

Задача 1. Найти цикл

Источник:	базовая
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Дан ориентированный граф. Требуется найти в этом графе простой цикл, если он существует.

Цикл — это замкнутый маршрут, идущий по рёбрам графа. Цикл называется простым, если он проходит через каждую вершину не более одного раза (за один круг).

Формат входных данных

В первой строке содержатся два целых числа N и M , где N — количество вершин, M — количество рёбер ($1 \leq N, M \leq 2 \cdot 10^5$).

В следующих M строках записаны ориентированные рёбра, по одному в строке. Для каждого ребра записано два целых числа: u_j — номер начальной вершины и v_j — номер конечной вершины ($1 \leq u_j \neq v_j \leq N$).

Формат выходных данных

Если в графе нет циклов, выведите одно целое число -1.

Иначе выведите **любой** простой цикл в виде последовательности вершин. В первой строке требуется записать число K — количество вершин в цикле ($2 \leq K \leq N$). Во второй строке должно быть записано K различных целых чисел X_i — номера вершин цикла в порядке их прохождения ($1 \leq X_i \leq N$).

Примеры

<code>input.txt</code>	<code>output.txt</code>
6 5 1 2 3 2 1 3 6 5 5 4	-1
6 5 1 2 2 3 3 1 6 5 5 4	3 2 3 1

Задача 2. Проверить граф на эйлеровость

Источник: базовая
Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дан неориентированный граф. Требуется проверить, содержит ли заданный граф эйлеров путь. И если это так, то найти этот путь.

Формат входных данных

В первой строке входного файла задано количество вершин N и количество ребер E ($1 \leq N, E \leq 20$).

В следующих E строках задаются ребра парой образующих вершин. Вершины нумеруются с 1 до N .

Формат выходных данных

Если заданный граф не содержит эйлеров путь, то в выходной файл требуется вывести одно слово NO.

В противном случае, в первую строку нужно вывести слово YES, а далее вывести номера ребер в порядке их эйлерова обхода, по одному в строке. Ребра нумеруются в порядке их появления во входном файле с 1.

Пример

input.txt	output.txt
5 8	YES
1 2	6
1 3	2
2 3	4
3 4	7
5 4	1
5 1	3
1 4	8
3 5	5

Задача 3. SMS счастья

Источник:	основная
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Студенты факультета информационных технологий Урюпинского государственного университета разрабатывают систему оповещения о различных событиях, таких, например, как назначение даты экзамена. В качестве транспортной среды для этой системы оповещения были выбраны SMS. Все студенты курса имеют сотовые телефоны, которые постоянно включены. Каждый студент в своей адресной книге помимо других записей имеет номера, по крайней мере, половины своих однокурсников. А те однокурсники, которые записаны в его адресной книге, имеют его номер тоже.

Всем понятно, что несправедливо заставлять инициатора рассылки сообщения платить за отправку SMS всем остальным студентам курса, поэтому было решено отправлять SMS по кругу: каждый получатель сообщения отправляет его следующему студенту. Последний студент в цепочке должен переслать SMS инициатору рассылки. Этим достигаются два полезных результата. Во-первых, инициатор рассылки таким образом узнает, дошло ли его сообщение до всех студентов. Во-вторых, благодаря этому любой студент может выступать в роли инициатора рассылки.

Вам передано содержимое адресных книг телефонов всех студентов курса. Вам необходимо написать программу, которая строит возможный маршрут рассылки сообщений, удовлетворяющий следующим требованиям:

1. Маршрут должен быть замкнутым.
2. Маршрут должен проходить через телефон каждого из студентов ровно один раз.
3. Следующим звеном в маршруте всегда должен быть телефон, записанный в адресной книге текущего телефона.

Формат входных данных

В первой строке входного файла задано количество студентов на курсе N ($2 < N \leq 300$).

В следующих N блоках задается информация о содержимом адресной книги каждого из студентов курса.

В первой строке i -го блока записаны фамилия i -го студента и через пробел целое число — количество записей в адресной книге его телефона ($(N + 1)/2 \leq K_i \leq 100, 1 \leq i \leq N$).

В следующих K_i строках перечислены записи его адресной книги. Заданы только фамилии, по одной в строке. Все фамилии имеют длину не более 16 символов латиницы, большие и маленькие буквы не различаются.

Нужно иметь в виду, что в адресных книгах могут встречаться телефоны людей, не являющихся студентами этого курса.

Формат выходных данных

В выходной файл необходимо вывести один из возможных маршрутов рассылки, удовлетворяющий указанным в условии задачи требованиям. Маршрут выводится в виде списка из N фамилий, по одной фамилии в строке.

