

# NOIP 模拟赛

WZMS

(请选手务必仔细阅读本页内容)

## 一、题目概况

中文题目名称	神奇的树	永远年轻	子序列	双重回文
子目录名	tree	young	subsequence	palindrome
可执行文件名	tree	young	subsequence	palindrome
输入文件名	tree.in	young.in	subsequence.in	palindrome.in
输出文件名	tree.out	young.out	subsequence.out	palindrome.out
每个测试点时限	1000ms	5000ms	2500ms	2000ms
内存上限	512M	512M	1G	512M
测试点数目	20	20	10	25
每个测试点分值	5	5	10	4
附加样例文件	有	有	有	有
结果比较方式	全文比较(过滤行末空格及文末回车)			
题目类型	传统	传统	传统	传统

## 二、提交源程序程序名

对于 C++ 语言	tree.cpp	young.cpp	subsequence.cpp	palindrome.cpp
-----------	----------	-----------	-----------------	----------------

## 三、优化开关

对于 C++ 语言	-O2 -std=c++17
-----------	----------------

### 注意事项:

1. 题目并不难, 请喧哗的同学不要大声 AK;
2. 文件名(程序名和输入输出文件名)必须使用英文小写;
3. C/C++ 中函数 main() 的返回类型必须是 int, 程序正常结束时的返回值必须是 0;
4. 保证各个题目的时间限制至少为标程运行时间的两倍;
5. 题目难度不一定升序排序;
6. 每道题目需单独建立对应子文件夹.

# 1 神奇的树

(tree.cpp)

## 1.1 问题描述

给定一个无向的树，希望你能求出这颗树字典序最小的**先序遍历**。

你可以任意选择一个点作为起点，当你在一个节点时，你只能访问你未被访问的邻居节点。你可以自由地安排访问邻居节点的顺序。当你访问完一个节点所有的邻居结点之后，将当前节点的标号写入当前的先序遍历序列的尾部。

下面给出访问过程的伪代码，G 用来储存一个节点的邻居：

```
1  vector<int> G[N], order;
2  bool vis[N];
3  void dfs(int u) {
4      vis[u] = true;
5      for (int v : G[u]) {
6          if (vis[v]) continue;
7          dfs(v);
8      }
9      order.push_back(u);
10 }
```

其中 order 则为所求的先序遍历序列。你要求出可能的字典序最小的先序遍历序列。

## 1.2 输入

第一行输入一个正整数  $T$ ，表示测试数据组数。

对于每个测试数据，第一行读入一个正整数  $n$  表示给定树的节点个数。

接下来  $n - 1$  行，每行读入两个正整数  $u, v$ ，表示树上有一条无向边连接  $u, v$  两个节点。保证给定图的形态为树。

## 1.3 输出

共  $T$  行，对每一个测试数据输出字典序最小的**先序遍历**。

## 1.4 输入输出样例

### 1.4.1 输入样例

```
3
3
1 3
```

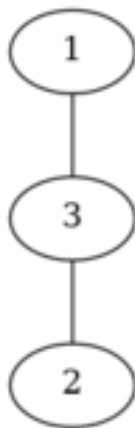
3 2  
3  
2 1  
1 3  
7  
1 2  
1 3  
2 4  
2 5  
3 6  
3 7

#### 1.4.2 输出样例

1 2 3  
2 1 3  
4 5 2 1 6 3 7

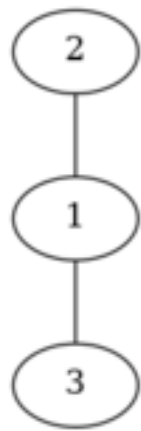
#### 1.5 样例解释

以下为第一个测试数据中给定的树的形态：



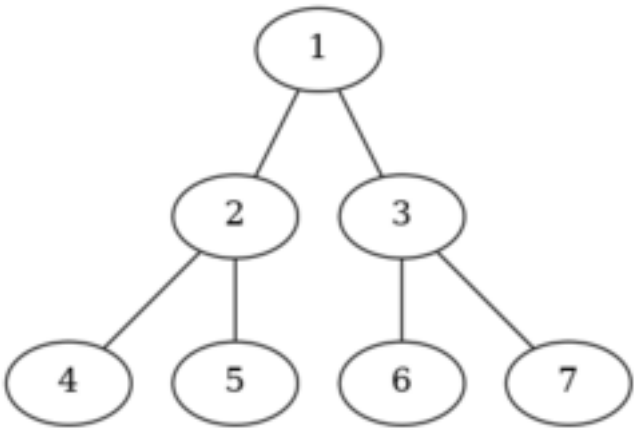
假如说我们从 1 开始遍历，我们只能得到序列 2 3 1，从 2 开始遍历只能得到 1 3 2，从 3 开始遍历，我们可以得到 1 2 3 或者 2 1 3。字典序最小的先序遍历序列为 1 2 3。

