Задача A. Точки сочленения

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

256 мегабайт

Дан неориентированный граф. Требуется найти все точки сочленения в нём.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m — количества вершин и рёбер графа соответственно ($1 \leqslant n \leqslant 20\,000$, $1 \leqslant m \leqslant 200\,000$).

Следующие m строк содержат описание рёбер по одному на строке. Ребро номер i описывается двумя натуральными числами b_i, e_i — номерами концов ребра ($1 \leqslant b_i, e_i \leqslant n$).

Формат выходных данных

Первая строка выходного файла должна содержать одно натуральное число b — количество точек сочленения в заданном графе. На следующей строке выведите b целых чисел — номера вершин, которые являются точками сочленения, в возрастающем порядке.

Пример

стандартный ввод	стандартный вывод
6 7 1 2 2 3 2 4 2 5 4 5 1 3 3 6	2 2 3

Задача B. Компоненты реберной двусвязности

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

Компонентой реберной двусвязности графа $\langle V, E \rangle$ называется подмножество вершин $S \subset V$, такое что для любых различных u и v из этого множества существует не менее двух реберно не пересекающихся путей из u в v.

Дан неориентированный граф. Требуется выделить компоненты реберной двусвязности в нем.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m — количества вершин и ребер графа соответственно ($1 \leqslant n \leqslant 20\,000$, $1 \leqslant m \leqslant 200\,000$).

Следующие m строк содержат описание ребер по одному на строке. Ребро номер i описывается двумя натуральными числами b_i, e_i — номерами концов ребра ($1 \leqslant b_i, e_i \leqslant n$).

Формат выходных данных

В первой строке выходного файла выведите целое число k — количество компонент реберной двусвязности графа.

Во второй строке выведите n натуральных чисел $a_1, a_2, \ldots, a_n$, не превосходящих k, где a_i — номер компоненты реберной двусвязности, которой принадлежит i-я вершина.

Компоненты требуется нумеровать в порядке возрастания минимального номера вершины, входящей в компоненту.

Пример

стандартный ввод	стандартный вывод
6 7 1 2 2 3 3 1 1 4 4 5 4 6 5 6	2 1 1 1 2 2 2

Задача C. Конденсация графа

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

Вам задан ориентированный граф с N вершинами и M ребрами ($1 \leqslant N \leqslant 200\,000$, $1 \leqslant M \leqslant 200\,000$). Найдите компоненты сильной связности заданного графа и топологически отсортируйте его конденсацию.

Формат входных данных

Граф задан во входном файле следующим образом: первая строка содержит числа N и M. Каждая из следующих M строк содержит описание ребра – два целых числа из диапазона от 1 до N – номера начала и конца ребра.

Формат выходных данных

На первой строке выведите число K – количество компонент сильной связности в заданном графе. На следующей строке выведите N чисел – для каждой вершины выведите номер компоненты сильной связности, которой принадлежит эта вершина. Компоненты сильной связности должны быть занумерованы таким образом, чтобы для любого ребра номер компоненты сильной связности его начала не превышал номера компоненты сильной связности его конца.

Пример

стандартный ввод	стандартный вывод
10 19 1 4 7 8 5 10 8 9 9 6 2 6 6 2 3 8 9 2 7 2 9 7 4 5 3 6 7 3 6 7 10 8 10 1 2 9 2 7	2 1 2 2 1 1 2 2 2 2 1

Задача D. Эйлеров путь

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

Дан неориентированный связный граф, не более трех вершин имеет нечетную степень. Требуется определить, существует ли в нем путь, проходящий по всем ребрам.

Если такой путь существует, необходимо его вывести.

Формат входных данных

Первая строка входного файла содержит натуральное число n — количество вершин графа ($1 \leqslant n \leqslant 100\,000$). Далее следуют n строк, задающих ребра. В i-й из этих строк находится число m_i — количество ребер, инцидентных вершине i. Далее следуют m_i натуральных чисел — номера вершин, в которые ведут ребро из i-й вершины.

Граф может содержать кратные ребра, но не содержит петель.

Граф содержит не более $300\,000$ ребер.

