

Задача А. Разбиение на пути

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Разбейте рёбра неориентированного графа на минимальное количество рёберно-простых путей.

Формат входных данных

Дан граф. На первой строке число вершин n ($1 \leq n \leq 20\,000$) и число рёбер m ($1 \leq m \leq 20\,000$).
Следующие m строк содержат описание рёбер графа. Каждая строка по два числа a_i и b_i ($1 \leq a_i, b_i \leq n$). Между каждыми двумя вершинами не более одного ребра. Граф связан.

Формат выходных данных

На первой строке минимальное число путей. На каждой следующей описании очередного пути – номера вершин в порядке прохождения.

Примеры

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

Задача В. 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 ($1 \leq n \leq 100\,000$). 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	5
2 2 2	1 2 4 3 2 1
4 1 4 3 1	
2 2 4	
2 3 2	

Задача C. Таня и пароль

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

Пока папа был на работе, маленькая девочка Таня решила поиграть с папиным паролем к секретной базе данных. Папин пароль представляет собой строку, состоящую из $n + 2$ символов. Она выписала все возможные n трёхбуквенных подстрок пароля на бумажки, по одной на каждую бумажку, а сам пароль выкинула. Каждая трёхбуквенная подстрока была выписана на бумажки столько раз, сколько она встречалась в пароле. Таким образом, в итоге у Тани оказалось n бумажек.

Потом Таня поняла, что папа расстроится, если узнает о ее игре, и решила восстановить пароль или, по крайней мере, хотя бы какую-то строку, соответствующую получившемуся набору трёхбуквенных строк. Вам предстоит помочь ей в этой непростой задаче. Известно, что папин пароль состоял из строчных и заглавных букв латинского алфавита, а также из цифр. Строчные и заглавные буквы латинского алфавита считаются различными.

Формат входных данных

В первой строке следует целое число n ($1 \leq n \leq 2 \cdot 10^5$), количество трёхбуквенных подстрок, которые получились у Тани.

Следующие n строк каждая содержат по три буквы, образующие подстроку пароля папы. Каждый символ во вводе — строчная или заглавная буква латинского алфавита или цифра.

Формат выходных данных

Если во время игры Таня что-то напутала, и строк, соответствующих данному набору подстрок, не существует, то выведите «NO».

Если же возможно восстановить строку, соответствующую данному набору подстрок, то выведите «YES», а затем любой подходящий вариант пароля.

Примеры

стандартный ввод	стандартный вывод
5 aca aba aba cab bac	YES abacaba
4 abc bCb cb1 b13	NO
7 aaa aaa aaa aaa aaa aaa aaa	YES aaaaaaaaa

Задача D. 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 ($1 \leq n \leq 100\,000$).

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 ($1 \leq m_i \leq 100\,000$, $0 \leq s_i \leq 10^9$). 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 ($0 \leq p_{i,1} < p_{i,2} < \dots < p_{i,m_i} \leq s_i$).

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	$n \leq 8, m_i = 1, s_i \leq 100$		первая ошибка
2	8	$n \leq 8, s_i \leq 100$	1	первая ошибка
3	21	$n \leq 1\,000$	1, 2	первая ошибка
4	21	$\sum m_i > n$		первая ошибка
5	21	$s_i \leq 100$	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 E. Почтальон

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

В городе есть N площадей, соединенных улицами. При этом количество улиц не превышает 100 000 и существует не более трех площадей, на которые выходит нечетное количество улиц. Для каждой улицы известна ее длина. По каждой улице разрешено движение в обе стороны. В городе есть хотя бы одна улица. От каждой площади до любой другой можно прийти по улицам.

Почтальону требуется пройти хотя бы один раз по каждой улице. Почтальон хочет, чтобы длина его пути была минимальна. Он может начать движение на любой площади и закончить также на любой (в том числе и на начальной).

Помогите почтальону составить такой маршрут.

Формат входных данных

Сначала записано число N — количество площадей в городе ($2 \leq N \leq 1000$).

Далее следуют N строк, задающих улицы. В i -й из этих строк находится число m_i — количество улиц, выходящих из площади i . Далее следуют m_i пар натуральных чисел: в j -й паре первое число — номер площади, в которую идет j -я улица с i -й площади, а второе число — длина этой улицы.

Между двумя площадями может быть несколько улиц, но не может быть улицы с площади на нее саму.

