

简单图数据库

时间限制：300.0 秒

空间限制：2048 MB

相关文件：题目目录

题目背景

北岛曾经写过一首名为《生活》的一字诗：网。这一个字便概括出了这个世界上各种事物之间错综复杂的关联。没有什么事物是孤立的，关联无处不在。而图正是描述事物之间关联性的一个强有力的工具。

例如，在社交网络上，对于给定的一个用户，找出他/她的朋友、他/她的朋友的朋友、他/她的朋友的朋友的朋友中，有多少人拥有和他同样的生肖。这类查询在实际应用中非常常见。

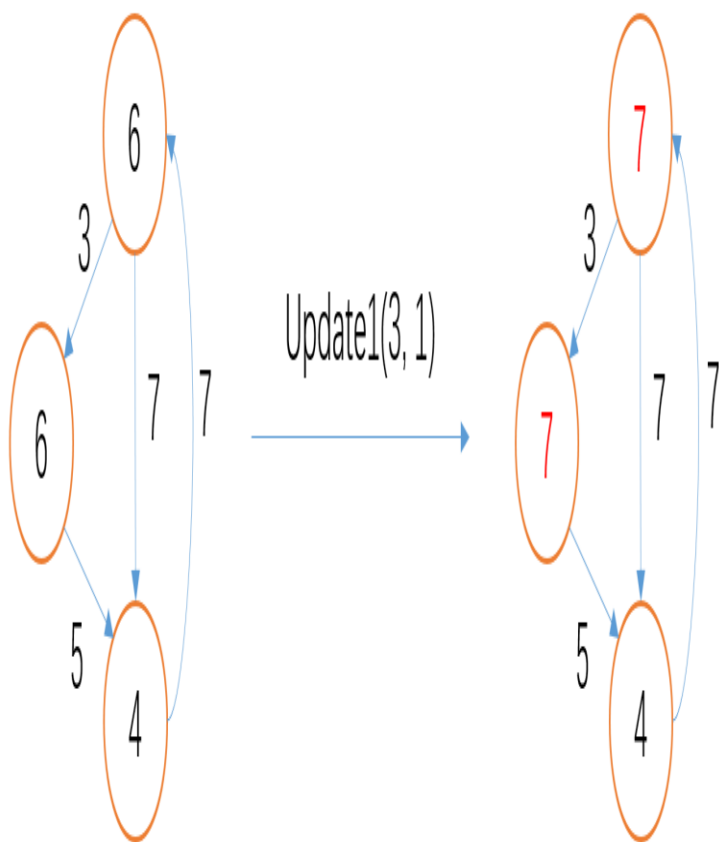
传统的关系型数据库在处理图上的查询时往往效率不高，而图数据库将图作为第一公民，在数据结构的组织和对查询的处理等方面有很多针对图的专门的优化，这使得图数据库在一些特定应用场合表现出远超传统关系型数据库的效率，从而成为了业内新宠。

