

# 2023 CSP七连测-day03

题目名称	杠上开花	公交星槎	记忆碎片	金人旧巷
题目类型	传统型	传统型	传统型	传统型
英文题目名称	bloom	bus	memory	street
输入文件名	<a href="#">bloom.in</a>	<a href="#">bus.in</a>	<a href="#">memory.in</a>	<a href="#">street.in</a>
输出文件名	bloom.out	bus.out	memory.out	street.out
每个测试点时限	1s	1s	1s	1s
内存限制	512MB	512MB	512MB	512MB
提交源文件名	bloom.cpp	bus.cpp	memory.cpp	street.cpp

【C++编译选项】 `-lm -std=c++14 -Wl,--stack=1000000000 -O2`

【试题下载地址】 <ftp://172.16.2.202/竞赛资料/20231007.zip> 匿名访问即可

【结果上传地址】 <ftp://172.16.2.202/20231007文件回收> 用户名密码均为test

【赛后补题地址】 <http://zhb.wms.edu/d/JH2023/> 训练

【提交文件夹格式】

--准考证号\ (平时训练用中文姓名)

```
--task1\  
|      |--task1.cpp  
--task2\  
|      |--task2.cpp  
--task3\  
|      |--task3.cpp  
--task4\  
|      |--task4.cpp
```

# T1 [2023CSP七连 Day3-01] 杠上开花

## 题目描述

帝垣琼玉是风靡仙舟的棋牌游戏。

青雀是帝垣琼玉的粉丝，甚至使用帝垣琼玉牌战斗。

现在青雀面对  $n$  个排成一排的敌人，从左到右第  $i$  个敌人有整数血量  $a_i$  ( $1 \leq a_i \leq 2$ )。每个回合，她可以选择一个敌人，对它造成 2 点伤害，并对与它相邻的敌人造成 1 点伤害。

当一个敌人的血量小于等于 0，它就会死亡。同时，若第  $i$  个敌人死亡，那么原来的第  $i + 1$  个敌人会变成第  $i$  个，第  $i + 2$  个会变成第  $i + 1$  个，以此类推。

此外，青雀攻击时一定会确保最左边的第一个敌人的血量减少，即攻击第一个敌人或第二个敌人。

小 X 想知道，青雀最少需要多少回合才能解决所有敌人。

## 输入格式

一行四个整数，表示  $n, s1, s2, s3$ 。

然后你可以用以下程序生成序列  $a_i$ ：

```
void Gen(int n, unsigned s1, unsigned s2, unsigned s3) {
    for(int i = 1; i <= n; ++i) {
        s1 ^= s3;
        s3 += 3055373123u;
        a[i] = (1 << ((s1 >> s2) & 1));
        s2 = (s2 ^ s3) & 31;
        s3 = (s3 >> s2) | ((s3 << (31 ^ s2)) << 1);
    }
}
```

## 输出格式

输出一个整数，表示最小回合数。

# 样例数据

## 样例输入 1

5 114514 3 1919810

## 样例输出 1

2

## 样例输入 2

114514 166032413 6 102134106

## 样例输出 2

49469

## 样例解释

对于第一组样例，敌人的血量依次是： 1 2 2 1 1 。

第一回合打一个血量为 2 的敌人，之后会剩下 1 1 1 ，可以一回合结束战斗。

# 数据范围

- 对于 5% 的数据， $n = 1$ 。
- 对于 20% 的数据， $n \leq 3$ 。
- 对于 30% 的数据， $n \leq 10$ 。
- 对于 50% 的数据， $n \leq 20$ 。
- 对于 70% 的数据， $n \leq 10^6$ 。
- 对于 100% 的数据， $n \leq 3 \times 10^7, 0 \leq s1, s3 < 2^{32}, 0 \leq s2 < 32$ 。

