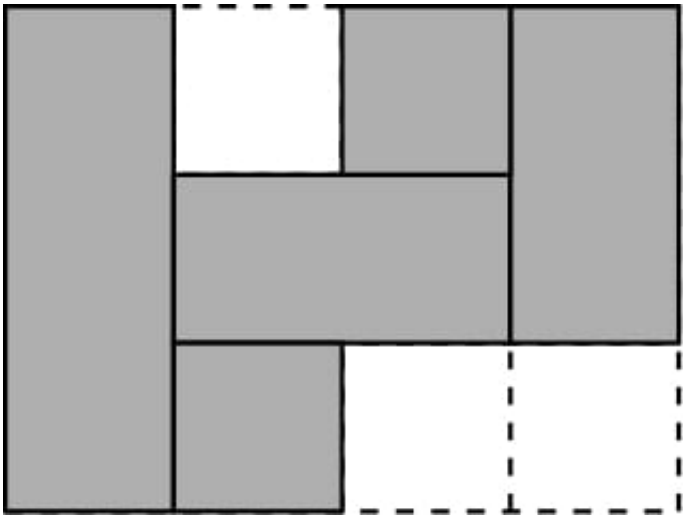
Задача A. Кирпичи

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

Кирпич — прямоугольник с целыми сторонами шириной 1 или высотой 1 (или и то и другое).

Дана сетка $n \times m$, и каждая ячейка окрашена в черный или белый цвет. Замощение — это способ поместить кирпичи на сетку так, чтобы каждая черная ячейка была покрыта **ровно одним** кирпичом, а каждая белая ячейка не была покрыта кирпичом. Другими словами, кирпичи размещаются только в черных ячейках, покрывают все черные ячейки, и **никакие два кирпича не перекрываются**.



Пример замощения с первого примера с использованием 5 кирпичей. Существует также замощение из 4 кирпичей.

Какое минимальное количество кирпичей необходимо для замощения данной сетки?

Формат входных данных

Первая строка содержит два целых числа n, m ($1 \leqslant n, m \leqslant 200$) — количество строк и столбцов соответственно.

Следующие n строки описывают сетку. i-я строка содержит строку длиной m, где j-я строка обозначает цвет ячейки в строке i, столбец j. Черная ячейка обозначается символом «#», а белая — символом «.».

Гарантируется, что есть хотя бы одна черная ячейка.

Формат выходных данных

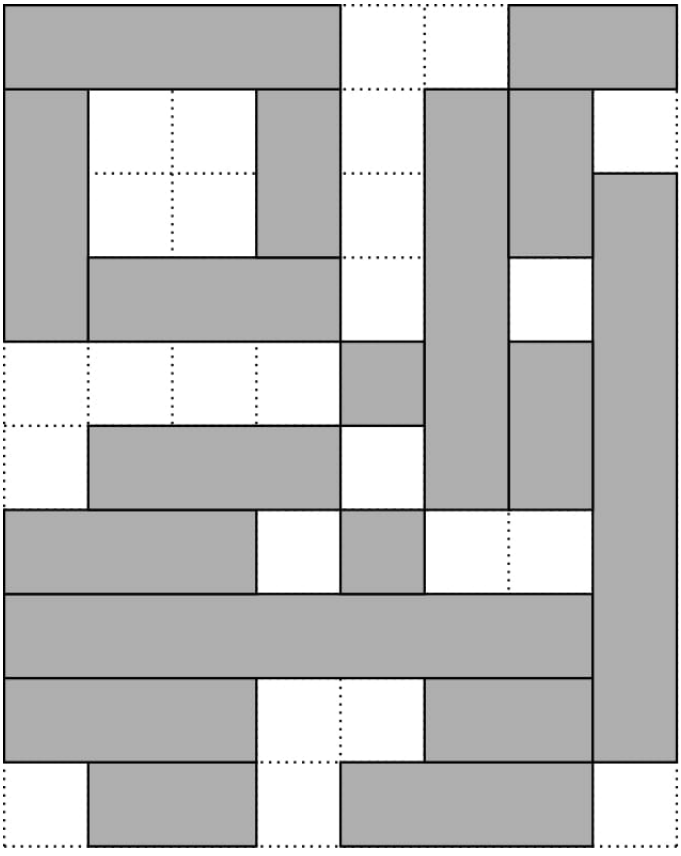
Выведите единственное целое число — минимальное количество требуемых кирпичей.