题目描述

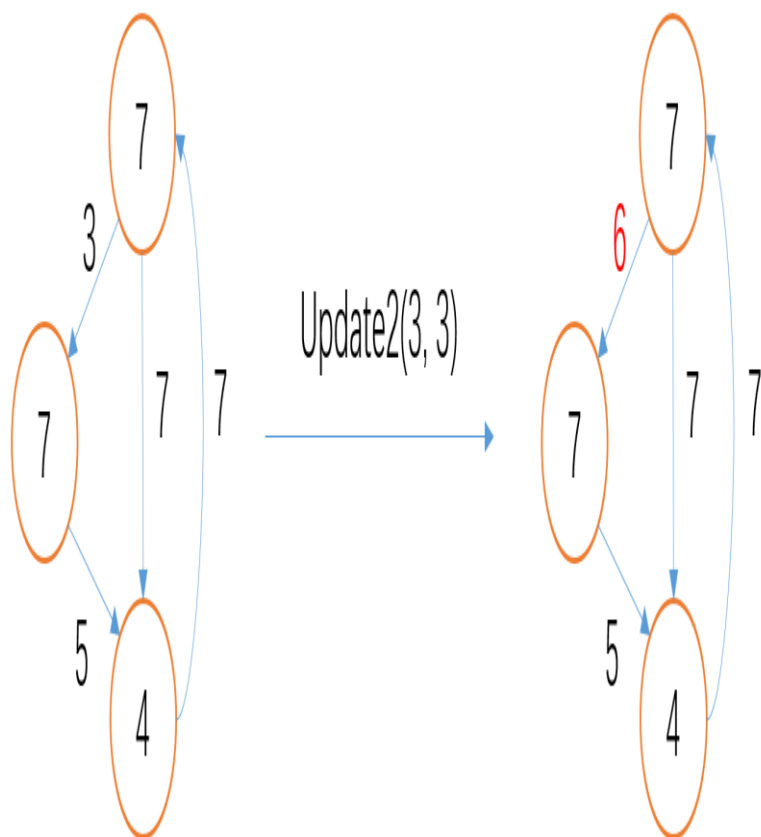
本题要求你实现一个单机图数据库上的几个简单功能。

假设数据库中存在一张有向图，图上有 n 个点，每个点有唯一的编号，编号依次为 $0, 1, 2, \dots, n-1$ ，图上的每个点都有个属性值，每条边也都有个属性值，你需要实现如下 5 个针对此图的操作：

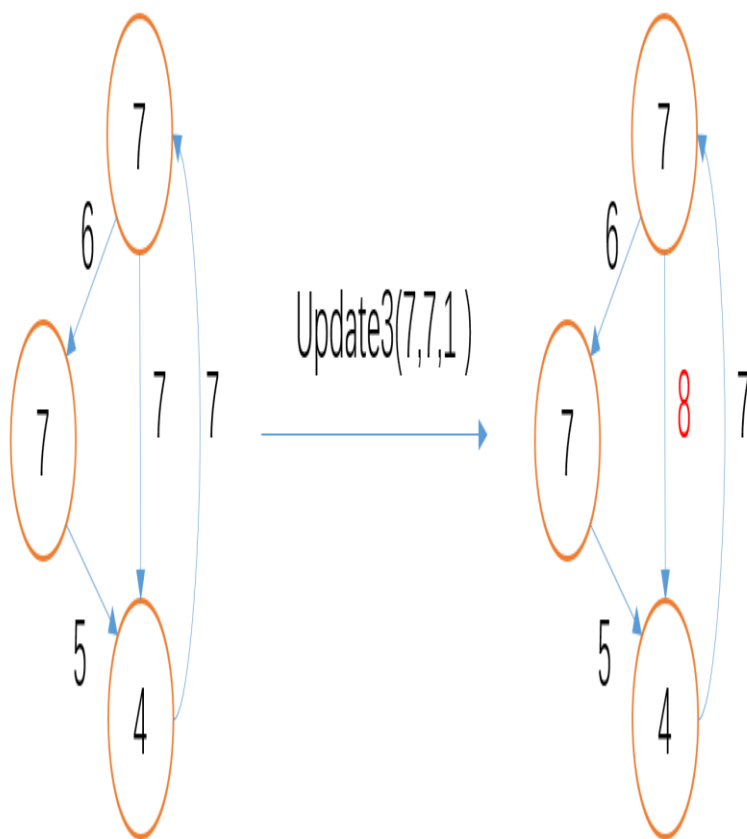
操作 1： $update1(a, b)$ ：表示将属性值为 a 的整数倍的点的属性值加上 b 。



