

CSP 2023 Div.3

2023 年 10 月 14 日

题目名称	数数	树树	数树	树数
题目类型	传统	传统	传统	传统
可执行文件名	calc	tree	graph	greedy
输入文件名	calc.in	tree.in	graph.in	greedy.in
输出文件名	calc.out	tree.out	graph.out	greedy.out
时间限制	1.0 sec	1.0 sec	1.0 sec	2.0 sec
是否捆绑测试	否	否	否	否
内存限制	1024 MiB	1024 MiB	1024 MiB	1024 MiB
结果比较方式	全文比较	全文比较	全文比较	全文比较
编译选项	-lm -std=c++14 -O2			

注意事项：

1. 评测在 Windows 11 下进行，处理器为 12th Gen Intel(R) Core(TM) i7-12700 2.10 GHz，采用 LemonLime 进行评测。时限以上述配置为准，编译时会开大栈空间；若配置有变化，将时限调整为标程 1.5 倍。
2. 文件名（包括程序名和输入输出文件名）必须使用英文小写；无需建立子文件夹。
3. C++ 中函数 main() 的返回值类型必须是 int，值必须为 0；
4. 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；
5. 全文比较忽略行末空格、文末回车；
6. 题目顺序以 std 长度升序排列。

数数 (calc)

【问题描述】

蛤蟆喜欢数数。

现在蛤蟆有一个 $n \times m$ 的网格，最开始所有格子都是白色的。记第 i 行第 j 列的格子为 (i, j) 。蛤蟆想将所有格子都涂黑。但是它有一个要求：

- (i, j) 这个格子在涂黑之前，必须满足前 i 行每行都至少有一个白色格子，前 j 列每列都至少有一个白色格子。

现在蛤蟆想请你求出将所有格子涂黑的方案数，对 998244353 取模。

蛤蟆想知道多种网格的答案，所以有多组数据。

【输入格式】

从文件 `calc.in` 中读入数据。

第一行一个整数 T 表示数据组数。

每组数据一行两个正整数 n, m 。

【输出格式】

输出到文件 `calc.out` 中。

输出 T 行，每行一个整数表示答案对 998244353 取模之后的值。

【样例 1 输入】

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

【样例 1 输出】

```
1
6
60
```

【样例 2】

见下发文件中 `calc2.in, calc2.ans`。

【数据规模与约定】

对于所有数据, $T = 3, 1 \leq n \times m \leq 10^7$ 。

测试点编号	特殊性质
1, 2	$n, m \leq 3$
3	$n, m \leq 5$
4	$n = 1$
5	$n, m \leq 10$
6	$n, m \leq 100$
7, 8	$n \times m \leq 10^5$
9, 10	无

树树 (tree)

【问题描述】

有一颗 n 个节点的有根树，树上节点的编号从 1 到 n ，且 1 为根节点。最开始，每个节点上都有一只蛤蟆。在每一秒末，蛤蟆都会从当前节点跳到其父亲节点上。如果蛤蟆当前在 1 号节点，那么他就会跳出这棵树。

同时，每个节点上都有一只水母。如果以一只水母所在的节点为根节点的子树内有 x 只蛤蟆，那就会带给它 a_x 的快乐度。现在你想知道第 $0 \sim n-1$ 秒时所有水母的快乐度之和。

由于森林里有很多树，所以你需要对每棵树都求出答案。

【输入格式】

从文件 `tree.in` 中读入数据。

第一行一个正整数 T 表示树的个数。

每棵树输入格式如下：

第一行一个正整数 n 表示树的节点数。

第 2 行 n 个整数，第 i 个整数 f_{i+1} 表示第 $i+1$ 号节点的父亲。

第 3 行 $n+1$ 的整数表示 $a_0 \sim a_n$ 。

【输出格式】

输出到文件 `tree.out` 中。

输出 T 行。

对于节点数为 n 的树，输出 n 个整数表示 $0 \sim n-1$ 秒时所有水母的快乐度之和。

【样例 1 输入】

```
4
6
1 2 3 2 2
3 5 7 4 7 8 1
6
1 1 3 2 2
3 8 6 5 7 7 10
6
1 2 1 1 3
3 9 2 7 9 9 5
2
```

```
1
0 2 3
```

【样例 1 输出】

```
31 29 24 20 18 18
45 30 20 18 18 18
41 29 23 24 18 18
5 2
```

【样例 2】

见下发文件中 tree2.in,tree2.ans。

【样例 3】

见下发文件中 tree3.in,tree3.ans。

【样例 4】

见下发文件中 tree4.in,tree4.ans。

【数据规模与约定】

对于所有数据, $1 \leq n, \sum n \leq 10^6, 0 \leq a_i \leq 10^9, 1 \leq f_i < i$ 。

测试点编号	$n \leq$	$\sum n \leq$	特殊性质
1 ~ 2	1000	10^4	无
3	5×10^5	5×10^5	$f_i = i - 1$
4 ~ 6	10^5	2×10^5	无
7 ~ 8	2×10^5	5×10^5	无
9 ~ 10	10^6	10^6	无

数树 (graph)

