

## Задача А. 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 В. 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 С. 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 D. 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 Е. Ковбой Джо и заброшенная шахта

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

Однажды ковбой Джо забрёл в жуткую заброшенную шахту, в которой обитают  $n$  духов с номерами  $1, 2, \dots, n$ . Сейчас каждый дух состоит в банде из самого себя. В один момент времени каждый дух может находиться **ровно** в одной банде. По одиночке духи слабы, поэтому вскоре банды начнут объединяться. Когда две банды объединяются, создаётся новая банда, в которую попадают все духи из объединяющихся банд, в то время как старые банды распадаются.

Ковбоя Джо оглушил душераздирающий крик, который сообщил ему о необходимости ответить на  $m$  вопросов. Если ковбой Джо откажется отвечать на вопросы крика или ошибётся, то навечно сгинет в заброшенной шахте.

Дух задаст  $m$  вопросов, каждый из которых относится к одному из трёх типов.

1. Духи  $x$  и  $y$  объединяются в банду. Если они уже находятся в одной банде, ничего не происходит.
2. Крик спрашивает ковбоя Джо, находятся ли духи  $x$  и  $y$  в одной банде.
3. Крик спрашивает ковбоя Джо, в скольких бандах побывал дух  $x$ .

Например, если  $n = 7$ , а банды выглядели так:  $[1, 3], [4, 6, 2], [5], [7]$ , и крик сообщил об объединении банд с духами 1 и 5, банды будут выглядеть так:  $[1, 5, 3], [4, 6, 2], [7]$ . Духи 1, 3, 5 побывали в двух бандах, духи 2, 4, 6, 7 в одной.

### Формат входных данных

В первой строке даны целые числа  $n, m$  ( $1 \leq n, m \leq 2 \cdot 10^5$ ) — количество духов и вопросов крика.

Следующие  $m$  строк содержат описание вопросов:

- 1  $x y$  ( $1 \leq x, y \leq n$ )
- 2  $x y$  ( $1 \leq x, y \leq n$ )
- 3  $x$  ( $1 \leq x \leq n$ )

### Формат выходных данных

Для каждого вопроса второго типа выведите «YES» или «NO».

Для каждого вопроса третьего типа выведите целое число — ответ на вопрос.

### Пример

стандартный ввод	стандартный вывод
7 13	NO
2 3 1	1
3 3	YES
1 2 4	2
2 1 1	NO
3 4	3
2 3 4	3
1 3 4	2
3 4	YES
1 7 3	
1 1 3	
3 7	
3 1	
2 7 4	

## Задача F. 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 G. Шарады

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

В ещё не изведанной части вселенной есть планета, на которой живут одни математики. На этой планете живут  $N$  математиков, каждый — в своём городе. Никакие два города не соединены дорогами, потому что математики могут общаться онлайн, оставляя комментарии о научных трудах друг друга.

Всё шло тихо и спокойно, пока один математик не решил написать научную работу со своего мобильного телефона. Автоисправление в телефоне заменило «очевидно» на «шарада». Не перечитав свою работу, математик так и опубликовал её. Совсем скоро об игре в шарады узнали все математики планеты, и им захотелось собраться и поиграть всем вместе. Поэтому в скором времени началась постройка дорог между городами. Строительство дорог будет идти  $M$  дней в соответствии со следующим расписанием: в первый день строятся дороги между всеми парами городов, у номеров которых наибольший общий делитель равен  $M$ . Во второй день строятся дороги между всеми парами городов, наибольший делитель номеров которых равен  $M - 1$ . И так далее до  $M$ -го дня, в который дороги строятся между всеми парами городов с взаимно простыми номерами. Говоря более формально, в  $i$ -й день (нумеруя дни с единицы) дороги строятся между всеми такими парами городов  $A$  и  $B$ , что  $\text{НОД}(A, B) = M + 1 - i$ .

Математики очень заняты постройкой дорог, поэтому они просят вас помочь определить минимальное число дней с начала строительства, через которое данная пара математиков сможет встретиться, чтобы поиграть в шарады.