Пример

input.txt	output.txt
4 Ivanov 3 Petrov Sidorov Pentyushkin Petrov 2 Sidorov Ivanov Sidorov 3 Pentyushkin Ivanov Petrov Pentyushkin 3 Ivanov Kuznetsov Sidorov	Ivanov Petrov Sidorov Pentyushkin

Задача 4. Сотовая связь

Источник:	основная
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Сеть сотовой связи состоит из нескольких базовых станций. Сигнал каждой из станций принимается на определенной территории, называемой зоной приема. Зоны приема станций частично пересекаются. Их объединение образует зону приема сети в целом. Размещение базовых станций определяется различными соображениями, прежде всего – рельефом местности и ожидаемым количеством абонентов на данной территории. При этом никакие две станции с пересекающимися зонами приема не могут работать в одном частотном диапазоне. Оператор сотовой связи оплачивает занятые им частотные диапазоны, поэтому заинтересован в минимизации их количества.

Вам дана карта размещения базовых станций и зоны их приема. Зона приема каждой станции представляет собой круг определенного радиуса. Считается, что зоны приема пересекаются, если пересечение соответствующих кругов содержит больше, чем одну точку. Требуется найти минимальное количество частотных каналов, необходимое для работы сети.

Формат входных данных

Первая строка входного файла содержит целое число N — количество базовых станций ($0 < N < 15$).

Следующие N строк содержат описания каждой станции: координаты X и Y и радиус зоны приема R , разделенные пробелами. Все координаты целочисленные, $0 < X, Y, R < 16000$.

Формат выходных данных

В выходной файл нужно вывести одно целое число — минимальное количество частотных каналов.

Пример

<code>input.txt</code>	<code>output.txt</code>
3 1 1 80 100 100 80 50 80 80	3

Задача 5. Списывание

Источник:	повышенной сложности
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Сегодня ученики пишут контрольную работу. Проблема состоит в том, что они любят списывать друг у друга. Ученик, сидящий за партой, может списать у своего соседа, а также у учеников, сидящих за партами, стоящими впереди и позади его парты в том же ряду, если таковые есть. Таким образом, он может списать максимум у пяти человек. Списывание у людей, сидящих за партами в соседних рядах невозможно. Есть одно обстоятельство, облегчающее жизнь учителю. Некоторые ученики поссорились друг с другом, так что ни за что не дадут списать у себя человеку, с которым они в плохих отношениях. Учитель знает, кто с кем поссорился и, естественно, знает, кто где сидит. У него есть всего два варианта контрольной, и он хочет раздать их так, чтобы никто не мог ни у кого списать. Списывать другой вариант бесполезно. На этот раз вы помогаете учителю раздать так варианты контрольной, чтобы никто ни у кого не смог списать.

В классе парты стоят в N рядов по M парт в каждом. Всего в классе K учеников, пронумерованных числами от 1 до K в классном журнале. Можно считать, что если ученик A поссорился с учеником B , то ученик B тоже находится в плохих отношениях с учеником A . Можно также считать, что у учителя достаточно экземпляров каждого варианта для раздачи всем ученикам.

Формат входных данных

В первой строке входного файла записаны числа N , M и K ($1 \leq K \leq 2 \cdot M \cdot N$).

Общее количество парт, равное $M \cdot N$, не превышает 20000.

В следующих M строках идет описание расположения учеников за партами. Первая строка описывает парты, стоящие впереди класса, последняя строка — парты, стоящие в конце. В каждой строке записано по N пар чисел. Каждая пара чисел описывает учеников, сидящих за партами каждого из N рядов. Первое число в паре задает номер ученика, сидящего за партой слева, второе число — сидящего справа. Если соответствующее место пусто, то вместо номера ученика записывается число 0.

Далее идет K строк, описывающих отношения учеников. В i -й строке сначала записано число n_i — количество человек, с которыми i -й ученик находится в конфликте, а далее n_i чисел — номера этих учеников. Общее число пар учеников, которые находятся в ссоре, не превосходит 100000. Числа в строках разделяются пробелами.

Формат выходных данных

Если раздать варианты контрольной требуемым образом невозможно, выходной файл должен содержать единственное слово NO.

Если же это возможно, в первую строку требуется вывести слово YES, а во вторую — K чисел, разделенных пробелами. i -е число в строке должно быть равно номеру варианта, который надо дать i -му ученику для выполнения требуемых условий.

Если способов раздать варианты несколько, достаточно выдать любой из них.

Примеры

input.txt	output.txt
1 2 4 1 2 4 3 1 3 0 1 1 0	NO
1 2 3 1 2 0 3 1 3 0 1 1	YES 1 2 1