操作 2 : $\text{update2}(a,b)$: 表示将属性值为 a 的整数倍的边的属性值加上 b 。

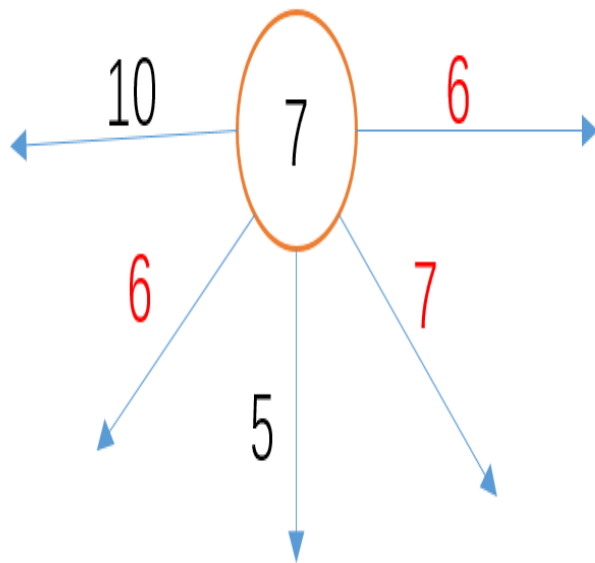


操作 3 : $\text{update3}(a,b,c)$: 表示将属性值为 a 的整数倍的点的出边中, 属性值为 b 的整数倍的边的属性值加上 c 。



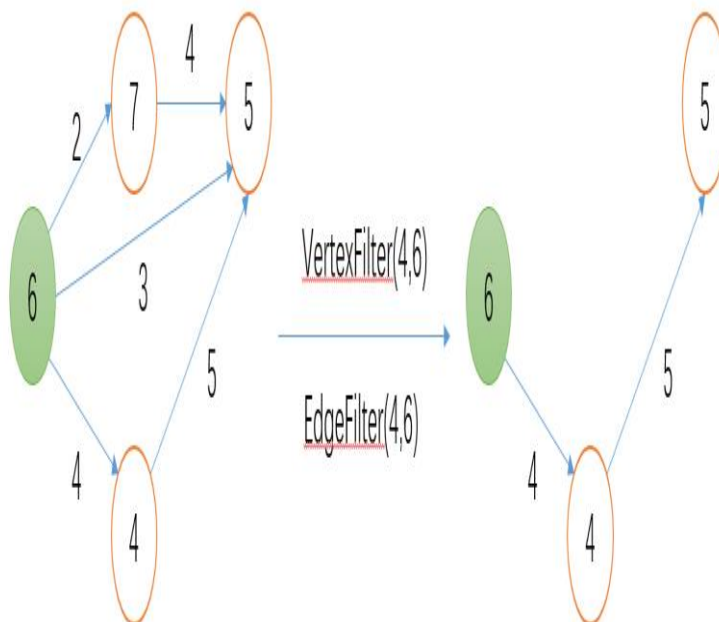
注意：在操作 1、2 和 3 中，如果一个属性加上另一个数后得到的新的值大于 M ，则需要将其减去 M 。例如，对于 $update1(3, M-1)$ ，如果一个点的属性值为 3，则执行此操作后新的属性值等于 2。

操作 4： $query1(u, v, x, y)$ ，表示查询点的编号在 u 到 v 之间的所有点的出边中，属性范围在 x 到 y 的边的数量之和。



假设上图中是 0 号节点及其所有出边，7 是其点的属性值，10,6,5,7,6 是其出边的属性值，那么 $query1(0,0,6,7)$ 的返回值就是 3。

操作 5 : $query2(s,d,u,v,x,y)$ ，表示查询点的属性范围在 u 到 v ，边的属性范围在 x 到 y 的子图中，点 s 在 d 跳以内可以达到的点的数目，如果 s 本身不在该子图中，则返回 0。



假设上图绿色的点是起点，编号为 0，那么 $query2(0,1,4,6,4,6)$ 的返回值是 2。对于属性值为 7 的那个点，因为该点本身不包含在子图中，所以其所有的邻边，不论属性值如何，都不