# T2 [2023CSP七连 Day3-02] 公交星槎

## 题目描述

为了方便居民的出行和提倡绿色出行，仙舟罗浮有一套公交星槎系统。

罗浮有  $n$  个公交站，一个公交站可以看作一个节点，编号为 1 到  $n$ 。它们由  $p$  条公交线路连通，每条线路途经若干个节点，第  $i$  条线路用它依次经过的节点  $s_{i,0}, s_{i,1}, \dots, s_{i,m_i}$  来表示。

一开始，这些线路上没有星槎在运行。

对于每条线路  $i$ ，每天的第 0 个时刻，会有一班公交星槎从  $s_{i,0}$  驶出，然后按照

$s_{i,0}, s_{i,1}, s_{i,2}, \dots, s_{i,m_i}, s_{i,m_i-1}, s_{i,m_i-2}, \dots, s_{i,0}$  的路径一直循环运行，从一个节点到下一个节点需要花费 1 个单位的时间。之后在  $g_i, 2g_i, 3g_i, \dots$  等  $g_i$  的倍数时刻，如果此时  $s_{i,0}$  的位置没有按照线路运行的星槎，会有一艘星槎从  $s_{i,0}$  驶出，按照上述路径运行（保证  $g_i$  是  $2m_i$  的约数）。

小 X 在这天的第 0 时刻来到 1 号节点，然后搭乘一条线路，之后如果有该线路上的公交星槎到达小 X 所在的节点，他可以选择乘坐该星槎（也可以不乘坐）并随该星槎移动到线路上的其他节点。小 X 可以在节点处更换搭乘的线路，但这会花费它  $t$  个单位的时间。即，如果小 X 由线路  $i$  在  $a$  时刻到达节点  $u$ ，之后想要搭乘线路  $j$ ，那么他只能搭乘  $a + t$  时刻以后的  $j$  线路上的星槎。

保证任意两个节点都是可以相互到达的。

小 X 想知道，他到达  $n$  号节点需要的最少时间是多少。

## 输入格式

第一行三个整数，表示  $n, p, t$ 。

接下来  $2p$  行，每两行表示一条线路：第一行两个整数  $m_i, g_i$ ，第二行  $m_i + 1$  个整数，表示  $s_{i,0}, s_{i,1}, \dots, s_{i,m_i}$ 。

## 输出格式

一行一个整数，表示从 1 号节点到达  $n$  号节点的最少时间。

## 样例数据

### 样例输入 1

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

### 样例输出 1

```
6
```

## 数据范围

对于 20% 的数据,  $n, p \leq 100, \sum m_i \leq 200$ 。

对于 40% 的数据,  $n, p \leq 1000, \sum m_i \leq 2000$ 。

对于另外 20% 的数据,  $\sum m_i = n - 1$ 。

对于 100% 的数据, 保证  $2 \leq n, p \leq 10^5, \sum m_i \leq 2 \times 10^5, 1 \leq t \leq n$ 。

# T3 [2023CSP七连 Day3] 记忆碎片

## 题目描述

小 X 有一个记忆仓库。它可以存储两类记忆碎片：事件与回忆。每一个记忆碎片是一段字符串，每个字符是  $a$  到  $z$  的小写字母。其中，**没有某个事件碎片的字符串为另一个事件碎片的字符串的前缀**。

一开始，仓库是空的。然后，小 X 每次会向仓库中添加一个事件碎片或者一个回忆碎片。

回忆是事件组成的。小 X 想知道，仓库中有哪些回忆碎片可以通过将若干个（可以是一个）仓库中的事件碎片（同一个可以使用多次）首尾连接得到。

你需要在每次加入操作后立刻回答小 X 的问题。**(强制在线)**

由于这样输出就太多了，所以小 X 只需要知道每次加入操作后，之前不能组成、现在可以组成的回忆碎片有哪些。

## 输入格式

第一行一个整数  $n$ ，表示要加入的碎片个数。

接下来  $n$  行，第  $i$  行由一个字符和一个字符串  $s'_i$  组成，字符是 + 或 ?，分别表示加入的是事件碎片和回忆碎片。该碎片的真实字符串由  $s'_i$  解密得到，实际的字符串  $s_i$  是  $s'_i$  做  $m_i - 1$  次向左循环移位的结果，其中  $m_i - 1$  的定义见输出格式， $m_0 = 0$ 。

