# 清华大学计算机系

# 2022 年“大中衔接”研讨与教学活动

## 第一试

时间：2022 年 5 月 21 日 14:00 ~ 19:00

| 题目名称 | 归程 | 最大连续和 | 寻宝 | 字符树 |
|---|---|---|---|---|
| 题目类型 | 传统型 | 传统型 | 传统型 | 传统型 |
| 输入 | 标准输入 | 标准输入 | 标准输入 | 标准输入 |
| 输出 | 标准输出 | 标准输出 | 标准输出 | 标准输出 |
| 每个测试点时限 | 2.0 秒 | 4.0 秒 | 5.0 秒 | 3.5 秒 |
| 内存限制 | 512 MiB | 512 MiB | 1024 MiB | 512 MiB |
| 子任务数目 | 5 | 3 | 25 | 10 |
| 测试点是否等分 | 否 | 否 | 否 | 否 |
| 预测试点数目 | 5 | 3 | 25 | 10 |

# 归程（return）

## 【题目背景】

小 S 是在幽燕之都的某著名大学上学的一名学生。考完期末考的当晚，小 S 独自一人出校犒劳一下多日来疲于复习的自己。没成想，在外面享用晚餐时，突然下起了雨。尽管幽燕之都是暖温带半湿润半干旱季风气候，但在一月突然下起有一定规模的雨（并且不是雪）实属罕见。

小 S 看了一下天气预报，发现雨可能会越下越大，便打算打车返回宿舍。屋漏偏逢连夜雨，当小 S 点完打车后，他的手机正好耗尽了所有电量，自动关机了。翻遍了书包口袋，小 S 也没有找到可以用来坐车的零钱，或是用来为手机续命的充电线。唯一能给小 S 一点安慰的，是他未雨绸缪在包里放的兔兔奶糖——源自神秘的魔力之都的这款零食是他的最爱。无奈之下，小 S 打算凭借他的两条双腿冒雨从餐馆赶回宿舍。

## 【题目描述】

从餐馆到宿舍的交通网络可以被抽象成一张有 $N$ 个点 $M$ 条边的**无向连通图**，图中没有重边也没有自环，点和边分别用从 1 开始的自然数编号。餐馆在第 $x$ 号点，而宿舍在 $y$ 号点。小 S 赶路时，通过第 $i$ 条边恰好需要 $l_i$ 分钟。尽管小 S 可以直接沿着最短路赶回宿舍，但他更希望自己被淋到的雨越少越好。受到路两旁建筑物等多方面因素的影响，在不同路上淋到的雨量可能会有所不同；而在不同天气状况下，在同一条路上淋到的雨量也可能会有所不同。具体来说，在当前的天气状况下，小 S 通过第 $i$ 条边时**每分钟**会被淋到 $a_i$ 单位的雨；但是如果雨下大了，那么通过第 $i$ 条边时小 S **每分钟**会被淋到 $b_i$ 单位的雨。

由于小 S 的手机没电关机了，他无法准确知道什么时候雨会开始下大，但是他瞅了一眼天空，掐指一算，给出了 $K$ 个雨可能会变大的时间点 $T_1, T_2, \cdots, T_K$，其中在他从餐馆出发后经过恰好 $T_i$ 分钟雨变大的概率是 $p_i$。小 S 希望，在这个给定的分布下，求出一个赶回宿舍的方案，使得他从出发到赶回宿舍为止被淋到的总雨量的期望最小。具体而言，这个方案由一组决策组成，每个决策根据当前时间、当前雨是否已经变大和小 S 所处的位置决定小 S 接下来应该沿着哪条边前进。由于小 S 急着赶回宿舍，我们规定，他不会在除了终点（即宿舍 $y$）以外的点或任意一条边上停留，并且也不会在沿着一条边走到中途时返回。

