

第 1 届论笔蹭唾飢藩洩踞環厝彤誹霰模拟赛

FOC-PINC

第一试

cdqz

时间：10 月 28 日

一、题目概况

题目名称	太空漫步	树的解构	小 T 与灵石	小 S 埋地雷
题目类型	传统型	传统型	传统型	传统型
目录	walking	deconstruct	stone	landmine
源程序文件名	walking.cpp	deconstruct.cpp	stone.cpp	landmine.cpp
可执行文件名	walking	deconstruct	stone	landmine
输入文件名	walking.in	deconstruct.in	stone.in	landmine.in
输出文件名	walking.out	deconstruct.out	stone.out	landmine.out
每个测试点时限	1s	1s	1s	1s
内存限制	256MB	256MB	256MB	256MB
子任务数目	7	4	5	4
编译选项	-lm -O2 -DONLINE_JUDGE			

二、注意事项

1. 选手提交的源文件应直接存放在对应的个人文件夹中，无需建立对应题目的子文件夹。
2. 选手个人文件夹名尽量使用英文字母、数字、下划线或短横，以免被评测机忽略。
3. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
4. 函数 `main()` 的返回值类型必须是 `int`，值必须为 `0`。
5. 若无特殊说明，输入文件中同一行内的多个整数、浮点数、字符串等均使用一个空格进行分隔。
6. 若无特殊说明，结果比较方式为忽略行末空格、文末回车后的全文比较。
7. 程序可使用的栈空间大小与该题内存空间限制一致。
8. 在终端下可使用命令 `ulimit -s unlimited` 将栈空间限制放大，但你使用的栈空间大小不应超过题目限制。

太空漫步 (walking)

【题目描述】

Do not go gentle into that good night.

Old age should burn and rave at close of day.

Rage, Rage against the dying light.

不要温和地走进那个良夜。

白昼将尽，暮年仍应燃烧咆哮。

怒斥吧——怒斥光的消逝。

Ame 是一位生活在 2078 年的女孩子。在她 16 岁时，她参加了 SpaceN 公司举行的知识竞赛并得到了特等奖，由此进入了 Space Explorer Program，Ame 由此获得了太空旅行的机会。我们的故事从这场太空旅行拉开。

这场旅行除了 Ame 外共有 n 位乘客，每位乘客都到飞船外尝试了太空漫步。太空没有重力的限制，因此每位乘客都可以自由地在三维空间中移动（使用 SpaceN 公司研发的 Space Jet 装置）——但在 Ame 眼中他们是在一个（投影的）二维空间移动。Ame 用一个只由 N（北）、S（南）、E（东）和 W（西）构成的非空字符串 s_i 来描述第 i 个乘客的移动轨迹（Ame 称之为轨迹串）。乘客每一单位时间移动一步，也就是说第 i 个乘客一共太空漫步了 $|s_i|$ 单位时间。

Ame 又想出了一个固定的模式串，这个模式串由 NSEW*?[] 构成，意义为：

- *：代表一个任意不定长度的可空串。
- ?：代表一个不定的动作（NSEW）。
- [A]：代表 A 可以出现也可以不出现，其中 A 也是一个模式串。
- NSEW：代表对应动作。

如果这个模式串能够匹配一个轨迹串，那么称这个轨迹是优美的。例如，当模式串是 NS*E，那么 NSWE 和 NSE 都是优美的，NSEW 则不是。

现在 Ame 想要问你，对于每一个乘客，各个时刻为止他走出的轨迹是不是优美的。

【输入格式】

从文件 *walking.in* 中读入数据。

输入共 $(n + 2)$ 行。

第一行一个字符串，表示 Ame 想的模式串。

第二行一个正整数 n ，代表除 Ame 外乘客的数量。

接下来 n 行，第 i 行给出第 i 个乘客的轨迹串 s_i 。

【输出格式】

输出到文件 *walking.out* 中。

输出共 n 行，第 i 行给出一个长度为 $|s_i|$ 的由 **01** 构成的字符串。该字符串的第 j 位 ($1 \leq j \leq |s_i|$) 代表第 i 个乘客在前 j 个单位时间走出的轨迹是不是**优美的** (**0** 代表不是**优美的**，**1** 代表是**优美的**)，即， s_i 的前 j 位前缀是不是**优美的**。

