2020 Canadian Computing Olympiad
Day 1, Problem 1
A Game with Grundy

Time Limit: 1 second

Problem Description

Grundy is playing his favourite game - hide and seek.

His N friends stand at locations on the x-axis of a two-dimensional plane - the i-th one is at coordinates $(x_i, 0)$. Each friend can see things in a triangular wedge extending vertically upwards from their position – the i-th friend's triangular wedge of vision will be specified by two lines: one with slope of v_i/h_i and the other with slope $-v_i/h_i$. A friend cannot see a point that lies exactly on one of these two lines.

Grundy may choose to hide at any location (a, Y), where a is an integer satisfying $L \leq a \leq R$, and L, R, and Y are given integer constants.

Each possible location may be in view of some of Grundy's friends (namely, strictly within their triangular wedge of vision).

Grundy would like to know in how many different spots he can stand such that he will be in view of at most i of his friends, for every possible value of i from 0 to N.

Input Specification

The first line of input contains the integer N ($1 \leq N \leq 100\,000$).

The next line contains three integers: L, R and Y ($-1\,000\,000\,000 \leq L \leq R \leq 1\,000\,000\,000, 1 \leq Y \leq 1\,000\,000$).

Each of the next N lines contain three integers: the i-th such line contains x_i ($L \leq x_i \leq R$), the x-value of the position of friend i followed by two integers v_i and h_i ($1 \leq v_i, h_i \leq 100$). The slopes v_i/h_i and $-v_i/h_i$ define the triangular wedge of vision for friend i.

For 15 of the 25 marks available, $-1\,000\,000 \leq L \leq R \leq 1\,000\,000$.

Output Specification

The output is $N + 1$ lines, where each line i ($0 \leq i \leq N$) contains the integer number of positions in which Grundy can stand and be in view of at most i of his friends.

