Мосты

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

Дан неориентированный граф. Требуется найти все мосты в нем.

Формат входного файла

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

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

Формат выходного файла

Первая строка выходного файла должна содержать одно натуральное число b — количество мостов в заданном графе. На следующей строке выведите b целых чисел — номера ребер, которые являются мостами, в возрастающем порядке. Ребра нумеруются с единицы в том порядке, в котором они заданы во входном файле.

Пример

bridges.in	bridges.out
6 7 1 2 2 3 3 4 1 3 4 5 4 6 5 6	1 3

Королевская задача

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

Король Артур любит читать философские трактаты. Недавно его верные рыцари добыли трактат, в котором рассказывается о том, как улучшить управление королевством. Его Величество с головой ушёл в чтение. Он узнал много нового, в частности, что королевством гораздо удобнее править, если для любых двух различных городов королевства u и v существует путь по дорогам либо из u в v, либо из v в u (а возможно и оба).

Теперь Артур хочет узнать, удовлетворяет ли его королевство этому условию. И задачу эту он поручил своему придворному магу, то есть Вам. К сожалению, Вы не маг, а всего лишь путешественник, затерянный во времени со своим ноутбуком. Поэтому для решения этой задачи Вам придётся написать программу.

Формат входного файла

В первой строке содержатся два числа N и M — число городов в королевстве Артура, и число дорог. Далее следуют M строк с описаниями дорог. К сожалению, местные маги здорово поколдовали в былые времена, поэтому по всем дорогам можно перемещаться только в одном направлении, так как неумолимые магические силы приносят погибель всем, кто осмелится пойти против них. Каждая дорога описывается двумя числами: u_i и v_i, она ведет из города u_i в город v_i. Между двумя городами может быть несколько дорог в различных направлениях, также бывают странные дороги, которые ведут из города в себя. Числа N и M не превосходят 100 000.

Формат выходного файла

Если королевство удовлетворяет требуемому условию, то выведите `YES`, иначе `NO`.

Пример

kingtask.in	kingtask.out
3 3 1 2 1 3 2 3	YES
3 2 1 2 1 3	NO
6 7 1 2 2 3 3 1 3 4 4 5 5 6 6 4	YES

Республика

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

В одной далекой-предалекой республике очень странная политическая система. Их конституция провозглашает *бюракратию* вместо демократии! Конечно, из-за этого появляются некоторые проблемы. Даже президентский приказ недействителен без 239 различных подписей.

Недавно президент понял, что идея сделать все дороги односторонними оказалась не такой удачной, как планировалось ранее. Существуют пары городов (a, b) такие, что нет

пути из a в b. Конечно, президент беспокоится о своих гражданах и хочет исправить эту проблему.

Но для строительства дороги требется подготовить, согласовать, подписать и проверить миллионы документов. Таким образом, президент хочет построить как можно меньше дорог (также односторонних) так, что можно будет добраться из каждого города в каждый (возможно, проходя через другие города). Вашему министерству поручили эту задачу. Напишите программу, которая будет определять минимальное количество дорог, которые необходимо построить.

Формат входного файла

Входной файл содержит не более 1 000 тестов. Каждый тест начинается с двух целых чисел: N (кодичество городов в республике) и M. Далее идут M Описаний дорог ($1 \le N \le 50\,000$, $0 \le M \le 100\,000$). Описание каждой дороги состоит из двух целых чисел a_i и b_i ($1 \le a_i, b_i \le N$), что означает наличие дороги из города a_i в город b_i. Входной файл завершается тестом $N = M = 0$. Этот тест не надо обрабатывать. Сумма всех M во входном файлк не превосходит 100 000.

Формат выходного файла

Для каждого теста выведите минимальное количество дорог, которые нужно построить, максимально точно следуя формату из примера.

Пример

republic.in	republic.out
3 3 1 2 1 1 1 2 3 4 1 2 2 3 3 1 3 2 2 1 1 2 1 0 2 0 0 0	Case 1: The minimal number of roads is 2. Case 2: The system is already connected. Case 3: The minimal number of roads is 1. Case 4: The system is already connected. Case 5: The minimal number of roads is 2.

Конденсация графа

Имя входного файла:	condense2.in
Имя выходного файла:	condense2.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Требуется найти количество ребер в конденсации (диаграмме порядка) ориентированного графа. Примечание: конденсация графа не содержит кратных ребер.

Формат входного файла

Первая строка входного файла содержит два натуральных числа n и m — количество вершин и ребер графа соответственно ($n \le 10\,000$, $m \le 100\,000$). Следующие m строк содержат описание ребер, по одному на строке. Ребро номер i описывается двумя натуральными числами b_i, e_i — началом и концом ребра соответственно ($1 \le b_i, e_i \le n$). В графе могут присутствовать кратные ребра и петли.

Формат выходного файла

Первая строка выходного файла должна содержать одно число — количество ребер в конденсации графа.

Пример

condense2.in	condense2.out
4 4 2 1 3 2 2 3 4 3	2

Точки сочленения

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

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

Формат входного файла

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

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

Формат выходного файла

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

points.in	points.out
9 12 1 2 1 3 2 3 1 4 4 5 1 5 2 6 6 7 2 7 3 8 8 9 3 9	3 1 2 3