【问题描述】

蛤蟆小时候无聊时喜欢数树，有一天它发现可以数星星，于是成为了一个喜欢研究星际旅行的天文学家。在它的想象世界里，星空可以抽象成一个 n 个点 m 条边的有向图，点的编号从 $1 \sim n$ ，边有边权，并且不存在自环。蛤蟆发现了一个神奇的性质，就是这些有向边都是从编号小的点指向编号大的点。

蛤蟆想象自己正在 1 号节点，正要前往 x 号节点。蛤蟆还有一个喜欢的数 r ，他希望自己到 x 节点走过的路径长度恰好是 r 。总所周知，天文学家允许有一定的误差。蛤蟆有一个误差参数 p 。假设实际走的路径长度是 t ，那么只要满足 $r \leq t \leq \frac{pr}{p-1}$ ，蛤蟆就会满意。

现在蛤蟆想问你，是否存在这样一条满足要求的路径？由于蛤蟆希望前往很多个点，所以会有多组询问。由于蛤蟆想象力丰富，所以会有多组数据。

【输入格式】

从文件 `graph.in` 中读入数据。

第一行一个整数 T 表示数据组数。

每组数据第一行四个整数 n, m, q, p ，含义如上所述。

接下来 m 行每行三个整数 u, v, w ，表示一条边。

接下来 q 行每行两个整数 x, r ，表示一组询问。

【输出格式】

输出到文件 `graph.out` 中。

对于每组数据，输出一个长度为 q 的 01 字符串表示答案。若第 i 组询问有解，那么字符串的第 i 位就是 1，否则就是 0。

【样例 1 输入】

```
1
3 3 5 20
1 2 20
2 3 1
1 3 10
2 19
2 20
3 20
3 21
```

3 9

【样例 1 输出】

11110

【样例 2】

见下发文件中 graph2.in,graph2.ans。

【样例 3】

见下发文件中 graph3.in,graph3.ans。

【样例 4】

见下发文件中 graph4.in,graph4.ans。

【数据规模与约定】

对于所有数据，满足 $1 \leq T \leq 1000, 1 \leq n, m, q, \sum n, \sum m, \sum q \leq 5 \times 10^5, 2 \leq p \leq 20, 1 \leq w \leq 10^{11}, 1 \leq r \leq 10^{17}$ 。

测试点编号	特殊限制	额外条件
1 ~ 2	$n, m, q \leq 10$	无
3 ~ 6	$\sum n, \sum m, \sum q \leq 5000$	$r \leq 5000$
7 ~ 10	$\sum n \leq 1000, \sum m \leq 2000$	一组数据所有 r 相等
11 ~ 15	无	一组数据所有 r 相等
15 ~ 20	无	无

树数 (greedy)

【问题描述】

给定一棵 n 个节点带边权的有根树，根节点为 1。

定义：

- $\text{dis}(u, v)$ 表示 u 到 v 的路径上的边权和 ($u = v$ 则为 0)。
- $\text{sub}(u)$ 表示 u 子树内的点所成的集合。

你需要执行以下三种操作 q 次：

- 1 u v : 将 u, v 路径上的边权变为原本的相反数 ($u = v$ 则不操作)。
- 2 x u : 询问 $\min_{v \in \text{sub}(x)} \text{dis}(u, v)$ 。
- 3 x y : 询问 $\min_{u \in \text{sub}(x)} \min_{v \in \text{sub}(y)} \text{dis}(u, v)$ 。

【输入格式】

从文件 `greedy.in` 中读入数据。

第一行两个整数 n, q 。

接下来 $n - 1$ 行，第 i 行两个整数 f_i, w_i ，表示 i 的父亲以及其到父亲的边的长度。

接下来 q 行，每行三个整数代表一次操作。保证至少有一次 2 或 3 操作。

【输出格式】

输出到文件 `greedy.out` 中。

对于每个 2 或 3 操作，输出一行一个整数表示答案。

【样例 1 输入】

```
3 3
1 114
1 514
2 1 2
1 1 3
2 1 2
```


【样例 1 输出】

0
-400

【样例 2】

见下发文件中 greedy2.in, greedy2.ans。

【样例 3】

见下发文件中 greedy3.in, greedy3.ans。

【样例 4】

见下发文件中 greedy4.in, greedy4.ans。

【样例 5】

见下发文件中 greedy5.in, greedy5.ans。

【数据规模与约定】

对于所有数据, $1 \leq n, q \leq 2 \times 10^5, 1 \leq f_i < i, -10^9 \leq w_i \leq 10^9$ 。

特殊性质 A: 没有 1 操作。

特殊性质 B: 没有 2 操作。

特殊性质 C: 没有 3 操作。

特殊性质 D: $f_i = i - 1$ 。

测试点编号	$n, q \leq$	特殊性质
1 ~ 2	100	无
3	3000	C
4 ~ 5	3000	无
6	10^5	ACD
7	10^5	ABD
8	10^5	AC
9	10^5	AB
10	10^5	AD
11	10^5	BD
12	10^5	CD
13	10^5	A
14	10^5	B
15	10^5	C
16 ~ 17	2×10^5	D
18 ~ 20	2×10^5	无