向左循环移位：即将字符串开头的字符放到字符串末尾。

## 输出格式

输出  $n$  行，第  $i$  行开头一个整数  $m_i$  表示第  $i$  次操作后新的能够被组成的回忆碎片个数。之后  $m_i$  个空格隔开的递增整数，表示这些回忆碎片的编号。

一个回忆碎片的编号为先于它加入的回忆碎片个数 +1。

## 样例数据

### 样例输入 1

```
5
? abcabd
+ abc
? abcabc
? dabdab
+ abd
```

### 样例输出 1

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

## 样例解释

实际加入的字符串：

记忆碎片 `abcabd`。

事件碎片 `abc`。

记忆碎片 `abcabc`。

记忆碎片 `abdabd`。

事件碎片 `abd`。

## 数据范围

令  $m$  为字符串总长度。

对于 20% 的数据，满足  $m \leq 500$ 。

对于 40% 的数据，满足  $n \leq 500, m \leq 10^6$ 。

对于另外 20% 的数据，满足所有 `+` 操作均在所有 `?` 前出现。

对于另外 20% 的数据，满足所有 `?` 操作均在所有 `+` 前出现。

对于 100% 的数据，满足  $n, m \leq 10^6$ 。

# T4 [2023CSP七连 Day3] 金人旧巷

## 题目描述

金人巷是仙舟罗浮上著名的夜市。

小 X 参与金人巷的复兴计划，增加它的热闹程度。其地图可以抽象为一棵  $n$  个点的无根树，每一个节点代表一个区域，这  $n$  个区域由  $n - 1$  条无向道路连接在一起，每个节点有一个热闹程度。在复兴计划中，记第  $u$  个节点增加的热闹程度为  $a_u$ ，一开始都为 0。

然后小 X 开始依次进行  $Q$  次操作。每一个操作可能是：

- `1 u w p` 小 X 在节点  $u$  上新建一个商铺。然后，对于所有节点  $v$ （包括  $u$  本身），记它与  $u$  之间的简单路径的边数为  $d$ ，那么  $a_v$  会增加  $\lfloor \frac{w}{p^d} \rfloor$ 。

- 2  $u\ v$  由于人流量限制，小 X 关闭了节点  $u$  和  $v$  之间的道路（保证存在）。他想知道，这条道路切断后，能到达  $u$  的所有节点  $p$  的  $a_p$  之和以及能到达  $v$  的所有节点  $q$  的  $a_q$  之和。注意：询问之间相互独立，即本次询问关闭的道路在下一次询问时不会关闭。

小 X 需要你帮忙回答所有的 2 操作。保证答案在 `long long` 范围内。

## 输入格式

第一行两个整数，表示  $n, Q$ 。

接下来  $n - 1$  行，每行两个整数，表示树上的一条边。

接下来  $Q$  行，每行一个询问。

## 输出格式

对每个操作 2 输出一行，每行两个整数，分别表示能到达  $u$  的所有节点  $p$  的  $a_p$  之和、能到达  $v$  的所有节点  $q$  的  $a_q$  之和。

## 样例数据

### 样例输入 1

```
6 10
5 2
6 1
3 2
4 2
6 2
1 4 5 2
2 6 2
1 2 13 3
1 6 9 2
2 4 2
2 6 1
1 1 14 2
1 3 3 1
1 4 4 1
2 6 1
```



## 样例输出 1

```
1 9
11 52
58 5
106 26
```

## 数据范围

对于 30% 的数据,  $n, q \leq 3000$ 。

对于另外 20% 的数据, 保证每个节点的度数不超过 2, 即树形成一条链。

对于另外 20% 的数据, 保证树的直径 (即树上最长简单路径) 长度 (边的条数) 不超过 20。

对于 100% 的数据, 保证  $1 \leq n, q \leq 10^5, 1 \leq w, p \leq 10^9$ 。