Формат выходных данных

Если решение существует, то в первую строку выходного файла выведите одно число k — количество ребер в искомом маршруте, а во вторую $k+1$ число — номера вершин в порядке их посещения.

Если решений нет, выведите в выходной файл одно число `-1`.

Если решений несколько, выведите любое.

Пример

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

Задача E. Компоненты вершинной двусвязности

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

Компонентой вершинной двусвязности графа $\langle V, E\rangle$ называется максимальный по включению подграф (состоящий из вершин и ребер), такой что любые два ребра из него лежат на вершинно простом цикле.

Дан неориентированный граф без петель. Требуется выделить компоненты вершинной двусвязности в нем.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m — количества вершин и ребер графа соответственно ($1 \leqslant n \leqslant 20\,000$, $1 \leqslant m \leqslant 200\,000$).

Следующие m строк содержат описание ребер по одному на строке. Ребро номер i описывается двумя натуральными числами b_i, e_i — номерами концов ребра ($1 \leqslant b_i, e_i \leqslant n$).

Формат выходных данных

В первой строке выходного файла выведите целое число k — количество компонент вершинной двусвязности графа.

Во второй строке выведите m натуральных чисел $a_1, a_2, \ldots, a_m$, не превосходящих k, где a_i — номер компоненты вершинной двусвязности, которой принадлежит i-е ребро. Ребра нумеруются с единицы в том порядке, в котором они заданы во входном файле.

Компоненты требуется нумеровать в порядке возрастания минимального номера ребра, входящего в компоненту.

Пример

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

Задача F. Алексей и Иван пишут код

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

Алексей и Иван — программисты из компании Yango. Алексей специализируется на Golang и работает с очередью задач `YGO` , а Иван — эксперт по C++ и работает с очередью `YCPP`. В компании есть 26 проектов, обозначенных буквами от «`a`» до «`z`».

Каждый день в очереди YGO появляется одна приоритетная задача из какого-то проекта. В очереди YCPP тоже появляется одна приоритетная задача (возможно, из другого проекта).

Алексей планирует весь ближайший спринт, состоящий из N дней, брать и выполнять приоритетные задачи из своей очереди `YGO`. Иван планирует делать то же самое со своей очередью `YCPP`.

Каждый мог бы работать только со своей привычной очередью, но у Алексея и Ивана есть общий KPI — они хотят, чтобы после N дней работы у них была абсолютно одинаковая статистика по проектам (то есть по каждому проекту от «`a`» до «`z`» Иван должен выполнить столько же задач, сколько и Алексей). Хоть программисты предпочитают работать на своем языке, ради достижения общего KPI в некоторые дни они готовы поменяться очередями задач.

Помогите программистам определить, можно ли им в какие-то дни поменяться очередями (то есть Алексей возьмет задачу из `YCPP`, а Иван из `YGO`), чтобы после N дней их статистика по всем проектам была одинаковой. И если можно, то помогите найти хотя бы один подходящий набор дней для обмена очередями.

Формат входных данных

Первая строка входного файла содержит число $1 \leqslant N \leqslant 2 \cdot 10^5$ — длина спринта, в рамках которого берут задачи Иван и Алексей.

Следующая строка содержит строку длины N, состоящую из маленьких латинских букв. 26 проектов закодированы маленькими латинскими буквами. i-й символ строки содержит код проекта, задача из которого в i-й день будет приоритетной в очереди `YGO`.

Следующая строка содержит строку длины N. Эта строка содержит аналогичную информацию для очереди `YCPP`.

Формат выходных данных

Выведите -1, если не существует набора дней, в который программисты могут поменяться своими очередями и при этом получить полностью одинаковую статистику по проектам.

Если такой набор существует, в первой строке выведите единственное число K — количество дней, в которые программистам следует поменяться очередями ($0 \leqslant K \leqslant N$). В следующей строке через пробел выведите K различных чисел от 1 до N — номера этих дней (дни нумеруются с единицы).

