

# NOIP 2024 模拟赛

(2024.09.15 8:00~12:30)

(请选手务必仔细阅读本页内容)

## 一、题目概况

中文题目名称	神奇的二叉查找树	神奇的字母	神奇的花园	神奇的区间
英文题目名称	bst	alpha	garden	interval
可执行文件名	bst	alpha	garden	interval
输入文件名	bst	alpha	garden	interval
输出文件名	bst	alpha	garden	interval
每个测试点时限	1 sec	1 sec	2 sec	1 sec
测试点数目	20	20	20	20
每个测试点分值	5	5	5	5
附加样例文件	有	有	有	有
结果比较方式	全文比较 (过滤行末空格与文末回车)			
题目类型	传统	传统	传统	传统
运行内存上限	512 MB	512 MB	512 MB	512 MB

## 二、提交源程序文件名

对于 C++ 语言	bst.cpp	alpha.cpp	garden.cpp	interval.cpp
对于 C 语言	bst.c	alpha.c	garden.c	interval.c
对于 Pascal 语言	bst.pas	alpha.pas	garden.pas	interval.pas

## 三、编译命令 (不包含任何优化开关)

对于 C++ 语言	<code>g++ -o [可执行文件名] [可执行文件名].cpp -lm</code>
对于 C 语言	<code>gcc -o [可执行文件名] [可执行文件名].c -lm</code>
对于 Pascal 语言	<code>fpc [可执行文件名].pas</code>

### 注意事项:

1. 文件名 (程序名与输入输出文件名) 必须使用英文小写。
2. C/C++ 中函数 `main()` 的返回值必须是 `int`, 程序正常结束时返回值必须是 `0`。
3. 全国统一评测时采用的机器配置为: Intel(R) Core(TM) i7-8700K CPU @ 3.70 GHz, 内存 32 GB, 上述时限以此配置为准。
4. 只提供 Linux 格式附加样例文件。
5. 特别提醒: 评测在当前最新公布的 NOI Linux , 其中C++编译器版本以C++14为准

# 1 神奇的二叉查找树 (bst.cpp/c/pas)

## 1.1 问题描述

小 A 穿越到了古老的数据结构国，这个国家的国树叫做二叉查找树。

数据结构国每年会进行一次神秘的祭祀活动，具体来说就是首先选出  $n$  位德高望重的国民，这些国民每人被钦定一个 1 至  $n$  的编号，不存在两个人有相同的编号；然后这  $n$  位国民随机站成一排；接着，国王将会将这  $n$  位国民以编号为权值，从左向右依次放到（插入）他们的国树——二叉查找树上，当然一开始二叉查找树为空。最后我们得到了一棵二叉查找树，称之为  $T$ 。

数据结构国的祖先们发现，国民排列是一个 1 到  $n$  的排列，而且随机站成一排的排列对最后得到的二叉查找树  $T$  的形态有较大的影响。假设当前的国民排列为  $\Pi$ ，数据结构国的祖先经过了多次实验，将来年的幸运值定义为满足下列条件的排列个数：

1. 排列与当前的国民排列  $\Pi$  不同；
2. 按照该排列，以国民编号为权值，从左向右依次放到（插入）一棵初始为空二叉查找树上，最终二叉树与  $T$  完全相同。

数据结构国同时发现，这个幸运值可能很大。但是经过实验他们发现，幸运值并不是越大越好，找规律后发现，幸运值对  $(10^9 + 7)$  取模后的结果才是真正重要的数据。

显然数据结构国的数据结构与算法并不发达，他们发现一共有  $n!$  种不同的排列，对这些排列进行统计并且计算来年的幸运值将会是一件困难的事情，于是他们找到了穿越来的小 A。可惜小 A 学艺不精，于是他希望来自未来的你们能够帮他统计一下，给定当前的国民排列  $\Pi$ ，计算来年的幸运值对  $(10^9 + 7)$  取模后的结果。

小 A 打算在数据结构国居住  $m$  年，于是你需要回答  $m$  个不同的询问。

## 1.2 输入描述

输入文件名为 bst.in。

输入文件第一行，一个数  $m$ ，表示小 A 在数据结构国居住的年数。

接下来  $m$  行，每行表示一年的数据。每行第一个数  $n$ ，表示德高望重的国民数量；接下来  $n$  个数，表示国民排列  $\Pi$ 。

## 1.3 输出描述

输出文件名为 bst.out。

输出文件共  $m$  行，对于每年的数据输出一行，一个数，表示来年的幸运值对  $(10^9 + 7)$  取模后的结果。

## 1.4 输入输出样例 1

### 1.4.1 输入 (bst1.in)

```
3
3 2 1 3
5 3 4 5 1 2
3 1 2 3
```

### 1.4.2 输出 (bst1.ans)

```
1
5
0
```

## 1.5 输入输出样例 2

见选手目录下的 bst/bst2.in 与 bst/bst2.ans。

## 1.6 输入输出样例 1 解释

第一年时，仅有排列  $(2, 3, 1)$  满足条件；

第二年时，5 个排列  $(3, 1, 2, 4, 5), (3, 1, 4, 2, 5), (3, 1, 4, 5, 2), (3, 4, 1, 2, 5), (3, 4, 1, 5, 2)$  满足条件；