### Формат входных данных

В первой строке даны три целых положительных числа  $N$ ,  $M$  и  $Q$  ( $1 \leq N, Q \leq 100000, 1 \leq M \leq N$ ) — количество городов, длительность строительства дорог и количество запросов соответственно.

В следующих  $Q$  строках вводятся по два целых числа  $A$  и  $B$  ( $1 \leq A, B \leq N$ ) — номера городов двух математиков, которым интересно, через сколько дней они смогут встретиться (добраться из одного города в другой, проехав по уже построенным дорогам).

### Формат выходных данных

На каждый из  $Q$  запросов выведите ответы —  $Q$  чисел, каждое в отдельной строке.

### Пример

стандартный ввод	стандартный вывод
8 3 3	3
2 5	1
3 6	2
4 8	

## Задача Н. 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
<pre>7 6 3 1 2 1 1 3 2 3 4 3 4 5 1 4 6 2 1 7 4</pre>	9
<pre>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</pre>	11



## Задача I. Плотное остовное дерево

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

Требуется найти в графе остовное дерево, в котором разница между весом максимального и минимального ребра минимальна.

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $m$  ( $n \leq 1\,000$ ,  $m \leq 10\,000$ ) — количество вершин и ребер в графе.

Каждая из следующих  $m$  строк содержит три целых числа  $b_i$ ,  $e_i$  и  $w_i$  ( $1 \leq b_i, e_i \leq n$ ,  $0 \leq |w_i| \leq 10^9$ ) — номера вершин, которые соединяет ребро, а также вес ребра.

### Формат выходных данных

Если остовное дерево существует, в первой строке выведите YES, а во второй строке одно целое число — минимальную разность между весом максимального и минимального ребра в дереве.

В противном случае в единственной строке выведите NO.

### Пример

стандартный ввод	стандартный вывод
4 5 1 2 1 1 3 2 1 4 1 3 2 2 3 4 2	YES 0

## Задача J. 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 К. Школы

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

С целью подготовки к проведению олимпиады по информатике мэр решил обеспечить надежным электроснабжением все школы города. Для этого необходимо провести линию электропередач от альтернативного источника электроэнергии "Майбутя" к одной из школ города (к какой неважно), а также соединить линиями электропередач некоторые школы между собой.

Считается, что школа имеет надежное электроснабжение, если она напрямую связана с источником "Майбутя" либо с одной из тех школ, которые имеют надежное электроснабжение.

Известна стоимость соединения между некоторыми парами школ. Мэр города решил выбрать одну из двух наиболее экономичных схем электроснабжения (стоимость схемы равняется сумме стоимостей соединений пар школ).

Напишите программу, которая вычисляет стоимость двух наиболее экономных схем альтернативного электроснабжения школ.

### Формат входных данных

В первой строке входного файла находятся два натуральных числа, разделенных пробелом:  $N$  ( $3 \leq N \leq 100$ ), количество школ в городе, и  $M$  — количество возможных соединений между ними.

В каждой из последующих  $M$  строк находятся по три числа:  $A_i, B_i, C_i$ , разделенных пробелами, где  $C_i$  — стоимость прокладки линии электроснабжения ( $1 \leq C_i \leq 300$ ) от школы  $A_i$  до школы  $B_i$ .

### Формат выходных данных

В единственной строке выходного файла должны содержаться два натуральных числа  $S_1$  и  $S_2$ , разделенных пробелом — две наименьшие стоимости схем ( $S_1 \leq S_2$ ).  $S_1 = S_2$  тогда и только тогда, когда существует несколько схем надежного электроснабжения наименьшей стоимости.

Гарантируется, что для входных данных существует две различные схемы надёжного электроснабжения.

### Пример

стандартный ввод	стандартный вывод
5 8 1 3 75 3 4 51 2 4 19 3 2 95 2 5 42 5 4 31 1 2 9 3 5 66	110 121

## Задача L. 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  $c_k$  ( $1 \leq r_k \leq n$ ,  $1 \leq c_k \leq m$ ).

### 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.