Примеры

стандартный ввод	стандартный вывод
4 aabc bcdd	2 2 4
3 abc abd	-1
7 players parsley	0

Задача H. Диаграммы Юнга выходят в интернет

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Рома и Сеня дали контест из 10 задач про диаграмму Юнга и теперь вынуждены скрываться от ДемидКомНадзора. Но они слишком любят диаграммы Юнга и хотят делиться новыми открытиями. Однако держаться вместе очень рискованно, так как в случае чего повяжут их обоих, поэтому они вынуждены общаться через Интернет. Но "обычный Интернет" полностью контролируется ДемидКомНадзором, поэтому они пользуются даркнетом.

В даркнете любое сообщение может проделать длинный запутанный путь до получателя через множество серверов. Более того, оно даже может проходить через один и тот же сервер несколько раз. За счет этого сообщение сложнее отследить.

Компьютер Ромы связан с сервером 1, а компьютер Сени — с сервером n.

ДемидКомНадзор хочет перехватить Ромино сообщение с диаграммой Юнга. Для этого ему необходимо взломать такой сервер, что сообщение, посланное Ромой, по любому пути к Сене пройдет через этот сервер **ровно один** раз.

Найдите все подходящие сервера.

Формат входных данных

В первой строке файла дано количество тестовых примеров t ($1 \leqslant t \leqslant 500$).

Каждый тестовый пример выглядит так: в первой строке даны два числа: n и m ($2 \leqslant n \leqslant 5 \cdot 10^5$, $0 \leqslant m \leqslant 10^6$), число серверов и число прямых соединений между серверами.

В каждой из последующих m строк содержится упорядоченная пара чисел a и b ($1 \leqslant a, b \leqslant n$), это означает, что с сервера a можно переслать сообщение напрямую на сервер b.

Гарантируется, что эти упорядоченные пары не повторяются внутри одного тестового примера.

Так же гарантируется, что и сумма по n, и сумма по m по всем тестовым примерам не превосходит 10^6.

Формат выходных данных

Для каждого тестового примера требуется вывести две строки: в первой число подходящих серверов, а во второй — номера этих серверов в порядке их следования на пути от a до b через пробел.

Пример

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

Задача I. Мосты

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

Владения короля Джулиана расположены на n островах, пронумерованных от 1 до n. Некоторые пары островов соединены друг с другом мостами, по которым можно перемещаться в две стороны. Всего между островами есть m мостов. От любого острова можно добраться до любого другого, перемещаясь по мостам.

Будем называть мост мост *критическим*, если в случае обрушения этого моста будут существовать такие две острова, что от одного из них нельзя добраться до другого, перемещаясь по оставшимся мостам.

Король Джулиан очень беспокоится о безопасности и доступности сообщения в своих владениях. Он хочет построить дополнительные мосты между некоторыми парами островов так, чтобы между островами не осталось критических мостов. Так как король в то же время еще и экономный, он хочет выяснить, какое минимальное количество дополнительных мостов можно построить, чтобы выполнить данное требование.

Формат входных данных

В первой строке даны два целых числа n и m — количество островов и количество мостов между ними ($2 \leqslant n \leqslant 100\,000$, $1 \leqslant m \leqslant 200\,000$).

В следующих m строках дано по два целых числа a_i и b_i — номера островов, соединенных i-м мостом ($1 \leqslant a_i, b_i \leqslant n$, $a_i \neq b_i$).

Гарантируется, что от любого острова можно добраться до любого другого, перемещаясь по мостам.

Формат выходных данных

Выведите одно целое число — минимальное количество дополнительных мостов, которое нужно построить, чтобы между островами не было критических мостов.

Примеры

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

Задача K. Тяжелый груз

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

Операторам склада необходимо переместить тяжелую коробку с использованием специального погрузчика. Склад можно схематически представить как n комнат, соединенных m коридорами. От любой комнаты можно добраться до любой другой, перемещаясь по коридорам. Комнаты пронумерованы от 1 до n. Коридор номер i непосредственно соединяет комнаты с номерами u_i и v_i, по коридору можно перемещаться в обоих направлениях.

