

CF 1094 C. Median Partition — 带教对话录

Me (zzy) & Mentor (Claude)

2026-04-26

第 1 步：理解题目

Claude

题面看过了。先做个具体模拟来确认你理解了题意。

题目第 8 个样例 $a = [2, 1, 3, 2, 2]$ ，标准答案是 3。请你手动写出这 3 段具体的划分（每段是 a 的哪几个下标），并验证：

- 每段长度都是奇数；
- 每段的中位数都相同（具体值是多少?）。

zzy

213

132

322

中位数是 2，所以说，不同的 subarray 是可以这个重叠的是吧？

Claude

不能重叠！你写的三段 $[2, 1, 3]$ 、 $[1, 3, 2]$ 、 $[3, 2, 2]$ 是滑动窗口，它们覆盖了同一个元素多次（比如 $a_3 = 3$ 同时出现在三段里），不是合法划分。

题面里那个 k 序列是**严格递增**的： $k_1 = 1, k_2, \dots, k_{p+1} = n+1$ 。第 i 段是 $a_{k_i}, a_{k_i+1}, \dots, a_{k_{i+1}-1}$ ，所以**第 i 段结束的位置 + 1 = 第 $i+1$ 段开始的位置**——子数组**首尾相接、不重不漏**地覆盖整个 a ，每个元素**恰好属于一段**。

修正这条理解之后，重新切一下 $a = [2, 1, 3, 2, 2]$ 看看，把 3 段长度 + 中位数都标出来。

zzy

那就是

2

1 3 2

3

呗

Claude

划分结构对了，但最后一段 $a_5 = 2$ 不是 3（应该是手滑写错了）。完整划分是：

段	下标范围	元素	长度	中位数
1	[1, 1]	[2]	1	2
2	[2, 4]	[1, 3, 2]	3	2
3	[5, 5]	[2]	1	2

题意 OK 了，进入第 2 步。

第 2 步：暴力解法思路

Claude

我先抛一个具体的探测问题给你。

你刚才那个划分里，所有段的中位数都是 2。现在我问你：能不能换个值？比如，能不能找一个 $a = [2, 1, 3, 2, 2]$ 的合法划分，让每一段的中位数都是 1？或者都是 3？

试一下，告诉我能不能做到，做不到的话为什么做不到。

zzy

3、1 肯定不行啊，因为 3、1 在原数组中都不是这个中位数啊。

Claude

直觉对的——原数组排好序是 $[1, 2, 2, 2, 3]$ ，中位数是 2，所以 1 和 3 都不行。

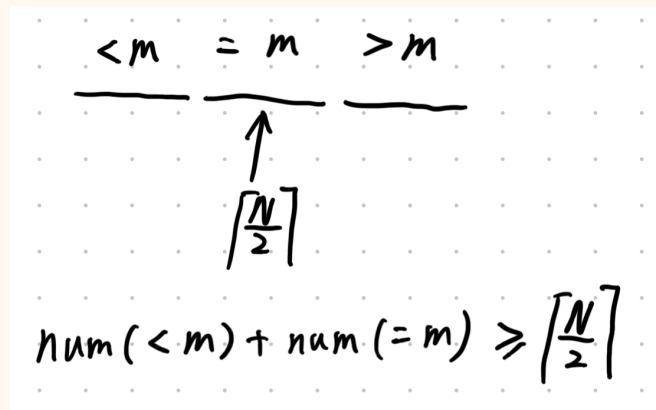
但你这是直觉——把它变成严谨证明试试：

命题：设 a 是长度为 n （奇）的数组，存在一个合法划分（所有段奇长度、公共中位数为 m ），那么 m 必须等于 a 整体的中位数。

提示用一个量做估计：「段内 $\leq m$ 的元素至少有多少个？ $\geq m$ 的元素至少有多少个？」把所有段加起来。试着推一下。

zzy

首先，如果 m 不是整体的中位数的话，那么肯定不能划分为 1 段，其次划分成更多段……一个数 m 是中位数的条件是：


$$\begin{array}{ccc} < m & = m & > m \\ \hline & \uparrow & \\ & \lfloor \frac{N}{2} \rfloor & \\ \text{num}(< m) + \text{num}(= m) & \geq & \lfloor \frac{N}{2} \rfloor \end{array}$$

