

# NOIP2024 模拟赛

—出题人: cbj

2024年 10 月 29 日      7: 30 ~ 12: 00

题目名称	随机游走	分发奖励	卡路里	传话游戏
题目类型	传统型	传统型	传统型	传统型
目录	walk	reward	calorie	message
可执行文件名	walk	reward	calorie	message
输入文件名	walk.in	reward.in	calorie.in	message.in
输出文件名walk	walk.out	reward.out	calorie.out	message.out
每个测试点时限	1.0 秒	4.0 秒	1.0 秒	1.0 秒
内存限制	512 MiB	2048 MiB	512 MiB	512 MiB
测试点数目	10	25	20	25
测试点是否等分	是	是	是	是

## 编译选项

-lm -O2 -std=c++14 -Wl,-stack=1000000000

# 随机游走

## 题目背景

本题读入量较大，请选用较为快速的读入方式。

## 题目描述

有一棵以 1 为根节点大小为  $n$  的树，小 A 目前在根节点，他的目标是随机游走直至走完树上的所有节点并回到根节点，每次他随机游走的方式如下：

1. 如果他存在没有走到的儿子，那就在没有走到的儿子中随机选择，然后花费的时间为该点与这个儿子连边的边权。
2. 否则回到父亲节点，由于某种神奇的力量，小 A 无需花费任何时间就可以返回，但前提是需走过该节点子树内的所有节点。

但是小 A 并不喜欢没完没了的随机游走，更不想在很晚的时候才走到他想经过的结点，于是他给每个节点设置了个重要度，其中第  $i$  个节点的重要度为  $w_i$ ，设他刚从根节点开始随机游走的时刻为 0，那么一种随机游走的路线给他造成的厌烦度计算方法如下：

设到达  $i$  号节点所用的时间为  $t_i$ ，那么他的厌烦度为  $\sum_{i=1}^n t_i w_i$ 。

显然小 A 希望他的厌烦度越小越好，所以他想知道，在众多随机游走的路线当中，厌烦度最小的路线的厌烦度是多少。

## 输入格式

第一行一个整数  $n$ 。

接下来  $n - 1$  行，第  $i$  行两个整数  $f_i, v_i$  分别代表  $i + 1$  的父亲以及  $i + 1$  与  $f_i$  连边的边权。相邻两个数以一个空格隔开（下同）。

最后一行  $n$  个整数，第  $i$  个整数为  $w_i$ 。

## 输出格式

共一行一个整数，代表代表答案。

## 样例 #1

### 样例输入 #1

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

### 样例输出 #1

150

### 样例 #2

#### 样例输入 #2

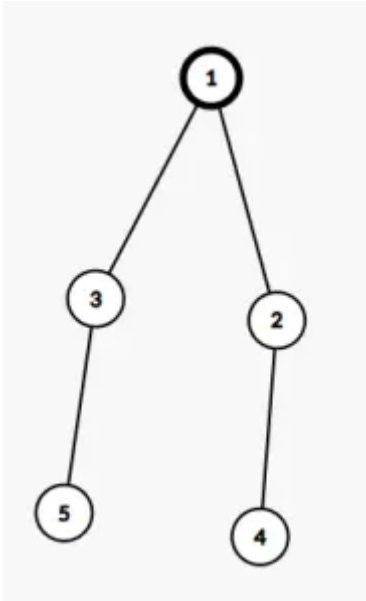
11  
1 5  
1 1  
3 4  
4 10  
3 8  
1 4  
3 7  
5 4  
8 3  
9 3  
9 10 10 6 8 10 3 4 4 6 10

### 样例输出 #2

1631

### 提示

【样例 #1 解释】



如上图，小 A 可以先走到 2 号节点，此时  $t_2 = 5$ ，再走到 4 号节点，此时  $t_4 = 9$ ，此时会先回到 2 号节点再回到 1 号节点，然后再走到 3 号节点，此时  $t_3 = 14$ ，再走到 5 号节点，此时  $t_5 = 19$ ，最后先回到 3 号节点再回到 1 号节点结束这段随机游走。

此时的厌烦度为  $0 \times 5 + 5 \times 4 + 14 \times 2 + 9 \times 5 + 19 \times 3 = 150$ 。

