Задача A. Биномиальная куча

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Реализуйте биномиальную кучу.

Формат входных данных

В первой строке содержится два целых числа: N — общее количество куч и M — количество операций ($1 \leqslant N \leqslant 1000, 1 \leqslant M \leqslant 1\,000\,000$). Изначально все кучи пусты.

Требуется поддерживать следующие операции:

- `0 v a` — добавить элемент со значением v в кучу с номером a. Вновь добавленный элемент имеет уникальный индекс равный порядковому номеру соответствующей операции добавления. Нумерация начинается с единицы.
- `1 a b` — переложить все элементы из кучи с номером a в кучу с номером b. После этой операции куча a становится пустой.
- `2 i` — удалить элемент с индексом i.
- `3 i v` — присвоить элементу с индексом i значение v. Гарантируется, что элемент существует.
- `4 a` — вывести на отдельной строке значение минимального элемента в куче с номером a. Гарантируется, что куча не пуста.
- `5 a` — удалить минимальный элемент из кучи с номером a. Если таковых несколько, то выбирается элемент с минимальным индексом. Гарантируется, что куча не пуста.

Формат выходных данных

Для каждой операции поиска минимального элемента выведите единственное число: значение искомого элемента.

Примеры

стандартный ввод	стандартный вывод
3 19 0 1 10 4 1 0 2 5 0 2 7 4 2 3 2 20 4 2 1 2 1 4 1 5 1 4 1 3 2 3 4 1 2 2 4 1 0 1 9 1 1 3 0 3 8 4 3	10 5 7 7 10 3 10 8

Задача B. K кратчайших путей

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

Дан ориентированный взвешенный граф из n вершин и m рёбер. В нем зафиксированы вершины $s = 0$ и $t = n - 1$. Найдите длины k кратчайших путей из s в t. Пути выбираются среди множества всех путей, не обязательно простых. В графе могут присутствовать кратные рёбра, но петель быть не может.

Необходимо для всех i от 1 до k найти длину i-го кратчайшего пути из s в t.

Формат входных данных

В первой строке заданы числа n, m и k — количество вершин, ребер и необходимое количество путей соотвественно ($2 \leqslant n \leqslant 50\,000$, $1 \leqslant m \leqslant 50\,000$, $1 \leqslant k \leqslant 50\,000$). Следующие m строк содержат описание рёбер в формате a_i, b_i, w_i. Это означает, что i-е ребро соединяет вершины a_i и b_i, и при этом имеет вес w_i ($1 \leqslant w_i \leqslant 100\,000$). Вершины нумеруются с нуля.

Формат выходных данных

Необходимо вывести k целых чисел — длины путей в порядке возрастания. Если очередного пути не существует, выведите вместо длины -1.

Примеры

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

Задача C. Персистентная приоритетная очередь

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

Требуется реализовать структуру данных, которая хранит мультимножество и умеет изменять любую свою предыдущую версию, выполняя одну из этих операций:

1. Заданы v и x, требуется добавить в множество v элемент со значением x, после чего вывести минимальный элемент в получившемся множестве.

2. Заданы v и u, требуется объединить множества с номерами v и u, после чего вывести минимальный элемент в получившемся множестве.

3. Задано v, требуется вывести минимальный элемент в множестве v, после чего удалить минимальный элемент из множества v. Если множество пустое, то вывести, что множество пустое, и создать новое пустое множество.

Изначально есть одно пустое множество с номером 0. После операции с номером i множество, получаемое во время этой операции, получает номер i.

Формат входных данных

Первая строка содержит число n — количество операций для выполнения.

От вас потребуется отвечать на запросы в онлайне, при этом поддерживая переменную s. Она изначально равна нулю. После каждой операции, она пересчитывается следующим образом через предыдущее значение: если ответ на запрос равен x, то $s = (s_{old} + x) \bmod 239017$. Если же ответом на запрос является слово `empty`, то s не изменяется.