Sample Input

```
3
-7 7 3
0 2 3
```

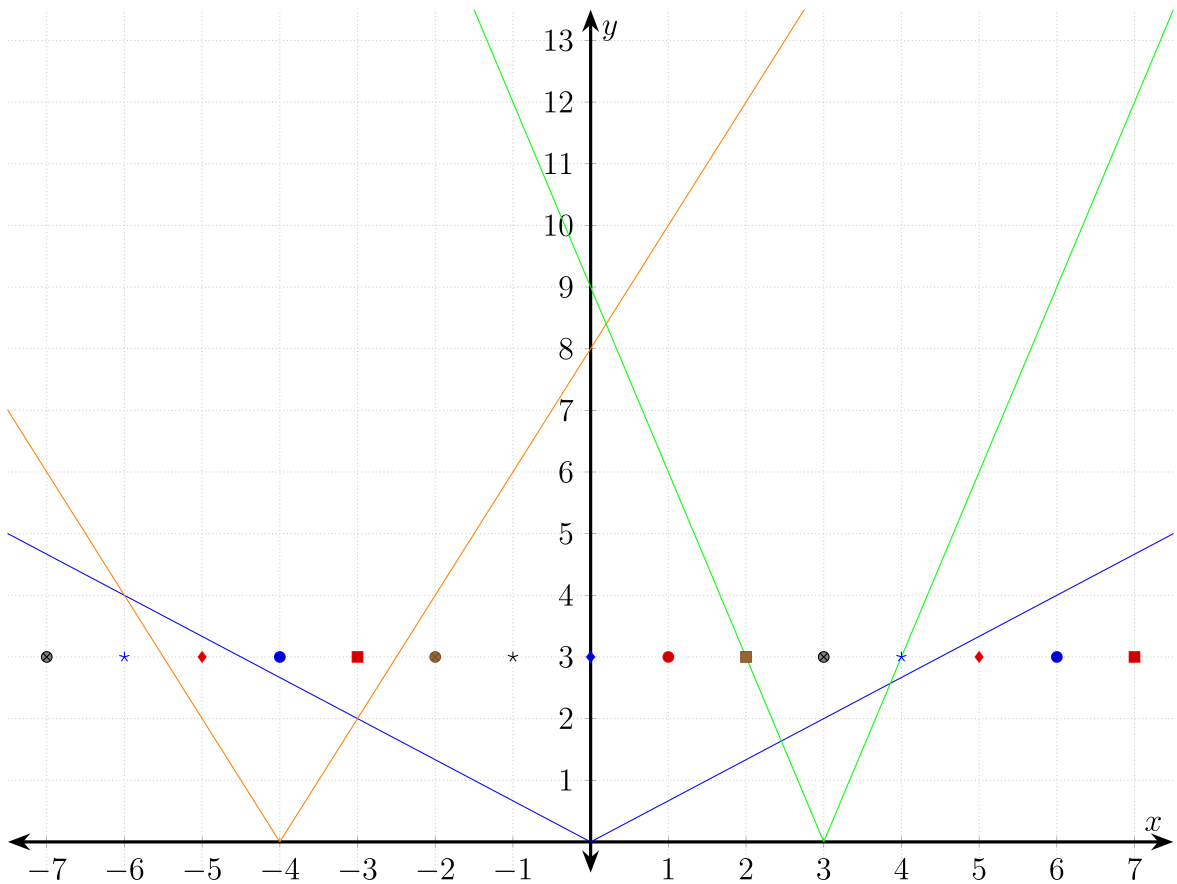
```
-4 2 1
3 3 1
```

Output for Sample Input

```
5
12
15
15
```

Explanation of Output for Sample Input

There are three friends with the following triangular wedges of vision, along with the possible positions that Grundy can be placed, as shown in the diagram below:



Notice the points $(2, 3)$ and $(4, 3)$ are visible only by the friend at position 0, since they lie on the boundary of the triangular wedge of vision for the friend at position 3.

2020 Canadian Computing Olympiad
Day 1, Problem 2
Exercise Deadlines

Time Limit: 1 second

Problem Description

Bob has N programming exercises that he needs to complete before their deadlines. Exercise i only takes one time unit to complete, but has a deadline d_i ($1 \leq d_i \leq N$) time units from now.

Bob will solve the exercises in an order described by a sequence $a_1, a_2, \ldots, a_N$, such that a_1 is the first exercise he solves, a_2 is the second exercise he solves, and so on. Bob's original plan is described by the sequence $1, 2, \ldots, N$. With one *swap* operation, Bob can exchange two adjacent numbers in this sequence. What is the minimum number of swaps required to change this sequence into one that completes all exercises on time?

Input Specification

The first line consists of a single integer N ($1 \leq N \leq 200\,000$). The next line contains N space-separated integers $d_1, d_2, \ldots, d_N$ ($1 \leq d_i \leq N$).

For 17 of the 25 marks available, $N \leq 5000$.

Output Specification

Output a single integer, the minimum number of swaps required for Bob to solve all exercises on time, or `-1` if this is impossible.

Sample Input 1

```
4
4 4 3 2
```

Output for Sample Input 1

```
3
```

Explanation of Output for Sample Input 1

One valid sequence is $(1, 4, 3, 2)$, which can be obtained from $(1, 2, 3, 4)$ by three swaps.

Sample Input 2

```
3
1 1 3
```

Output for Sample Input 2

```
-1
```

Explanation of Output for Sample Input 2

There are two exercises that are due at time 1, but only one exercise can be solved by this time.

2020 Canadian Computing Olympiad
Day 1, Problem 3
Mountains and Valleys

Time Limit: 7 seconds

Problem Description

You are planning a long hiking trip through some interesting, but well-known terrain. There are N interesting sites you would like to visit and M trails connecting pairs of sites. Each trail has a difficulty level indicated as a positive integer.

The trail system is a bit special, however. There are exactly $N-1$ trails with difficulty level 1 (these are completely flat trails), and the rest of the trails all have a difficulty level of at least $\left\lceil \frac{N}{3} \right\rceil$ (these are very mountainous trails). (The ceiling of x, denoted as $\lceil x \rceil$, is the smallest integer greater than or equal to x.)

Additionally, it is possible to travel between any two sites using only the trails with difficulty level 1.

You would like to visit every site, starting your walk from any site of your choice and finishing at some other site, such that you visit each site at least once and the total sum of difficulty levels is minimum among all walks. Note that walking a trail k times with difficulty level d contributes a value of $k \cdot d$ to the sum of difficulty levels.

Input Specification

The first line of input contains two space-separated integers N ($4 \leq N \leq 500\,000$) and M ($N-1 \leq M \leq 2\,000\,000$).

The next M lines contain three space-separated integers x_i, y_i, and w_i describing the trail between sites x_i and y_i with difficulty level w_i ($1 \leq i \leq M$; $0 \leq x_i, y_i \leq N-1$; $x_i \neq y_i$). Note that there is at most one trail between every pair of sites, and that $w_i = 1$ or $\left\lceil \frac{N}{3} \right\rceil \leq w_i \leq N$.