Погрузчик может поднимать и опускать коробку, а также, если он не держит коробку, перемещаться по свободным комнатам и коридорам. Изначально погрузчик находится в комнате номер 1, и держит поднятую коробку. Погрузчику доступны следующие действия:

1. Если погрузчик находится в комнате a и держит поднятую коробку, он может, не сдвигаясь с места, поставить коробку в комнату b, если комнаты a и b непосредственно соединены коридором. После этого действия погрузчик не держит коробку и может перемещаться.
2. Если погрузчик находится в комнате a, а коробка стоит в комнате b, и комнаты a и b непосредственно соединены коридором, погрузчик может, не перемещаясь, поднять коробку. После этого действия погрузчик остается в комнате a и держит поднятую коробку, он не может перемещаться, пока не поставит коробку.
3. Если погрузчик не держит коробку, он может перемещаться по коридорам и комнатам, однако он не может проходить через комнату, где лежит коробка.

Пустой погрузчик перемещается между комнатами очень быстро, гораздо быстрее, чем он поднимает или опускает коробку. Поэтому будем считать, что на выполнение первого или второго действия погрузчик тратит одну единицу времени, а третье действие выполняется мгновенно. Ваша задача — для каждой комнаты p ($2 \leqslant p \leqslant n$) определить, за какое минимальное время погрузчик может из изначального положения — в первой комнате с поднятой коробкой, оказаться в комнате p с поднятой коробкой. Либо определить, что это сделать невозможно.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке дано одно целое число t ($1 \leqslant t \leqslant 100\,000$) — количество наборов входных данных. Далее следуют описания наборов входных данных.

В первой строке каждого набора входных данных даны два целых числа n и m ($2 \leqslant n \leqslant 500\,000$, $1 \leqslant m \leqslant 500\,000$) — количество комнат и коридоров на складе.

В следующих m строках даны по два целых числа u_i и v_i ($1 \leqslant u_i, v_i \leqslant n$, $u_i \neq v_i$) — номера комнат, соединенных i-м коридором. Гарантируется, что каждая пара комнат, соединенных коридором, упомянута ровно один раз. Гарантируется, что если все комнаты свободны, от любой комнаты можно добраться до любой другой, перемещаясь по коридорам.

Обозначим за $\sum n$ сумму n, а за $\sum m$ сумму m по всем наборам входных данных в одном тесте. Гарантируется, что $\sum n \leqslant 500\,000$, $\sum m \leqslant 500\,000$.

Формат выходных данных

Для каждого набора входных данных выведите $n-1$ чисел: i-е из них должно быть равно минимальному количеству подъемов и опусканий коробки, которые нужно сделать погрузчику, чтобы оказаться в комнате $i+1$ с поднятой коробкой. Если это сделать невозможно, то i-е число должно быть равно -1.

Пример

стандартный ввод	стандартный вывод
4 4 4 1 2 2 3 3 4 4 1 5 5 1 2 2 3 3 4 4 5 5 1 5 6 1 2 3 2 1 3 3 5 5 4 3 4 9 12 1 2 2 3 3 1 4 5 5 6 6 4 7 8 8 9 9 7 3 6 6 9 9 3	-1 2 -1 4 2 2 4 2 2 4 4 2 2 6 6 4 6 6 4

Замечание

В четвертом наборе входных данных погрузчик может выполнить следующие действия, чтобы из комнаты 1 с поднятой коробкой быстрее всего оказаться в комнате 4 с поднятой коробкой:

- Поставить коробку в комнату 2. Тратится одна единица времени.
- Переместиться в комнату 3. Время не тратится.
- Поднять коробку из комнаты 2. Тратится одна единица времени.
- Поставить коробку в комнату 9. Тратится одна единица времени.
- Переместиться в комнату 6. Время не тратится.
- Поднять коробку из комнаты 9. Тратится одна единица времени.
- Поставить коробку в комнату 5. Тратится одна единица времени.
- Переместиться в комнату 4. Время не тратится.
- Поднять коробку из комнаты 5. Тратится одна единица времени.

Всего будет потрачено 6 единиц времени.

Задача M. Блогеры-путешественники

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	3 секунды
Ограничение по памяти:	1024 мегабайта

Ян и Татьяна решили стать блогерами-путешественниками и публиковать ролики о поездках по городам своей страны.