第三年时，没有排列满足条件。

## 1.7 数据范围与约定

对于 30% 的数据，满足  $1 \leq n \leq 10$ ；

对于 50% 的数据，满足  $1 \leq n \leq 10^2$ ；

对于 100% 的数据，满足  $1 \leq n \leq 10^3, 1 \leq m \leq 10^2$ 。

## 2 神奇的字母 (alpha.cpp/c/pas)

### 2.1 问题描述

小 A 穿越到了古老的字母国，这个国家的语言仅由两个字母  $\{a, b\}$  组成。

小 A 对这样的语言很是头疼，所以他通过远程电话询问了某位不知名的语言大师。这位大师告诉小 A，这个语言有一个神奇的地方，就是有许多单词表示的是同一个意思。

具体来说，假设小 A 现在得到了一个由字符集  $\{a, b\}$  构成的单词  $s$ ，那么将  $s$  进行下列操作得到的单词与  $s$  具有相同的含义。

1. 将两个连续的  $b$  替换为  $a$ ，例如  $bba$  变为  $aa$ ；
2. 将两个连续的  $a$  替换为  $b$ ，例如  $aab$  变为  $bb$ 。

这样的操作可以进行无限次，得到的单词均与  $s$  具有相同的含义。

现在小 A 想知道，对于自己拿到的这个单词  $s$ ，经过若干次上述操作后（可以不进行操作），有多少个不同的单词与单词  $s$  有相同的含义。小 A 通过远程电话找到了你，不过小 A 只需要你给出答案对于  $(10^9 + 7)$  取模后的结果即可。

由于小 A 还可能遇到很多单词，所以你需要支持多组询问。

### 2.2 输入描述

输入文件名为 `alpha.in`。

输入文件第一行，一个数  $T$ ，表示数据组数。

接下来  $T$  行，一个字符串  $s$ ，表示小 A 拿到的单词。

### 2.3 输出描述

输出文件名为 `alpha.out`。

输出文件共  $T$  行。每行回答一个询问，一个数，表示经过若干次操作后（可以不进行操作），有多少个不同的单词与单词  $s$  有相同的含义，答案对  $(10^9 + 7)$  取模。

### 2.4 输入输出样例 1

#### 2.4.1 输入 (alpha1.in)

```
2
aaaa
aabb
```

#### 2.4.2 输出 (alpha1.ans)

```
6
5
```

### 2.5 输入输出样例 2

见选手目录下的 `alpha/alpha2.in` 与 `alpha/alpha2.ans`。

### 2.6 输入输出样例 1 解释

经过若干次操作，以下 6 个不同的单词与 `aaaa` 有相同的含义： $\{aaaa, aab, aba, baa, bb, a\}$ 。

经过若干次操作，以下 5 个不同的单词与 `aabb` 有相同的含义： $\{aabb, aaa, bbb, ab, ba\}$ 。

## 2.7 数据范围与约定

对于 30% 的数据,  $|s| \leq 10$ ;

对于 50% 的数据,  $|s| \leq 10^3$ ;

对于 100% 的数据,  $|s| \leq 10^5$ ,  $1 \leq T \leq 100$ , 保证  $s$  中只包含  $\{a, b\}$ 。

## 3 神奇的花园 (garden.cpp/c/pas)

### 3.1 问题描述

小 A 在四处游历的时候,发现了一个花园,这个花园是由  $n$  个亭子和  $m$  条无向小路组成的。每个亭子有  $c_i$  种类的花无限朵;每条小路连接两个亭子  $u_i$  与  $v_i$ ,且每条小路有一个长度  $l_i$ 。

小 A 想要逛逛这个花园,于是他制定了  $q$  次旅行计划。每次他从花园的一个亭子  $a_i$  出发,由于他不想走太远的小路,于是在这次旅行计划中,他只能走长度不超过  $b_i$  的小路。他将到达所有他能够到达的亭子,并在每个到达过的亭子采一朵花(同一个亭子只采一束花)。