## 【输入格式】

从标准输入读入数据。

输入的第一行包含五个正整数 $N, M, K, x, y$，分别表示交通网络的点数、交通网络的边数、雨可能会变大的时间点数量、餐馆的位置及宿舍的位置。保证 $2 \le N \le 1000, 1 \le M \le 4000, 1 \le K \le 1000, 1 \le x, y \le N$ 且 $x \ne y$。

接下来输入 $M$ 行，其中第 $i$ 行包含五个正整数 $u_i, v_i, l_i, a_i, b_i$，分别表示第 $i$ 条边的两个端点、小 S 通过第 $i$ 条边所需时间、雨小时在第 $i$ 条路上每分钟会被淋到的雨量及雨大时在第 $i$ 条路上每分钟会被淋到的雨量。保证 $1 \le u_i, v_i \le N, 1 \le l_i \le 20, 1 \le a_i \le b_i \le 100,000$，且输入的无向图不包含重边或自环。

最后输入 $K$ 行，其中第 $i$ 行包含两个正整数 $T_i, w_i$。其中，$T_i$ 表示第 $i$ 个雨可能变大的时刻，对应的发生概率 $p_i = w_i / \sum_{j=1}^{K} w_j$。保证 $1 \le T_1 < T_2 < \cdots < T_K \le 10000$ 且 $1 \le w_i \le 1000$。

## 【输出格式】

输出到标准输出。

输出一个正实数，表示在最优方案下，小 S 赶回宿舍过程中被淋到的总雨量。当你的输出和标准答案的相对误差或绝对误差在 $10^{-6}$ 以内时，我们认为你的输出是正确的。

## 【样例 1 输入】

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

## 【样例 1 输出】

```
13.00000000000000000
```

## 【样例 1 解释】

小 S 认为，有 1/2 的概率雨在他出发后 3 分钟时变大，另外有 1/2 的概率是在出发后 6 分钟时变大。

如果已知雨一定在出发后 3 分钟时变大，那么赶回宿舍的最优方案为：先沿着边 $(1,3)$ 到达点 3，由于途中雨变大，小 S 会被淋到 7 单位的雨；接着沿着边 $(3,4)$ 回到宿舍，被淋到 9 单位的雨。路上小 S 总共被淋到 16 单位的雨。

如果已知雨一定在出发后 6 分钟时变大，那么赶回宿舍的最优方案为：先沿着边 $(1,2)$ 到达点 2，被淋到 3 单位的雨；接着沿着边 $(2,4)$ 回到宿舍，被淋到 6 单位的雨。路上小 S 总共被淋到 9 单位的雨。这同时也是花费时间最少的方案。

根据小 S 推测的分布，最优的方案应该为：先沿着边 $(1,2)$ 到达点 2；此时恰为从餐馆出发后 3 分钟，如果雨此时变大，则沿着边 $(2,3)$ 和 $(3,4)$ 回到宿舍，这种情况下淋到的雨量为 $3\times1+1\times5+3\times3=17$；否则，（根据所给分布）可推测雨一定是从餐馆出发后 6 分钟时变大，则可以直接沿着 $(2,4)$ 回到宿舍，这种情况下淋到的雨量为 $3\times1+2\times3=9$。期望被淋到雨量为 $17\times\frac{1}{2}+9\times\frac{1}{2}=13$ 。

## 【样例 2】

见题目目录下的 ***2.in*** 与 ***2.ans***。

这个样例和第 1 个子任务的性质相同。

## 【样例 3】

见题目目录下的 ***3.in*** 与 ***3.ans***。

这个样例和第 3 个子任务的性质相同。

## 【样例 4】

见题目目录下的 ***4.in*** 与 ***4.ans***。

这个样例和第 4 个子任务的性质相同。

## 【子任务】

对于 100% 的数据，保证：

- $2\le N\le1000, 1\le M\le4000, 1\le K\le1000$；
- $1\le x,y\le N$，且 $x\ne y$；
- 输入的无向图无重边无自环，图是连通的；
- 对于第 $i$ 条边，$1\le u_i,v_i\le N, 1\le l_i\le20, 1\le a_i\le b_i\le100{,}000$；
- 对于给出的时间点，$1\le T_1<T_2<\cdots<T_K\le10000$ 且 $\forall 1\le i\le K, 1\le w_i\le 1000$。