В стране есть n городов, пронумерованных от 1 до n. Город 1 — столица их страны. Города соединены m двусторонними дорогами, пронумерованными от 1 до m, каждая из которых соединяет два различных города. При этом одну и ту же пару городов могут соединять несколько различных дорог. Из любого города по дорогам можно доехать до любого другого города страны.

Путешественники планируют отправиться из столицы в какой-то другой город, но пока не выбрали в какой. Маршрут путешествия в город k будет состоять из городов $s_1, s_2, \ldots, s_q$ и дорог $r_1, r_2, \ldots, r_{q-1}$, таких что:

- $s_1 = 1$, $s_q = k$;
- дорога r_i соединяет города s_i и s_{i+1};
- ребята не проезжают по одной и той же дороге дважды, поэтому все r_i различны. Допускается проезжать несколько раз через один и тот же город, в том числе через город 1, где путешествие начинается, и город k, в котором путешествие заканчивается.

Для каждой дороги Ян и Татьяна посчитали длительность ролика, который получится при съемке путешествия по этой дороге, длительность ролика для дороги с номером i равна t_i.

В процессе путешествия каждый из ребят выберет одну из дорог маршрута и снимет ролик, посвящённый этой дороге. При этом Ян любит снимать короткие ролики, поэтому выберет на маршруте дорогу с наименьшим значением t_i, а Татьяна предпочитает длинные ролики, поэтому выберет дорогу с наибольшим значением t_i.

Суммарная длина двух роликов будет равна $\min\limits_{1 \leqslant i \leqslant q-1} t_{r_i} + \max\limits_{1 \leqslant i \leqslant q-1} t_{r_i}$.

Ребята планируют выложить ролики на известную платформу, где большей популярностью пользуются короткие ролики, поэтому они хотят минимизировать суммарную длину двух роликов. Чтобы выбрать конечный город и маршрут для путешествия, блогеры хотят для каждого конечного города k подсчитать минимальную по всем возможным маршрутам из города 1 в город k суммарную длину двух роликов.

Формат входных данных

В первой строке даны два целых числа n, m ($2 \leqslant n \leqslant 300\,000$, $1 \leqslant m \leqslant 300\,000$) — количество городов и дорог.

Следующие m строк содержат описания дорог. В i-й из этих строк находятся три целых числа u_i, v_i, t_i ($1 \leqslant u_i, v_i \leqslant n$, $u_i \neq v_i$, $0 \leqslant t_i \leqslant 10^9$) — номера городов, соединённых дорогой, и длительность ролика про эту дорогу.

Гарантируется, что по имеющимся дорогам можно проехать из любого города в любой другой, возможно, через другие города.

Формат выходных данных

Для каждого $2 \leqslant k \leqslant n$ выведите минимальную суммарную длину роликов Яна и Татьяны для путешествия, заканчивающегося в городе k.

Примеры

стандартный ввод	стандартный вывод
3 3 1 2 2 1 3 1 2 3 1	2 2
7 10 1 2 2 1 2 8 2 3 3 3 4 5 3 5 4 4 5 4 6 5 7 6 4 4 1 7 6 6 7 9	4 5 6 6 6 10
4 4 1 2 2 3 2 0 2 4 3 4 3 1	3 2 2

Замечание

В первом примере возможные оптимальные маршруты:

- $1 \overset{t=1}{\rightarrow} 3 \overset{t=1}{\rightarrow} 2$. Длина роликов в маршруте $1 + 1 = 2$.
- $1 \overset{t=1}{\rightarrow} 3$. Длина роликов в маршруте $1 + 1 = 2$.

