

## Задача А. Остовное дерево

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 0.5 секунд  
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. 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 ответ —  $Q$  чисел, каждое в отдельной строке.

### Пример

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

## Задача С. Кукушки

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 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:

(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.

(Практический) Закончится ли процесс, если подложить яйцо  $(x, y)$  в гнездо  $x$ ? Если процесс закончится, то яйцо добавляется в реальности согласно описанному процессу.

(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  $n, m, q$ , ( $2 \leq n \leq 200000, 0 \leq m \leq n, 1 \leq q \leq 600000$ ), 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.

**Пример**

стандартный ввод	стандартный вывод
5 3 8	3
1 2	3
5 1	2
2 4	0
1 1 2	3
3	3
2 1 2	3
3	2
2 4 2	
2 5 3	
3	
1 4 5	

## Задача D. 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 Е. Всем чмоки в этом чатике!

Имя входного файла:	chat.in
Имя выходного файла:	chat.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Сегодня Мэри, как программисту социальной сети «Телеграфчик», предстоит реализовать сложную систему управления чатами.

Задача Мэри усложняется тем, что в социальную сеть «Телеграфчик» внедрена продвинутая система шифрования «ZergRus», простая, как всё гениальное. Суть её в том, что в системе хранится одна переменная  $zerg$ , которая принимает значения от 0 (включительно) до  $p = 10^6 + 3$  (исключая  $p$ ) и меняется в зависимости от событий в системе.

В социальной сети всего  $n$  пользователей ( $1 \leq n \leq 10^5$ ). В начале дня каждый пользователь оказывается в своём собственном чате, в котором больше никого нет. Переменная  $zerg$  в начале дня устанавливается равной 0.

В течение дня происходят события типов:

1. Участник с номером  $(i + zerg) \bmod n$  посылает сообщение всем участникам, сидящим с ним в чате (в том числе и себе самому), при этом переменная  $zerg$  заменяется на  $(30 \cdot zerg + 239) \bmod p$ .
2. Происходит слияние чатов, в которых сидят участники с номерами  $(i + zerg) \bmod n$  и  $(j + zerg) \bmod n$ . Если участники и так сидели в одном чате, то ничего не происходит. Если в разных, то чаты объединяются, а переменной  $zerg$  присваивается значение  $(13 \cdot zerg + 11) \bmod p$ .
3. Участник с номером  $(i + zerg) \bmod n$  хочет узнать, сколько сообщений он не прочитал, и прочитать их. Если участник прочитал  $q$  новых сообщений, то переменной  $zerg$  присваивается значение  $(100\,500 \cdot zerg + q) \bmod p$ .

Вы поможете Мэри реализовать систему, обрабатывающую эти события?

### Формат входных данных

В первой строке входного файла записаны натуральные числа  $n$  ( $1 \leq n \leq 10^5$ ) — число пользователей социальной сети, и  $m$  ( $1 \leq m \leq 3 \cdot 10^5$ ) — число событий, произошедших за день. В следующих  $m$  строках содержится описание событий. Первое целое число в строке означает тип события  $t$  ( $1 \leq t \leq 3$ ). Если  $t = 1$ , далее следует число  $i$  ( $0 \leq i < n$ ), по которому можно вычислить, какой участник послал сообщение. Если  $t = 2$ , далее следуют числа  $i$  и  $j$  ( $0 \leq i, j < n$ ), по которым можно вычислить номера участников, чаты с которыми должны объединиться. Если  $t = 3$ , далее следует число  $i$  ( $0 \leq i < n$ ), по которому можно вычислить номер участника, желающего узнать, сколько у него сообщений, и прочитать их.

### Формат выходных данных

Для каждого события типа 3 нужно вывести число непрочитанных сообщений у участника.

### Пример

chat.in	chat.out
4 10	1
1 0	1
1 2	2
1 1	
1 2	
3 1	
2 1 2	
1 3	
3 3	
2 3 2	
3 2	

## Задача F. День Объединения

Имя входного файла: `unionday.in`  
Имя выходного файла: `unionday.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

64 мегабайта

В Бейтландии есть целых  $n$  городов, но нет ни одной дороги. Король страны, Вальдемар де Беар, решил исправить эту ситуацию и соединить некоторые города дорогами так, чтобы по этим дорогам можно было добраться от любого города до любого другого. Когда строительство будет завершено, король планирует отпраздновать День Объединения. К сожалению, казна Бейтландии почти пуста, поэтому король требует сэкономить деньги, минимизировав суммарную длину всех построенных дорог.

### Формат входных данных

Первая строка входного файла содержит натуральное число  $n$  ( $1 \leq n \leq 5\,000$ ) — количество городов в Бейтландии. Каждая из следующих  $n$  строк содержит по два целых числа  $x_i, y_i$  — координаты  $i$ -го города ( $-10\,000 \leq x_i, y_i \leq 10\,000$ ). Никакие два города не расположены в одной точке.

### Формат выходных данных

Первая строка выходного файла должна содержать минимальную суммарную длину дорог. Выведите число с точностью не менее  $10^{-4}$ .

### Пример

<code>unionday.in</code>	<code>unionday.out</code>
6	9.6568542495
1 1	
7 1	
2 2	
6 2	
1 3	
7 3	

## Задача G. Разрезание графа

Имя входного файла: `cutting.in`  
 Имя выходного файла: `cutting.out`  
 Ограничение по времени: 2 секунды  
 Ограничение по памяти: 256 мегабайт

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

cutting.in	cutting.out
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	