本题共有 5 个子任务，你需要通过一个子任务中的所有测试点才能获得该子任务的相应分数。下表为各子任务的数据规模及性质。

| 子任务编号 | 对应分数 | $N$ | $M$ | $K$ | 特殊性质 |
|---|---|---|---|---|---|
| 1 | 15 | $\leq 100$ | $= N-1$ | $\leq 50$ | 无 |
| 2 | 10 | $\leq 100$ | $\leq 400$ | $=1$ | $T_1 = 10000$ |
| 3 | 15 | $\leq 1000$ | $\leq 4000$ | $=1$ | 无 |
| 4 | 25 | $\leq 100$ | $\leq 400$ | $\leq 50$ | 无 |
| 5 | 35 | $\leq 1000$ | $\leq 4000$ | $\leq 1000$ | 无 |

## 【提示】

北京市紫荆区酒井蒜协提醒您：道路千万条，电量第一条；外出不充电，关机两行泪。

如果你不熟悉条件概率相关的内容，以下是一些对你做题可能有所帮助的提示。

我们记 $P(A|B)$ 表示在已知事件 $B$ 发生时事件 $A$ 发生的**条件概率**。如果事件 $B$ 发生的概率不为 0，这一条件概率可以使用以下公式进行计算：

$$P(A|B) = \frac{P(AB)}{P(B)}$$

其中，$P(AB)$ 表示时间 $A$ 和 $B$ 同时发生的概率，而 $P(B)$ 表示事件 $B$ 发生的概率。在本题中，可以认为在每个时间点下雨是互斥事件，且它们组成了样本空间 $\Omega$。我们称这样的一组事件为样本空间的一个分割（或者划分）。记事件 $A_i$ 表示 $T = T_i$，即“从餐馆出发后经过恰好 $T_i$ 分钟雨变大”，则有：

- 所有的 $A_i$ 组成了样本空间 $\Omega$，即 $\cup_{i=1}^{k} A_i = \Omega$，对应概率 $P\left(\cup_{i=1}^{k} A_i\right) = P(\Omega) = 1$；
- $A_i$ 之间两两互斥，即 $\forall 1 \leq i, j \leq K, i \neq j, A_i \cap A_j = \varnothing$，对应概率 $P\left(A_i A_j\right) = P\left(\varnothing\right) = 0$。

由以上性质可以推出，在已知不是从餐馆出发后经过恰好 $T_i$ 分钟雨变大（对应事件 $\bar{A}_i$，即 $A_i$ 不发生）的条件下，出发后 $T_j (j \neq i)$ 分钟时下雨，也即事件 $A_j$ 发生的**条件概率**为：

$$P\left(A_j|\bar{A}_i\right) = \frac{P\left(A_j \bar{A}_i\right)}{P\left(\bar{A}_i\right)} = \frac{P\left(A_j\right)}{P\left(\bar{A}_i\right)} = \frac{p_j}{\sum\limits_{k \neq i} p_k} = \frac{w_j}{\sum\limits_{k \neq i} w_k}$$

更进一步，对于本题中的三个事件 $A_i, A_{j_1}, A_{j_2}$（其中 $i, j_1, j_2$ 互不相同），可以证明：

$$\frac{P\left(A_{j_1}|\bar{A}_i\right)}{P\left(A_{j_2}|\bar{A}_i\right)} = \frac{\frac{w_{j_1}}{\sum_{k \neq i} w_k}}{\frac{w_{j_2}}{\sum_{k \neq i} w_k}} = \frac{w_{j_1}}{w_{j_2}} = \frac{P\left(A_{j_1}\right)}{P\left(A_{j_2}\right)}$$

即在一组分割中，两个事件发生概率的比值与是否已知分割中**其它**事件发生与否无关。尽管在本题中，已知的信息是前缀的一系列事件（即 $A_1 \cup A_2 \cup \cdots \cup A_i$）是否发生，但这一性质仍然成立。

