

Задача А. Разрезание графа

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

Дан неориентированный граф. Над ним в заданном порядке производят операции следующих двух типов:

- **cut** — разрезать граф, то есть удалить из него ребро;
- **ask** — проверить, лежат ли две вершины графа в одной компоненте связности.

Известно, что после выполнения всех операций типа **cut** рёбер в графе не осталось. Найдите результат выполнения каждой из операций типа **ask**.

Формат входных данных

Первая строка входного файла содержит три целых числа, разделённые пробелами — количество вершин графа n , количество рёбер m и количество операций k ($1 \leq n \leq 50\,000$, $0 \leq m \leq 100\,000$, $m \leq k \leq 150\,000$).

Следующие m строк задают рёбра графа; i -я из этих строк содержит два числа u_i и v_i ($1 \leq u_i, v_i \leq n$), разделённые пробелами — номера концов i -го ребра. Вершины нумеруются с единицы; граф не содержит петель и кратных рёбер.

Далее следуют k строк, описывающих операции. Операция типа **cut** задаётся строкой “**cut** u v ” ($1 \leq u, v \leq n$), которая означает, что из графа удаляют ребро между вершинами u и v . Операция типа **ask** задаётся строкой “**ask** u v ” ($1 \leq u, v \leq n$), которая означает, что необходимо узнать, лежат ли в данный момент вершины u и v в одной компоненте связности. Гарантируется, что каждое ребро графа встретится в операциях типа **cut** ровно один раз.

Формат выходных данных

Для каждой операции **ask** во входном файле выведите на отдельной строке слово “**YES**”, если две указанные вершины лежат в одной компоненте связности, и “**NO**” в противном случае. Порядок ответов должен соответствовать порядку операций **ask** во входном файле.

Пример

стандартный ввод	стандартный вывод
3 3 7	YES
1 2	YES
2 3	NO
3 1	NO
ask 3 3	
cut 1 2	
ask 1 2	
cut 1 3	
ask 2 1	
cut 2 3	
ask 3 1	

Задача В. Реструктуризация компании

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

В жизни даже самой успешной компании может наступить кризисный период, когда приходится принимать тяжёлое решение о реструктуризации, распускать и объединять отделы, увольнять работников и заниматься прочими неприятными делами. Рассмотрим следующую модель компании.

В Большой Софтверной Компании работают n человек. Каждый человек принадлежит какому-то *отделу*. Исходно каждый человек работает над своим проектом в своём собственном отделе (таким образом, в начале компания состоит из n отделов по одному человеку).

Однако, в жизни компании наступили тяжёлые времена, и руководство было вынуждено нанять кризисного менеджера, который начал переустраивать рабочий процесс для повышения эффективности производства. Обозначим за $team(person)$ команду, в которой работает человек $person$. Кризисный менеджер может принимать решения двух типов:

1. Объединить отделы $team(x)$ и $team(y)$, сформировав из них один большой отдел, содержащий всех сотрудников $team(x)$ и $team(y)$, где x и y ($1 \leq x, y \leq n$) — номера каких-то двух сотрудников компании. Если $team(x)$ совпадает с $team(y)$, ничего делать не требуется.
2. Объединить отделы $team(x), team(x + 1), \dots, team(y)$, где x и y ($1 \leq x \leq y \leq n$) — номера каких-то двух сотрудников компании.

При этом кризисный менеджер иногда может интересоваться, работают ли в одном отделе сотрудники x и y ($1 \leq x, y \leq n$).

Помогите кризисному менеджеру, ответив на все его запросы.

Формат входных данных

Первая строка входных данных содержит два целых числа n и q ($1 \leq n \leq 200\,000, 1 \leq q \leq 500\,000$) — количество сотрудников компании и количество запросов кризисного менеджера.

В последующих q строках находятся запросы кризисного менеджера. Каждый запрос имеет вид $type\ x\ y$, где $type \in \{1, 2, 3\}$. Если $type = 1$ или $type = 2$, то запрос представляет собой решение кризисного менеджера об объединении отделов соответственно первого или второго вида. Если $type = 3$, то требуется определить, работают ли в одном отделе сотрудники x и y . Обратите внимание, что x может равняться y в запросе любого типа.

Формат выходных данных

На каждый запрос типа 3 выведите «YES» или «NO» (без кавычек), в зависимости от того, работают ли в одном отделе соответствующие люди.

Примеры

стандартный ввод	стандартный вывод
8 6	NO
3 2 5	YES
1 2 5	YES
3 2 5	
2 4 7	
2 1 2	
3 1 7	

Задача С. Подсчет опыта

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

В очередной онлайн игре игроки, как обычно, сражаются с монстрами и набирают опыт. Для того, чтобы сражаться с монстрами, они объединяются в кланы. После победы над монстром, всем участникам клана, победившего его, добавляется одинаковое число единиц опыта. Особенностью этой игры является то, что кланы никогда не распадаются и из клана нельзя выйти. Единственная доступная операция — объединение двух кланов в один.

