

恩欧挨批摸你尸体

题目名称	有脚就行	有手就行	有头就行	有脑就行
英文题目名称	foot	hand	head	brain
源程序文件名	foot.cpp	hand.cpp	head.cpp	brain.cpp
输入文件名	foot.in	hand.in	head.in	brain.in
输出文件名	foot.out	hand.out	head.out	brain.out
每个测试点时限	1s	4s	1s	2s
内存上限	512MB	512MB	512MB	512MB
测试点数目	10	10	10	10
每个测试点分值	10	10	10	10
题目类型	传统	传统	传统	传统
比较方式	忽略行末空格及 多余制表符	忽略行末空格及 多余制表符	忽略行末空格及 多余制表符	忽略行末空格及 多余制表符
编译选项	-lm	-lm	-lm	-lm

Q:题目水不水啊?

A:题目很良心，是真正意义上的良心，AK了请不要大声喧哗，可以提前离场(

Q:代码长度有没有限制，我可以打表吗?

A:每道题源代码长度限制均为 50KB。

Q:卡不卡常啊

A:每道题的时限都是标程时间的两倍以上（

Q:有没有部分分啊

A:虽然并不一定每道题的数据都有梯度，但请尽量优化算法，争取拿高分(

Q:网上有没有题解啊我想闷声发大财

A:题目基本不是原创的，如有雷同，纯属巧合(

Q:题目出锅怎么办

A:如果对于题目有任何疑问可以PM deaf | wangdy

Q:有没有什么别的事要注意啊

A:建议最后 10 分钟不要再编程，检查一下提交的文件夹中的代码是否符合要求，检查文件名，输入输出文件名，数据类型，数据精度，空间限制，赋初值等是否按试卷上的要求来做的，一定要杜绝一切的不小心的人为错误，显然这种错误是致命的。做题时，审题是关键，必须深入与全面，学过的知识与做过的题都是分析问题的有利武器；编写代码要细致，多写函数，便于调试，只有这样，才能达到你的期望（

有脚就行

题目描述

给你一个句子，这个句子由 n 个只包含 $a - z$ 的单词组成。

定义两个句子 S, T 相似，当且仅当：

- 两个句子中的单词构成的可重集合(*multiset*)相等。
- 对于 S 和 T 中出现过的任意一个单词 W 以及任意一个 $k(k \leq W$ 在 S 和 T 中出现的次数), W 在 S 中第 k 次出现的位置与 W 在 T 中第 k 次出现的位置相差不超过1。

现在给出一个句子 S ，求与 S 相似的句子个数。对 $10^9 + 7$ 取模。

输入格式

第一行，一个整数 n 。

第二行， n 个单词，表示一个句子。

输出格式

输出一行一个整数，表示答案。

样例输入1

```
4
n o i p
```

样例输出1

```
5
```

样例解释1

noip,

onip,

onpi,

niop,

nopi.

样例输入2

```
6
yao yao si wu yao si
```

样例输出2

```
8
```

样例输入/输出3

见下发文件中的foot_sample_3.in/out

说明

对于60%测试点, $n \leq 3000$ 。

对于所有测试点, $n \leq 10^5$, 单词长度小于等于10。

有手就行

题目描述

你有 n 个 $galgame$ 。因为你觉得同一个 $galgame$ 玩多了就没意思了要雨露均沾，所以每个 $galgame$ 你打算最多玩3次。对于第 i 个，当你第一次玩的时候会获得 a_i 点快乐，当你第二次玩的时候会获得 b_i 点快乐，当你第三次玩的时候又会获得 a_i 点快乐。

我们记玩 k 次 $galgame$ 能获得的最大快乐值为 $f(k)$ 。现在给定 m ，请求出 $f(1) \text{ xor } f(2) \text{ xor } \dots \text{ xor } f(m)$ 。 xor 表示异或，在 $C++$ 中其运算符为 \wedge 。

输入格式

为了防止本题输入量过大，我们使用一种随机算法构造数据。你只需要读入一行四个整数： n, m 和随机数种子 $k1, k2$ ，然后调用以下程序中的 $gen(n, k1, k2)$ ；就可以得到数组 a 和 b 。

```
#define ull unsigned long long
const int threshold=10000000;
ull k1,k2;
ull Rand(){
    ull k3=k1,k4=k2;
    k1=k4;
    k3^=(k3<<23);
    k2=k3^k4^(k3>>17)^(k4>>26);
    return k2+k4;
}
int a[maxn],b[maxn]; //数组a和b
void gen(int n,ull _k1,ull _k2){
    k1=_k1,k2=_k2;
    for(int i=1;i<=n;i++){
        a[i]=Rand()%threshold+1;
        b[i]=Rand()%threshold+1;
    }
}
```

