

## Задача А. Наркоконтроль

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

В маленьком городке «М» начала действовать служба наркоконтроля. Первая задача службы — выяснить, сколько наркоторговцев работает в окрестности города. Агенты службы опросили всех наркозависимых в городе и составили список случаев продажи травки, произошедших за одни сутки, с указанием места и времени наблюдения. Теперь аналитики хотят понять, сколько же на самом деле есть наркоторговцев. Из данных разведки известна максимальная скорость, с которой может двигаться наркоторговец. Аналитики просят вас узнать, какое минимальное количество наркоторговцев могли участвовать во всех зафиксированных случаях продажи травки.

### Формат входных данных

На первой строке входного файла содержатся целые числа  $n$  и  $v$  — количество случаев продажи травки и максимальная скорость наркоторговца ( $1 \leq n \leq 100, 1 \leq v \leq 10000$ ). Следующие  $n$  строк содержат описания случаев продажи травки в формате «ЧЧ:ММ  $x$   $y$ », где ЧЧ:ММ — время продажи,  $x$  и  $y$  — координаты места, в котором продавалась травка (для простоты будем считать, что всё происходило на плоскости). Координаты по модулю не превышают 1000. Скорость выражена в км/ч, координаты — в км.

### Формат выходных данных

Выведите в выходной файл одно число — минимальное возможное количество наркоторговцев.

### Пример

стандартный ввод	стандартный вывод
4 1 12:00 0 0 13:10 0 1 14:00 1 0 15:00 1 1	2

## Задача В. Экспедиция

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

Винни-Пух отправился в очередную Экспедицию по Лесу!

Лес представляет собой  $n$  полянок, соединённых  $m$  двусторонними тропинками. При этом возможно, что для некоторых полянок не существует способа добраться из одной в другую по тропинкам. За половину суток (день или ночь) Пух проходит ровно по одной тропинке.

Когда Пух пришёл на рассвете на очередную полянку, ему показалось, что он вернулся к полянке, с которой он начал свое путешествие. Он также уверен, что не проходил ни по одной тропинке дважды, и что все полянки до этого были различными. Также он помнит, что он отправился в Экспедицию на рассвете, то есть число дней в пути было равно числу ночей.

Вы решили помочь медвежонку и хотите определить по карте Леса, могло ли такое случиться.

### Формат входных данных

Первая строка содержит число  $T$  — число наборов входных данных, которые вам предстоит обработать. Далее следуют  $T$  наборов входных данных. Каждый набор задаётся в следующем формате.

В первой строке набора входных данных заданы два числа  $n$  и  $m$  ( $2 \leq n \leq 500\,000$ ,  $1 \leq m \leq 500\,000$ ) — количество полянок и количество тропинок в соответствующем Лесу.

В следующих  $m$  строках содержатся описания тропинок: каждая строка состоит из двух целых чисел  $u_i, v_i$  ( $1 \leq u_i, v_i \leq n$ ,  $u_i \neq v_i$ ), обозначающих номера полянок, соединённых очередной тропинкой.

Гарантируется, что никакие две полянки не соединены двумя или более тропинками.

Гарантируется, что и суммарное число полянок, и суммарное число тропинок по всем наборам входных данных не превзойдёт 500 000.

### Формат выходных данных

Для каждого набора входных данных выведите в отдельной строке «YES», если в Лесу существует маршрут, удовлетворяющий описанию Пуха, и «NO» в противном случае.

### Пример

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

### Замечание

Тест из условия состоит из двух наборов входных данных.

В первом примере Пух мог начать Экспедицию на полянке с номером 2, в первый день перейти на полянку 5, в первую ночь перейти на полянку 3, во второй день перейти на полянку 4 и во вторую ночь вернуться обратно на полянку 2.

Во втором примере, независимо от выбора начальной полянки, Пух может вернуться на неё, но ему в любом случае потребуется три перехода, а значит дней в пути будет больше, чем ночей.

## Задача С. Цветные волшебники

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

Сказочная страна представляет собой множество городов, соединенных дорогами с двухсторонним движением. Причем из любого города страны можно добраться в любой другой город либо непосредственно, либо через другие города. Известно, что в сказочной стране не существует дорог, соединяющих город сам с собой и между любыми двумя разными городами, существует не более одной дороги.