【样例 1 输入】

N[E]W
2
NW
NEW

【样例 1 输出】

01
001

【样例 2 输入】

?E
2
SE
WE

【样例 2 输出】

01
01

【样例 3 输入】

N*E
2
NE
NWEWEWSSSE

【样例 3 输出】

01
0010100001

【样例 4 输入】

?W[W[E]E]ES
3
SWWEEES
SWWEES
SWES

【样例 4 输出】

0000001
000001
0001

【样例 5】

见选手目录下的 *walking/walking5.in* 与 *walking/walking5.ans*。

【测试点约束】

约定 *?[[]] 为特殊字符。

对于所有测试点, 满足 $n \leq 100$, 模式串非空且长度小于等于 100, $|s_i| > 0, \sum_{i=1}^n |s_i| \leq 10^6$ 。
保证模式串是合法的, 没有相邻的特殊字符, 且 [和] 总是两两配对的。

每个子任务的具体限制见下表:

子任务编号	分值	可能含有的特殊字符	特殊限制
1	2	无	无
2	8	<u>?</u>	
3	20	<u>*?</u>	A
4	20	<u>[]</u>	B
5	20	<u>*</u>	C
6	28	<u>*?[[]]</u>	D
7	2		无

特殊限制 A: 模式串中最多只会出现一个 *。

特殊限制 B: 模式串长度小于等于 10, $\sum_{i=1}^n |s_i| \leq 10^4$, [] 中出现的是一个轨迹串, 即 [] 不会嵌套。

特殊限制 C: 模式串中最多会出现 10 个 *。

特殊限制 D: 模式串与所有乘客的轨迹串在所有可能的字符串中随机生成。

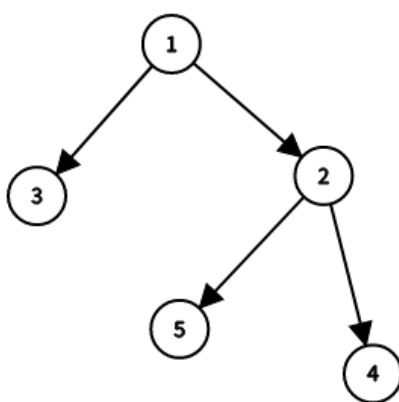
树的解构 (deconstruct)

【题目描述】

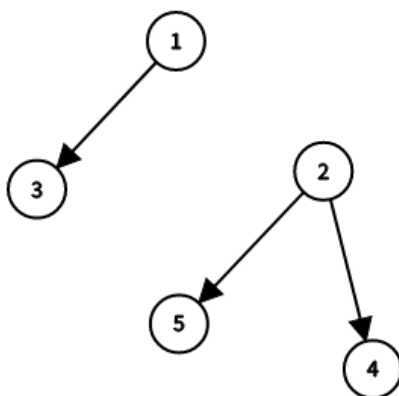
Mivik 喜欢 Eprom 的解构俱乐部，于是他想解构一棵树。

Mivik 找到了一棵以 1 为根的有 n 个结点的有根外向树。Mivik 会进行 $(n - 1)$ 次操作，每次 Mivik 都会从未删掉的边中**等概率**选择一条边将其删去。记这条边为 $a \rightarrow b$ ，则删去这条边的代价是删边时 b 的子树大小（包括 b 自己）；删去这条边后 b 为根的子树会形成一棵新的以 b 为根的有根树。

