

软件包管理器

【问题描述】

Linux 用户和 OS X 用户一定对软件包管理器不会陌生。通过软件包管理器，你可以通过一行命令安装某一个软件包，然后软件包管理器会帮助你从软件源下载软件包，同时自动解决所有的依赖（即下载安装这个软件包的安装所依赖的其它软件包），完成所有的配置。Debian/Ubuntu 使用的 `apt-get`，Fedora/CentOS 使用的 `yum`，以及 OS X 下可用的 `homebrew` 都是优秀的软件包管理器。

你决定设计你自己的软件包管理器。不可避免地，你要解决软件包之间的依赖问题。如果软件包 A 依赖软件包 B ，那么安装软件包 A 以前，必须先安装软件包 B 。同时，如果想要卸载软件包 B ，则必须卸载软件包 A 。现在你已经获得了所有的软件包之间的依赖关系。而且，由于你之前的工作，除 0 号软件包以外，在你的管理器当中的软件包都会依赖一个且仅一个软件包，而 0 号软件包不依赖任何一个软件包。依赖关系不存在环（若有 m ($m \geq 2$) 个软件包 $A_1, A_2, A_3, \dots, A_m$ ，其中 A_1 依赖 A_2 ， A_2 依赖 A_3 ， A_3 依赖 A_4 ， $\dots$ ， A_{m-1} 依赖 A_m ，而 A_m 依赖 A_1 ，则称这 m 个软件包的依赖关系构成环），当然也不会有一个软件包依赖自己。

现在你要为你的软件包管理器写一个依赖解决程序。根据反馈，用户希望在安装和卸载某个软件包时，快速地知道这个操作实际上会改变多少个软件包的安装状态（即安装操作会安装多少个未安装的软件包，或卸载操作会卸载多少个已安装的软件包），你的任务就是实现这个部分。注意，安装一个已安装的软件包，或卸载一个未安装的软件包，都不会改变任何软件包的安装状态，即在此情况下，改变安装状态的软件包数为 0。

【输入格式】

从文件 `manager.in` 中读入数据。

输入文件的第 1 行包含 1 个整数 n ，表示软件包的总数。软件包从 0 开始编号。

随后一行包含 $n - 1$ 个整数，相邻整数之间用单个空格隔开，分别表示 $1, 2, 3, \dots, n - 2, n - 1$ 号软件包依赖的软件包的编号。

接下来一行包含 1 个整数 q ，表示询问的总数。

之后 q 行，每行 1 个询问。询问分为两种：

- `install x`: 表示安装软件包 x
- `uninstall x`: 表示卸载软件包 x

你需要维护每个软件包的安装状态，一开始所有的软件包都处于未安装状态。对于每个操作，你需要输出这步操作会改变多少个软件包的安装状态，随后应用这个操作（即改变你维护的安装状态）。

【输出格式】

输出到文件 *manager.out* 中。

输出文件包括 q 行。

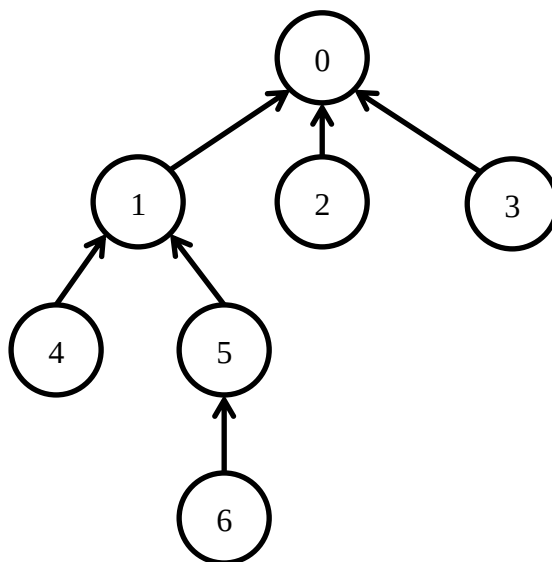
输出文件的第 i 行输出 1 个整数, 为第 i 步操作中改变安装状态的软件包数。

【样例输入 1】

```
7
0 0 0 1 1 5
5
install 5
install 6
uninstall 1
install 4
uninstall 0
```

【样例输出 1】

```
3
1
3
2
3
```

【样例说明 1】

一开始所有的软件包都处于未安装状态。

安装 5 号软件包, 需要安装 0,1,5 三个软件包。

之后安装 6 号软件包, 只需要安装 6 号软件包。此时安装了 0,1,5,6 四个软件包。

卸载 1 号软件包需要卸载 1,5,6 三个软件包。此时只有 0 号软件包还处于安装状态。

之后安装 4 号软件包，需要安装 1,4 两个软件包。此时 0,1,4 处在安装状态。
最后，卸载 0 号软件包会卸载所有的软件包。

【样例输入 2】

```
10
0 1 2 1 3 0 0 3 2
10
install 0
install 3
uninstall 2
install 7
install 5
install 9
uninstall 9
install 4
install 1
install 9
```

【样例输出 2】

```
1
3
2
1
3
1
1
1
1
0
1
```

【样例输入输出 3】

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

【数据规模与约定】

测试点编号	n 的规模	q 的规模	备注
1	$n = 5,000$	$q = 5,000$	
2			
3	$n = 100,000$	$q = 100,000$	数据不包含卸载操作
4			
5	$n = 100,000$	$q = 100,000$	编号为 i 的软件包所依赖的软件包编号在 $[0, i - 1]$ 内均匀随机 每次执行操作的软件包编号在 $[0, n - 1]$ 内均匀随机
6			
7			
8			
9	$n = 100,000$	$q = 100,000$	
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			