

## Задача А. Реструктуризация компании (простая версия)

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 одного типа:

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.

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  $type \in \{1, 3\}$ . Если  $type = 1$ , то запрос представляет собой решение кризисного менеджера об объединении отделов. 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	NO
3 2 5	
1 4 7	
1 1 2	
3 1 7	

## Задача В. Остовное дерево

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

Найти в связном графе остовное дерево минимального веса.

### Формат входных данных

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количество вершин и ребер графа соответственно ( $1 \leq n \leq 20000, 0 \leq m \leq 100000$ ). Следующие  $m$  строк содержат описание ребер по одному на строке. Ребро номер  $i$  описывается тремя натуральными числами  $b_i$ ,  $e_i$  и  $w_i$  — номера концов ребра и его вес соответственно ( $1 \leq b_i, e_i \leq n, 0 \leq w_i \leq 100000$ ).

Граф является связным.

### Формат выходных данных

Выведите единственное целое число — вес минимального остовного дерева.

### Пример

стандартный ввод	стандартный вывод
4 4 1 2 1 2 3 2 3 4 5 4 1 4	7

## Задача С. 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 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. 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 F. Прыжки между вселенными

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

Для путешествий между вселенными в *Паучьих мирах* существуют специальные порталы. Сеть из  $m$  порталов для  $n$  вселенных представляет из себя граф из  $n$  вершин и  $m$  ребер. Каждый портал связывает какие-то две вселенные, причем никакие два портала не связывают одну и ту же пару вселенных.

Для перемещения Мигель О'Хара использует свои высокотехнологичные порталные часы. Но самые первые версии этих часов были не настолько развиты, и не позволяли сразу путешествовать в произвольные вселенные. А именно, у часов был параметр *энергетического уровня*, который изначально был равен 0.

Каждый портал характеризуется некоторой характеристикой  $w$  — минимальным значением энергетического уровня, необходимым для использования этого портала. Если значение энергетического уровня часов меньше  $w$ , то воспользоваться этим порталом сейчас нельзя. К счастью, есть способ увеличить энергетический уровень: **при первом попадании** во вселенную  $i$  энергетический уровень часов **навсегда** увеличивается на  $a_i$ .

Мигель задумался, какое максимальное значение энергетического уровня он может получить на старом прототипе часов, если посетит все вселенные, в которые сможет попасть, начав свой путь из вселенной номер  $s$ ? Помогите ему ответить на этот вопрос.

### Формат входных данных

В первой строке входных данных даны три целых числа  $n$ ,  $m$  и  $s$  — количество вселенных, количество порталов между этими вселенными, и номер вселенной, из которой Мигель начинает свои перемещения ( $1 \leq s \leq n \leq 10^5$ ;  $1 \leq m \leq 2 \cdot 10^5$ ).

Во второй строке входных данных через пробел перечислены  $n$  целых чисел  $a_i$  — значения, на которые увеличивается энергетический уровень часов при первом посещении каждой вселенной ( $1 \leq a_i \leq 10^9$ ).

В следующих  $m$  строках дано описание порталов. Описание портала под номером  $i$  содержит три целых числа  $u_i$ ,  $v_i$  и  $w_i$ , и определяет портал между вселенными  $u_i$  и  $v_i$ , для использования которого нужно иметь силу часов не меньше  $w_i$  ( $1 \leq u_i, v_i \leq n$ ;  $0 \leq w_i \leq 10^9$ ).

Граф связей между вселенными не содержит кратных ребер и петель, однако не обязательно связан.

### Формат выходных данных

В единственной строке выведите одно целое число — максимальное значения энергетического уровня часов, которое может получить Мигель.

### Примеры

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

## Задача G. Расстояние X

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

Вам дан неориентированный взвешенный граф из  $N$  вершин и  $M$  ребер. Назовем ценой пути между двумя вершинами вес максимального ребра на этом пути. Найдите количество пар с минимальной ценой пути между ними, равной  $X$ .