小 A 还可以先走到 3 号节点，此时  $t_3 = 5$ ，再走到 5 号节点，此时  $t_5 = 10$ ，此时会先回到 3 号节点再回到 1 号节点，然后再走到 2 号节点，此时  $t_2 = 15$ ，再走到 4 号节点，此时  $t_4 = 19$ ，最后先回到 2 号节点再回到 1 号节点结束这段随机游走。

此时的厌烦度为  $0 \times 5 + 15 \times 4 + 5 \times 2 + 19 \times 5 + 10 \times 3 = 195$ 。

显然小 A 只有这两种随机游走的方式，最小厌烦度为 150。

【其他样例】

所有样例包含在文件夹 `A walk` 中，其中：

- 1. `ex_walk1.in/out` 为样例组 #1。
- 2. `ex_walk2.in/out` 为样例组 #2。
- 3. `ex_walk3.in/out` 满足测试点编号 6 ~ 7 的限制。

【数据范围】

对于全部测试点，满足  $1 \leq n \leq 5 \times 10^5, 1 \leq f_i \leq i, 1 \leq v_i, w_i \leq 10^3$ ，每个测试点 10 分。

测试点编号	$n \leq$	$f_i$	$w_i$
1	11	无特殊限制	无特殊限制
2	$5 \times 10^4$	$\geq i - 5$	无特殊限制
3 ~ 4	1000	$\geq i - 15$	无特殊限制
5	$5 \times 10^5$	$= i$	无特殊限制
6 ~ 7	$5 \times 10^5$	$= 1$	无特殊限制
8 ~ 9	$5 \times 10^5$	无特殊限制	$= 1$
10	$5 \times 10^5$	无特殊限制	无特殊限制

分发奖励

题目背景

本题输入输出量较大，请选择较为快速的输入输出方式。

题目描述

小 A 开了家公司，身为这家公司的老板，因为一个人管理不了太多的事务，所以雇了  $n$  个员工来发展该公司，其中 1 号员工因为是公司的元老且最勤劳能干，因此他被聘为**总经理**，负责掌管公司内的所有事务。

公司有很多部门，一开始除了总经理（即 1 号员工），所有员工都会被分到其中一个部门里（需保证每个部门里都**至少有一个人**），如果这个部门的人数  $> 1$ ，那么这个部门就需要**一位部长**来主导，其余的人将会再被分到若干**分部门**里。特别地，你可以认为**整个公司也算一个部门**，其中的**部长**就是总经理（1 号员工），同样如果一个部门只有 1 个人，那么则认为这个人就是这个部门的**部长**，同时这个部门**不存在分部门**。

这样有一个好处是每个部门的部长只需要管理每个分部门即可，而他们也只需要跟分部门的部长联络交流，我们也称该部长是**他的分部门部长的上级**；同理，该部长的**每个分部门部长都是他的下级**。

而本年小 A 的公司业绩非常突出，他决定给员工奖励，奖励一共有  $q$  种，由于每个部门分管的事务不同，同时经常会出现两个部门互相合作的情况，所以在分发奖励的时候，他会把第  $i$  种奖励分发给  $a_i$  和  $b_i$  号员工（**请注意  $a_i$  和  $b_i$  可能相等**），并要求  $a_i, b_i$  把该奖励**一部分留给自己**，另一部分分给他们的**所有下级**，当  $a_i, b_i$  把该奖励传给下级之后，也会提醒他们把奖励拿出一部分**分给他们的下级**.....以此类推，直到某个员工没有下级，奖励分发才会到他那停止，此时他会拿走分发到他手里的所有奖励。

现在小 A 很好奇，对于每个员工  $i$ ，有多少员工（不包括自己）拿到的奖励和他**有一样的**？小 A 当然不可能再多花时间去问每个员工他们拿到的奖励有哪些，所以他希望你快速解决这个问题。

## 输入格式

第一行两个整数  $n, q$ 。

接下来  $n - 1$  个整数，第  $i$  个整数  $p_i$  代表  $i + 1$  号员工的**上级**，显然 1 号员工**没有上级**。

接下来  $q$  行，第  $i$  行两个整数  $a_i, b_i$ 。

## 输出格式

共一行  $n$  个整数，**每两个整数之间需要用一个空格隔开**，其中第  $i$  个整数代表拿到的奖励存在和员工  $i$  一样的员工数量。

## 样例 #1

### 样例输入 #1

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

### 样例输出 #1

```
0 2 1 3 2
```

## 提示

### 【数据范围】

对于全部测试点，保证  $1 \leq n \leq 5 \times 10^5, 0 \leq q \leq 5 \times 10^5, 1 \leq p_i \leq i - 1, 1 \leq a_i, b_i \leq n$ ，**请注意  $a_i$  可能等于  $b_i$** ，每个测试点 4 分。