以下为第二个测试数据中给定的树的形态：



假如说我们从 1 开始遍历，我们能得到序列 2 3 1 或者 3 2 1，从 2 开始遍历只能得到 3 1 2，从 3 开始遍历，我们只能得到 2 1 3。字典序最小的先序遍历序列为 2 1 3。

以下为第三个测试数据中给定的树的形态：



其中字典序最小的先序遍历序列 4 5 2 1 6 3 7 可以从 7 号节点开始访问。

1.6 约定和数据范围

对所有测试点数据, 保证  $1 \leq T \leq 10^5, 2 \leq n \leq 2 \times 10^5, \sum n \leq 2 \times 10^5, 1 \leq u, v \leq n, u \neq v$ 。

每个测试点的具体限制见下表：

测试点编号	$n \leq$	$\sum n \leq$	特殊性质
1~2	10	100	无
3~6	300	3000	无
7~10	2000	10000	无
11~12	$10^5$	$2 \times 10^5$	特殊性质 A
13~14	$10^5$	$2 \times 10^5$	特殊性质 B
15~20	$10^5$	$2 \times 10^5$	无

特殊性质 A: 给定的树是一条链。

特殊性质 B: 给定的树是一个菊花图。即总共有  $n - 1$  个叶子连接在一个节点下。

## 2 永远年轻

(young.cpp)

### 2.1 问题描述

给定一个由  $n$  个节点构成的无向有根树，其中根为 1 号节点。

我们称一个点  $u$  为另一个节点  $v$  的祖先当且仅当  $u$  在从  $v$  到根的路径上。

然后我们定义集合  $\text{subtree}(u) = \{v | u \text{ 为 } v \text{ 的祖先}\}$ 。

树上每一个节点  $u$  有两个权值  $c_u, e_u$ ，定义一个点  $u$  的  $\text{val}_u$  为：

$$\text{val}_u = \sum_{x \in \text{subtree}(u)} \sum_{y \in \text{subtree}(u)} \min\{|c_x - c_y|, |e_x - e_y|\}$$

请输出树上每一个节点  $u$  输出  $\text{val}_u$ ，按节点标号从小到大输出。由于计算结果可能很大，请将结果对  $10^9 + 7$  取模后输出。

### 2.2 输入

第一行一个正整数  $n$ ，表示树的节点个数。

接下来  $n - 1$  行，每行两个正整数  $u, v$ ，表示树上有一条无向边链接节点  $u, v$ 。保证给定图的形态是树。

接下来  $n$  行，每行读入两个正整数  $c_i, e_i$  表示节点  $i$  的两个权值。

### 2.3 输出

共  $n$  行，第  $i$  行  $\text{val}_i$  对  $10^9 + 7$  取模之后的结果。

### 2.4 输入输出样例

#### 2.4.1 输入样例

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

### 2.4.2 输出样例

44  
12  
0  
0  
0

## 2.5 约定和数据范围

对所有测试点数据, 保证  $1 \leq n \leq 5 \times 10^5, 1 \leq u, v \leq n, u \neq v, 1 \leq c_i, e_i \leq 10^9$ .

每个测试点的具体限制见下表:

测试点编号	$N \leq$	特殊性质
1 ~ 4	2000	无
5 ~ 8	5000	无
9 ~ 10	$5 \times 10^5$	特殊性质 A
11 ~ 12	$5 \times 10^5$	特殊性质 B
13 ~ 16	$10^5$	无
17 ~ 20	无	无

特殊性质 A: 给定的树是一条链。

特殊性质 B: 给定的树是一个菊花图。即总共有  $n - 1$  个叶子连接在一个节点下。

### 3 子序列

(subsequence.cpp)

给定一个长度为  $n$  的正整数序列  $a$ ，和一个长度为  $m$  的正整数序列  $b$ 。我们需要从  $a$  中选出一个非空子序列  $p_a$ ，从  $b$  中选出一个非空子序列  $p_b$ 。我们这里定义一个长度为  $k$  的子序列  $p_1, p_2, \dots, p_k$  的权值为：

$$\sum_{i=1}^k p_i \times 1000^{k-i}$$

现在想知道有多少种在  $a, b$  中分别选择子序列的方案，能使得  $p_a$  的权值严格大于  $p_b$  的权值。

我们认为两种选择子序列的方案  $(A_1, B_1), (A_2, B_2)$  不同，当且仅当  $A_1 \neq A_2$  或者  $B_1 \neq B_2$ 。两个子序列  $p, q$  不同当且仅当其权值不同。