### Формат входных данных

Первая строка содержит три целых числа  $N$ ,  $M$  и  $X$  ( $1 \leq N \leq 10^5, 1 \leq M \leq 3 \cdot 10^5, 1 \leq X \leq 10^9$ ). Следующие  $M$  строк содержат по три числа  $a_i$ ,  $b_i$  и  $w_i$ , обозначающие ребро между вершинами  $a_i$  и  $b_i$  веса  $w_i$ . ( $1 \leq w_i \leq 10^9$ )

### Формат выходных данных

Выведите одно число — ответ на задачу

### Примеры

стандартный ввод	стандартный вывод
7 6 3 1 2 1 1 3 2 3 4 3 4 5 1 4 6 2 1 7 4	9
8 8 4 1 3 2 2 4 1 1 5 1 6 7 3 5 8 4 8 4 4 6 5 5 7 8 6	11

## Задача Н. Легендарное списывание

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

В Самой Честной Школе планируется проведение контрольной работы!

В учебном классе этой школы  $n$  рядов одиночных парт по  $m$  штук в каждом. Известно, что некоторые ученики не смогут написать контрольную работу (разумеется, по уважительной причине!!!).

Ученики Самой Честной Школы ожидают, что эта контрольная работа будет очень сложной! Поэтому они *тщательно* готовились к ней, и сейчас ученик, сидящий за партой в  $i$ -м ряду на  $j$ -м месте, обладает бесконечным количеством шпаргалок (достаточно хорошо спрятанных от учителя) с эффективностью  $s_{i,j}$  каждая. Но, кажется, этого может не хватить! Потому ученики придумали гениальный план помощи друг другу: ученик, сидящий за партой в  $i$ -м ряду на  $j$ -м месте, может передать свои шпаргалки соседям на местах  $(i+1, j)$ ,  $(i, j-1)$ ,  $(i-1, j)$ ,  $(i, j+1)$ , если у него таковые имеются. А соседи, в свою очередь, тоже могут передать полученные шпаргалки другим рядом сидящим. Но есть *нюанс*...

По ходу контрольной работы учитель может удалить  $q$  учеников из класса за «академическую нечестность»! За одно удаление преподаватель выводит школьника из класса и забирает все **чужие** шпаргалки у других учеников!

Чтобы получить хорошие оценки, ученики делятся между собой шпаргалками так, чтобы эффективность шпаргалки у каждого ученика была максимально возможной в текущей обстановке в классе. Помогите им узнать минимальную эффективность шпаргалки у ученика в классе сразу же после начала работы и после каждого из  $q$  удалений!

### Формат входных данных

В первой строке входного файла вводятся два числа  $n$  и  $m$  — размеры класса ( $1 \leq n \cdot m \leq 10^5$ ).

В следующих  $n$  строках вводится по  $m$  целых чисел  $s_{i,j}$  — эффективность шпаргалок ученика, сидящего за партой в ряду  $i$  на месте  $j$ . Если число  $s_{i,j}$  равно нулю, то ученика не будет на контрольной! ( $0 \leq s_{i,j} \leq 10^9$ ).

Далее вводится целое число  $q$  — количество учеников, которые будут удалены из класса по ходу работы ( $0 \leq q \leq p-1$ ,  $p$  — количество занятых мест в классе в начале контрольной работы).

В следующих  $q$  строках вводятся пары чисел  $r_k, c_k$  — удаление ученика из класса, сидящего за партой в ряду  $r_k$  на месте  $c_k$  ( $1 \leq r_k \leq n$ ,  $1 \leq r_c \leq m$ ).

### Формат выходных данных

В  $q+1$  строке выходного файла выведите минимальную эффективность шпаргалки, которой может обладать ученик класса в начале КР и после каждого из  $q$  удалений.

### Пример

стандартный ввод	стандартный вывод
3 3	9
3 1 2	5
1 5 1	3
1 1 9	1
3	
3 3	
2 2	
1 1	

### Замечание

Сначала ученик за третьей партой третьего ряда может помочь всем (в т. ч. опосредованно), поэтому минимальная эффективность шпаргалки у любого ученика в классе равна 9.

После удаления ученика за третьей партой третьего ряда, школьник со шпаргалкой эффективностью 5 за второй партой второго ряда все так же может помочь всему классу. После его удаления, ученик за первой партой первого ряда может поделиться с классом своими шпаргалками эффективностью 3.

