

NWERC 2020 Practice

Problem B Broken Gearbox

The mechanical Turk was an 18th-century sham robot designed to give the illusion of artificial intelligence by playing chess. More importantly, the Turk inspired us to make our own fake robot that appears to solve programming contest problems.

We put some gears on axles inside an uncovered panel to make our machine look more realistic. Since the gears are just for visual effect, they have been placed so as to create an impressive meshing pattern; however, the placement is without any regard to gear ratios or turning direction, so it's possible that none of the gears can really move.



Several interlocking gears by Thomas Claveirole, CC BY-SA

It is guaranteed that every axle was connected to every other axle by meshing, either directly or indirectly. Two axles a and b are said to have directly meshing gears if their distance from one another is equal to the sum of the radii of their gears ($d = r_{g_a} + r_{g_b}$).

Sadly, the gears fell off our machine. We think we collected all of them back up again, but now we're not sure which should go back on which axle. Please find a way of putting the gears on the axles to make them mesh the way they did originally.

Input

The input consists of:

- One line with an integer n ($2 \leq n \leq 10^5$), the number of gears and axles.
- One line with n integers r_1, \dots, r_n ($2 \leq r_i \leq 10^8$ for each i), the radius of each gear.
- One line with an integer m ($n - 1 \leq m \leq 10^5$), the number of pairs of gear axles that originally had meshing gears.
- m lines, the i th of which contains three integers a_i , b_i and d_i ($1 \leq a_i < b_i \leq n$, $1 \leq d_i \leq 10^8$), the indices of two connected gear axles and the distance between them.

Output

If it is possible to fix the machine with the given gears, output n integers g_1, \dots, g_n , where g_i is the 1-based index of the gear to be put on the i th axle.

Otherwise, output "impossible".

NWERC 2020 Practice

Sample Input 1

```
4
20 10 5 15
4
1 2 15
2 3 30
3 4 35
1 4 20
```

Sample Output 1

```
3 2 1 4
```

Sample Input 2

```
6
10 55 80 5 60 50
6
2 4 60
3 4 65
4 5 90
1 6 65
5 6 85
1 4 70
```

Sample Output 2

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

Sample Input 3

```
4
1 2 1 1
4
1 4 3
2 4 3
3 4 3
1 3 2
```

Sample Output 3

```
3 1 4 2
```

Sample Input 4

```
3
3 4 5
2
1 2 7
1 3 7
```

Sample Output 4

```
impossible
```