

编译命令为-O2 -std=c++11 -lm

## 1 树上四次求和

文件名称: `sumsumsumsum.cpp`/in/out

时间限制: 2s

空间限制: 512MB

### 1.1 题目背景

对于一棵树, 我们定义  $dis(i, j)$  为节点  $i, j$  之间最短路径上的边数。

对于一个长度为  $n$  的序列  $a$ , 我们定义  $w(l, r) (1 \leq l \leq r \leq n)$  为  $\sum_{i=l}^r \sum_{j=i}^r dis(a_i, a_j)$ 。

### 1.2 题目描述

给你一棵  $n$  个点的树以及一个  $1-n$  的排列  $a$ , 有  $q$  次询问, 每次给出  $k$ , 求  $\sum_{i=1}^k \sum_{j=i}^k w(i, j)$  对 998244353 取模的值。

### 1.3 输入格式

第一行两个正整数  $n, q$ 。

接下来  $n-1$  行, 每行两个正整数  $u, v$ , 表示  $u, v$  间有一条树边。

接下来一行  $n$  个数字描述排列  $a$ 。

接下来  $q$  行每行一个正整数  $k_i$  表示询问。

### 1.4 输出格式

输出  $q$  行, 第  $i$  行表示第  $i$  次询问的答案。

### 1.5 样例数据

#### 1.5.1 样例 1 输入

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

### 1.5.2 样例 1 输出

0  
1  
6  
21

### 1.6 数据范围

对于前 30% 的数据,  $n \leq 1000$ ;  
对于另 20% 的数据,  $q_i = n$ ;  
对于另 20% 的数据, 第  $i$  条边连接点  $i$  与点  $i + 1$ ;  
对于 100% 的数据,  $n, q \leq 10^5, u, v, k_i \leq n$ 。

## 2 Cubelia

文件名称: cubelia.cpp/in/out

时间限制: 5s

空间限制: 512MB

### 2.1 题目背景

Cubelia 喜欢思考关于最大前缀和的问题。她认为一个长度为  $n$  的序列  $A$  的最大前缀和为  $\forall j \in [1, n], \sum_{i=1}^j A_i$  的最大值。

同时, 她喜欢用  $a_{l,r}$  表示序列  $a$  的连续子序列  $[a_l, a_{l+1}, \dots, a_r]$ 。

### 2.2 题目描述

Cubelia 有一个长度为  $n$  的序列  $a$  和  $q$  次询问, 每次询问给出  $l, r$ 。对于每次询问, 她想要知道  $a_{l,r}$  的所有连续子序列的最大前缀和的和。

由于 Cubelia 无法与你直接交流, 她的询问将以一种特殊的方式给出, 你的回答也需要以特殊形式输出。具体见输入输出格式。

### 2.3 输入格式

第一行两个正整数  $n, q$ 。

第二行  $n$  个整数, 描述 Cubelia 的序列  $a$ 。

第三行五个正整数  $S, A, B, P, tp$ 。

对于每次询问的  $l, r$ , Cubelia 定义了一个随机函数如下:

```
int S, A, B, P, tp;
long long lastans;
int Rand()
{
    S = ( S * A % P + ( B ^ ( tp * lastans ) ) ) % P;
    S = S < 0 ? -S : S;
    return S;
}
```

其中 lastans 表示上一次询问的答案, 初值为 0。

每次询问时, 你需要以如下方式生成  $l, r$ :

```
int l = Rand() % n + 1, r = Rand() % n + 1;
if ( l > r )
    swap ( l, r );
```

为了方便, 你可以 (务必) 以下发的 cubelia\_1.cpp 或 cubelia\_2.cpp 为模板编写你的程序。

## 2.4 输出格式

一行一个正整数，表示所有询问的答案的和对 998244353 取模的值。如果所有询问的答案和是负数，你需要将其加上  $k * 998244353$  后输出其对 998244353 取模的值。其中  $k$  是一个远大于所有询问的答案和的正整数。

## 2.5 样例数据

### 2.5.1 样例 1 输入

```
5 3
55419 228586 -483578 -148471 -100617
907442592 999221847 909035853 910239943 1
```

### 2.5.2 样例 1 输出

```
996966677
```

### 2.5.3 样例 1 解释

三次询问的端点和答案分别如下：

```
2 3
-26406
3 5
-1848293
1 3
597023
```

注意不要对 lastans 进行取模。

## 2.6 数据范围

对于前 10% 的数据,  $q \leq 10$ ;

对于前 40% 的数据,  $tp = 0$ ;

对于前 70% 的数据,  $q \leq 10^5$ ;

对于前 90% 的数据,  $n \leq 10^5$ ;

对于 100% 的数据,  $n \leq 2 * 10^6, q \leq 10^7, S, A, B, P \leq 10^9, tp \leq 1$ , 保证  $a$  的任意一个连续子序列的最大前缀和不超过  $10^6$ .

## 3 Cuvelia

文件名称: skyhigh.cpp/in/out

时间限制: 2s

空间限制: 512MB

### 3.1 题目背景

对于一棵树, Cuvelia 定义  $dis(i, j)$  为节点  $i, j$  之间最短路径上的边数。

### 3.2 题目描述

Cuvelia 有一棵  $n$  个节点的树, 和  $q$  个询问, 第  $i$  个询问中给出了  $k_i$  个节点。对于每次询问, Cuvelia 想要知道有多少不同的节点  $m$  满足如果在这次询问中给出的点中选择任意两点  $u, v$ , 都有  $dis(m, u) = dis(m, v)$ 。

### 3.3 输入格式

第一行两个正整数  $n, q$ 。

接下来  $n - 1$  行, 每行两个正整数  $u, v$ , 表示  $u, v$  间有一条树边。

接下来  $q$  行每行先是一个正整数  $k_i$ , 之后是  $k_i$  个互不相同的正整数描述这次询问的点。

### 3.4 输出格式

输出  $q$  行, 第  $i$  行表示第  $i$  次询问的答案。

### 3.5 样例数据

#### 3.5.1 样例 1 输入

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

### 3.5.2 样例 1 输出

10  
6  
1  
1  
0

### 3.6 数据范围

对于前 10% 的数据,  $k_i \leq 1$ ;

对于前 40% 的数据,  $k_i \leq 2$ ;

对于前 70% 的数据,  $k_i \leq 3$ ;

对于 100% 的数据,  $n \leq 3 * 10^5, \sum k_i \leq 3 * 10^5$ .

### 3.7 提示

注意读入效率对程序运行时间的影响。