Во втором примере возможные оптимальные маршруты:

- $1 \overset{t=2}{\rightarrow} 2$. Длина роликов в маршруте $2 + 2 = 4$.
- $1 \overset{t=2}{\rightarrow} 2 \overset{t=3}{\rightarrow} 3$. Длина роликов в маршруте $2 + 3 = 5$.
- $1 \overset{t=2}{\rightarrow} 2 \overset{t=3}{\rightarrow} 3 \overset{t=4}{\rightarrow} 5 \overset{t=4}{\rightarrow} 4$. Длина роликов в маршруте $2 + 4 = 6$.
- $1 \overset{t=2}{\rightarrow} 2 \overset{t=3}{\rightarrow} 3 \overset{t=4}{\rightarrow} 5$. Длина роликов в маршруте $2 + 4 = 6$.
- $1 \overset{t=2}{\rightarrow} 2 \overset{t=3}{\rightarrow} 3 \overset{t=4}{\rightarrow} 5 \overset{t=4}{\rightarrow} 4 \overset{t=4}{\rightarrow} 6$. Длина роликов в маршруте $2 + 4 = 6$.
- $1 \overset{t=2}{\rightarrow} 2 \overset{t=8}{\rightarrow} 1 \overset{t=6}{\rightarrow} 7$. Длина роликов в маршруте $2 + 8 = 10$.

В третьем примере возможные оптимальные маршруты:

- $1 \overset{t=2}{\rightarrow} 2 \overset{t=0}{\rightarrow} 3 \overset{t=1}{\rightarrow} 4 \overset{t=3}{\rightarrow} 2$. Длина роликов в маршруте $0 + 3 = 3$.
- $1 \overset{t=2}{\rightarrow} 2 \overset{t=0}{\rightarrow} 3$. Длина роликов в маршруте $0 + 2 = 2$.
- $1 \overset{t=2}{\rightarrow} 2 \overset{t=0}{\rightarrow} 3 \overset{t=1}{\rightarrow} 4$. Длина роликов в маршруте $0 + 2 = 2$.

Система оценки

Подз.	Баллы	Ограничения			Необх. подзадачи	Информация о проверке
		n	m	дополнительно		
1	9	$n \leqslant 300\,000$	$m \leqslant 300\,000$	$m = n - 1$		первая ошибка
2	17	$n \leqslant 300\,000$	$m \leqslant 300\,000$	$t_i = 0$ для всех дорог i из города 1		первая ошибка
3	12	$n \leqslant 300\,000$	$m \leqslant 300\,000$	$t_i = 10^9$ для всех дорог i из города 1		первая ошибка
4	9	$n \leqslant 10$	$m \leqslant 10$	каждая пара городов соединена не более чем одной дорогой		первая ошибка
5	6	$n \leqslant 20$	$m \leqslant 20$	каждая пара городов соединена не более чем одной дорогой	4	первая ошибка
6	6	$n \leqslant 2000$	$m \leqslant 2000$	$\lvert u_i - v_i \rvert = 1$ для всех дорог		первая ошибка
7	9	$n \leqslant 2000$	$m \leqslant 2000$		У, 4–6	первая ошибка
8	8	$n \leqslant 5000$	$m \leqslant 300\,000$		У, 4–7	только баллы
9	10	$n \leqslant 300\,000$	$m \leqslant 300\,000$	для всех a существует дорога между парой городов a и $a+1$; для любой пары дорог i и j, для которых $\lvert u_i - v_i \rvert = 1$ и $\lvert u_j - v_j \rvert > 1$ выполнено $t_i \leqslant t_j$	6	только баллы
10	14	$n \leqslant 300\,000$	$m \leqslant 300\,000$		У, 1–9	только баллы

Задача N. Игра в слова

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

Юрий и Олег играли в «слова». Игра проходит по следующим правилам:

- Игроки по очереди называют слова, первый ход делает Олег;
- Первое слово может быть любым, но любое последующее слово должно начинаться на ту же букву, что начиналось предыдущее;
- Слова не должны повторяться.

Юрий и Олег долго играли в эту игру, в итоге назвали по n слов каждый. Известно, что Олег назвал слова $s_1, \ldots, s_n$ в произвольном порядке, а Юрий — $t_1, \ldots, t_n$, также необязательно именно в таком порядке. Также известно, что игроки не нарушали правила игры.

По известным данным восстановите, как могла проходить игра. А точнее, от вас требуется найти последовательность слов $u_1, \ldots, u_{2n}$, которые произносили игроки.

Формат входных данных

В первой строке указано число n ($1 \leqslant n \leqslant 40\,000$).

В последующих n строках указаны слова, сказанные Олегом.

