**512MB,2S,osbt.xxx**

**硬盘自带的OS中的B树**

**问题描述**

作为胆大心细的未来科技有限公司下属小王电脑店的员工，你注意到所有硬盘都自带OS，一天你决定研究一下OS的架构，你被OS中的一种叫做B树的数据结构所吸引，B树结构十分复杂，执行任务却十分重要，所以很有必要花10分钟完全了解B树，最好的学习就是应用，所以董事长决定让你写出一棵B树。

B树由N个带权节点，编号1到N，每个点有颜色，N-1条带权边构成。

B树是一棵有根树，初始根为1

B树支持加边，将边加入后，与之前B树的N-1条边，生成一棵最小生成树，若存在多棵最小生成树，选择编号和尽可能小的一棵（边按输入顺序依次编号）。

B树支持将某个以a为根的子树变为某个节点b的儿子。

B树支持换根。

B树支持修改点权。

B树支持修改链上点颜色。

B树支持修改边权。

B树支持查询a到b链上的连续颜色段数。（例如1,1,3,3,1,2的颜色段数为4）

B树支持查询某个子树权值之和。

B树支持查询含某个节点，并且节点均在其子树中的最大权子连通块权值。

B树拥有一种特殊查询——查询从根节点到所有询问点的公共路经长度。

$\text{#pragma GCC optimize(2)}$

**输入描述**

第一行两个整数N, M表示B树有N个点，有M次操作

接下来一行2\*N个用空格隔开的整数，vi, ci 分别表示i号节点的权值和颜色。

接下来N-1行，每行3个整数a b v 表示一条连接a点和b点的权值为v的树边。

接下来M行，表示操作。

1 a b v 有一条权值为v的连接a和b的边加入。

2 a b v将以a为根的子树（含a）变为节点b的儿子,a与b之间的边权为v（若b在a的子树中则忽略这次操作）。

3 a 将根设为a。

4 a v 将a点权值改为v。

5 a b c 将a点到b点颜色值赋为c。

6 a b v将a b之间的边的权值改为v（如果a b之间没有边直接相连则忽略）。

7 a b 查询a到b链上的连续颜色段数。

8 a 查询a子树的权值和。

9 a 查询包含a节点，且其他节点均在其子树中的最大权自连通块权值。

10 k a1 a2 .. ak 有k个询问点，查询从根节点到所有询问点的公共路经长度。

**输出描述**

对于每个7,8,9,10操作，输出一行一个整数表示询问的答案。

**样例输入**

3 11

2 2 2 5 -3 1

2 1 2

2 3 2

1 1 3 2

2 1 3 1

3 2

4 3 -5

5 2 2 1

6 2 3 3

7 1 2

8 3

9 2

10 2 1 3

10 1 1

**样例输出**

2

-3

3

4

**数据范围及提示**

2 <= N <= 10,0000; 0 <= M <= 10,0000; 1 <= c <= 10,0000,0000;

|v| <= 1,0000; 0 <= Σk <= 10,0000; 1 <= k <= 10,0000;