Примеры

стандартный ввод	стандартный вывод
3 4 #.## #### ##..	4
6 6 ###### ##.... ###### ##...# ##...# ######	6
10 8 ####..## #..#.##. #..#.### ####.#.# ....#### .###.### ###.#..# ######## ###..### .##.###.	18

Замечание

Сетка с первого примера может быть замощена 4-мя кирпичами, размещенными вертикально.
Сетка с третьего примера может быть замощена такими 18 кирпичами:



Задача B. Радиус взвешенного дерева

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	6 секунд
Ограничение по памяти:	512 мегабайт

Вам задано дерево из n вершин и $n-1$ ребра. Первоначальный вес i-й вершины равен a_i.

Назовем *расстоянием* $d_v(u)$ от вершины v до вершины u количество ребер на пути из v в u. Заметим, что $d_v(u) = d_u(v)$ и $d_v(v) = 0$.

Назовем *взвешенным* расстоянием $w_v(u)$ от v до u значение $w_v(u) = d_v(u) + a_u$. Заметим, что $w_v(v) = a_v$ и $w_v(u) \neq w_u(v)$, если $a_u \neq a_v$.

Аналогично обычному расстоянию, назовем *эксцентриситетом* $e(v)$ вершины v наибольшее взвешенное расстояние от v до какой-либо вершины (включая саму v), или $e(v) = \max\limits_{1 \leqslant u \leqslant n} w_v(u)$.

Наконец, назовем *радиусом* r дерева наименьший из эксцентриситетов его вершин, или $r = \min\limits_{1 \leqslant v \leqslant n} e(v)$.

Вам нужно обработать m запросов следующего вида:

- v_j x_j — присвоить $a_{v_j} = x_j$.

После обработки каждого запроса выведите радиус r текущего дерева.

Формат входных данных

В первой строке задано одно целое число n ($2 \leqslant n \leqslant 2 \cdot 10^5$) — количество вершин в дереве.

Во второй строке заданы n целых чисел $a_1, \ldots, a_n$ ($0 \leqslant a_i \leqslant 10^6$) — первоначальные веса вершин.

В следующих $n-1$ строках заданы ребра дерева. В i-й строке заданы два целых числа u_i и v_i ($1 \leqslant u_i, v_i \leqslant n$; $u_i \neq v_i$) — соответствующее ребро. Заданные ребра образуют дерево.

В следующей строке задано одно целое число m ($1 \leqslant m \leqslant 10^5$) — количество запросов.

В следующих m строках заданы сами запросы — по одному в строке. В j-м запросе заданы два целых числа v_j и x_j ($1 \leqslant v_j \leqslant n$; $0 \leqslant x_j \leqslant 10^6$) — вершина и ее новый вес.

Формат выходных данных

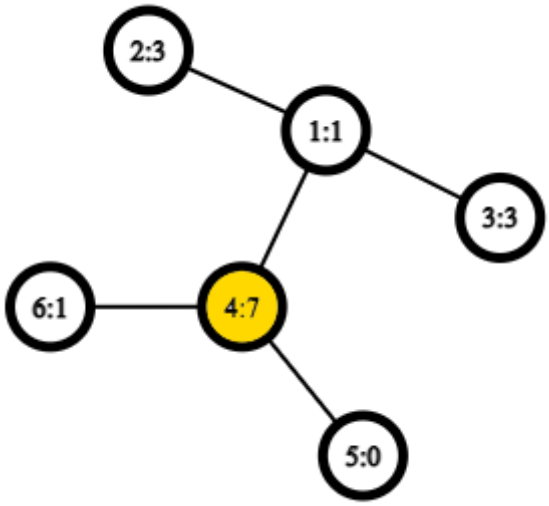
Выведите m целых чисел — радиус r дерева после обработки каждого запроса.