Все числа во входном файле не превосходят 100 000.

Формат выходных данных

Если решение существует, то в первую строку выходного файла выведите одно число — количество улиц в искомом маршруте, а во вторую — номера площадей в порядке их посещения.

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

Если решений несколько, выведите любое.

Пример

стандартный ввод	стандартный вывод
4	5
2 2 1 2 2	1 2 4 3 2 1
4 1 2 4 4 3 5 1 1	
2 2 5 4 8	
2 3 8 2 4	

Задача F. 2-SAT

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

Формулировка 2-SAT: нужно подобрать значения n булевых переменных так, чтобы все m утверждений вида $(x_{i_1} = e_1) \vee (x_{i_2} = e_2)$ обратились в истину. В данной задаче вам гарантируется, что решение существует.

Формат входных данных

Входной файл состоит из одного или нескольких тестов.

Каждый тест описывается следующим образом. На первой строке число переменных n и число утверждений m . Каждая из следующих m строк содержит числа i_1, e_1, i_2, e_2 , задает утверждение $(x_{i_1} = e_1) \vee (x_{i_2} = e_2)$ ($0 \leq i_j < n, 0 \leq e_j \leq 1$). Ограничения: сумма всех n не больше 100 000, сумма всех m не больше 300 000.

Формат выходных данных

Для каждого теста выведите строку из n нулей и единиц — значения переменных. Если у данной задачи 2-SAT есть несколько решений, выведите любое.

Пример

стандартный ввод	стандартный вывод
1 0	0
2 2	01
0 0 1 0	000
0 1 1 1	
3 4	
0 1 1 0	
0 0 2 1	
1 1 2 0	
0 0 0 1	

Задача G. Экскурсия

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

Группа из n человек решила поехать на экскурсию. В процессе экскурсии можно заехать в некоторые из m городов.

Экскурсовод попросила каждого человека высказать свои пожелания по поводу посещения городов. Каждый человек может про какие-то города заявить, что он хочет их непременно посетить, а про какие-то — что точно не хочет.

Группа всегда путешествует вместе. Если группа заезжает в город, то все люди, заявившие, что точно не хотят его посетить, расстраиваются. Если группа не заезжает в город, то расстраиваются все люди, которые заявили, что хотят его непременно посетить.

Экскурсовод понимает, что удовлетворить все пожелания не всегда можно. Она хочет составить план, чтобы каждый человек расстроился не более одного раза.

Помогите экскурсоводу справиться с этой нелегкой задачей и составьте план посещения городов или выясните, что это невозможно.

Формат входных данных

В первой строке входных данных содержатся три целых числа n , m , k — количество человек, количество городов и количество пожеланий ($1 \leq n, m, k \leq 100\,000$).

В каждой из последующих k строк содержатся по два целых числа a и b , обозначающих пожелания ($1 \leq a \leq n, 1 \leq |b| \leq m$). Если $b > 0$, то человек под номером a хочет посетить город под номером b . Если же $b < 0$, то человек под номером a не хочет посетить город под номером $-b$. Каждое пожелание указано во вводе не более одного раза, ни для кого из участников нет одновременно пожелания посетить и не посещать один и тот же город.

Формат выходных данных

Если решения не существует, то выведите -1 .

Иначе, в первой строке выходных данных выведите единственное целое число k — количество городов, которые войдут в план.

Во второй строке выведите k целых чисел — номера городов, которые следует посетить. Номера городов можно выводить в любом порядке.

Если существует несколько возможных ответов, можно вывести любой из них. Обратите внимание, что не требуется искать максимальное или минимальное k , можно вывести любой корректный ответ.

Примеры

стандартный ввод	стандартный вывод
3 5 6 1 2 1 3 1 -4 2 3 2 4 2 5	3 2 3 5
3 3 6 1 -1 1 2 2 -2 2 3 3 -3 3 1	0

Задача Н. Раскраска в три цвета

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

Петя нарисовал на бумаге n кружков и соединил некоторые пары кружков линиями. После этого он раскрасил каждый кружок в один из трех цветов — красный, синий или зеленый.

Теперь Петя хочет изменить их раскраску. А именно — он хочет перекрасить каждый кружок в некоторый другой цвет так, чтобы никакие два кружка одного цвета не были соединены линией. При этом он хочет обязательно перекрасить каждый кружок, а перекрашивать кружок в тот же цвет, в который он был раскрашен исходно, не разрешается.