После удаления школьника за первой партой первого ряда, класс «разбивается» на две части, которые никак не могут помочь друг другу. Поэтому минимальная по эффективности шпаргалка, которой может обладать ученик при оптимальной помощи одноклассников, имеет эффективность 1.

## Задача I. 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 J. Реактивные поезда

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

3019 год. Всё человечество давно живёт в городах, парящих в атмосфере на аэростатах и реактивных двигателях. На планете есть  $n$  городов, между некоторыми парами городов курсируют реактивные поезда. Если между городами  $a$  и  $b$  курсируют реактивные поезда, то пассажиры могут перемещаться между этими городами в любом направлении. Будем говорить, что из города  $a$  достижим город  $b$ , если пассажир может добраться от  $a$  до  $b$  с использованием реактивных поездов, возможно с пересадками.

Люди в 3019 любят дружить, и дружат не парами, а целыми городами. Отношение дружбы взаимно: если город  $a$  дружит с городом  $b$ , то город  $b$  также дружит с городом  $a$ .

Время от времени в каком-нибудь городе решают организовать праздник и зовут всех своих друзей к себе в гости. Если праздник проходит в городе  $a$ , то гости из всех городов, которые дружат с  $a$ , и из которых достижим город  $a$ , прибывают на мероприятие, пользуясь реактивными поездами.

Для того, чтобы оценивать количество гостей на празднике, решено было разработать специальную информационную систему «Праздник 3019». Задан список текущих рейсов реактивных поездов, а также информация о том, какие города дружат. Информационная система должна обрабатывать запросы вида: «Если в городе  $v$  случится праздник, гости из сколько городов попадут на него?» Кроме того, должна поддерживаться возможность добавить информацию о новом рейсе реактивного поезда, а также о новой паре городов, которые начинают дружить. К счастью, рейсы не отменяются, и однажды подружившись, города навсегда остаются друзьями.

Помогите человечеству разработать описанную информационную систему.

### Формат входных данных

В первой строке заданы три целых числа  $n$ ,  $m$  и  $k$  — количество городов, пар исходно дружащих городов и рейсов реактивных поездов ( $1 \leq n \leq 10^5$ ,  $0 \leq m, k \leq 10^5$ ).

В следующих  $m$  строках находятся по два целых числа  $a$ ,  $b$  ( $1 \leq a, b \leq n$ ,  $a \neq b$ ) — пары дружащих городов. Каждая пара дружащих городов указана во вводе ровно один раз.

В следующих  $k$  строках находятся по два целых числа  $a$ ,  $b$  ( $1 \leq a, b \leq n$ ,  $a \neq b$ ) — пары городов, соединённых рейсами реактивных поездов. Каждая пара городов соединена не более, чем одним рейсом.

Следующая строка содержит число  $q$  ( $0 \leq q \leq 10^5$ ) — количество запросов, которые следует обработать, а в следующих  $q$  строках заданы сами запросы.

- Запрос «T  $a$   $b$ » означает, что между городами  $a$  и  $b$  запущен рейс реактивного поезда ( $1 \leq a, b \leq n$ ,  $a \neq b$ ). Гарантируется, что до этого запроса между этими городами рейса не было.
- Запрос «F  $a$   $b$ » означает, что города  $a$  и  $b$  начинают дружить ( $1 \leq a, b \leq n$ ,  $a \neq b$ ). Гарантируется, что до этого запроса эти города не дружили.
- Запрос «?  $v$ » означает, что к информационной системе сделан запрос: «Если в городе  $v$  случится праздник, гости из сколько городов попадут на него?» ( $1 \leq v \leq n$ )

### Формат выходных данных

Для каждого запроса «?  $v$ » выведите ответ на него в отдельной строке.

## Пример

стандартный ввод	стандартный вывод
4 2 2	1
1 2	2
1 3	3
1 2	
1 4	
5	
? 1	
F 4 1	
? 1	
T 4 3	
? 1	

## Замечание

Ответ на первый запрос равен единице, потому что только один из городов 2, 3 соединен с городом 1 рейсом поездов. Ответ на второй запрос равен двойке, потому что перед этим города 1 и 4 подружились, а они были соединены рейсом напрямую. Ответ на третий вопрос равен тройке, потому что от города 3 стало возможным добраться на поездах до города 1 следующим образом: сначала доехать до города 4, а потом от него до города 1.