

Задача А. Array

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

Дан массив a длины k . Так же, имеются n массивов длины k . Массив x называется хорошим, если существует перестановка p длины k , такая что для каждого i ($1 \leq i \leq k$) выполняется $x_{p_i} \equiv 0 \pmod{a_i}$.

Вам необходимо посчитать количество пар (l, r) ($1 \leq l \leq r \leq n$), таких что количество хороших массивов в отрезке больше чем нехороших.

Формат входных данных

В первой строке входных данных даются числа n и k ($1 \leq n \leq 10^5, 1 \leq k \leq 20$).

Во второй строке входных данных даётся массив a длины k ($1 \leq a_i \leq 10^9$).

В следующих n строк даются массивы длины k , где в i -ой содержится массив b_i ($1 \leq b_{ij} \leq 10^9$, где $1 \leq i \leq k$).

Формат выходных данных

В единственной строке выходных данных выведите одно число - ответ на задачу.

Система оценки

Данная задача содержит четыре подзадачи:

1. $1 \leq n \leq 1000, k = 1$. Оценивается в 15 баллов.
2. $1 \leq n \leq 1000, 1 \leq k \leq 8$. Оценивается в 19 баллов.
3. $1 \leq n \leq 100, 1 \leq k \leq 20$. Оценивается в 31 баллов.
4. $1 \leq n \leq 10^5, 1 \leq k \leq 8$. Оценивается в 35 баллов.

Пример

стандартный ввод	стандартный вывод
3 1 2 1 2 2	4

Замечание

$a \equiv 0 \pmod{b}$ - означает что число a делится на b без остатка.

В первом тестовом примере, пары $(1, 1)$ и $(1, 2)$ не входят в ответ, так как в одном количество хороших массивов в одном равно 0 ($0 \leq 1$), а во втором равно 1 ($1 \leq 1$).

Задача В. Занимательная игра

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

Недавно Айдос изучая культуру древних народов горных местностей, обнаружил одну очень занимательную игру. Как верили создатели этой игры, она играла очень важную роль в культуре и помогала развивать дедуктивное мышление людей всех возрастов. Айдос проникся интересностью игры и решил перевести ее на современный лад. Вот, что у него получилось.

Изначально вам дается неориентированный граф с n вершинами и m ребрами. У каждой вершины v есть значение a_v . Предлагается произвести над этим графом q операций. Операции могут быть четырех типов:

1. Добавить ребро между существующими вершинами
2. Удалить существующее ребро из графа
3. Изменить значение одной вершины
4. Среди соседей заданной вершины найти k -ю по значению вершину.

Замечание. В графе могут быть кратные ребра и **не гарантируется**, что граф связный.

Формат входных данных

Первая строка входных данных содержит три целых числа n, m, q через пробел — количество вершин, ребер и операций ($1 \leq n, m, q \leq 2 \cdot 10^5$).

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n через пробел — значения вершин ($1 \leq a_i \leq 10^9$).

Следующие m строк содержат описания ребер графа, по два целых числа u, v в каждой строке — неориентированное ребро между вершинами u и v ($1 \leq u, v \leq n$).

В каждой из следующих q строк содержатся описания запросов. Каждое описание соответствует одному из следующих видов:

- $1 \ u \ v$ — добавить ребро между вершинами u и v .
- $2 \ u \ v$ — удалить ребро между вершинами u и v . Если их несколько, то удаляется только одно ребро.
- $3 \ u \ x$ — поменять значение вершины u на x .
- $4 \ u \ k$ — выведите номер вершины по следующему правилу: отсортируйте соседние вершины u сначала по значению вершин, затем по номеру (оба параметра — по неубыванию). Из отсортированного списка выведите k -ю вершину ($1 \leq u, v \leq 2 \cdot 10^5, 1 \leq x \leq 10^9$).

Формат выходных данных

На каждый запрос типа 4 выведите в новой строке номер вершины согласно требованиям в запросе.