包含在子图中。所以符合的要求的点有两个，一个是 0 点本身，一个是其相邻的属性为 4 的点。

输入格式

读入有向图

你需要从指定的文本文件中读取有向图的数据。正式测试时的文件路径为 `./graph.data`，文件的格式如下：

第 1 行是一个正整数 n ，表示图中点的个数。点的编号为 0 到 $n-1$ 。

第 2 行是 n 个正整数，第 i 个数表示编号为 $i-1$ 的点的初始属性值。保证这 n 个正整数均不超过 M 。

第 3 行是一个整数，表示图中总的边数。

第 4 行到第 $n+3$ 行，第 $4+i$ 行表示编号为 i 的点的所有出边。其中每行开始有一个非负整数 x 表示该点有 x 条出边，接下来有 x 对整数，第 j 对整数 (d,l) 表示该点的第 j 条边的终点编号为 d ，该边的初始属性值为 l 。保证 $0 \leq d < n$ ，且 $0 < l \leq M$ 。

读入操作序列

你需要从标准输入流中读取你需要完成的操作。格式如下：

第 1 行是 1 个正整数 q ，表示操作的个数。

第 2 行到 $n+1$ 行中，第 $i+1$ 行表示第 i 个操作。每行第一个字符串表示操作的类型， $u1$ ， $u2$ ， $u3$ 分别表示 `update1,2,3`， $q1$ ， $q2$ 别表示 `query1,2`。接下来的若干整数则是该操作的参数。每个操作的格式如下：

- `update1(a,b)` : $u1 \ a \ b$
- `update2(a,b)` : $u2 \ a \ b$
- `update3(a,b,c)` : $u3 \ a \ b \ c$
- `query1(u,v,x,y)` : $q1 \ u \ v \ x \ y$
- `query2(s,d,u,v,x,y)` : $q2 \ s \ d \ u \ v \ x \ y$

上述操作各自参数的含义参见题目描述。

本题中 M 等于 106。

`update1` 和 `update2` 中的 a ，`update3` 中的 a 和 b ，在大多数情况下不超过 10。

`update1` 和 `update2` 中的 a,b ，`update3` 中的 a,b,c ，`query1` 中的 x,y ，`query2` 中的 u,v,x,y 均为不超过 M 的正整数。

对于同一个操作中的 u,v 或 x,y , 有 $u \leq v, x \leq y$ 。特别地, $query1$ 中 $0 \leq v - u \leq 10000$ 。

$query1$ 中的 u,v 和 $query2$ 中的 s,d 均为小于 n 的非负整数。

输出格式

你需要输出到标准输出流。

对于 $update$, 你不需要输出任何东西。对于每个 $query1$ 或 $query2$, 你需要输出一个整数, 表示所求的答案。每个整数单独占据一行, 请不要输出多余的空格和空行。

样例 1 输入

有向图文件:

```
7
6 7 4 5 6 6 4
9
3 1 2 2 4 3 3
1 3 4
1 3 5
0
2 5 6 6 7
1 6 5
1 4 7
```

操作序列:

```
7
q2 0 1 4 6 4 6
q1 4 6 3 6
u1 3 1
u2 3 3
u3 7 7 1
q1 4 6 3 5
q1 4 6 8 8
```

样例 1 输出

```
2
2
1
1
```

评分方式

测试点：一共有 2 个测试点，每个测试点有多个操作。

实际评测中用到的有向图和参赛选手本机的有向图完全相同，各类操作的数目和参赛选手本机的测试数据大致相同。

计分方法：总分 100 分，每个测试点 50 分。对于每个测试点，如果该测试点的输出有错、超时、超内存限制或运行错误等则该测试得 0 分；若全部正确则首先得到 10 分的正确分，剩余 40 分是性能得分，通过和其他参赛者的计算时间的对比算出（用 40 分乘上最快的参赛者的程序耗时，再除以该参赛者的程序耗时）。

提示

评测机器为多核机器，所以你可以用并行化的方法来加速计算。但是你输出的结果必须和串行依次执行每个操作的输出结果完全相同。