Затем идут n слов, которые сказал Юрий.

Известно, что игроки называли слова, состоящие из строчных латинских букв и каждое слово имело длину не более 6 символов. Также гарантируется, что такие множества слов действительно могли быть названы в ходе игры в слова. Обратите внимание, что слова необязательно осмыслены и представляют собой произвольные строки.

Формат выходных данных

Выведите последовательность слов $u_1, \ldots, u_{2n}$ — последовательность слов, которая подходит под условие задачи.

Пример

стандартный ввод	стандартный вывод
2 tinkov rich var hack	tinkov var rich hack

Задача P. Антенна

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

Для связи с Землёй членам экспедиции на Марс необходимо собрать антенну. Антенна в разобранном состоянии представляет собой n фрагментов, i-й фрагмент представляет собой штангу длиной s_i сантиметров, на которой закреплены m_i перекладин. Каждый фрагмент содержит хотя бы одну перекладину.

У каждой штанги есть начало, в котором расположен штекер, и конец, в котором расположено гнездо. Любые две штанги можно последовательно соединить, присоединив начало одной к концу другой. Для каждой перекладины известно расстояние от начала её штанги в сантиметрах. Для i-го фрагмента это расстояние может быть от 0 до s_i, значение 0 означает, что перекладина находится непосредственно в начале штанги, значение s_i — что она находится непосредственно в конце штанги. Толщиной перекладин и размерами штекера и гнезда следует пренебречь.

Чтобы корректно собрать антенну, необходимо соединить в некотором порядке все n фрагментов, при этом расстояние между любыми двумя соседними перекладинами должно быть одинаковым.

К сожалению, члены экспедиции забыли инструкцию по сборке антенны на Земле, а передать её на Марс не представляется возможным — ведь антенна ещё не собрана. Помогите исследователям!

Требуется определить, в каком порядке необходимо соединить фрагменты антенны, чтобы установить связь с Землей.

Формат входных данных

В первой строке дано одно число n — количество фрагментов ($1 \leqslant n \leqslant 100\,000$).

Далее дано описание n фрагментов. В первой строке описания фрагмента даны два целых числа m_i и s_i — количество перекладин и длина штанги в i-м фрагменте ($1 \leqslant m_i \leqslant 100\,000$, $0 \leqslant s_i \leqslant 10^9$). В следующей строке даны m_i целых чисел $p_{i,j}$ — позиции перекладин, $p_{i,j}$ равно расстоянию в сантиметрах от начала штанги до j-й перекладины на ней ($0 \leqslant p_{i,1} < p_{i,2} < \cdots < p_{i,m_i} \leqslant s_i$).

Сумма всех m_i не превышает $100\,000$.

Формат выходных данных

Если собрать антенну указанным образом возможно, в первой строке выведите «`Yes`», а во второй строке выведите перестановку чисел от 1 до n — номера фрагментов в порядке, в котором их следует соединить, начало каждого следующего фрагмента в этом порядке присоединяется к концу предыдущего фрагмента. Если существует несколько подходящих ответов, можно вывести любой из них.

Если собрать антенну невозможно, в единственной строке выведите «`No`».

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	**Баллы**	**Доп. ограничения**	**Необходимые подзадачи**	**Информация о проверке**
1	8	$n \leqslant 8$, $m_i = 1$, $s_i \leqslant 100$		первая ошибка
2	8	$n \leqslant 8$, $s_i \leqslant 100$	1	первая ошибка
3	21	$n \leqslant 1\,000$	1, 2	первая ошибка
4	21	$\sum m_i > n$		первая ошибка
5	21	$s_i \leqslant 100$	1, 2	первая ошибка
6	21	нет	1–5	первая ошибка

Примеры

стандартный ввод	стандартный вывод
3 1 7 3 1 8 6 2 8 1 6	Yes 2 1 3
1 1 7 5	Yes 1
1 3 10 2 5 9	No
3 1 5 3 1 3 3 1 6 3	No
4 1 5 0 1 0 0 1 3 3 1 0 0	Yes 3 2 4 1

Задача Q. Два проспекта

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	8 секунд
Ограничение по памяти:	512 мегабайт

Для того чтобы сделать столицу Берляндии более привлекательным туристическим местом, великий король придумал следующий план: выбрать две улицы города и назвать их проспектами. Естественно, эти проспекты будут объявлены чрезвычайно важными историческими местами, что должно привлечь туристов со всего мира.