В сказочной стране живут желтый и синий волшебники. Желтый волшебник, пройдя по дороге, перекрашивает ее в желтый цвет, синий — в синий. Как известно, при наложении желтой краски на синюю, либо синей краски на желтую, краски смешиваются и превращаются в краску зеленого цвета, который является самым нелюбимым цветом обоих волшебников.

В этом году в столице страны (городе  $f$ ) проводится конференция волшебников. Поэтому желтый и синий волшебники хотят узнать, какое минимальное количество дорог им придется перекрасить в зеленый цвет, чтобы добраться в столицу. Изначально все дороги не покрашены.

Начальное положение желтого и синего волшебников заранее не известно. Поэтому необходимо решить данную задачу для  $k$  возможных случаев их начальных расположений.

### Формат входных данных

Первая строка входного файла содержит целые числа:  $n$  ( $1 \leq n \leq 10^5$ ) и  $m$  ( $1 \leq m \leq 10^5$ ) — количество городов и дорог в волшебной стране соответственно.

Вторая строка содержит одно целое число  $f$  ( $1 \leq f \leq n$ ) — номер города, являющегося столицей сказочной страны.

В следующих  $m$  строках, находится описание дорог страны. В этих  $m$  строк записано по два целых числа  $a_i$  и  $b_i$ , означающих, что существует дорога, соединяющая города  $a_i$  и  $b_i$ .

Следующая строка содержит целое число  $k$  ( $1 \leq k \leq 10^5$ ) — количество возможных начальных расположений волшебников.

Далее следуют  $k$  строк, каждая из которых содержит два целых числа — номера городов, в которых изначально находится желтый и синий волшебники соответственно.

### Формат выходных данных

Для каждого из  $k$  случаев, ваша программа должна вывести в выходной минимальное количество дорог, которое придется покрасить в зеленый цвет волшебникам для того, чтобы добраться в столицу

### Пример

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

## Задача D. Обновление дата-центров

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

У компании BigData Inc. есть  $n$  дата-центров, пронумерованных от 1 до  $n$ , расположенных по всему миру. В этих дата-центрах хранятся данные клиентов компании (как можно догадаться из названия — большие данные!)

Основой предлагаемых компанией BigData Inc. услуг является гарантия возможности работы с пользовательскими данными даже при условии выхода какого-либо из дата-центров компании из доступности. Подобная гарантия достигается путём использования *двойной репликации* данных. Двойная репликация — это подход, при котором любые данные хранятся в двух идентичных копиях в двух различных дата-центрах.

Про каждого из  $m$  клиентов компании известны номера двух различных дата-центров  $c_{i,1}$  и  $c_{i,2}$ , в которых хранятся его данные.

Для поддержания работоспособности дата-центра и безопасности данных программное обеспечение каждого дата-центра требует регулярного обновления. Релизный цикл в компании BigData Inc. составляет один день, то есть новая версия программного обеспечения выкладывается на каждый компьютер дата-центра каждый день.

Обновление дата-центра, состоящего из множества компьютеров, является сложной и длительной задачей, поэтому для каждого дата-центра выделен временной интервал длиной в час, в течение которого компьютеры дата-центра обновляются и, как следствие, могут быть недоступны. Будем считать, что в сутках  $h$  часов. Таким образом, для каждого дата-центра зафиксировано целое число  $u_j$  ( $0 \leq u_j \leq h-1$ ), обозначающее номер часа в сутках, в течение которого  $j$ -й дата-центр недоступен в связи с плановым обновлением.

Из всего вышесказанного следует, что для любого клиента должны выполняться условия  $u_{c_{i,1}} \neq u_{c_{i,2}}$ , так как иначе во время одновременного обновления обоих дата-центров, компания будет не в состоянии обеспечить клиенту доступ к его данным.

В связи с переводом часов в разных странах и городах мира, время обновления в некоторых дата-центрах может сдвинуться на один час вперёд. Для подготовки к непредвиденным ситуациям руководство компании хочет провести учения, в ходе которых будет выбрано некоторое непустое подмножество дата-центров, и время обновления каждого из них будет сдвинуто на один час позже внутри суток (то есть, если  $u_j = h-1$ , то новым часом обновления будет 0, иначе новым часом обновления станет  $u_j + 1$ ). При этом учения не должны нарушать гарантии доступности, то есть, после смены графика обновления должно по-прежнему выполняться условие, что данные любого клиента доступны хотя бы в одном экземпляре в любой час.