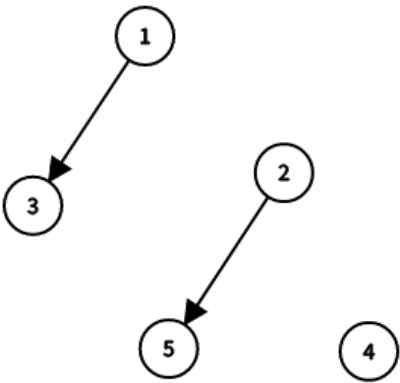
例如，下图是 Mivik 找到的有根树：



如果 Mivik 选了 $1 \rightarrow 2$ 这条边并将其删去，那么代价是 3（2 所在的子树内共有 2、4、5 三个结点），而后情况会变成这样：



如果 Mivik 此时再删去 $2 \rightarrow 4$ 这条边，那么代价是 1（4 所在的子树内只有 4 一个结点），随后情况会变成这样：



Mivik 想知道，他进行这 $(n - 1)$ 次操作后期望的代价总和是多少。由于 Mivik 不喜欢太大的数，你只需要输出期望的值对 $10^9 + 7$ 取模的结果。

【输入格式】

从文件 *deconstruct.in* 中读入数据。
输入共两行。
第一行一个正整数 n ，表示给出的有根树的大小。
第二行给出 $(n - 1)$ 个正整数，第 i 个数 a_i 表示 $(i + 1)$ 号结点在有根树上的父亲是 a_i 。

【输出格式】

输出到文件 *deconstruct.out* 中。
输出一行一个非负整数，表示代价的期望对 $10^9 + 7$ 取模的结果。

【样例 1 输入】

4
1 2 2

【样例 1 输出】

4

【样例 1 解释】

解构的总代价有 $1/3$ 的概率为 3，有 $1/3$ 的概率为 4，有 $1/3$ 的概率为 5，所以代价的期望为 4。

【样例 2 输入】

5
1 1 2 2

【样例 2 输出】

5

【样例 3】

见选手目录下的 *deconstruct/deconstruct3.in* 与 *deconstruct/deconstruct3.ans*。

【测试点约束】

对于所有测试点，满足 $1 \leq n \leq 2 \times 10^6$ 。保证给出的有根树合法。
每个子任务的具体限制见下表：

子任务编号	分值	特殊限制
1	10	$a_i = 1$
2	15	$a_i = i$
3	25	$n \leq 500$
4	50	无

【提示】

本题输入量较大，请使用较快的读入方式。

小 T 与灵石 (stone)

【题目描述】

穿过繁花森林，小 T 来到一个瀑布前。小 T 像往常一样戏水，但今天小 T 在瀑布后发现了一个密室。小 T 小心翼翼地走了进去，看见一块灵石悬浮在空中，小 T 突然明白了为什么这座山上的花开得这么茂盛。小 T 正想上前查看，却被一股神秘的力量挡住，紧接着不知从何处传来了声音：“想要接近我，先回答一个问题……”

给你一棵 n 个结点的有根树，树的根为 1。一共有 q 次操作，第 i 次操作选定 k_i 个点 p_1, p_2, \dots, p_{k_i} ，对每个点 x 定义 $f_{x,i} = \max_{j=1}^{k_i} \{\text{dist}(x, p_j)\}$ ，其中 $\text{dist}(x, y)$ 表示树上 x, y 两点间最短路经过的边数。再对每个点 x 定义 $g_x = \min_{i=1}^q \{f_{x,i}\}$ ，对于每个点你需要求出 g_x 。

听完这个问题后，小 T 顿时就蒙圈了。但小 T 对灵石非常地好奇，你能帮帮小 T 解决这个问题吗？

【输入格式】

从文件 `stone.in` 中读入数据。

输入共 $(q+3)$ 行。

第一行一个正整数 n ，表示树上的结点数。

第二行 $(n-1)$ 个正整数，第 i 个数表示 $i+1$ 号结点的父亲 a_{i+1} 。

第三行一个正整数 q ，表示操作次数。

接下来 q 行，每行最开始有一个正整数 k_i ，接下来输入 k_i 个互不相同的正整数 p_1, p_2, \dots, p_{k_i} ，其意义见【题目描述】。

