Redhat. Красная Шапочка (*)

Имя входного файла: `redhat.in`
Имя выходного файла: `redhat.out`

В непроходимом лесу имеется N полянок и M тропинок между ними. Каждая тропинка соединяет две различные полянки. Две полянки могут быть соединены несколькими тропинками.

На двух разных полянках живут Красная Шапочка и ее бабушка. Домик Красной Шапочки находится на полянке с номером 1, а домик бабушки — на полянке с номером N. Красная Шапочка хорошо ориентируется в лесу и знает, какое минимальное время ей потребуется для прохождения каждой тропинки. Когда Красная Шапочка идет по лесу, она переходит с тропинки на тропинку только на полянках. На каждой полянке есть укрытие, в котором Красная Шапочка может спрятаться на некоторое время.

В этом же лесу живет Волк. Время, за которое Волк пробегает какую-либо тропинку, может отличаться от времени, за которое по ней проходит Красная Шапочка. Кроме того, если Волк пробегает по одной и той же тропинке несколько раз, то каждый раз он может тратить на это разное время.

С края полянки, где живет Красная Шапочка, Волк увидел, что она собирается нести пирожки бабушке и побежал по тропинкам привычного ему пути от дома Красной Шапочки к дому бабушки. Волк начинает бежать от домика Красной Шапочки в тот момент, когда она решила выйти из дома, его путь заканчивается как только он окажется на полянке с домиком бабушки. Ни на одной полянке Волк не задерживается.

Чтобы застать бабушку в целости и сохранности, Красной Шапочке необходимо обогнать Волка. При этом ей нельзя оказаться с Волком на одной тропинке, даже если Волк уже покидает ее, а она только появляется на ней, или наоборот. Чтобы избежать встречи с Волком на полянке, Красная Шапочка использует имеющееся там укрытие. Красной Шапочке нельзя появляться на полянке одновременно с Волком или покидать укрытие на полянке в тот момент, когда на ней появляется Волк. При необходимости Красная Шапочка может идти по тропинке дольше минимально возможного времени, а также выйти из дома позже, чем она исходно решила.

Необходимо написать программу, которая поможет Красной Шапочке добраться к бабушке раньше Волка, если известна последовательность тропинок, по которым побежал Волк.

Формат входного файла

Первая строка входного файла содержит числа N, M и K ($2 \le N \le 2\,000$, $1 \le M \le 100\,000$, $1 \le K \le 100\,000$). Следующие M строк содержат по три числа: B_i, E_i — номера полянок, которые соединяет i-я тропинка, и T_i — минимальное время, за которое Красная Шапочка может по ней пройти ($1 \le T_i \le 10\,000$).

В следующих K строках находится последовательное описание пути Волка, по два числа в строке: P_i — номер тропинки, по которой он побежит, и V_i — время, которое он на это затратит ($1 \le V_i \le 10\,000$). Путь волка всегда начинается на полянке 1 и заканчивается на полянке N.

Все числа во входном файле целые и в пределах одной строки разделены пробелами.

Формат выходного файла

В том случае, если Красная Шапочка не может добраться до домика бабушки быстрее Волка, выходной файл должен содержать слово “`NO`”.

Если Красная Шапочка сможет добраться до домика бабушки быстрее волка, в первой строке выходного файла должно быть слово “`YES`”. Во второй строке в этом случае должно содержаться число тропинок в пути Красной Шапочки. В третью строку следует вывести номера тропинок в том порядке, в котором Красная Шапочка должна по ним пройти. Числа должны быть разделены пробелами.

Информацию о времени прохождения по тропинкам и остановках на полянках в выходной файл выводить не нужно.

Пример

<table>
<tr><th>redhat.in</th><th>redhat.out</th></tr>
<tr><td><pre>4 4 5
1 3 6
1 2 2
2 3 2
3 4 1
2 1
2 2
2 1
3 4
4 1</pre></td><td><pre>YES
2
1 4</pre></td></tr>
<tr><td><pre>4 3 4
1 2 2
2 3 1
2 4 3
1 2
2 1
2 2
3 5</pre></td><td><pre>NO</pre></td></tr>
</table>

Hilander. Горец (*)

Имя входного файла: `hilander.in`
Имя выходного файла: `hilander.out`

ЛКШата играют в игру «Горец». Каждый из них получил бумажку с именем ЛКШонка, которого он должен «убить», причем имя каждого из n игроков напечатано ровно на одной табличке.

Игрок может подкараулить свою «жертву» в укромном месте и «убить» ее. В этом случае, «убийца» надрывает бумажку с именем «жертвы» и забирает все ее накопленные бумажки. Таким образом, у всех живых игроков всегда есть ровно одна ненадорванная бумажка.

Если два игрока в какой-то момент имеют бумажки с именами друг друга, то «убьет» другого тот, кто первым предъявит свою бумажку. Будем считать, что спорных ситуаций никогда не возникает.

Если игроку досталась неразорванная бумажка с его именем, то для него игра закончилась.

Игра идет до тех пор, пока есть игрок, на чьей бумажке напечатано чужое имя.

Выигрывает тот, у кого в конце игры на руках будет больше всего разорванных бумажек. Если игроков с таким количеством бумажек несколько, то все они объявляются победителями.

Преподаватели собрали со всей ЛКШ информацию о том, у кого чье имя было на бумажке в начале игры.

Определите список игроков, которые теоретически могут победить в этой игре.

Формат входного файла

Первая строка входного файла содержит натуральное число n — количество ЛКШат, играющих в «Горца» ($n \le 210$).

Каждая из следующих n строк содержат информацию об очередном игроке в следующем виде:

`<Имя игрока>: <Имя, напечатанное на его бумажке>`

Формат выходного файла

В выходной файл выведите список игроков, которые теоретически могут победить, отсортированный в лексикографическом порядке.

Пример

hilander.in

```
10
Лев Толмачев: Сергей Низовцев
Алексей Вихарев: Лев Толмачев
Виктор Матюхин: Виктор Матюхин
Сергей Низовцев: Дмитрий Васин
Василий Болгар: Александр Лебедкин
Дмитрий Васин: Мария Дуженко
Александр Лебедкин: Тимур Гарифзянов
Мария Дуженко: Филипп Рухович
Тимур Гарифзянов: Василий Болгар
Филипп Рухович: Алексей Вихарев
```

hilander.out

```
Алексей Вихарев
Дмитрий Васин
Лев Толмачев
Мария Дуженко
Сергей Низовцев
Филипп Рухович
```