Учения — полезное мероприятие, но трудоёмкое и затратное, поэтому руководство компании обратилось к вам за помощью в определении минимального по размеру непустого подходящего подмножества дата-центров, чтобы провести учения только на этом подмножестве.

### Формат входных данных

В первой строке находятся три целых числа  $n$ ,  $m$  и  $h$  ( $2 \leq n \leq 100\,000$ ,  $1 \leq m \leq 100\,000$ ,  $2 \leq h \leq 100\,000$ ) — число дата-центров компании, число клиентов компании и количество часов в сутках.

Во второй строке вам даны  $n$  чисел  $u_1, u_2, \dots, u_n$  ( $0 \leq u_j < h$ ),  $j$ -е из которых задаёт номер часа, в который происходит плановое обновление программного обеспечения на компьютерах дата-центра  $j$ .

Далее в  $m$  строках находятся пары чисел  $c_{i,1}$  и  $c_{i,2}$  ( $1 \leq c_{i,1}, c_{i,2} \leq n$ ,  $c_{i,1} \neq c_{i,2}$ ), задающие номера дата-центров, на которых находятся данные клиента  $i$ .

Гарантируется, что при заданном расписании обновлений в дата-центрах любому клиенту в любой момент доступна хотя бы одна копия его данных.

## Формат выходных данных

В первой строке выведите минимальное количество дата-центров  $k$  ( $1 \leq k \leq n$ ), которые должны затронуть учения, чтобы не потерять гарантию доступности. Во второй строке выведите  $k$  различных целых чисел — номера кластеров  $x_1, x_2, \dots, x_k$  ( $1 \leq x_i \leq n$ ), на которых в рамках учений обновления станут проводиться на час позже. Номера кластеров можно выводить в любом порядке.

Если возможных ответов несколько, разрешается вывести любой из них. Гарантируется, что хотя бы один ответ, удовлетворяющий условиям задачи, существует.

## Примеры

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

## Замечание

Рассмотрим первый тест из условия. Приведённый ответ является единственным способом провести учения, затронув только один дата-центр. В таком сценарии третий сервер начинает обновляться в первый час дня, и никакие два сервера, хранящие данные одного и того же пользователя, не обновляются в один и тот же час.

С другой стороны, например, сдвинуть только время обновления первого сервера на один час вперёд нельзя — в таком случае данные пользователей 1 и 3 будут недоступны в течение нулевого часа.

## Задача E. Xor the Graph

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

Дан неориентированный граф, состоящий из  $N$  вершин и  $M$  рёбер. Каждому ребру присвоено значение, 0 или 1. Операция состоит в том, чтобы выбрать путь (не обязательно простой), и изменить значения, записанные на каждом его ребре на противоположные. Сделайте значения на всех рёбрах нулями за минимальное число операций.

### Формат входных данных

На первой строке заданы два целых числа  $N$  и  $M$  ( $1 \leq N \leq 10^5$ ,  $0 \leq M \leq 10^5$ ).

В каждой из следующих  $M$  строк записаны по три целых числа  $a$ ,  $b$  и  $c$ . Они означают, что существует ребро между вершинами  $(a, b)$ , изначальное значение которого равно  $c$  ( $1 \leq a, b \leq n$ ,  $0 \leq c \leq 1$ ).

### Формат выходных данных

Выведите количество операций  $K$  на первой строке.

В последующих  $K$  строках выведите пути, которые вы выбрали:

- Первым числом выведите  $P$  — количество вершин в пути
- Затем выведите  $P$  целых чисел  $v_1, v_2, \dots, v_P$  — индексы вершин. Для всех  $1 \leq i \leq P - 1$   $(v_i, v_{i+1})$  должно являться ребром в заданном графе.

Решение будет считаться верным, если оно использует минимальное количество операций, и при этом  $\sum P \leq 4 * M$ .

### Пример

стандартный ввод	стандартный вывод
13 14	3
1 2 1	5 1 2 3 4 6
2 3 1	8 4 2 7 8 9 7 2 5
3 4 1	3 12 11 13
2 4 1	
1 4 0	
4 6 1	
4 10 0	
2 5 1	
2 7 0	
7 8 1	
8 9 1	
9 7 1	
11 12 1	
11 13 1	

### Замечание

Ниже приведён граф в первом тесте.