Пример

стандартный ввод	стандартный вывод
6 1 3 3 7 0 1 2 1 1 3 1 4 5 4 4 6 5 4 7 4 0 2 5 5 10 5 5	7 4 5 10 7

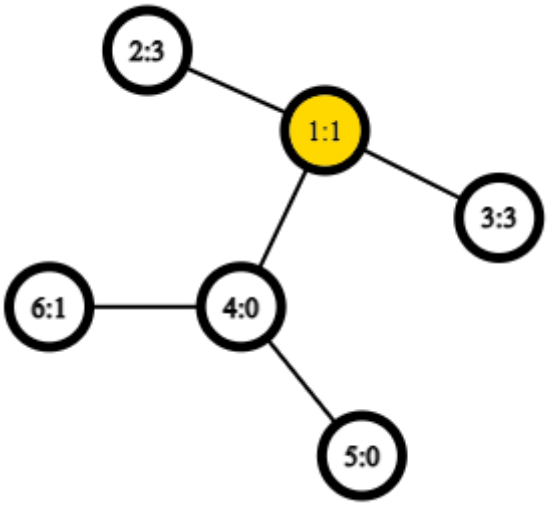
Замечание

После первого запроса дерево выглядит следующим образом:



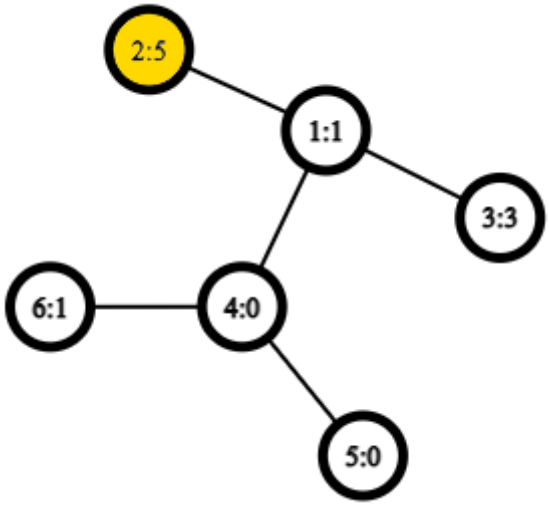
Цветом на изображении отмечена вершина с наименьшим $e(v)$, соответственно $r = e(4) = 7$. Эксцентриситеты других вершин равны: $e(1) = 8$, $e(2) = 9$, $e(3) = 9$, $e(5) = 8$, $e(6) = 8$.

Дерево после второго запроса:



Радиус $r = e(1) = 4$.

После третьего запроса радиус $r = e(2) = 5$:


2:5
1:1
3:3
6:1
4:0
5:0

Задача C. Комбокамень

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Одна большая компания разрабатывает компьютерную игру «Комбокамень», которая должна мигом перевернуть всю индустрию. Правила игры достаточно сложные, и на реализацию серверного движка, моделирующего ход игры, был объявлен конкурс. От вас требуется реализовать подобный серверный движок.

Суть игры — игроки умеют призывать на арену и усиливать существ, используя заклинания, а также заставлять их сражаться друг с другом. Каждое существо имеет два параметра — численное значение атаки a и численное значение оставшегося здоровья h. Для краткости будем обозначать параметры существа как (a, h). Исходно на арене нет существ.

Игроку доступны следующие заклинания:

- *Призыв существа*: Призвать новое существо с характеристиками $(1, 1)$. Если уже в игру было введено k существ, то новое существо получает номер $k + 1$.
- *Благословение силы*: Удвоить атаку выбранного существа. Если до применения этого заклинания оно имело характеристики (a, h), то после этого действия оно будет иметь характеристики $(2a, h)$.
- *Божественный дух*: Удвоить здоровье выбранного существа. Если до применения этого заклинания оно имело характеристики (a, h), то после этого действия оно будет иметь характеристики $(a, 2h)$.
- *Копия из лавы*: Призвать новое существо, которое будет иметь такие же характеристики, как и выбранное заклинанием существо. Если уже в игру было введено k существ, то новое существо получает номер $k + 1$.
- *Сражайся!*: Заставить двух различных существ сразиться. Во время сражения оба существа одновременно наносят друг другу по одному удару, уменьшая количество здоровья соперника на значение своей атаки. Так, если сражаются два существа с характеристиками (a_1, h_1) и (a_2, h_2), то после сражения они будут иметь характеристики $(a_1, h_1 - a_2)$ и $(a_2, h_2 - a_1)$, соответственно. Если после сражения у существа остается 0 или меньше единиц здоровья, оно умирает и больше не может участвовать в игре.

От серверного движка, на реализацию которого объявлен конкурс, требуется способность промоделировать все события и для каждого созданного во время игры существа вывести номер хода, на котором оно погибло, либо определить, что оно осталось живо к концу игры.