即 $\text{num}(< m) + \text{num}(= m) \geq \lfloor N/2 \rfloor$ 。如果总体都不满足这个条件，那么分段当中，被分的 $\text{num}(< m)$ 和 $\text{num}(= m)$ 少的段更不可能满足这个条件。

Claude

对，核心就是这两条：(1) 中位数判定 $\text{num}(\leq m) \geq \lfloor N/2 \rfloor$ 且 $\text{num}(\geq m) \geq \lfloor N/2 \rfloor$ ；(2) 每段都满足这俩条件，把所有段相加，整体也满足，所以 m 必须等于整体中位数。

结论：公共中位数 m 唯一确定，等于 a 整体中位数 ($O(n \log n)$ 排序求出)。

接下来问题就转化成了：

已知 m ，把 a 切成最多段奇长度子数组，每段中位数都等于 m 。求最大段数。

题目 tag 是 dp， $n < 5000$ ， $\text{sum of } n^2 \leq 5000^2$ 。所以 $O(n^2)$ 暴力 DP 就够。

追问：你来设计这个 DP。

- 状态 $f[i]$ 是什么意思（用一句中文话讲清楚）？

- 转移怎么写（从哪些 j 转移过来， j 满足什么条件）？

zzy

$f[i]$ 嘛肯定就是表示到 i 为止最多能被切成多少块嘛，从哪些 j 转移就看 $a[j+1, i]$ 的中位数是不是这个 m 呗。

Claude

对，框架就是这个。补两个你没明说但要写到代码里的细节：

- 段长 $i - j$ 必须奇（题目要求）。
- 初值 $f[0] = 0$ ，其余 $-\infty$ （不可达）；答案是 $f[n]$ 。

转移式：

$$f[i] = \max \left\{ f[j] + 1 \mid 0 \leq j < i, (i - j) \text{ 奇}, \text{med}(a[j+1..i]) = m \right\}.$$

追问：现在 DP 框架定了，瓶颈就是中间那个判定——**怎么判断一段子数组（奇长度）中位数 = m ?**

要求：能 $O(1)$ 单次判定（预处理可以 $O(n)$ ）。给我一个具体方法。

zzy

怎么 $O(1)$ 判定吗？这个有点难度啊，不太会啊。

哦，我们实际上就是要判定这个区间内 m 的个数是不是大于 $\lceil N/2 \rceil$ 嘛，这个简单啊，整个前缀和就行。

Claude

前缀和方向对，但**数错了东西**。你的判定是 $\text{count}(=m) > \lceil \text{len}/2 \rceil$ ，这是**严格多数**，比中位数 $=m$ 强。反例： $a = [1, 2, 3]$ ， $m = 2$ ，中位数是 2，但 $\text{count}(=2) = 1 < \lceil 3/2 \rceil = 2$ ，会漏掉。

正解（直接给）：用 **± 1 编码 + 两套前缀和**。

$$b_1[k] = \begin{cases} +1, & a_k \leq m \\ -1, & a_k > m \end{cases}, \quad B_1[k] = \sum_{j \leq k} b_1[j],$$

$$b_2[k] = \begin{cases} +1, & a_k \geq m \\ -1, & a_k < m \end{cases}, \quad B_2[k] = \sum_{j \leq k} b_2[j].$$

对于奇长度区间 $[l, r]$ ($\text{len} = r - l + 1$ 奇)， $\text{num}(\leq m) - \text{num}(> m) = B_1[r] - B_1[l - 1]$ ， ≥ 1 等价于 $\text{num}(\leq m) \geq \lceil \text{len}/2 \rceil$ 。 B_2 同理。所以：

$$\text{med}(a[l..r]) = m \text{ 当且仅当 } B_1[r] - B_1[l - 1] \geq 1 \text{ 且 } B_2[r] - B_2[l - 1] \geq 1.$$

预处理 $O(n)$ ，单次判 $O(1)$ 。整个 DP 复杂度 $O(n^2)$ ，给 $\sum n^2 \leq 5000^2$ 够用，这就是**标程**（不需要进一步优化）。