小 A 想知道,每次旅行结束后,采摘的哪种花最多?如果有若干种花一样多,那么输出种类编号最小的那一种。小 A 还要继续他的旅程,于是他把这个事情交给了你。

特殊地,由于小 A 需要依赖上次旅行计划的结果来制定下次旅行计划,所以你可能需要实时回答他的问题。

### 3.2 输入描述

输入文件名为 garden.in。

第一行,三个数  $n, m, type$ , 表示有  $n$  个亭子和  $m$  条无向小路,  $type$  表示数据种类;

第二行,  $n$  个数,  $c_1, c_2, \dots, c_n$ , 表示每个亭子花的种类;

接下来  $m$  行,每行三个数  $u_i, v_i, l_i$ , 分别表示小路连接的两个亭子和小路的长度;

接下来一行,一个数  $q$ , 表示小 A 的旅行计划总数;

接下来  $q$  行,每行两个数  $a_i, b_i$ , 描述一次旅行计划,分别表示旅行计划的起点和旅行计划中,对于可经过的小路长度的限制。

**【注意】**当  $type = 2$  时,设上一个询问的答案为  $lastans$ ,你需要将输入的  $a_i, b_i$  进行如下操作后得到真实的  $a_i^*, b_i^*$ , 其中  $\oplus$  表示按位异或操作。当  $type = 1$  时不需要进行任何处理。

$$a_i^* = a_i \oplus lastans, b_i^* = b_i \oplus lastans$$

初始时,  $lastans = 0$ 。

### 3.3 输出描述

输出文件名为 garden.out。

对于每个询问,输出一行,一个数,表示手上数量最多的花的种类。如果有若干种花一样多,那么输出种类编号最小的那一种。

### 3.4 输入输出样例 1

#### 3.4.1 输入 (garden1.in)

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

### 3.4.2 输出 (garden1.ans)

2  
1  
3  
1

## 3.5 输入输出样例 2

见选手目录下的 garden/garden2.in 与 garden/garden2.ans。

## 3.6 输入输出样例 1 解释

真正询问的  $(a_i, b_i)$  为:  $\{(1, 1), (2, 2), (4, 4), (5, 8)\}$ 。

## 3.7 数据范围与约定

对于 30% 的数据,  $1 \leq n, m, q \leq 10^3$ ;

对于 100% 的数据,  $1 \leq n \leq 131071, 1 \leq m, q \leq 2 \cdot 10^5, 1 \leq u_i, v_i, c_i, a_i \leq n, 1 \leq l_i, b_i \leq 10^6$ 。

有 45% 的数据分散在各档次数据中, 满足  $type = 1$ ; 其余 55% 数据  $type = 2$ 。

有 25% 的数据分散在各档次数据中, 满足  $1 \leq c_i \leq 20$ 。

## 4 神奇的区间 (interval.cpp/c/pas)

### 4.1 问题描述

小 A 穿越到了古老的区间国，这个国家的由一维数轴上的许多区间构成。

对于一维数轴上的两个区间  $(a, b)$  与  $(c, d)$ ，若  $c < a < d$  或  $c < b < d$ ，则可以从  $(a, b)$  单向走到  $(c, d)$ 。区间国有两种操作：

1. 在区间集合中添加  $(x, y)$  区间，保证新加入的区间的长度一定比之前所有的区间长度要长；
2. 询问是否存在一条路径从第  $x$  个区间走到第  $y$  个区间。

初始时，区间集合为空。小 A 想探究区间国的形成，于是他给你发了一系列操作，想让你模拟一下区间国形成的过程，并回答所有的询问。

### 4.2 输入描述

输入文件名为 interval.in。

输入文件第一行，一个数  $n$ ，表示操作数目；

接下来  $n$  行，每行三个整数  $op, x, y$ ；其中  $op$  表示操作的类别， $op = 1$  表示操作 1， $op = 2$  表示操作 2； $x, y$  表示每个操作的操作数，参见问题描述。

### 4.3 输出描述

输出文件名为 interval.out。

输出文件有若干行，对于每个询问操作，输出一行一个字符串，若存在这样的路径，则输出“YES”，否则输出“NO”（均不含引号）。

### 4.4 输入输出样例 1

#### 4.4.1 输入 (interval1.in)

```
5
1 1 5
1 5 11
2 1 2
1 2 9
2 1 2
```

#### 4.4.2 输出 (interval1.ans)

```
NO
YES
```

### 4.5 输入输出样例 2

见选手目录下的 interval/interval2.in 与 interval/interval2.ans。

### 4.6 输入输出样例 1 解释

只有前两个区间时，区间 1 和区间 2 没有路径相连，故回答“NO”；加入区间 3 后，区间 1 可通过区间 3 到达区间 2，故回答“YES”。



## 4.7 数据范围与约定

对于 30% 的数据,  $1 \leq n \leq 10^3$ ;

对于 100% 的数据,  $1 \leq n \leq 10^5$ , 所有数字绝对值  $\leq 10^9$ , 保证操作合法。