测试点编号	$n \leq$	$q \leq$	$p_i$	$a_i, b_i$
1 ~ 4	500	500	无特殊限制	无特殊限制
5 ~ 6	3000	3000	无特殊限制	无特殊限制
7 ~ 9	6000	6000	无特殊限制	无特殊限制
10 ~ 12	20000	20000	无特殊限制	无特殊限制
13 ~ 15	$5 \times 10^5$	100	无特殊限制	无特殊限制
16	$5 \times 10^5$	$5 \times 10^5$	$= i - 1$	无特殊限制
17	$5 \times 10^5$	$5 \times 10^5$	$= 1$	无特殊限制
18 ~ 21	$5 \times 10^5$	$5 \times 10^5$	无特殊限制	$a_i = b_i$
22 ~ 25	$5 \times 10^5$	$5 \times 10^5$	无特殊限制	无特殊限制

# 卡路里

## 题目背景

本题输入量较大，请选用较为快速的读入方式。

## 题目描述

小 A 喜欢喝奶茶，但现在的他处于减肥阶段，但是嘴馋的他还是防不住奶茶的诱惑，有一天他花了 114514 元买了  $n$  张不同的奶茶免单券，其中第  $i$  张免单券能用于第  $i$  款奶茶的免单（且只能一次），有  $m$  家同一连锁的奶茶店顺次分布在一条街上，从左往右数第  $i$  家店距离它的下一家第  $i + 1$  家店为  $d_i$ ，小 A 每走 1 单位距离的路程都会消耗 1 单位的卡路里。

虽然这  $m$  家店都是同一连锁店，但是在商家制作奶茶的时候可能会有偏差，例如有一家会把奶茶的料放多（不管是有意还是无意），这会导致同一款奶茶在各个店能使人体增加的卡路里数不同，具体地，从左往右数第  $i$  家店的第  $j$  款奶茶的卡路里数为  $a_{i,j}$ 。

小 A 今天的目标是在尽量不长胖的情况下喝光能免单的  $n$  款奶茶，他会从任意一家奶茶店出发（可以先在当前奶茶店使用免单券并喝掉对应的奶茶），每次走到一家店然后使用免单券并喝掉对应的奶茶，直至  $n$  张免单券全部用光并喝完这  $n$  款奶茶为止，小 A 可以在一家奶茶店用任意多张免单券。

但是小 A 很害怕长胖，所以他想知道，在所有可能的喝奶茶方案中，最坏情况下会使体内增加多少卡路里？

## 输入格式

第一行两个整数  $n, m$ 。

第二行  $m - 1$  个整数  $d_1, d_2, \dots, d_{m-1}$ 。

接下来  $m$  行，第  $i$  行  $n$  个整数  $a_{i,1}, a_{i,2}, \dots, a_{i,n}$ 。

## 输出格式

共一行一个整数代表答案。

# 样例 #1

## 样例输入 #1

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

## 样例输出 #1

```
19
```

# 提示

### 【其它样例】

所有样例包含在文件夹 `c calorie` 中，其中：

- 1. `ex_calorie1.in/out` 为**样例组 #1**。
- 2. `ex_calorie2.in/out` 满足测试点编号 6 ~ 7 的限制。
- 3. `ex_calorie3.in/out` 满足测试点编号 8 ~ 10 的限制。
- 4. `ex_calorie4.in/out` 满足测试点编号 11 ~ 20 的限制。

### 【数据范围】

对于全部测试点，满足  $1 \leq n \leq 200, 2 \leq m \leq 5000, 1 \leq d_i, a_{i,j} \leq 10^9$ ，每个测试点 5 分。