第 3 步：实现

Claude

算法已经齐了：

1. 排序求整体中位数 m 。
2. 预处理两套前缀和 B_1, B_2 。
3. DP: $f[i] = \max\{f[j] + 1 : (i - j) \text{ 奇}, B_1[i] - B_1[j] \geq 1, B_2[i] - B_2[j] \geq 1\}$, $f[0] = 0$ 。
4. 答案 $f[n]$ 。

去 `src/C_Median_Partition.cpp` 里把 `Solve()` 写完，写完告诉我。

ZZY

暴力的实现我们直接跳过吧，之前已经把所有细节都搞清楚了（关于暴力的）。哦，不对，这个题目本身就是 $O(n^2)$ 的，那就已经能做了，绷不住了。

Claude

对，这题约束 $\sum n^2 \leq 5000^2$ 直接为 $O(n^2)$ DP 量身定制，暴力 = 标程，本题没有「暴力 → 优化」两步走的必要。直接去写代码就行。

ZZY

```
1 for (int i = 0; i < N; ++i) {
2     for (int j = 0; j <= i; ++j) {
3         if (checkLr(preBtm, preLtm, i, j)) {
4             dp[i] = max(dp[i], dp[i] + dp[j - 1]);
5         }
6     }
7 }
```

我在想 0-based 的话这个 dp 转移怎么实现比较优雅？我当然知道这里可以用三元运算符，但那本质上还是 if-else。

Claude

$j - 1$ 不优雅是因为 dp 的索引语义混了——“ $a[0..i]$ 的答案”和“ $a[0..i-1]$ 的答案”这俩约定哪种都行，但同一份代码里只能选一种。

最干净的：让 $f[i]$ 表示「前 i 个元素」的答案（前缀长度，不是末元素下标）。这样 $f[0] = 0$ 是天然 base，不需要 $f[-1]$ 。

```
1 vector<ll> dp(N + 1, LLONG_MIN);
2 dp[0] = 0;
3 for (int i = 1; i <= N; ++i)
4     for (int j = 0; j < i; ++j)
5         if (((i - j) & 1) && checkLr(j, i - 1)
6             && dp[j] != LLONG_MIN)
7             dp[i] = max(dp[i], dp[j] + 1);
8 cout << dp[N] << "\n";
```

「前缀长度索引」是专治 base case 越界的标准 trick。

首先，我问你的，是一劳永逸拒绝 dp 0-based 方法，不是针对这道题目。
其次，你又开始摆 mentor 的架子，你应该理解我的 dp 定义，而不是套用你自己的定义。

Claude

都吃下。两条都对——上面那段是按我自己的语义重写了你的 dp，确实越界。重新答你的通用问题：

只要 $dp[i]$ 按「末元素下标」索引，转移到 $dp[j-1]$ 时 $j=0$ 必然撞 $dp[-1]$ （空前缀的 base）。这个 base 不是某题特例，是所有按位置索引的 DP 的共性。通用治法只有两种范式：

(A) Sentinel 包装，保留你的语义

不改 dp 的语义，包一层访问器：

```
1 auto D = [&](int x) { return x < 0 ? 0 : dp[x]; };
2 // 转移:  $D(j - 1) + 1$  代替  $dp[j - 1] + 1$ 
```

或直接内联三元：

```
1 dp[i] = max(dp[i], (j == 0 ? 0 : dp[j-1]) + 1);
```

你说“三元本质上还是 if-else”——对，是。但 base 的判断逻辑上必须存在：要么写出来要么藏到访问器里。Sentinel 的好处是包装一次、之后转移式干净没补丁。

(B) 索引整体右移 1，吃掉分支

机械地把 $dp[i]$ 全部重命名成 $dp'[i+1]$ ：分配大小 $N+1$ ， $dp'[0]$ 就是空前缀的 base，不需要负下标。这不是改你的语义，是纯下标偏移：

```
1 vector<ll> dp(N + 1, ...);
2 dp[0] = 0; // 原 dp[-1]
3 // 原 dp[i] 写成 dp[i+1]
4 // 转移用 dp[j] 代替 dp[j-1]
```

很多人看到这写法叫“按前缀长度索引”——其实只是机械的 +1 偏移，不是换语义。它的好处是热循环没分支。

两种等价，挑顺手的。(A) 贴近你的思维模型，(B) 编译出的代码可能稍快一丁点（其实差不多）。