Поскольку игроков стало уже много, вам поручили написать систему учета текущего опыта игроков.

Формат входных данных

В первой строке входного файла содержатся числа n ($1 \leq n \leq 200000$) и m ($1 \leq m \leq 200000$) — число зарегистрированных игроков и число запросов.

В следующих m строках содержатся описания запросов. Запросы бывают трех типов:

- `join X Y` — объединить кланы, в которые входят игроки X и Y (если они уже в одном клане, то ничего не меняется).
- `add X V` — добавить V единиц опыта всем участникам клана, в который входит игрок X ($1 \leq V \leq 100$).
- `get X` — вывести текущий опыт игрока X .

Изначально у всех игроков 0 опыта и каждый из них состоит в клане, состоящим из него одного.

Формат выходных данных

Для каждого запроса `get X` выведите текущий опыт игрока X .

Пример

стандартный ввод	стандартный вывод
3 6	150
add 1 100	0
join 1 3	50
add 1 50	
get 1	
get 2	
get 3	

Задача D. Парковка

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

На кольцевой парковке есть n мест пронумерованных от 1 до n . Всего на парковку приезжает n машин в порядке нумерации. У i -й машины известно место p_i , которое она хочет занять. Если машина приезжает на парковку, а её место занято, то она едет далее по кругу и встаёт на первое свободное место.

Формат входных данных

В первой строке входного файла находится число n ($1 \leq n \leq 300\,000$) — размер парковки и число машин. Во второй строке записаны n чисел, i -е из которых p_i ($1 \leq p_i \leq n$) — место, которое хочет занять машина с номером i .

Формат выходных данных

Выведите n чисел: i -е число — номер парковочного места, которое было занято машиной с номером i .

Пример

стандартный ввод	стандартный вывод
3 2 2 2	2 3 1

Задача Е. Телескопы

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

Нина работает в исследовательском центре, сотрудники которого наблюдают за движением звезд.

В центре есть n телескопов, которые можно использовать для наблюдения за звездами. Каждый из телескопов может работать в одном из k режимов. С предыдущей серии наблюдений телескоп с номером i находится в режиме p_i .

Для качественного наблюдения за одной звездой необходимо одновременно использовать 2 различных телескопа, включенных в **одинаковом** режиме. Следующей ночью ученые планируют провести наблюдения за m звездами, причем звезду i исследователи хотят увидеть с помощью телескопов a_i и b_i .

Нине поручили переключить минимальное количество телескопов в нужные режимы заранее так, чтобы можно было качественно пронаблюдать все m звезд, **не меняя** режимы во время следующих наблюдений. Конечно же, Нина справилась со своей задачей и знает, сколько телескопов ей пришлось переключить. А сможете ли найти это число вы?

Формат входных данных

В первой строке дано три целых числа n, m, k — количество телескопов, количество звезд и количество режимов у телескопа ($2 \leq n \leq 10^5$; $0 \leq m \leq 10^5$; $1 \leq k \leq 10^5$).

Во второй строке дано n разделённых пробелами целых чисел p_1, \dots, p_n — режимы, в которых остались телескопы с предыдущего наблюдения ($1 \leq p_i \leq k$).

В каждой из следующих m строк находится по два целых числа a_i и b_i — номера телескопов, которые необходимо использовать для наблюдения за звездой i ($1 \leq a_i, b_i \leq n, a_i \neq b_i$).

Формат выходных данных

Выведите одно целое число — минимальное количество телескопов, которые необходимо перевести в другой режим, чтобы пронаблюдать все m звезд.

Примеры

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

Задача F. Кукушки

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

Британские учёные решили заняться орнитологией и понаблюдать за жизнью необычных кукушек. Для этого они вырастили дерево и построили на нём n гнёзд, в каждом из которых живёт кукушка. Наблюдение за деревом состоит в том, что в некоторые моменты времени учёные оценивают, можно ли подложить определённое яйцо в гнездо к некоторой кукушке или нет.

Каждое яйцо может вынашиваться только в двух определённых гнёздах. Каждое яйцо задаётся неупорядоченной парой различных чисел (x, y) . Яйцо (x, y) может вынашиваться в любом из гнёзд x и y и не может вынашиваться в других гнёздах. Обратите внимание, яйцо (x, y) не отличается от яйца (y, x) .

Теперь опишем процесс подкладывания яйца в имеющиеся гнезда: пусть учёные хотят подложить яйцо (x, y) в гнездо x . Если в гнезде x нет яйца, то яйцо (x, y) просто остаётся в этом гнезде, и процесс на данном шаге завершается. Если же в гнезде x лежит какое-то яйцо (x, p) , то кукушка кладёт яйцо (x, y) в данное гнездо, а яйцо (x, p) пытается подложить в гнездо p аналогичным образом, и процесс продолжается.