For 1 of the 25 marks available, $N \leq 6$ and $M \leq 10$.

For an additional 2 of the 25 marks available, $N \leq 20$ and $M \leq 40$.

For an additional 2 of the 25 marks available, $N \leq 5\,000$, $M \leq 10\,000$, and either $w_i = 1$ or $\left\lceil \frac{N}{2} \right\rceil \leq w_i \leq N$.

For an additional 6 of the 25 marks available, $N \leq 100$ and $M \leq 200$.

For an additional 2 of the 25 marks available, $N \leq 500$ and $M \leq 1\,000$.

For an additional 3 of the 25 marks available, $N \leq 5\,000$ and $M \leq 10\,000$.

For an additional 5 of the 25 marks available, $N \leq 80\,000$ and $M \leq 160\,000$.

Output Specification

Output one integer, which is the minimum sum of difficulty levels taken for all trails walked to visit each site at least once.

Sample Input

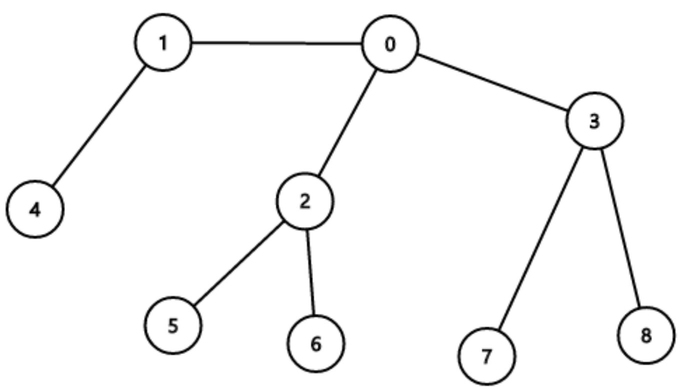
```
9 10
0 1 1
0 2 1
0 3 1
1 4 1
2 5 1
2 6 1
3 7 1
3 8 1
2 4 5
6 7 3
```

Output for Sample Input

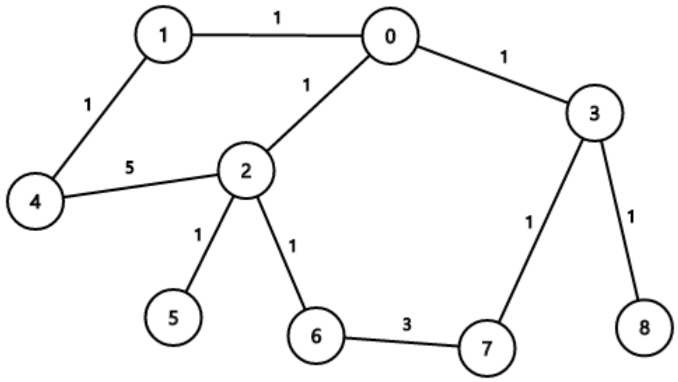
```
11
```

Explanation of Output for Sample Input

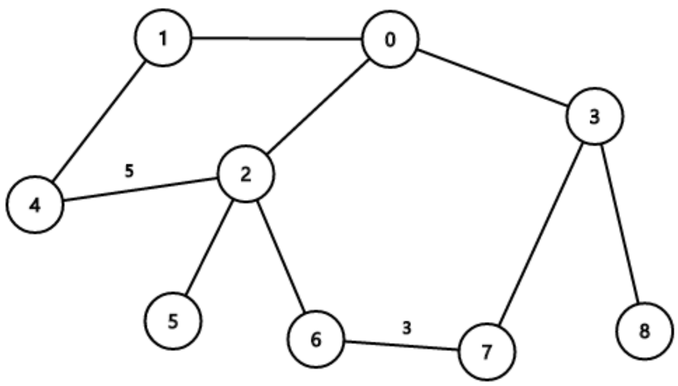
This is the set of flat trails:



This is the entire set of trails with all the difficulty levels.



This is the entire set of trails, with trails with difficulty level 1 being omitted.



An optimal walk for this set of trails is $4 \to 1 \to 0 \to 2 \to 5 \to 2 \to 6 \to 7 \to 3 \to 8$ with a total cost of $1 + 1 + 1 + 1 + 1 + 1 + 3 + 1 + 1 = 11$. There is no way to make a walk that visits all the sites at least once with a lower total difficulty level cost.