阵 A 满足四边形不等式。

设 $A_{i,j}$ 为最后一段开头在 i，选 j 段的权值最大值。我们可以进一步观察到如下的性质：

定理

矩阵 A 满足四边形不等式。

证明.

$A_{i,j+1} - A_{i,j}$ 等于前 $i-1$ 个数第 j 大的平方。$A_{i+1,j+1} - A_{i+1,j}$ 等于前 i 个数第 j 大的平方。后者 $\geq$ 前者，变形可得四边形不等式。 □

- 由四边形不等式得 A 每一列的最大值位置单调。

- 由四边形不等式得 A 每一列的最大值位置单调。
- 分治，对中间列求出最大值位置后递归到两侧限制其最大值位置。

- 由四边形不等式得 A 每一列的最大值位置单调。
- 分治，对中间列求出最大值位置后递归到两侧限制其最大值位置。
- 由于需要反复移动，常数较大，可能需要用双堆模拟对顶堆才能过。

- 由四边形不等式得 A 每一列的最大值位置单调。
- 分治，对中间列求出最大值位置后递归到两侧限制其最大值位置。
- 由于需要反复移动，常数较大，可能需要用双堆模拟对顶堆才能过。
- 时间复杂度 $O(n \log n \log q)$，期望得分：$70 \sim 100$。

- 我们其实可以在 $O(\log n)$ 内直接求出 $A_{i,j}$ 的值。

- 我们其实可以在 $O(\log n)$ 内直接求出 $A_{i,j}$ 的值。
- 只需维护前缀前 k 大的平方和以及末尾段的最大值，前者可以用持久化线段树直接维护。

- 我们其实可以在 $O(\log n)$ 内直接求出 $A_{i,j}$ 的值。
- 只需维护前缀前 k 大的平方和以及末尾段的最大值，前者可以用持久化线段树直接维护。
- 这个做法常数显著更小。

- 我们其实可以在 $O(\log n)$ 内直接求出 $A_{i,j}$ 的值。
- 只需维护前缀前 k 大的平方和以及末尾段的最大值，前者可以用持久化线段树直接维护。
- 这个做法常数显著更小。
- 时间复杂度 $O(n \log n \log q)$，期望得分：100。

- 我们其实可以在 $O(\log n)$ 内直接求出 $A_{i,j}$ 的值。
- 只需维护前缀前 k 大的平方和以及末尾段的最大值，前者可以用持久化线段树直接维护。
- 这个做法常数显著更小。
- 时间复杂度 $O(n \log n \log q)$，期望得分：100。
- 可能可以通过一些修改-查询的平衡得到更好的复杂度，但没有必要。

- 能够快速求出 monge 单点点值给了我们极大的方便。

- 能够快速求出 monge 单点点值给了我们极大的方便。
- 我们可以直接使用 SMAWK 算法在 $O(n)$ 次查询内求出每一列的最大值。

- 能够快速求出 monge 单点点值给了我们极大的方便。
- 我们可以直接使用 SMAWK 算法在 $O(n)$ 次查询内求出每一列的最大值。
- 时间复杂度 $O(n \log n)$，期望得分：100。

- 能够快速求出 monge 单点点值给了我们极大的方便。
- 我们可以直接使用 SMAWK 算法在 $O(n)$ 次查询内求出每一列的最大值。
- 时间复杂度 $O(n \log n)$，期望得分：100。
- 但 SMAWK 的常数没有很优秀，$n = 10^6$ 跑出来可能比上一个解法还慢。