

2026 全国青少年信息学奥林匹克竞赛

CCF NOI 2026

第一试

时间：2026 年 7 月 20 日 08:00 ~ 13:00

题目名称	线段	传送	布丁
题目类型	传统型	传统型	交互型
目录	segment	teleport	pudding
每个测试点时限	2.5 秒	3.5 秒	10.0 秒
内存限制	512 MiB	1024 MiB	1024 MiB
测试点数目	25	20	3
测试点是否等分	是	是	否
预测试点数目	25	20	3

提交程序源文件名

对于 C++ 语言	segment.cpp	teleport.cpp	pudding.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。

线段 (segment)

【题目描述】

小 L 有 n 条包含于 $[1, m]$ 的线段, 其中第 i ($0 \leq i < n$) 条为 $[l_i, r_i]$ ($1 \leq l_i \leq r_i \leq m$)。

小 L 认为过于复杂的线段相交关系不够优美, 于是对于每一个线段的集合 $S \subseteq \{0, 1, \dots, n-1\}$, 小 L 定义 S 是**优美的**, 如果满足如下要求:

- 构造一个顶点集合对应 S 的图, 其中顶点 u 与顶点 v 间存在一条边, 当且仅当线段 u 与线段 v 相交, 即存在 $x \in [1, m]$, 满足 $l_u \leq x \leq r_u$ 且 $l_v \leq x \leq r_v$ 。称 S 是**优美的**, 当且仅当构造出的图恰好为一棵树。

小 L 想知道有多少线段集合是优美的, 因此他给定了一个正整数 k ($k \leq n$), 你需要计算, 对于每个 $s = 1, 2, \dots, k$, 有多少个大小为 s 的集合是优美的。由于答案可能较大, 只需求出答案对 998,244,353 取模后的结果。

【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 `segment.h`, 即在程序开头加入以下代码:

```
1 #include "segment.h"
```

选手需要在提交的程序源文件 `segment.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 std::vector<int> segment(int n, int m, int k,
    std::vector<int> l, std::vector<int> r);
```

- n, m, k 分别表示线段的数量、坐标范围上限及需要计算的集合的大小上限。
- l, r 分别表示每条线段的左端点与右端点。
- 该函数需要返回一个长度恰好为 $k+1$ 的序列 a , 其中 $a_0 = 0$, a_s ($1 \leq s \leq k$) 表示大小为 s 的优美的集合数量对 998,244,353 取模后的结果。
- 对于每个测试点, 该函数会被评测程序调用恰好 t 次。

本试题目录下的 `template_segment.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 segment.cpp -o segment -O2 -std=c++14 -static
```

对于编译得到的可执行文件 `segment`:

- 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:
 - * 第一行包含三个正整数 n, m, k 。
 - * 第 $i + 2$ ($0 \leq i < n$) 行包含两个正整数 l_i, r_i 。
- 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 k 个非负整数 $a_1, a_2, \dots, a_k$ 。

【样例 1 输入】

```
1 0 3
2 3 3 3
3 1 2
4 2 3
5 1 3
6 4 5 4
7 1 2
8 2 3
9 3 4
10 4 5
11 4 2 3
12 1 2
13 1 2
14 1 2
15 1 1
```

【样例 1 输出】

```
1 3 3 0
2 4 3 2 1
3 4 6 0
```

【样例 1 解释】

对于第一组测试数据,

- 大小为 1 的集合有 $\{0\}, \{1\}, \{2\}$, 均是优美的。
- 大小为 2 的集合有 $\{0, 1\}, \{1, 2\}, \{0, 2\}$, 均是优美的。
- 大小为 3 的集合有 $\{0, 1, 2\}$, 构造出的图是一个三元环, 不是优美的。

因此答案分别为 3, 3, 0。

对于第二组测试数据。

- 大小为 1 的集合中, 所有 4 个集合均是优美的。
- 大小为 2 的集合中, $\{0, 1\}, \{1, 2\}, \{2, 3\}$ 是优美的。
- 大小为 3 的集合中, $\{0, 1, 2\}, \{1, 2, 3\}$ 是优美的。
- 大小为 4 的集合 $\{0, 1, 2, 3\}$ 是优美的。

因此答案分别为 4, 3, 2, 1。

【样例 2】

见选手目录下的 *segment/segment2.in* 与 *segment/segment2.ans*。

该样例满足测试点 6 ~ 8 的约束条件。

【样例 3】

见选手目录下的 *segment/segment3.in* 与 *segment/segment3.ans*。

该样例满足测试点 9, 10 的约束条件。

【样例 4】

见选手目录下的 *segment/segment4.in* 与 *segment/segment4.ans*。

该样例满足测试点 11 ~ 15 的约束条件。

【样例 5】

见选手目录下的 *segment/segment5.in* 与 *segment/segment5.ans*。

该样例满足测试点 16 ~ 18 的约束条件。

【样例 6】

见选手目录下的 *segment/segment6.in* 与 *segment/segment6.ans*。

该样例满足测试点 22, 23 的约束条件。

【样例 7】

见选手目录下的 `segment/segment7.in` 与 `segment/segment7.ans`。

该样例满足测试点 24, 25 的约束条件。

【数据范围】

设 K 为单个测试点内所有测试数据的 k 的和。对于所有测试数据，均有：

- $1 \leq t \leq 20$;
- $1 \leq n \leq 3,000$, $1 \leq m \leq 10^3$, $1 \leq k \leq n$, $K \leq 200$;
- 对于所有 $0 \leq i < n$, 均有 $1 \leq l_i \leq r_i \leq m$ 。

测试点编号	$n \leq$	$m \leq$	$K \leq$	$k \leq$	特殊性质
1 ~ 3	20	10^2	20	20	无
4, 5	3,000	10^3	200	2	
6 ~ 8				3	
9, 10	500			200	A
11 ~ 15	3,000				B
16 ~ 18	200	500	50	50	C
19 ~ 21	500	10^3	200	200	
22, 23	10^3	10^2	30	30	无
24, 25	3,000	10^3	200	200	

特殊性质 A：对于所有 $0 \leq i, j < n$ 且 $i \neq j$, 均有线段 i 不包含线段 j , 即 $l_i > l_j$ 或 $r_i < r_j$ 。

特殊性质 B：对于所有 $0 \leq i < j < n$, 均有线段 i 包含线段 j 或线段 i 与线段 j 不相交, 即 $l_i \leq l_j \leq r_j \leq r_i$ 或 $r_i < l_j$ 或 $l_i > r_j$ 。

特殊性质 C： n 条线段的全部 $2n$ 个端点互不相同, 即 $l_0, l_1, \dots, l_{n-1}, r_0, r_1, \dots, r_{n-1}$ 两两不同。