# 最大连续和（sequence）

## 【题目描述】

给定长度为 $N$ 的序列 $a_1, a_2, ..., a_N$，和长度为 $M$ 的序列 $b_1, b_2, ..., b_M$。接着你需要构造出一个长度为 $M$ 的整数序列 $c_1, c_2, ..., c_M$，满足:

- $0 \le c_n \le N$。
- 每个非 0 元素在 $c$ 中出现次数不能超过 1 次。

构造完序列 $c$ 后，你将依序做以下的操作：

- 若 $c_i \neq 0$ ，将 $a_{c_i}$ 替换成 $b_i$。

试求在经由你构造的序列 c 替换完以后，序列 $a$ 的最大连续和（$\max\limits_{1 \le l \le r \le N}(\sum_{i=l}^{r} a_i)$）能多大。

## 【输入格式】

输入的第一行包含两个整数 $N$ 和 $M$，代表序列 $a$ 和序列 $b$ 的长度。

接下来的一行包含 $N$ 整数 $a_1, a_2, ..., a_N$。

接下来的一行包含 $M$ 整数 $b_1, b_2, ..., b_M$。

## 【输出格式】

输出一行一个整数表示经由你构造的序列 c 替换完以后，序列 $a$ 的最大连续和能多大。

## 【样例 1 输入】

```
3 1
3 -6 3
-2
```

## 【样例 1 输出】

```
4
```

## 【样例 1 解释】

令 $c_1 = 2$ ，操作时 $a_{c_1}(a_2)$ 被替换为 $-2$ 。

$\max\limits_{1 \le l \le r \le N}(\sum_{i=l}^{r} a_i) = \sum_{i=1}^{3} a_i = 3 + (-2) + 3 = 4$

## 【子任务】

- $1 \le N \le 10^5$
- $0 \le M \le 10^5$
- $-10^9 \le a_i, b_i \le 10^9$

前 20% 的分数满足 $N, M \le 500$。

其余有 30% 的分数满足 $N, M \le 2000$。

# 寻宝（treasure）

## 【题目描述】

你是一名寻宝爱好者，这天，你听说在神秘的藏宝大学（**T**reasure **H**olding **U**niversity）里藏有巨量的神秘宝藏，于是你打算前去一探究竟。

你在网上查到了一些关于藏宝大学的信息：学校里共有 $n$ 个标识点，$m$ 条双向道路连接这些标识点，使得任意两个标识点之间都可以通过道路互相到达。而所谓的神秘宝藏，就埋藏在某一个标识点下方。

你又联系到了一位藏宝大学的学长，据说他知道神秘宝藏的具体位置，但是他却不能直接告诉你——这算是泄露学校机密了。不过，你还是从他嘴里旁敲侧击到了一些有用的信息：

1、藏宝大学的道路是经过精心设计的，使得你从任意一个标识点出发，沿着任意的路径在学校里行走，最后回到原来的标识点，所经过的道路条数**一定都是偶数**。据说，这样可以有效缓解校园内的自行车拥堵问题。

2、学校所在的行政区域刚刚因为确诊病例升级为中风险，而学校规定 14 天内途径过中高风险地区的人员一律不得入校。因此，你只能乘坐直升机空降入校了。不过既然你都有直升机了，你也不打算沿着道路探索学校的每一个标识点，而是可以任选标识点降落，并在标识点之间任意跳跃。

3、在你进入校园之前，学长可以帮你在校园里留下一些提示信息——他会在学校里的每条道路上放置一个箭头，指向这条边连接的两个标识点中之一。但在你到达校园之后，就无法再联系上学长，只能靠你自己了。