Столицу Берляндии можно представить в виде графа, вершинами которого являются перекрестки, а ребрами являются улицы, соединяющие два перекрестка напрямую. Всего в графе n вершин и m ребер, по любой улице можно двигаться в обоих направлениях, от любого перекрестка можно добраться до любого другого, перемещаясь только по улицам, каждая улица соединяет два различных перекрестка, и никакие две улицы не соединяют одинаковую пару перекрестков.

Чтобы снизить поток обычных горожан, перемещающихся по великим проспектам, было решено ввести платный проезд по каждому их них в обе стороны. Теперь за один проезд по проспекту нужно заплатить 1 тугрик. За проезд по остальным улицам платить не нужно.

Аналитики собрали выборку из k горожан, i-му из них надо ездить на работу от перекрестка a_i к перекрестку b_i. После выбора двух проспектов каждый горожанин будет добираться на работу вдоль пути, стоимость которого будет минимальна.

Для того чтобы заработать как можно больше денег, было решено выбрать в качестве двух проспектов две такие улицы, что суммарное количество тугриков, которые заплатят эти k горожан, будет максимально возможным. Помогите королю: по заданной схеме города и выборке горожан найдите, какие две улицы нужно сделать проспектами, и сколько тугриков заплатят горожане при таком выборе.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число t ($1 \leqslant t \leqslant 10^5$) — количество наборов входных данных.

В первой строке описания каждого набора входных данных находятся два целых числа n и m ($3 \leqslant n \leqslant 500\,000$, $n-1 \leqslant m \leqslant 500\,000$, $m \leqslant \frac{n(n-1)}{2}$) — количество перекрестков и улиц города.

В следующих m строках содержатся описания улиц, в i-й строке находятся два целых числа s_i и f_i ($1 \leqslant s_i, f_i \leqslant n$, $s_i \neq f_i$) — номера перекрестков, которые соединяет i-я улица. Гарантируется, что никакие две улицы не соединяют одну и ту же пару перекрестков, и что от любого перекрестка можно добраться до любого другого, перемещаясь только по улицам.

В следующей строке находится единственное целое число k ($1 \leqslant k \leqslant 500\,000$) — количество горожан в выборке.

В следующих k строках содержатся описания горожан, в i-й строке находятся два целых числа a_i и b_i ($1 \leqslant a_i, b_i \leqslant n$, $a_i \neq b_i$) — i-й горожанин едет на работу от перекрестка a_i до перекрестка b_i.

Пусть M обозначает сумму значений m по всем наборам входных данных, а K означает сумму значений k по всем наборам входных данных. Гарантируется, что $M, K \leqslant 500\,000$.

Формат выходных данных

Для каждого набора входных данных в выведите ответ на задачу.

В первой строке ответа данных выведите суммарное количество тугриков, которое заплатят горожане.

Во второй строке ответа выведите два целых числа x_1 и y_1 — номера перекрёстков, дорогу между которыми нужно сделать первым проспектом.

В третей строке ответа выведите два целых числа x_2 и y_2 — номера перекрёстков, дорогу между которыми нужно сделать вторым проспектом.

Номера перекрестков, соединенных улицей, можно выводить в произвольном порядке, каждая из двух выведенных улиц должна встречаться среди m улиц города, выбранные улицы должны быть различными.

Пример

стандартный ввод	стандартный вывод
3 6 5 1 2 2 3 2 4 4 5 4 6 3 1 6 5 3 2 5 5 5 1 2 2 3 3 4 4 5 5 1 6 1 5 1 3 1 3 2 4 2 5 5 3 8 10 1 2 2 3 3 4 4 5 5 6 6 7 7 8 7 1 1 8 3 6 4 2 5 3 7 2 5 7 8	5 4 2 5 4 5 1 5 3 2 3 7 6 2 3