

2026 全国青少年信息学奥林匹克竞赛

CCF NOI 2026

第二试

时间：2026 年 7 月 22 日 08:00 ~ 13:00

题目名称	中位数	木棉	彩虹树
题目类型	传统型	传统型	传统型
目录	median	kapok	rainbow
每个测试点时限	1.0 秒	2.0 秒	1.0 秒
内存限制	1024 MiB	1024 MiB	1024 MiB
测试点数目	20	25	25
测试点是否等分	是	是	是
预测试点数目	20	25	25

提交程序源文件名

对于 C++ 语言	median.cpp	kapok.cpp	rainbow.cpp
-----------	------------	-----------	-------------

编译选项

对于 C++ 语言	-O2 -std=c++14 -static
-----------	------------------------

注意事项（请仔细阅读）

- 赛后正式测试时将以选手留在题目目录下的程序源文件为准。对程序源文件以外文件的修改不会影响评测结果。
- 选手提交的程序源文件大小不得超过 100 KiB。
- 程序可使用的栈空间内存限制与题目的内存限制一致。
- 禁止在提交的程序源文件中改变编译器参数（如使用 `#pragma` 命令），禁止使用系统结构相关指令（如内联汇编）或其他可能造成不公平的方法。
- 因违反上述规定而出现的问题，申诉时一律不予受理。
- 选手可使用快捷启动页面中的工具 selfEval 进行自测。选手需将待测程序的源文件置于相应题目目录下。每次自测时可选择全部或部分题目进行自测。注意：自测有次数限制，且自测结果仅供选手测试参考，不作为最终正式成绩。

中位数 (median)

【题目描述】

对于大小为 m 的可重集 $S = \{x_0, x_1, \dots, x_{m-1}\}$, 设其所有元素从大到小排序后的结果为 $y_0 \geq y_1 \geq \dots \geq y_{m-1}$ 。定义其中位数 $\text{Median}(S)$ 为其中第 $\lceil m/2 \rceil$ 大的数, 即 $\text{Median}(S) = y_{\lceil m/2 \rceil - 1}$ 。注意: 本题中的中位数定义与常规定义可能有所不同。

给定长度为 n 的序列 $[a_0, a_1, \dots, a_{n-1}]$, 以及一个正整数 k ($k \leq n$)。定义一个划分如下: 选择一个长度为 $k-1$ 的递增下标序列 $0 < b_1 < \dots < b_{k-1} < n$, 即可将原序列划分为 k 个段, 对应的下标区间依次为 $[0, b_1), [b_1, b_2), \dots, [b_{k-1}, n)$ 。

对于一个划分, 定义该划分的平衡度如下: 对于划分出的每个段, 计算其中所有元素形成的可重集的中位数, 则这 k 个中位数形成的可重集的中位数即为该划分的平衡度。形式化地, 对于划分 $b_1, \dots, b_{k-1}$, 记 $b_0 = 0$, $b_k = n$, 设第 i ($0 \leq i < k$) 个段中所有元素形成的可重集的中位数为 $c_i = \text{Median}(\{a_{b_i}, a_{b_i+1}, \dots, a_{b_{i+1}-1}\})$, 则该划分的平衡度为 $\text{Median}(\{c_0, \dots, c_{k-1}\})$ 。

请求出所有划分中平衡度的最大值。

【实现细节】

选手不需要, 也不应该实现 `main` 函数。

选手需要确保提交的程序源文件包含头文件 `median.h`, 即在程序开头加入以下代码:

```
1 #include "median.h"
```

选手需要在提交的程序源文件 `median.cpp` 中实现以下两个函数:

```
1 void init(int c, int t);
```

- c, t 分别表示测试点编号与测试数据组数。 $c = 0$ 表示该测试点为样例。
- 对于每个测试点, 该函数会在程序开始运行时被评测程序调用恰好一次。

```
1 int median(int n, int k, std::vector<int> a);
```

- n, k, a 分别表示序列长度、划分的段数与给定的序列。
- 该函数需要返回平衡度的最大值。
- 对于每个测试点, 该函数会被评测程序调用恰好 t 次。

本试题目下的 `template_median.cpp` 是提供的示例代码, 选手可参考并实现自己的代码。

【测试程序方式】

选手可以在本题目目录下使用如下命令编译得到可执行文件:

```
1 g++ grader.cpp median.cpp -o median -O2 -std=c++14 -static
```

对于编译得到的可执行文件 `median`:

- 可执行文件将从标准输入读入以下格式的数据:
 - 第一行包含两个非负整数 c, t 。
 - 接下来依次为每组测试数据, 对于每组测试数据:
 - * 第一行包含两个正整数 n, k 。
 - * 第二行包含 n 个正整数 $a_0, a_1, \dots, a_{n-1}$ 。
- 可执行文件将输出以下格式的数据至标准输出:
 - 对于每组测试数据, 输出一行一个正整数, 表示平衡度的最大值。

【样例 1 输入】

```
1 0 2
2 10 4
3 6 5 1 9 2 3 10 7 4 8
4 10 5
5 5 7 3 10 8 2 9 1 6 4
```

【样例 1 输出】

```
1 9
2 8
```

【样例 1 解释】

对于第一组测试数据, 一种平衡度最大的划分为 $b_1 = 3, b_2 = 5, b_3 = 7$, 其将原序列划分为 4 个段 $[6, 5, 1], [9, 2], [3, 10], [7, 4, 8]$, 段中所有元素的中位数分别为 5, 9, 10, 7, 因此该划分的平衡度为 $\text{Median}(\{5, 9, 10, 7\}) = 9$ 。

对于第二组测试数据, 一种平衡度最大的划分为 $b_1 = 2, b_2 = 4, b_3 = 6, b_4 = 8$, 其将原序列划分为 5 个段 $[5, 7], [3, 10], [8, 2], [9, 1], [6, 4]$, 段中所有元素的中位数分别为 7, 10, 8, 9, 6, 因此该划分的平衡度为 $\text{Median}(\{7, 10, 8, 9, 6\}) = 8$ 。

【样例 2】

见选手目录下的 `median/median2.in` 与 `median/median2.ans`。

该样例满足测试点 6 的约束条件。

【样例 3】

见选手目录下的 `median/median3.in` 与 `median/median3.ans`。
该样例满足测试点 9 的约束条件。

【样例 4】

见选手目录下的 `median/median4.in` 与 `median/median4.ans`。
该样例满足测试点 14 的约束条件。

【样例 5】

见选手目录下的 `median/median5.in` 与 `median/median5.ans`。
该样例满足测试点 18 ~ 20 的约束条件。

【数据范围】

设 N 为单个测试点内所有测试数据的 n 的和。对于所有测试数据，均有：

- $1 \leq t \leq 20$;
- $5 \leq n \leq 10^6$, $2 \leq k \leq n$, $N \leq 10^6$;
- 对于所有 $0 \leq i < n$, 均有 $1 \leq a_i \leq n$ 。

测试点编号	$N \leq$	$n \leq$	k	特殊性质
1, 2	40	20	$\leq n$	无
3 ~ 5	800	80		A
6				无
7, 8	8, 000	800		A
9				无
10	2×10^5	2×10^5	$= 2$	
11			$= 3$	
12, 13			$= 5$	
14			≤ 10	
15			$\equiv 0 \pmod{2}$	
16, 17	10^6	10^6	> 5	
18 ~ 20			$\leq n$	

特殊性质 A：对于所有 $0 \leq i < n$, 均有 $a_i \leq 2$ 。