Система оценки

Данная задача содержит шесть подзадач:

1. $1 \leq n, q \leq 100$. Оценивается в 6 баллов.
2. $1 \leq n, q \leq 10000$, также в этой подзадаче нет второго типа запроса. Оценивается в 7 баллов.
3. $1 \leq n, q \leq 50000$. Оценивается в 22 баллов.

4. $1 \leq n, q \leq 200000$, $k = 1$, также в этой подзадаче нет третьего типа запроса. Оценивается в 9 баллов.
5. $1 \leq n, q \leq 200000$, также в этой подзадаче нет третьего типа запроса. Оценивается в 12 баллов.
6. Ограничения из условия. Оценивается в 43 баллов.

Пример

стандартный ввод	стандартный вывод
3 2 8	3
3 1 2	1
1 2	1
2 3	1
4 2 1	2
4 2 2	
1 1 3	
4 3 2	
3 1 2	
4 3 2	
2 1 3	
4 3 1	

Задача С. Маглы наступают

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

Интернет в мире представляется в виде n вершин и $n - 1$ ребер, каждое ребро соединяет две вершины, и возможно добраться из каждой вершину в каждую, передвигаясь по ребрам.

В каждой вершине живет Волшебник. Волшебники — очень дружелюбные существа, но они могут существовать только парами. 2 Волшебника являются парой, если они находятся в соседних вершинах и решили быть парой. У Волшебника не может быть большой одной пары.

Маглы решили заблокировать некоторые вершины дерева, а заодно и забрать Волшебников, живущих в них. Каждая блокировка представляет собой k вершин дерева, которые будут заблокированы. Из каждой заблокированной вершины Маглы забирают Волшебника. После блокировки наступает паника и Волшебники ищут себе пару. Если хотя бы один Волшебник (среди тех, кого не забрали) не найдет себе пару, наступает Хаос. **Обратите внимание**, Волшебники разбиваются на пары так, чтобы образовать как можно большее количество пар.

У Волшебников есть q предположений того — какие вершины будут блокировать. И на каждый из этих предположений, Волшебникам интересно, будет ли Хаос если случится такая блокировка.

Формат входных данных

В первой строке входных данных дается два числа n, q ($1 \leq n, q \leq 5 * 10^5$) — количество вершин и количество предположений блокировок соответственно.

В следующих $n - 1$ строках даются два числа u, v ($1 \leq u, v \leq n$) — вершины, между которыми есть ребро.

В следующих q строках подаются описания блокировок в следующем виде:

k, a_1, a_2, \dots, a_k ($0 \leq k \leq n, 1 \leq a_i \leq n$) — количество и номера заблокированных вершин.

Формат выходных данных

Выведите q чисел, разделенных пробелами. Для каждой блокировки в порядке входных данных выведите 1, если все Волшебники смогут разбиться на пары (Хаос не наступит), и 0, если нет (Хаос наступит).

Система оценки

Определим $sum(k)$ как суммарное количество вершин во всех блокировках.

1. $1 \leq n, q, sum(k) \leq 3000$, в дереве нет вершин, у которых больше 2 соседей. Оценивается в 13 баллов.

2. $1 \leq n, q, sum(k) \leq 3000$. Оценивается в 17 баллов.

3. $1 \leq n, q, sum(k) \leq 100000$. Оценивается в 22 баллов.

4. $1 \leq n, q, sum(k) \leq 100000$. $k_i \leq 1$. Оценивается в 19 баллов.

5. $1 \leq n, q, sum(k) \leq 500000$. Оценивается в 29 баллов.

Пример

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

Замечание

После блокировки первой вершины или третьей вершины, остаются лишь 2 вершины, которые связаны ребром и поэтому могут образовать пару, что и делают. В случае удаления 2ой вершины, даже если Волшебники хотят образовать пару, они не могут, так как не связаны прямым ребром.