Кроме того, движок должен корректно обрабатывать случаи, когда игрок пытается взаимодействовать с мертвыми по мнению сервера существами: если заклинание *Благословение силы*, *Божественный дух* или *Сражайся!* обращено к уже мертвому существу, то не должно произойти ничего. Если заклинание *Копия из лавы* применено к мертвому существу, создается его мертвая копия с такими же характеристиками, но умершая на текущем ходу, в момент копирования.

Формат входных данных

В первой строке дано число n — количество совершенных ходов ($1 \leqslant n \leqslant 250\,000$).

В следующих n строках даны ходы, пришедшие к серверному движку, в следующем формате:

- 1 — применить заклинание *Призыв существа*;
- 2 i — применить заклинание *Благословение силы* к существу с номером i;
- 3 i — применить заклинание *Божественный дух* к существу с номером i;
- 4 i — применить заклинание *Копия из лавы* к существу с номером i;
- 5 i j — применить заклинание *Сражайся!* к существам с номерами i и j.

Гарантируется, что любые упомянутые в запросах существа к моменту запроса уже были призваны, но, возможно, могут уже быть мертвы.

Формат выходных данных

В первой строке выведите одно целое число k — количество существ, призванных за время игры.

В следующей строке выведите k целых чисел $t_1, t_2, \ldots, t_k$ — если существо с номером i осталось живо к концу игры, то t_i должно быть равно -1, иначе t_i должно быть равно номеру хода, на котором оно погибло.

Пример

стандартный ввод	стандартный вывод
16 1 2 1 3 1 1 5 1 2 3 1 1 3 3 3 3 4 1 5 1 3 3 3 5 1 3 5 4 3 5 4 3 4 1	5 13 5 14 -1 16

Замечание

В таблице можно увидеть, как изменялись характеристики существ в первом примере.

ход	1	2	3	4	5
0	-	-	-	-	-
1	(1, 1)	-	-	-	-
2	(2, 1)	-	-	-	-
3	(2, 2)	-	-	-	-
4	(2, 2)	(1, 1)	-	-	-
5	(2, 1)	мертво	-	-	-
6	(2, 2)	мертво	-	-	-
7	(2, 2)	мертво	(1, 1)	-	-
8	(2, 2)	мертво	(1, 2)	-	-
9	(2, 2)	мертво	(1, 4)	-	-
10	(2, 2)	мертво	(1, 4)	(2, 2)	-
11	(2, 1)	мертво	(1, 2)	(2, 2)	-
12	(2, 1)	мертво	(1, 4)	(2, 2)	-
13	мертво	мертво	(1, 2)	(2, 2)	-
14	мертво	мертво	мертво	(2, 1)	-
15	мертво	мертво	мертво	(2, 1)	-
16	мертво	мертво	мертво	(2, 1)	мертво

Задача D. Black and White Tree

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Вам дано дерево из N вершин, пронумерованных от 1 до N. i-е из $N-1$ ребер соединяет вершины a_i и b_i. Изначально, каждая вершина не покрашена.

Takahashi и Aoki играют в игру, в которой они красят вершины. Первым ходит Takahashi. В этой игре они последовательно выполняют данную операцию:

- Выберите вершину, которая еще не покрашена.
- Если это ход Takahashi, то покрасьте вершину в белый цвет, если же это ход Aoki, то покрасьте ее в черный.

После того, как все вершины становятся покрашенными происходит данная процедура:

- Перекрасьте каждую белую вершину соседнюю с черной в черный цвет.

Заметьте, что все вершины перекрашиваются одновременно.

Если остается хотя бы одна белая вершина, то побеждает Takahashi. Если же все оставшиеся вершины черного цвета, то побеждает Aoki. Назовите победителя игры, если Aoki и Takahashi играют оптимально.

Формат входных данных

Первая строка содержит число N ($2 \leqslant N \leqslant 10^5$)

Следующие $N-1$ строк содержат пары чисел a_i и b_i ($1 \leqslant a_i, b_i \leqslant n$)

Гарантируется, что данный граф - дерево.

Формат выходных данных

Выведите "`First`" если Takahashi побеждает; выведите "`Second`" если Aoki побеждает.