4、学校的所有标识点是**无标号**的，这可能会增大你寻宝的难度。

不幸的是，你并没有拿到藏宝大学的地图，你甚至不知道道路条数 $m$ 的值，只知道标识点数 $n$ 的值。当然，学长肯定是知道全部信息的。

在引起校领导的注意之前，你只有有限的时间访问学校的若干个**不同**的标识点，并只能选择**一个**标识点进行宝藏挖掘。能否寻宝成功，就要看你和学长的配合了。

### 实现细节

你不需要，也不应该实现主函数，你只需要实现如下两个函数（这也是你需要提交到 OJ 的内容）：

```
void Alice(const int testid, const int n, const int m, const int
    x, const int u[], const int v[], bool dir[]);

int Bob(const int testid, const int n);
```

`Alice`函数表示学长的提示。其中 $testid$ 为子任务编号，$n$ 为标识点数，$m$ 为道路数，$x$ 为宝藏所在的标识点的编号。数组 $u$ 和 $v$ 的大小均为 $m$ ，描述校园中的道路，第 $i$ 条道路连接标识点 $u[i]$ 和 $v[i]$ 。（所有编号和下标均从 0 开始，下同。）

`Alice`函数需要将输出写入至 $dir$ 数组中，数组大小为 $m$ ，表示学长提示的每一条边的箭头指向。$dir[i]=0$ 表示第 $i$ 条边的箭头指向 $v[i]$ ，反之则表示指向 $u[i]$ 。

`Bob`函数表示你的寻宝过程。其中 $testid$ 和 $n$ 的含义同上。该函数需要返回一个标识点编号，表示对该标识点进行宝藏挖掘。

`Bob`函数执行的过程中，可以调用如下函数：

```cpp
vector<pair<int,bool>> discover(int pos);
```

该函数表示你访问编号为 $pos$ 的标识点，需要满足 $0 \leq posn$ 。通过访问你可以获取到与该点相连的所有边的信息，用一个 `vector`来存储。每条边的信息用一个`pair<int,bool>`来表示，分别表示这条边指向的另一个点的编号以及这条边上的箭头方向（为 0 表示指向另一个点，为 1 表示指向自己）。另外，在同一个测试点中，你不能访问同一个标识点多于一次。

在一个测试点中，将会先执行一次 `Alice`函数，再执行一次 `Bob`函数。为了体现标识点的无标号性，执行完 `Alice`函数之后，所有标识点的编号将被**随机打乱**。`Bob`函数中涉及到的一切点的编号，包括调用 `discover`函数传入的 $pos$ 、调用 `discover`函数得到的出边指向的标识点编号、以及 `Bob`函数的返回值等，均指打乱后的编号。另外，`discover`函数返回的边的顺序也并非按照 `Alice`函数中传入的顺序，而是乱序给出的。（在部分测试点中，标识点的编号不会随机打乱，详见“数据范围”。）

另外，你也可以根据自己的需要，定义其他变量或者函数。**但你定义的全局变量在`Alice`函数中修改的值将不能应用于 `Bob`函数**。

你的 `Alice`函数只能访问数组 $u$ $v$ 的 $0\sim m-1$ 下标且不能对其进行修改，并只能访问和修改数组 $dir$ 的 $0\sim m-1$ 位置，越界访问可能出现意料不到的错误。

你的程序中，需要包含本题的头文件：

```cpp
#include "treasure.h"
```

### 如何测试你的程序

下载链接中包含 4 个文件：`treasure.h`、`grader.cpp`、`Alice.cpp`和 `Bob.cpp`。

你需要在本地将自己的程序命名为 `treasure.cpp`，其中包含你编写的 `Alice`和 `Bob`两个函数。

将上述所有程序放在同一个文件夹里，然后直接编译运行 `grader.cpp`。

上述交互库将从标准输入按照以下格式输入数据：

- 第一行，4 个非负整数 $testid,n,m,x$ ，含义如题面所属，其中 $testid$ 决定交互库是否会进行对标识点编号的随机打乱，具体参见”数据范围“部分；
- 接下来 $m$ 行，每行 2 个非负整数 $u[i],v[i]$ ，描述一条道路。