由于答案可能很大，请将其对 998244353 取模后输出。

#### 3.1 输入

第一行读入两个正整数  $n, m$  表示  $a$  的长度和  $b$  的长度。

接下来一行读入  $n$  个正整数，其中第  $i$  个数表示  $a_i$ 。

接下来一行读入  $m$  个正整数，其中第  $i$  个数表示  $b_i$ 。

#### 3.2 输出

输出一个非负整数，表示答案对 998244353 取模之后的结果。

#### 3.3 输入输出样例 1

##### 3.3.1 输入样例

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

##### 3.3.2 输出样例

```
22
```

#### 3.4 样例解释

在  $a$  能选出的子序列有  $\{1\}, \{2\}, \{1, 2\}, \{2, 1\}, \{2, 2\}, \{2, 1, 2\}$ ，他们对应的权值为 1, 2, 1002, 2001, 2002, 2001002。在  $b$  中能选出的子序列有  $\{1\}, \{2\}, \{1, 1\}, \{1, 2\}, \{2, 1\}, \{2, 2\}, \{1, 1, 2\}, \{1, 2, 1\}, \{1, 2, 2\}, \{2, 1, 2\}, \{2, 2, 2\}$ ，他们对应的权值为 1, 2, 1001, 1002, 2001, 2002, 1001002, 1002001, 1002002, 2001002, 2002002。



那么其中有  $0 + 1 + 3 + 4 + 5 + 9 = 22$  种选择子序列的方案，能使得  $p_a$  的权值严格大于  $p_b$  的权值。

### 3.5 约定和数据范围

对所有测试点数据, 保证  $1 \leq n, m \leq 5000, 1 \leq a_i, b_i \leq 100$ 。

每个测试点的具体限制见下表：

测试点编号	$N \leq$	$M \leq$
1 ~ 2	10	10
3 ~ 4	100	100
5 ~ 6	300	300
7 ~ 8	1000	1000
9 ~ 10	5000	5000

## 4 双重回文

(palindrome.cpp)

### 4.1 问题描述

定义一个字符串是回文的，当且仅当该字符串的正串等于其反串。比如说 aba 和 oppo 是回文的，但 wzms 不是回文的。

定义一个字符串是双重回文的，当且仅当该字符串本身是回文的，或者该字符串可以看成两个非空回文串的顺序拼接（这两个回文串可以相同）。比如说 abacdda, potato 和 xjxx 是双重回文的。而 wzmstxdy 和 abaabb 不是双重回文的。

现在你要求出长度不超过  $n$  的仅用前  $k$  个小写字母构成的非空字符串中双重回文串的数量。由于答案可能很大，请对 998244353 取模后输出。

### 4.2 输入

第一行两个正整数  $n, k$ 。表示最长的字符串长度，和字符集的大小。

### 4.3 输出

输出一个整数表示长度不超过  $n$  的仅用前  $k$  个小写字母构成的非空字符串中双重回文串的数量。答案对 998244353 取模。

### 4.4 输入输出样例 1

#### 4.4.1 输入样例

3 3

#### 4.4.2 输出样例

33

### 4.5 输入输出样例 2

#### 4.5.1 输入样例

6 2

#### 4.5.2 输出样例

114

## 4.6 输入输出样例 3

### 4.6.1 输入样例

42 7

### 4.6.2 输出样例

83419789

## 4.7 样例解释

第一个样例中的双重回文字符串为: a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, aaa, aab, aac, aba, abb, aca, acc, baa, bab, bba, bbb, bbc, bcb, bcc, caa, cac, cbb, cbc, cca, ccb, ccc.

## 4.8 约定和数据范围

对所有测试点数据, 保证  $1 \leq n \leq 10^5, 1 \leq k \leq 26$ 。

每个测试点的具体限制见下表:

测试点编号	$n \leq$	$k \leq$
1 ~ 2	10	2
3 ~ 4	10	26
5 ~ 6	20	2
7 ~ 8	20	26
9 ~ 10	1000	2
11 ~ 12	1000	26
13 ~ 14	5000	26
15 ~ 17	$5 \times 10^4$	26
18 ~ 20	$10^5$	26