В следующих n строках заданы запросы.

Запросы первого типа описываются строкой `1 a b`, где a и b — неотрицательные целые числа, которые описывают v и x для соответствующего запроса, как $v = (a + s) \bmod i$ и $x = (b + 17s) \bmod (10^9 + 1)$, где i — номер соответствующего запроса.

Запросы второго типа описываются строкой `2 a b`, где a и b — неотрицательные целые числа, которые описывают v и для соответствующего запроса, как $v = (a + s) \bmod i$ и $u = (b + 13s) \bmod i$, где i — номер соответствующего запроса.

Запросы третьего типа описываются строкой `3 a`, где a — неотрицательное целое число, которые описывает v для соответствующего запроса, как $v = (a + s) \bmod i$, где i — номер соответствующего запроса.

Число запросов не превышает 200 000. Гарантируется, что мощность любого созданного мультимножества не превышает 2^{63}.

Формат выходных данных

Требуется вывести ровно n строк, в каждой строке должно находиться неотрицательное целое число либо слово `empty`.

Для запросов первого и второго типа требуется вывести значение минимального элемента в только что созданном множестве, либо слово `empty`, если множество пустое.

Для запросов третьего типа требуется вывести минимальный элемент в множестве, либо слово `empty`, если множество пустое.

Примеры

стандартный ввод	стандартный вывод
9 1 0 2 1 0 999999970 2 2 0 3 0 2 4 4 3 0 3 0 3 0 3 8	2 3 2 2 2 2 2 3 empty

Задача D. Легчайшая

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

Имеется N наборов чисел, которые изначально известны.

Поступают Q запросов вида x y k, для выполнения запроса нужно взять все числа из набора под номером x, которые не меньше числа k, и переместить их в набор номер y. После выполнения всех запросов необходимо вывести конечные состояния наборов.

Формат входных данных

В первой строке дано два числа — N и Q ($2 \leqslant N \leqslant 10^5$, $1 \leqslant Q \leqslant 10^5$).

В последующих N строках заданы наборы. Каждый набор задается строкой вида: число k, за ним k чисел в неубывающем порядке. Суммарный размер наборов не превышает 10^5. Все числа в наборах от 1 до 10^6.

Далее в Q строках заданы запросы — по три числа x, y и k ($1 \leqslant x, y \leqslant N$, $x \neq y$, $1 \leqslant k \leqslant 10^6$).

Формат выходных данных

Выведите N строк — конечные состояния наборов, выводить числа набора следует в неубывающем порядке.

Примеры

стандартный ввод	стандартный вывод
3 2 3 1 2 3 3 1 2 4 0 1 2 2 2 3 3	1 1 3 1 2 2 2 3 4

Задача E. «Чапаев» на дереве

Имя входного файла:	`game.in`
Имя выходного файла:	`game.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Вова и Марина любят играть в игры, а особенно — придумывать к ним свои правила. Недавно они открыли для себя веселую игру «Чапаев», в которой игроки должны сбивать щелчками шашки вражеского цвета с шахматной доски (также эта игра известна под названием «Щелкунчики»). Вдоволь наигравшись, они решили модифицировать правила, добавив игре математическую сложность.

Теперь они играют в «Чапаева» не на шахматной доске, а на доске в форме дерева! Их дерево состоит из N вершин. Вершина 1 является корнем дерева, а из каждой из оставшихся вершин проведено ребро в некоторую вершину с меньшим номером — ее непосредственного предка.

В игре участвуют шашки одного цвета, изначально расположенные в некоторых вершинах дерева. За один ход игрок выбирает некоторую шашку и щелчком отправляет ее к корню по ребрам дерева, сбивая при этом с доски все встреченные на пути шашки. Сама шашка, по которой производился удар, после попадания в корень дерева также слетает с доски.

Игроки делают ходы по очереди. Проигрывает тот игрок, к ходу которого на доске не остается шашек.