请务必确保输入数据格式符合题面以及”数据范围“部分的要求，否则交互库可能出现意想不到的错误。

上述交互库会将你的程序运行正确与否的相关信息输出至标准输出。

注意：上述交互库运行过程中会创建并使用文件 `treasure.tmp`。

你可以适当修改 `grader.cpp`来实现交互库向文件的输入输出。

注意：考虑到选手本地系统配置不同，我们下发了 `Windows` 和 `Linux` 两个文件夹，分别包含能在两种系统上运行的交互库。如果你在本地仍然无法正常编译运行对应的交互库，你也可以采用如下的方法：

1、将你的源程序粘贴至 `Alice.cpp` 的文件末尾，将 `Alice.cpp` 的输出重定向至 `treasure.tmp`；

2、将你的源程序粘贴至 `Bob.cpp` 的文件末尾,将 `Bob.cpp` 的输入重定向至 `treasure.tmp` ；

3、先编译运行 `Alice.cpp` 再编译运行 `Bob.cpp` ，你只需在运行 `Alice` 程序时按上述格式输入数据，在运行 `Bob` 程序时即会按照上述格式输出相应信息。

## 【样例 1 输入】

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

## 【样例 1 输出】

```
Correct.
cnt = 1
```

## 【样例 1 解释】

以下是一个**并不保证正确**，但可能通过本例的编码和解码程序实现。你可以参照这一实现熟悉交互库的使用方法。

```cpp
void Alice(const int testid, const int n, const int m, const int
    x, const int u[], const int v[], bool dir[])
{
    dir[0] = true;
    for(int i = 1; i < m; ++i)
        dir[i] = false;
}

int Bob(const int testid, const int n)
{
```

```
    vector<pair<int,bool>> e = discover(0);
    return 0;
}
```

当标识点的编号不进行随机打乱时，该程序可通过本样例，且调用 `discover(0)`的返回值 $e$ 的信息如下：$e.size()$ 为 1 ，$e[0].first$ 为 1 ，$e[0].second$ 为 $true$ 。

## 【评分标准】

本题共 25 个子任务，每个子任务满分 4 分，由多个测试点组成。

对于每个测试点，如果你的程序不能正常结束，或 Bob函数的返回值不正确（即不是重新标号后的藏宝地点），或调用了超过 $5 \times 10^5$ 次 `discover`函数，或调用 `discover`函数不符合题目要求，或有其他违反题目要求的行为，将得 0 分；否则你的得分将由该测试点调用 `discover`函数的次数 $cnt$ 决定：

$cnt \le 5000$ ： 4 分；

$5000 < cnt \le 10^4$ ： 3 分；

$10^4 < cnt \le 5 \times 10^4$ ： 2 分；

$5 \times 10^4 < cnt \le 5 \times 10^5$ ： 1 分；

$cnt > 5 \times 10^5$ ： 0 分。

你在每个子任务的得分是该子任务中所有测试点得分的最小值。

## 【数据范围】

| 子任务编号 | $n \le$ | $m \le$ | 特殊条件 |
|---|---|---|---|
| 1 | 5000 | $10^4$ | 无 |
| $2 \sim 3$ | $10^5$ | $2 \times 10^5$ | |
| $4 \sim 5$ | $10^6$ | $2 \times 10^6$ | $m = n-1, u[i] = i+1, v[i] = i$ |
| $6 \sim 8$ | | | $m = n-1, u[i] = i+1, v[i]$ 在 $[\max(i-1000,0), i]$ 中均匀随机 |
| $9 \sim 13$ | | | $m = n-1$ |
| $14 \sim 15$ | | | 与每个标识点相连的边均不超过 100 条 |
| $16 \sim 18$ | | | 标识点的编号不会随机打乱 |
| $19 \sim 25$ | | | 无 |

对于所有测试点，保证 $2 \le n \le 10^6$ ，$n-1 \le m \le 2 \times 10^6$ ，$0 \le u[i], v[i], x < n$ ，给出的地图为无自环无重边的连通图，且每个环的大小均为偶数。

