

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）

1. 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。
2. 选手提交的程序源文件大小不得超过 100 KiB。
3. 程序可使用的栈空间内存限制与题目的内存限制一致。
4. 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。
5. 因违反上述规定而出现的问题，申诉时一律不予受理。
6. 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。

木棉 (kapok)

【题目描述】

故园中的那几棵木棉，仍旧生长在小 N 逐渐朦胧的记忆中。小 N 对故园的记忆，可以用一个长度为 n 的序列 $[a_0, a_1, \dots, a_{n-1}]$ 表示。

故园中的每棵木棉，都形如一棵结点有标号的无根树。小 N 对一棵木棉的印象，可以用她的故园记忆的一个区间 $[l, r)$ 描述：

- 这棵树的结点数目为 $k = r - l + 2$ ，结点编号为 $0 \sim k - 1$ ；
- $[\min(a_l, k - 1), \min(a_{l+1}, k - 1), \dots, \min(a_{r-1}, k - 1)]$ 是这棵树的 Prüfer 序列，其中 Prüfer 序列的定义详见【提示】一节。

在回忆往事时，小 N 也向你提出了 m 次询问。其中第 i ($0 \leq i < m$) 次询问为：

- 在故园记忆的区间 $[l_i, r_i)$ 所对应的木棉上，结点 x_i, y_i 是否相邻？

【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 kapok.h，即在程序开头加入以下代码：

```
1 #include "kapok.h"
```

选手需要在提交的程序源文件 kapok.cpp 中实现以下函数：

```
1 std::vector<bool> kapok(int c, int n, int m,
    std::vector<int> a, std::vector<int> l, std::vector<int>
    r, std::vector<int> x, std::vector<int> y);
```

- c, n, m 分别表示测试点编号、故园记忆序列的长度、询问次数， $c = 0$ 表示该测试点为样例。
- a 表示故园记忆序列。
- l, r 分别表示每次询问所给定的区间的两个端点。
- x, y 分别表示每次询问所给定的两个结点的编号。
- 该函数需要返回一个长度恰好为 m 的序列 $f_0, f_1, \dots, f_{m-1}$ ，其中 f_i ($0 \leq i < m$) 表示第 i 次询问的结果。
- 对于每个测试点，该函数会被评测程序调用恰好一次。

本试题目录下的 template_kapok.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】

选手可以在本题目录下使用如下命令编译得到可执行文件：

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

对于编译得到的可执行文件 **kapok**:

- 可执行文件将从标准输入读入以下格式的数据:
 - 第一行包含三个非负整数 c, n, m 。
 - 第二行包含 n 个非负整数 $a_0, a_1, \dots, a_{n-1}$ 。
 - 第 $i+3$ ($0 \leq i < m$) 行包含四个非负整数 l_i, r_i, x_i, y_i 。
- 可执行文件将输出以下格式的数据至标准输出:
 - 第 $i+1$ ($0 \leq i < m$) 行包含一个非负整数, 其中 0 表示 f_i 为 **false**, 1 表示 f_i 为 **true**。

【样例 1 输入】

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

【样例 1 输出】

```
1 1
2 0
3 1
```

【样例 1 解释】

- 区间 $[0, 3)$ 对应的树有 5 个结点, Prüfer 序列为 $[2, 0, 2]$, 边集为 $\{(1, 2), (0, 3), (0, 2), (2, 4)\}$, 因此结点 0, 2 相邻。
- 区间 $[3, 5)$ 对应的树有 4 个结点, Prüfer 序列为 $[3, 0]$, 边集为 $\{(1, 3), (0, 2), (0, 3)\}$, 因此结点 1, 2 不相邻。
- 区间 $[0, 0)$ 对应的树有 2 个结点, Prüfer 序列为空, 唯一一条边为 $(0, 1)$, 因此结点 0, 1 相邻。

【样例 2】

见选手目录下的 *kapok/kapok2.in* 与 *kapok/kapok2.ans*。

该样例满足测试点 3 ~ 5 的约束条件。

【样例 3】

见选手目录下的 *kapok/kapok3.in* 与 *kapok/kapok3.ans*。

该样例满足测试点 3 ~ 5 的约束条件。

【数据范围】

对于所有测试数据，均有：

- $1 \leq n, m \leq 2 \times 10^5$;
- 对于所有 $0 \leq i < n$ ，均有 $0 \leq a_i < n + 2$;
- 对于所有 $0 \leq i < m$ ，均有 $0 \leq l_i \leq r_i \leq n$ 且 $0 \leq x_i, y_i < r_i - l_i + 2$ 。

测试点编号	$n, m \leq$	特殊性质
1, 2	500	无
3 ~ 5	5,000	
6, 7	2×10^5	对于所有 $0 \leq i < n$ ，均有 $a_i < 10$
8, 9		对于所有 $0 \leq i < n$ ，均有 $a_i < 10^3$
10, 11		对于所有 $0 \leq i < m$ ，均有 $x_i, y_i < 10$
12, 13		对于所有 $0 \leq i < m$ ，均有 $x_i, y_i < 10^3$
14 ~ 16		A
17 ~ 19		对于所有 $0 \leq i < m$ ，均有 $r_i = n$
20 ~ 22	10^5	无
23 ~ 25	2×10^5	

- 特殊性质 A：对于所有 $0 \leq i < m$ ，在给定 l_i, r_i 后， x_i, y_i 均从所有满足限制的有序点对中独立均匀随机生成。

【提示】

对于一棵结点编号为 $0 \sim c - 1$ ($c \geq 2$) 的无根树 T ：

- 依次执行 $c - 2$ 次操作，其中第 i ($0 \leq i < c - 2$) 次操作如下：
 - 找出当前编号最小的叶子 v_i ；
 - 记录其唯一相邻点的编号 p_i ；
 - 然后从 T 中删去 v_i 及其唯一邻边。
- 如此得到的长度为 $c - 2$ 的序列 $[p_0, p_1, \dots, p_{c-3}]$ ，即为 T 的 Prüfer 序列。