

$$\frac{n}{O} \leq 20$$

$$\frac{n}{O} \leq 80$$

$$\frac{n}{O} \leq 800$$

$$\frac{k}{O} \bmod 2 = 0$$

$$\frac{k}{O} > 5$$

$$\frac{n}{O} \leq 10^6$$

$$\frac{n}{O} \leq 10^6$$

NOI 2026 第二试《中位数》解题报告

NOI 2026 命题组

2026 年 7 月 22 日

$$\bullet \quad n \leq 20$$

$$\circ \quad n \leq 80$$

$$\circ \quad n \leq 800$$

$$\circ \quad k \bmod 2 = 0$$

$$\circ \quad k > 5$$

$$\circ \quad n \leq 10^6$$

$$\circ \quad n \leq 10^6$$

$$n \leq 20$$

- 直接枚举所有可能的划分并计算平衡度即可。
- 时间复杂度 $\mathcal{O}(n2^n) \sim \mathcal{O}(n^2 2^n)$ 。

$$\begin{array}{c} n \leq 20 \\ \bigcirc \end{array}$$

$$\begin{array}{c} n \leq 80 \\ \bullet \end{array}$$

$$\begin{array}{c} n \leq 800 \\ \bigcirc \end{array}$$

$$\begin{array}{c} k \bmod 2 = 0 \\ \bigcirc \end{array}$$

$$\begin{array}{c} k > 5 \\ \bigcirc \end{array}$$

$$\begin{array}{c} n \leq 10^6 \\ \bigcirc \end{array}$$

$$\begin{array}{c} n \leq 10^6 \\ \bigcirc \end{array}$$

$$n \leq 80$$

- 特殊性质 A 提示了可以将值域划分为两部分，然后判断落于哪一部分内。由此不难想到原问题可以通过二分答案转化为特殊性质 A 的情形。
- 在特殊性质 A 的限制下，一个段的中位数是 2 当且仅当 2 的数量不少于 1 的数量。将 1, 2 分别改写为 $-1, 1$ 后，条件变为区间和不少于 0。
- 由此可以设计一个简单的动态规划：设 $dp_{i,j}$ 表示以 i 结尾的前缀划分为 j 个段时，区间和不少于 0 的最大段数。
- 直接转移的复杂度是 $\mathcal{O}(n^2k)$ 的，总时间复杂度为 $\mathcal{O}(n^2k \log n)$ 。

$$\begin{array}{c} n \leq 20 \\ \bigcirc \end{array}$$

$$\begin{array}{c} n \leq 80 \\ \bigcirc \end{array}$$

$$\begin{array}{c} n \leq 800 \\ \bullet \end{array}$$

$$\begin{array}{c} k \bmod 2 = 0 \\ \bigcirc \end{array}$$

$$\begin{array}{c} k > 5 \\ \bigcirc \end{array}$$

$$\begin{array}{c} n \leq 10^6 \\ \bigcirc \end{array}$$

$$\begin{array}{c} n \leq 10^6 \\ \bigcirc \end{array}$$

$$n \leq 800$$

- 考虑上述动态规划转移的特性：每次转移只会增加至多 1 的贡献。因此大多数的转移都是无效的。
- 考虑贪心，尽可能只最近的可能增加贡献的位置转移，其余情况直接取前缀 max。
- 经观察或讨论可以发现，按如下方式选取最后一段一定是最优的：
 - 若 $a_i = 0$ ，则取最大的满足 $[j, i]$ 区间和为 0 的 j 。
 - 若 $a_i = 1$ ， $a_{i-1} = 1$ ，则取 $[i, i]$ 。
 - 若 $a_i = 1$ ， $a_{i-1} = 0$ ，则取 $[i-1, i]$ 。
- 找到区间和为 0 的一段是容易的，因为只需要找出前缀和相同的端点。所有信息可以线性预处理，最终转移可以优化到 $\mathcal{O}(nk)$ ，总时间复杂度为 $\mathcal{O}(nk \log n)$ 。因此该算法也能通过 $k \leq 10$ 的测试点。