## 【提示】

本题时限 $5s$ ，空间限制 $1024MB$ 。保证对于任何合法的数据及在限制范围内的调用，交互库运行所用的时间不会超过 $2s$ ，运行所用的空间不会超过 $256MB$ ，也就是说，你实际可用的时间至少为 $3s$ ，实际可用的空间至少为 $768MB$ 。但需要注意的是，对于每个测试点，你的 `Alice` 函数和 `Bob` 函数所用时间和空间将会合并计算。

交互库正常运行需包含的头文件已经给出，你的程序可以包含你需要的头文件。

注意交互库使用了 `using namespace std`，你的程序需小心可能的变量名冲突问题。

为符合选手本地调试和评测的实际情况，本题实际评测使用的交互库与下发的交互库不完全相同。这可能导致下发交互库在输入数据规模较大时运行较慢，请不必担心。

你的程序不能进行任何读入、输出和文件操作。

通过访问输入输出文件、攻击评测系统或攻击评测库等方式得分属于作弊行为，所得分数无效。

# 字符树（tree）

## 【题目描述】

给定一棵树，树上的每条边上有一个字符。给定一个常数 $X$。

有两种操作，每次操作输入三个整数与一个字符串：

`1 x y S`：对于 $x$ 到 $y$ 的有向简单路径上的边，将路径上的第 $i$ 条边上的字符替换为 $S$ 上的对应字符 $S_i$，保证这条 $x$ 到 $y$ 之间路径的边数为 $X$。

`2 x y S`：查询 $x$ 到 $y$ 构成的有向简单路径所形成的字符串上，$S$ 匹配的次数（这里的匹配即：将 $S$ 当做模式串，在这条路径构成的字符串上逐位置匹配。）例如：$S = 121$，路径上的字符串为 1212122，则匹配了 2 次，分别在第 $[1, 3]$ 处和 $[3, 5]$ 处。

上述所有字符串 $S$ 的下标从 1 开始，保证每个输入的字符串长度均为 $X$。

## 【输入格式】

第一行三个以空格分隔的整数 $n, m, X$。

之后一行包含 $n-1$ 个数，第 $i$ 个数表示第 $i+1$ 个点的父亲 $f_{i+1}$，保证每个点父亲的编号比这个点的编号小。

之后一行包含 $n-1$ 个字符，第 $i$ 个字符表示第 $i+1$ 个点到父亲的边上的字符 $a_{i+1}$。

之后 $m$ 行，每行输入以空格隔开的三个整数 $opt, x, y$ 与一个长为 $X$ 的字符串 $S$，表示一次操作。

## 【输出格式】

若干行，每行一个整数，表示每次 2 操作的答案。

## 【样例 1 输入】

```
10 7 2
1 2 3 2 3 3 3 3 7
212111121
2 1 4 21
1 10 3 21
1 9 7 22
2 2 10 12
2 6 8 11
1 9 8 12
2 4 7 11
```

**【样例 1 输出】**

```
1
1
1
0
```

**【子任务】**

对于 10% 的数据，满足 $1 \le n, m \le 250$。

对于另外 10% 的数据，不存在 1 操作。

对于另外 10% 的数据，满足 $X = 1$。

对于另外 10% 的数据，满足 $X \le 3$。

对于另外 10% 的数据，满足 $X \ge 20000$。

对于另外 10% 的数据，满足 $f_i = i - 1$。

对于另外 10% 的数据，满足 $a_i \le 1$。

对于另外 10% 的数据，满足 $1 \le n, m \le 2.5 \times 10^4$，$mX \le 2.5 \times 10^4$。

对于另外 10% 的数据，满足 $1 \le n, m \le 2.5 \times 10^5$，$mX \le 2.5 \times 10^5$。

对于 100% 的数据，满足 $1 \le n, m, X \le 5 \times 10^5$，字符集为 $[1, 9]$ 以内的整数，$mX \le 5 \times 10^5$。