

Description

temporaryDO 是一个很菜的 Oler。在 4 月，他在省队选拔赛的考场上见到了《林克卡特树》一题，其中 $k = 0$ 的部分是求树 T 上的最长链。可怜的 temporaryDO 并不会做这道题，他在考场上抓猫耳挠猫腮都想不出一思路。

这时，善良的板板出现在了空中，他的身上发出璀璨却柔和的光芒，荡漾在考场上。“题目并不难。”板板说。那充满磁性的声音，让 temporaryDO 全身充满了力量。他决定：写一个枚举点对求 LCA 算距离的 $k = 0$ 的 $O(n^2 \log n)$ 的部分分程序！于是，temporaryDO 选择以 1 为根，建立了求 LCA 的树链剖分结构，然后写了二重 for 循环枚举点对。

然而，菜菜的 temporaryDO 不小心开小了数组，于是数组越界到了一片神秘的内存区域。但恰好的是，那片内存区域存储的区域恰好是另一棵树 T' 。这样一来，程序并没有 RE，但他求 x 和 y 的距离的时候，计算的是

$$\text{depth}(x) + \text{depth}(y) - (\text{depth}(\text{LCA}(x, y)) + \text{depth}'(\text{LCA}'(x, y)))$$

最后程序会输出每一对点对 $i, j (i \leq j)$ 的如上定义的“距离”的最大值。

temporaryDO 的程序在评测时光荣地爆零了。但他并不服气，他决定花好几天把自己的程序跑出来。请你根据 T 和 T' 帮帮可怜的 temporaryDO 求出他程序的输出。

Input

第一行包含一个整数 n ，表示树上的节点个数；

第 2 到第 n 行，每行三个整数 x, y, v ，表示 T 中存在一条从 x 到 y 的边，其长度为 v ；第 $n + 1$ 到第 $2n - 1$ 行，每行三个整数 x, y, v ，表示 T' 中存在一条从 x 到 y 的边，其长度为 v 。

Output

输出一行一个整数，表示 temporaryDO 的程序的输出。

Sample Input

```
6
1 2 2
1 3 0
2 4 1
2 5 -7
3 6 0
1 2 -1
2 3 -1
2 5 3
2 6 -2
3 4 8
```

Sample Output

```
5
```

Sample Explanation

点对 $(3, 4)$ 的距离计算为 $3 + 0 - (0 + (-2)) = 5$ 。

Task

下发文件包含一份源代码: `code3.cpp` , 你的任务是构造一组满足 $1 \leq n \leq 10^5, |v| \leq 2017011328$ 的合法输入, 使得 `code3.cpp` 无法在时间限制内得出正确的答案。

该子任务时间限制为 1 秒, 请注意读入消耗的时间不会被计入程序的运行时间。

在该子任务中, `code3.cpp` 将会被运行 3 次, 你只要在任意一次运行中成功即可。

source : [「CTSC2018」暴力写挂](#)