Примеры

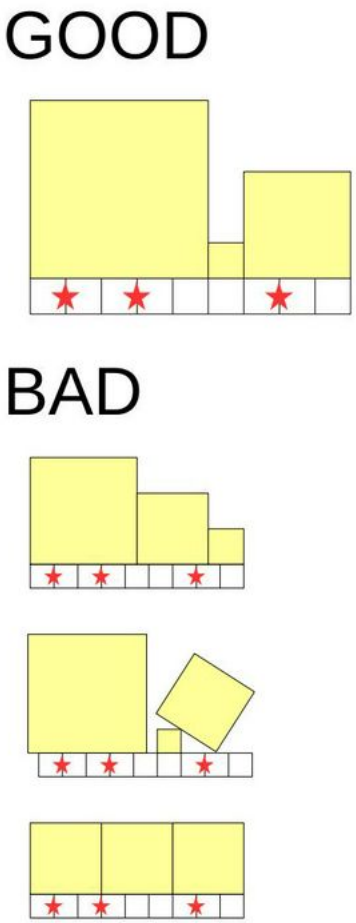
стандартный ввод	стандартный вывод
3 1 2 2 3	First
4 1 2 2 3 2 4	First
6 1 2 2 3 3 4 2 5 5 6	Second

Задача E. Баобаб в поисках квадратов

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

У Баобаба есть полоска длины N, на которой есть M отметок. Расстояние от левого конца до i-й отметки равняется X_i. Он хочет положить на эту полоску квадраты так, чтобы удовлетворялись следующие условия:

- На полоске будут квадраты с целочисленными размерами сторон.
- Каждый квадрат должен лежать так, чтобы его нижняя сторона касалась полоски.
- Полоска должна быть полностью покрыта квадратами.
- Границы никаких двух квадратов не могут быть над отметками.



Красотой расстановки корректного расположения квадратов будем называть произведение площадей квадратов. Баобаб хочет посчитать сумму красот по всем корректным способам расположить квадраты. Так как это число может быть довольно большим, найдите его по модулю $10^9 + 7$.

Формат входных данных

Первая строка входных данных содержит два целых числа N и M ($1 \leqslant N \leqslant 10^9$, $1 \leqslant M \leqslant 10^5$). В следующей строке содержатся M целых чисел X_i ($1 \leqslant X_1 < X_2 < \ldots < X_{M-1} < X_M \leqslant N - 1$).

Формат выходных данных

Выходные данные должны содержат одно целое число — сумму красот расстановок квадратов по модулю $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
3 1 2	13
5 2 2 3	66
10 9 1 2 3 4 5 6 7 8 9	100

Задача F. Алгоритм жа дфс

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Леон и Шелли нашли по корневому дереву. В каждом из этих двух деревьев по N вершин. Чтобы победить в матче, они решили найти такой набор целых чисел $X_1, X_2, \ldots, X_N$ такой, что следующее условие выполняется:

- Для любой вершины v любого дерева пусть $i_1, i_2, \ldots, i_k$ — индексы ее детей (считаются не только непосредственные дети). Леон и Шелли хотят, чтобы $abs(X_v + X_{i_1} + X_{i_2} + \ldots + X_{i_n}) = 1$.

Скажите, возможно ли построить такую последовательность $X_1, X_2, \ldots X_N$, чтобы все условия выполнялись. В случае, если такая последовательность существует, приведите пример любой корректной последовательности X, такой что $|X_i| \leqslant 10^9$.

Формат входных данных

В первой строке входных данных содержится целое число N ($1 \leqslant N \leqslant 10^5$) — количество вершин дерева.

Во второй строке содержатся N целых чисел par_i. Если i является корнем дерева Леона, то $par_i = -1$, иначе $1 \leqslant par_i \leqslant N$ и par_i является предком вершины i в дереве Леона.

Во третьей строке содержатся N целых чисел par_i. Если i является корнем дерева Шелли, то $par_i = -1$, иначе $1 \leqslant par_i \leqslant N$ и par_i является предком вершины i в дереве Шелли.

Формат выходных данных

Если невозможно построить последовательность X, выведите «IMPOSSIBLE», иначе выведите в первой строке «POSSIBLE», а во второй строке выведите любую корректную последовательность X.

Примеры

стандартный ввод	стандартный вывод
5 3 3 4 -1 4 4 4 1 -1 1	POSSIBLE -1 1 1 -1 1
6 -1 5 1 5 1 3 6 5 5 3 -1 3	IMPOSSIBLE
8 2 7 1 2 2 1 -1 4 4 -1 4 7 4 4 2 4	POSSIBLE -1 0 1 -2 1 1 0 1