$$k \bmod 2 = 0$$

- 之前的贪心提示了尽可能选取短的段是很优的。考虑直接选出 $k/2$ 个 1 作为单独的段。若其中有至少两个 1 相邻，则剩下的段一定不超过 k 个，无论怎么划分都是可行的。因此问题的关键实际在于，选出 $k/2$ 个 1 后，需要删去剩下的至多 $k/2 + 1$ 段区间和 < 0 中的一段。
- 对于开头和结尾，可以在开头增加 $[1, -1]$ ，结尾增加 $[-1, 1]$ 以简化问题。此时只需要考虑消去两个 1 之间的段。这只需要存在两个 1 之间的 -1 个数不超过 2 即可。
- 时间复杂度为 $\mathcal{O}(n \log n)$ 。

$$\frac{n}{0} \leq 20$$

$$\frac{n}{0} \leq 80$$

$$\frac{n}{0} \leq 800$$

$$\frac{k}{0} \bmod 2 = 0$$

$$\frac{k}{\bullet} > 5$$

$$\frac{n}{0} \leq 10^6$$

$$\frac{n}{0} \leq 10^6$$

$$k > 5$$

- 对于奇数的情况，还需要多消去一段。直接找两组距离 ≤ 3 的 1 的问题是，可能存在 $[1, -1, -1, 1, -1, -1, 1]$ 的情况。排除这种情况后，剩下的情况一定均有解。
- 然而，这样的划分需要 4 段和 ≥ 0 的段，因此对于 $k = 3, 5$ 的情况，该做法并不适用。同理，偶数的做法也不适用于 $k = 2$ ，需要单独判断。
- 因此该贪心做法仅能通过 $k > 5$ 的测试点。

$$\frac{n}{0} \leq 20$$

$$\frac{n}{0} \leq 80$$

$$\frac{n}{0} \leq 800$$

$$\frac{k}{0} \bmod 2 = 0$$

$$\frac{k}{0} > 5$$

$$\frac{n}{\bullet} \leq 10^6$$

$$\frac{n}{0} \leq 10^6$$

$$n \leq 10^6$$

- 直接合并 $k \leq 10$ 的做法与 $k > 5$ 的做法即可。
- 也可以对 $k = 2, 3, 5$ 单独分类讨论：
 - $k = 2$: 贪心选出第一段和 > 0 的位置，或翻转后再求一次；
 - $k = 3$: 贪心选出第一段和 > 0 的位置，然后转化为 $k = 2$ ，翻转后同理；
 - $k = 5$: 与 $k = 3$ 类似地转化，变为 $k = 4$ 的情况。唯一多的情况是两侧的段和 < 0 ，中间三段和均 ≥ 0 ，这可以通过枚举第一段位置，然后中间贪心选三次判断。
- 时间复杂度 $\mathcal{O}(n \log n)$ ，足以通过本题。

$$\begin{array}{c} n \leq 20 \\ \bigcirc \end{array}$$

$$\begin{array}{c} n \leq 80 \\ \bigcirc \end{array}$$

$$\begin{array}{c} n \leq 800 \\ \bigcirc \end{array}$$

$$\begin{array}{c} k \bmod 2 = 0 \\ \bigcirc \end{array}$$

$$\begin{array}{c} k > 5 \\ \bigcirc \end{array}$$

$$\begin{array}{c} n \leq 10^6 \\ \bigcirc \end{array}$$

$$\begin{array}{c} n \leq 10^6 \\ \bullet \end{array}$$

$$n \leq 10^6$$

- 进一步地，可以发现，二分的过程不是必要的。
- 例如，对于偶数的情况，一对距离不超过 3 的 1 可以判定可行，这意味着对于所有距离不超过 3 的点对的 $\min$ 求 $\max$ 即为答案。
- 除此之外还需满足 1 的个数足够，这可以通过线性求第 k 大的算法实现。
- 奇数的情况也是类似的。此时不止需要求出最大值，还要求出次大值等，但均可以在线性的时间复杂度内求出。
- $k = 2, 3, 5$ 的情况仍需要单独讨论。与二分的做法类似，此处需要一定的枚举。在线性时间内求出每个前缀的中位数是不可行的，但是可以求出这个值的前缀最大值。此外单独处理 $k = 5$ 的特殊情况即可。
- 最终的时间复杂度为 $\mathcal{O}(n)$ 。