测试点编号	$n \leq$	$m \leq$	特殊性质
1	200	5000	$a_{1,j}, a_{2,j}, \dots, a_{m,j}$ 完全相同
2 ~ 3	5	5	无
4 ~ 5	200	5000	$a_{i,1}, a_{i,2}, \dots, a_{i,n}$ 完全相同
6 ~ 7	50	50	无
8 ~ 10	200	500	无
11 ~ 20	200	5000	无

# 传话游戏

## 题目描述

小 A 召集了  $n$  位同学来玩传话游戏，游戏规则非常简单，首先由小 A 生成一个长度为  $m$  的数字序列然后传给第 1 位同学，第 1 位同学接收到这个序列之后有一定的记忆时间，记忆完之后将会传给第 2 位同学，第 2 位同学接收到这个序列之后再进行一次一定时间的记忆后传给第 3 位同学.....以此类推。直到第  $n$  位同学接收到这个序列。显然传话游戏的最大看点就是每个人在传话的时候**可能会遗忘部分东西**。此时

我们会将第  $n$  位同学接收到的序列与第 1 位同学接收到的序列做比较，看看哪些信息是从头完好无损地传到尾的。

设第  $i$  位同学接收到的序列为  $S_i$ ，由于每位同学传话都非常的谨慎，宁愿**少传**也不愿**多传或错传**，所以每位同学在传话的时候，如果有些数字记忆但凡有一点点不清楚，他是不会将它们传给下一位同学的。所以每位同学接收到的序列一定是上一位同学接收到的序列里**保留的若干元素**（可能**全保留**也可能**一个都保留不了**，同时这些元素被传出去的**相对顺序不会发生变化**）所组成的。

现在小 A 不想知道最终的游戏结果，他更想知道在整个过程中有多少**本质不同的**用序列组成的序列  $S$ （即  $[S_1, S_2, \dots, S_n]$ ），两个长度相同的**用序列组成的序列**  $S, T$  不同，当且仅当存在一个  $i$  使得  $S_i \neq T_i$ ，由于答案可能非常大，所以你只需要输出答案对  $10^9 + 7$  取模的结果即可。

## 输入格式

共两行，第一行两个整数  $n, m$ 。

由于  $S_1$  始终是已知且固定的，所以第二行共  $m$  个整数，其中第  $i$  个整数代表  $(S_1)_i$ 。

## 输出格式

共一行一个整数，代表答案对  $10^9 + 7$  取模的结果。

## 样例 #1

### 样例输入 #1

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

### 样例输出 #1

```
12
```

## 样例 #2

### 样例输入 #2

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

### 样例输出 #2

```
176
```

## 样例 #3

### 样例输入 #3

```
1000000000 4
1 2 1 2
```



# 样例输出 #3

1617

## 提示

### 【样例 #1 解释】

$S_1 = [1, 2, 1, 2]$ , 所有可能的  $S_2$  如下:

$[], [1], [2], [1, 1], [1, 2], [2, 1], [2, 2], [1, 1, 2], [1, 2, 1], [1, 2, 2], [2, 1, 2], [1, 2, 1, 2]$ 。

综上, 所有可能的  $[S_1, S_2]$  一共有 12 种。

### 【其它样例】

所有样例包含在文件夹 `D message` 中, 其中:

- 1. `ex_message1.in/out` 为**样例组 #1**。
- 2. `ex_message2.in/out` 为**样例组 #2**。
- 3. `ex_message3.in/out` 为**样例组 #3**。
- 4. `ex_message4.in/out` 满足测试点编号 9 ~ 12 的限制。
- 5. `ex_message5.in/out` 满足测试点编号 13 的限制。

### 【数据范围】

对于全部测试点, 满足  $1 \leq n \leq 10^9, 1 \leq (S_1)_i \leq m \leq 200$ , 每个测试点 4 分。

测试点编号	$n \leq$	$m \leq$	$(S_1)_i$
1	$10^9$	200	互不相同
2	200	200	$= 1$
3	$10^9$	200	$= 1$
4 ~ 5	2	200	无特殊限制
6	5	5	无特殊限制
7	5000	7	无特殊限制
8	$10^9$	7	无特殊限制
9 ~ 12	30	30	无特殊限制
13	$10^9$	30	无特殊限制
14 ~ 17	70	70	无特殊限制
18	$10^9$	70	无特殊限制
19 ~ 23	200	200	无特殊限制
24 ~ 25	$10^9$	200	无特殊限制

