Cycle. Поиск цикла

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

Дан ориентированный невзвешенный граф. Необходимо определить есть ли в нём циклы, и если есть, то вывести любой из них.

Формат входного файла

В первой строке входного файла находятся два натуральных числа N и M ($1 \le N \le 100\,000, M \le 100\,000$) — количество вершин и рёбер в графе соответственно. Далее в M строках перечислены рёбра графа. Каждое ребро задаётся парой чисел — номерами начальной и конечной вершин соответственно.

Формат выходного файла

Если в графе нет цикла, то вывести «NO», иначе — «YES» и затем перечислить все вершины в порядке обхода цикла.

Пример

cycle.in	cycle.out
2 2 1 2 2 1	YES 2 1
2 2 1 2 1 2	NO

Island2. Островные государства-2

Имя входного файла:	`island2.in`
Имя выходного файла:	`island2.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Суровые феодальные времена переживала некогда великая островная страна Байтландия. За главенство над всем островом борются два самых сильных барона. Таким образом, каждый город страны контролируется одним из правителей. Как водится издревле, некоторые из городов соединены двусторонними дорогами. Бароны очень не любят друг друга и стараются делать как можно больше пакостей. В частности, теперь для того чтобы пройти по дороге, соединяющей города различных правителей, надо заплатить пошлину — один байтландский рубль. Кроме этого за выезд из городов с четными номерами береться удвоенная пошлина.

Программист Вася живет в городе номер 1. С наступлением лета он собирается съездить в город N на Всебайтландское сборище программистов. Разумеется, он хочет затратить при этом как можно меньше денег и помочь ему здесь, как обычно, предлагается Вам.

Формат входного файла

В первой строке входного файла записано два числа N и M ($1 \le N, M \le 100\,000$) — количество городов и количество дорог соответсвенно.

В следующий строке содержится информация о городах — N чисел 1 или 2 — какому из баронов принадлежит соответствующий город.

В последних M строках записаны пары $1 \le a, b \le N$, $a \ne b$. Каждая пара означает наличие дороги из города a в город b. По дорогам Байтландии можно двигаться в любом направлении.

Формат выходного файла

Если искомого пути не существует, выведите единственное слово `impossible`. В противном случае в первой строке напишите минимальную стоимость и количество посещенных городов, а во вторую выведите эти города в порядке посещения. Если минимальных путей несколько, выведите любой.

Пример

island2.in	island2.out
7 8 1 1 1 1 2 2 1 1 2 2 5 2 3 5 4 4 3 4 7 1 6 6 7	0 5 1 2 3 4 7
5 5 1 1 1 2 1 1 2 2 3 3 4 4 5 2 4	3 5 1 2 3 4 5

TopSort. Топологическая сортировка

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

Дан ориентированный невзвешенный граф. Необходимо его топологически отсортировать.

Формат входного файла

В первой строке входного файла даны два натуральных числа N и M ($1 \le N \le 100\,000, M \le 100\,000$) — количество вершин и рёбер в графе соответственно. Далее

в M строках перечислены рёбра графа. Каждое ребро задаётся парой чисел — номерами начальной и конечной вершин соответственно.

Формат выходного файла

Вывести любую топологическую сортировку графа в виде последовательности номеров вершин. Если граф невозможно топологически отсортировать, вывести -1.

Пример

topsort.in	topsort.out
6 6 1 2 3 2 4 2 2 5 6 5 4 6	4 6 3 1 2 5
3 3 1 2 2 3 3 1	-1

Экскурсия

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

ЛКШата собираются на экскурсию в Кострому. Кострома — красивый старый город, в котором площади соединяются друг с другом короткими улицами, каждую из которых можно пройти не более чем за десять минут. ЛКШата хотят составить интересный маршрут экскурсии. Так как они поедут на автобусах, то маршрут экскурсии должен начинаться и оканчиваться на одной и той же площади. К сожалению, у ЛКШат будет очень мало времени. Поэтому они решили выбрать наиболее короткий кольцевой маршрут, не проходящий ни по какой улице дважды.

Помогите ЛКШатам найти такой маршрут.

Формат входного файла

Первая строка входного файла содержит два натуральных числа n и m — количество площадей и улиц в Костроме ($1 \le n \le 1\,000$, $1 \le m \le 10\,000$). Площади занумерованы от 1 до n.

Последующие m строк содержат описания улиц. Каждая улица описывается тремя целыми числами — номерами площадей, которые она соединяет, и количеством минут, которые требуются ЛКШатам на то, чтобы пройти по ней (от одной до десяти минут). Между двумя площадями может быть более одной улицы. Улица соединяет две различные площади.

Гарантируется, что в Костроме существует как минимум один кольцевой маршрут.

Формат выходного файла

Первая строка выходного файла должна содержать единственное число — продолжительность минимального маршрута в минутах.

Пример

excurs.in	excurs.out
5 6 1 2 1 2 3 10 1 3 1 2 4 1 3 4 1 1 5 1	4

Ancestor. Предок

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

Напишите программу, которая для двух вершин дерева определяет, является ли одна из них предком другой.

Формат входного файла

Первая строка входного файла содержит натуральное число n ($1 \le n \le 100\,000$) — количество вершин в дереве. Во второй строке находится n чисел. При этом i-ое число второй строки определяет непосредственного родителя вершины с номером i. Если номер родителя равен нулю, то вершина является корнем дерева.

В третьей строке находится число m ($1 \le m \le 100\,000$) — количество запросов. Каждая из следующих m строк содержит два различных числа a и b ($1 \le a, b \le n$).

Формат выходного файла

Для каждого из m запросов выведите на отдельной строке число 1, если вершина a является одним из предков вершины b, и 0 в противном случае.

Пример

ancestor.in	ancestor.out
6 0 1 1 2 3 3 5 4 1 1 4 3 6 2 6 6 5	0 1 1 0 0