输出格式

一行，表示答案。

样例输入1

```
2 6 123456789 987654321
```

样例输出1

```
6935157
```

样例解释1

$$a_1 = 406905, a_2 = 491922$$

$$b_1 = 1803337, b_2 = 4734236$$

$$f(1) = 491922, f(2) = 5226158, f(3) = 5718080, f(4) = 7436400, f(5) = 7928322, f(6) = 8335227$$

你可以据此检查你的数据生成是否正确。

样例输入2

```
10 20 19260817 71806291
```

样例输出2

```
41506271
```

样例输入/输出3

见下发文件中的hand_sample_3.in/out

说明

对于60%的数据, $n \leq 2000$

对于100%的数据, $1 \leq n \leq 5 \times 10^6, m \leq 3n, k1, k2 \leq 10^{12}$

有头就行

题目描述

你有一棵树， S 是在树上的一个点集，且这个点集构成一个连通块。

定义 S 的价值 $f(S)$ 为：取出一个路径集合，这个路径集合内的路径上的所有点的并集构成的点集为 T ，要求 $T \cap S = \emptyset$ 且在树上与 S 中的点有边相连且不属于 S 的点必须属于 T 。满足条件的路径集合的个数就是 $f(S)$ 。注意，路径是无向的，即路径 (a, b) 与路径 (b, a) 完全等价，视为同一条路径。允许路径 (a, a) 。

现在，请求出所有满足条件的 S 的 $f(S)$ 之和。空集不计入答案。

输入格式

第一行，一个整数 n ，表示树的大小。

接下来 $n - 1$ 行，每行两个整数 x, y ，表示树上的一条边。

输出格式

一行，表示答案对998244353取模的结果。

样例输入1

```
2
1 2
```

样例输出1

```
3
```

样例输入2

```
3
1 2
2 3
```

样例输出2

```
16
```

样例输入/输出3

见下发文件中的head_sample_3.in/out

说明

对于20%的数据， $n \leq 10$ 。

对于所有的数据， $n \leq 2 \times 10^5$ 。

由于wangdy || deaf并没有想到 n^2 的暴力怎么写，所以就没有 n^2 的暴力分，如果你想到了可以D wangdy || deaf，让他们加上。

有脑就行

题目描述

你有一棵树，树上的边分为三类:1, 2, 3。

从树上某个点 s 开始行走，不能走重复的边，在走过第一个3类边之前，1类边和2类边都可以走，在走过第一个3类边后，就只能走1类边了。在这种规则下，如果能走到点 t ，就称 s 可达 t 。换句话说，对于树上两个点 s, t ，如果 s, t 之间的路径上只有1, 2类边，或者只有一条3类边且路径在这条3类边靠近 s 的这一部分仅包含1, 2类边，路径在这条3类边靠近 t 的这一部分仅包含1类边，则称 s 可达 t 。特殊地， s 可达 s 。

注意，可达关系是单向的， s 可达 t 并不意味着 t 可达 s 。

现在你需要支持一些操作。每个操作包含4个数 a, b, t, s ，其含义为：首先把树上节点 a, b 之间的边的类别减一，即3类边变成2类边，2类边变成1类边，1类边还是1类边。保证树上 a, b 之间有边。然后一行输出两个整数，第一个输出0或1，表示 s 是否可达 t ，第二个输出一个整数，表示满足 x 可达 t 的 x 有多少个，包含 t 本身。

输入格式

第一行，两个整数 n, m ，表示树的大小和操作个数。

接下来 $n - 1$ 行，每行两个整数，表示树上的一条边。

接下来 m 行，每行四个整数 a, b, t, s ，表示一个操作。

输出格式

对于每个操作，输出一行两个整数表示这次操作的答案。

样例输入1

```
6 5
1 2 3
2 3 3
1 4 3
3 5 3
2 6 3
2 6 1 5
3 5 2 4
2 3 4 4
2 3 2 4
1 2 6 2
```

样例输出1

```
0 4
0 5
1 2
0 5
1 5
```

样例输入/输出2

见下发文件中的brain_sample_2.in/out

说明

对于20%的数据, $1 \leq n, m \leq 2000$ 。

另有20%的数据, 满足只存在形如 $(i, i + 1)$ 的边。

另有20%的数据, 满足只存在形如 $(1, i)$ 的边。

对于所有数据, $1 \leq n, m \leq 10^5$ 。