Вам предлагается отвечать на вопросы учёных. Всего есть три типа вопросов:

1. (Теоретический) Закончится ли процесс, если подложить яйцо (x, y) в гнездо x ? Так как вопрос чисто теоретический, оно **не добавляется** на самом деле, и состояние гнёзд не меняется.
2. (Практический) Закончится ли процесс, если подложить яйцо (x, y) в гнездо x ? Если процесс закончится, то яйцо **добавляется** в реальности согласно описанному процессу.
3. (Теоретический) Сколько существует **упорядоченных** пар различных чисел (x, y) , таких что яйцо (x, y) можно подложить в гнездо x с учётом имеющихся в гнёздах яиц? При этом для каждого яйца ответ определяется независимо от других добавляемых яиц.

Формат входных данных

В первой строке вводятся три целых числа n, m, q , ($2 \leq n \leq 200\,000$, $0 \leq m \leq n$, $1 \leq q \leq 600\,000$), где n — количество гнёзд на дереве, m — количество яиц, которые учёные уже положили, q — количество вопросов, которые задают учёные.

В каждой из m последующих строк следуют по два числа x_i, y_i , означающих, что в гнезде x_i лежит яйцо (x_i, y_i) . Гарантируется, что все x_i различны и что $x_i \neq y_i$ для всех i .

В следующих q строках описаны вопросы учёных. Вопросы даны в том порядке, в котором на них требуется отвечать. Первое число t_j в строке описывает тип вопроса.

Если $t_j = 1$ или $t_j = 2$, то далее идут два различных числа x_j и y_j , описывающих яйцо, которое фигурирует в соответствующем вопросе.

Если $t_j = 1$, то яйцо не требуется добавлять в текущую расстановку.

Если $t_j = 2$, то яйцо требуется добавить, если процесс добавления потребует конечного числа перекладываний.

Если $t_j = 3$, то требуется определить количество упорядоченных пар (x, y) , таких что яйцо (x, y) можно добавить в гнездо x с тем, чтобы процесс когда-нибудь завершился. В реальности никакие яйца в расстановку не добавляются.

Формат выходных данных

Для каждого вопроса первого и второго типа выведите единственное слово «Yes» или «No» в зависимости от того, закончится ли процесс перекладывания.

Для каждого запроса третьего типа выведите количество искомых упорядоченных пар.

Система оценки

Тесты к этой задаче проходятся в случае прохождения всех тестов. Пусть t_1 — количество запросов третьего уровня, которое запущено для данной группы станут до

Группа	Баллы	Дополнительные ограничения				Необх. группы	Комме
		n	t_1	t_2	t_3		
0	0	–	–	–	–	–	При
1	13	$n \leq 2000$	$t_1 \leq 2000$	$t_2 = 0$	$t_3 = 0$	–	
2	14	$n \leq 2000$	$t_1 \leq 2000$	$t_2 = 0$	$t_3 \leq 1$	1	
3	12	$n \leq 2000$	$t_1 \leq 2000$	$t_2 \leq 2000$	$t_3 \leq 2000$	0 – 2	
4	12	–	$t_1 \leq 2 \cdot 10^5$	$t_2 = 0$	$t_3 = 0$	1	
5	18	–	$t_1 \leq 2 \cdot 10^5$	$t_2 = 0$	$t_3 \leq 1$	1 – 2, 4	
6	31	–	$t_1 \leq 2 \cdot 10^5$	$t_2 \leq 2 \cdot 10^5$	$t_3 \leq 2 \cdot 10^5$	0 – 5	Offline-p

Пример

стандартный ввод	стандартный вывод
5 3 8	Yes
1 2	20
5 1	Yes
2 4	8
1 1 2	No
3	Yes
2 1 2	0
3	No
2 4 2	
2 5 3	
3	
1 4 5	

Замечание

Изначальное расположение яиц в тесте из условия такое: в первом гнезде лежит яйцо (1, 2), во втором — (2, 4), в пятом — (5, 1), а в третьем и четвёртом яиц нет.

Яйцо (1, 2) добавить можно, несмотря на то что подобное яйцо на дереве уже есть, это приведёт к перекладыванию имеющегося яйца (1, 2) в другое гнездо.

Также в начальную конфигурацию можно добавить любое из 10 яиц, существующих для дерева с пятью гнёздами, и каждое яйцо можно положить в любое из двух гнёзд, ему отвечающих, и для любого из добавляемых яиц и гнёзд это потребует конечное количество шагов. Таким образом, ответ на второй запрос — 20.

В результате следующего запроса яйцо (1, 2) будет добавлено реально, и распределение яиц будет таким: в первом гнезде лежит яйцо (1, 2), во втором — также (1, 2), в четвёртом — (2, 4), в пятом (5, 1).

Теперь уже можно добавить только яйца (1, 3), (2, 3), (4, 3) и (5, 3), причём по-прежнему любое яйцо можно положить в каждое из двух упомянутых на нём гнёзд, поэтому ответ на запрос — 8.

Яйцо (4, 2) добавить на дерево нельзя, поэтому состояние гнёзд не изменится.

Для добавления яйца (5, 3) понадобится 5 перекладываний яиц, а после этого никакое новое яйцо за конечное количество шагов добавить уже нельзя.