Придуманная ими игра замечательна также тем, что на одной и той же доске можно играть, начиная с разных начальных позиций шашек. Практика показала, что самые интересные партии получаются, если исходно расставить фишки во все вершины, являющиеся потомками (непосредственными или косвенными) некоторой вершины $Root$, при этом в саму вершину $Root$ фишка не ставится.

Дети решили сыграть N партий, перебрав в качестве вершины $Root$ каждую вершину дерева по одному разу. Если у очередной вершины $Root$ нет потомков, и на доске исходно не оказывается ни одной фишки, то игры не происходит, и дети переходят к следующей расстановке. В каждой партии Марина ходит первой.

Вова интересуется у вас, в скольких партиях Марина сможет одержать победу, если игроки будут действовать оптимально.

Формат входных данных

В первой строке находится целое число N ($1 \leqslant N \leqslant 500\,000$) — количество вершин в дереве.

Во второй строке следуют целые числа $p_2, p_3, \ldots, p_N$, разделенные пробелами, где p_i — это номер вершины, являющейся предком вершины i ($1 \leqslant p_i < i$).

Формат выходных данных

Выведите единственное целое число — количество партий, в которых Марина одержит победу.

Примеры

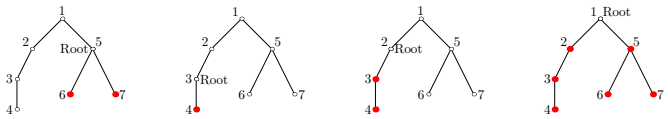
game.in	game.out
7 1 2 3 1 5 5	3

Замечание

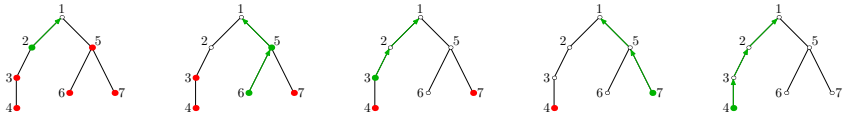
Разберем тест из условия. Доска для игры показана на рисунках ниже. Дети сыграют четыре партии, выбирая в качестве $Root$ вершины 1, 2, 3 и 5. Если выбрать в качестве $Root$ любую из трех оставшихся вершин, на доске исходно не окажется ни одной фишки, поэтому игры не произойдет.

Если выбрать в качестве $Root$ вершину 5, фишки будут исходно находиться в вершинах 6 и 7. В такой партии Марина проигрывает: после того, как она сбивает любую из этих двух фишек с доски, Вова сбивает оставшуюся и заканчивает партию.

Если выбрать в качестве $Root$ вершину 2 или 3, у Марины будет возможность выиграть игру за один ход, щелкнув по фишке из вершины 4 (при этом, в случае $Root = 2$, она по пути также собьет фишку из 3 вершины по правилам игры).



Можно убедиться, что если выбрать в качестве *Root* вершину 1, у Марины также будет выигрышная стратегия. Для этого первым ходом Марина должна сбить фишку из вершины 2. Пример партии с таким начальным расположением показан ниже.



Таким образом, Марина выигрывает в трех партиях.

Тесты к этой задаче состоят из пяти групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп.

0. Тест 1. Тест из условия, оценивается в ноль баллов.
1. Тесты 2–17. В тестах этой группы $N \leqslant 20$. Эта группа оценивается в 20 баллов.
2. Тесты 18–38. В тестах этой группы $N \leqslant 200$. Эта группа оценивается в 20 баллов.
3. Тесты 39–59. В тестах этой группы $N \leqslant 5\,000$. Эта группа оценивается в 20 баллов.
4. В тестах этой группы дополнительные ограничения отсутствуют. Эта группа оценивается в 40 баллов. Решение будет тестироваться на тестах этой группы **offline,** т. е. после окончания тура.