【输出格式】

输出到文件 `stone.out` 中。

输出 n 行，第 i 行输出一个整数 g_i ，其意义见【题目描述】。

【样例 1 输入】

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

【样例 1 输出】

3
2
4
1
3
5
2
2

【样例 1 解释】

第一次操作的 f 值为 3,3,4,4,4,5,5,5；第二次操作的 f 值为 3,2,4,1,3,5,2,2。

【样例 2】

见选手目录下的 `stone/stone2.in` 与 `stone/stone2.ans`。

【测试点约束】

对于所有测试点，满足 $1 \leq n, q \leq 3 \times 10^5$ ， $\sum_{i=1}^q k_i \leq 10^6$ ， $1 \leq a_i, p_i, k_i \leq n$ 。
每个子任务的具体限制见下表：

子任务编号	分值	$n \leq$	$\sum_{i=1}^q k_i \leq$	特殊限制
1	10	400	400	无
2	20	3000	3000	
3	20	3×10^5	10^6	树是一条链
4	20	10^5	10^5	无
5	30	3×10^5	10^6	

小 S 埋地雷 (landmine)

【题目描述】

L 国马上就要发动战争了，战士小 S 被提前派往敌对城市埋下了一排地雷。

具体来说，这排地雷由 n 个小地雷组成，编号到从 1 到 n 。敌方的工兵可以任意选择拆弹顺序，每次只能选择一个地雷排除。由于地雷的特殊性，每当排除一个地雷后会这个地雷左边和右边的地雷会连接到一起。每个地雷拥有四个属性 p_i, q_i, r_i, s_i ，排除当前第 i 个地雷的代价为 $(p_{i-1} - q_i)^2 + (p_i - r_{i+1})^2 + (p_{i+1} - s_{i+2})^2$ ，其中第 $(i-1)$ 个地雷是当前时刻位于 i 前的地雷，第 $(i+1)$ ， $(i+2)$ 个地雷分别为当前时刻位于 i 后第一个，第二个的地雷，若不存在这样的地雷则属性 p, q, r, s 均为 0。

现在小 S 已经埋好了一排地雷了，小 S 想知道敌方工兵在运气最坏的情况下拆掉这些地雷的代价，你能帮帮小 S 吗？

【输入格式】

从文件 *landmine.in* 中读入数据。

输入共 5 行。

第一行一个整数 n ，表示地雷的数量。

第二行 n 个整数，第 i 个整数表示 p_i 。

第三行 n 个整数，第 i 个整数表示 q_i 。

第四行 n 个整数，第 i 个整数表示 r_i 。

第五行 n 个整数，第 i 个整数表示 s_i 。

所有输入的意义见【题目描述】。

【输出格式】

输出到文件 *landmine.out* 中。

输出一行一个整数，表示最坏情况下拆除所有地雷所花的代价。

【样例 1 输入】

```
4
1 0 0 1
0 1 0 0
1 0 0 1
0 0 0 1
```

【样例 1 输出】

8

【样例 1 解释】

当按顺序拆除第 3, 1, 2, 4 个地雷时，代价最大为 8。

【样例 2】

见选手目录下的 *landmine/landmine2.in* 与 *landmine/landmine2.ans*。

【测试点约束】

对于所有测试点，满足 $1 \leq n \leq 70$ ， $-1000 \leq p_i, q_i, r_i, s_i \leq 1000$ 。
每个子任务的具体限制见下表：

子任务编号	分值	特殊限制
1	20	$n \leq 20$
2	20	$s_i = 0$
3	30	$n \leq 40$
4	30	无