Помогите Пете решить, в какие цвета следует перекрасить кружки, чтобы выполнялось указанное условие.

Формат входных данных

Первая строка содержит два целых числа n и m — количество кружков и количество линий, которые нарисовал Петя, соответственно ($1 \leq n \leq 1\,000$, $0 \leq m \leq 20\,000$).

Следующая строка содержит n символов из множества $\{\text{'R'}, \text{'G'}, \text{'B'}\}$ — i -й из этих символов означает цвет, в который раскрашен i -й кружок ('R' — красный, 'G' — зеленый, 'B' — синий).

Следующие m строк содержат по два целых числа — пары кружков, соединенных отрезками.

Формат выходных данных

Выведите в выходной файл одну строку, состоящую из n символов из множества $\{\text{'R'}, \text{'G'}, \text{'B'}\}$ — цвета кружков после перекраски. Если решений несколько, выведите любое.

Если решения не существует, выведите в выходной файл слово "Impossible".

Примеры

стандартный ввод	стандартный вывод
4 5 RRRG 1 3 1 4 3 4 2 4 2 3	BBGR
4 5 RGRR 1 3 1 4 3 4 2 4 2 3	Impossible

Задача I. Установка модулей GAIA

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2.5 секунд
Ограничение по памяти:	256 мегабайт

Всего есть n модулей системы GAIA, таких как MINERVA, AETHER и другие. Модули пронумерованы от 1 до n , и для них есть ровно n слотов для подключения их к GAIA. Изначально модуль номер i подключен к слоту номер i .

Существуют только две операции, позволяющие оперировать назначением модулей GAIA по слотам. Эти операции могут быть описаны перестановками p и q длины n . В соответствии с операцией p , модуль, подключенный к слоту p_i , перемещается в слот i . Аналогично для q : при применении операции q , модуль, подключенный к слоту q_i , переподключается к слоту i .

Чтобы GAIA функционировала корректно, требуется назначить каждому модулю слот, используя *частичную композицию* операций p и q . Это означает, что для каждого i к слоту номер i должен быть подключен

- либо модуль, подключаемый к нему применением операции p ;
- либо модуль, подключаемый к нему применением операции p , а затем операции q .

Иными словами, к слоту номер i может быть подключен либо модуль с номером p_i , либо модуль с номером $(q \circ p)_i = q_{p_i}$. Для каждого i этот выбор можно сделать независимо от других.

Помимо этого, известны также m системных ограничений вида «модуль номер a_i не может располагаться на соседнем слоте с модулем b_i ».

Определите, существует ли частичная композиция перестановок p и q , обеспечивающая корректное функционирование GAIA, то есть при которой

- каждый модуль подключен к своему слоту, и каждый слот занят только одним модулем;
- и удовлетворены все ограничения на расположение модулей в соседних слотах.

Формат входных данных

В первой строке ввода через пробел даны два целых числа n и m — количество модулей системы и количество ограничений ($1 \leq n \leq 2 \cdot 10^5$; $0 \leq m \leq 2 \cdot 10^5$).

Во второй и третьей строках через пробел перечислены элементы перестановок p и q , описывающих операции ($1 \leq p_i, q_i \leq n$). Гарантируется, что каждое число от 1 до n встречается в описании каждой операции ровно один раз.

В следующих m строках даны ограничения на расположение модулей: в i -й из них через пробел даны два целых числа a_i и b_i — номера модулей, которые не должны располагаться в соседних слотах ($1 \leq a_i, b_i \leq n$; $a_i \neq b_i$).

Формат выходных данных

Если невозможно построить удовлетворяющее условию назначение, выведите «-1» (без кавычек).

Иначе выведите через пробел n целых чисел, i -е из которых равно 1, если для i -го слота выбрано назначение, соответствующее p , и 2, если выбрано назначение, соответствующее $q \circ p$.

Если есть несколько подходящих вариантов назначений, выведите любой из них.

Система оценки

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

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	9	$n \leq 3$	–	полная
2	14	$n \leq 18$	1	полная
3	14	$p_i = i$ для всех i	–	полная
4	17	$m \leq 1$	–	полная
5	21	$q_{p_i} = i$ для всех i	–	полная
6	25	нет	1 – 5	первая ошибка

Примеры

стандартный ввод	стандартный вывод
3 1 3 1 2 2 1 3 1 3	1 2 2
3 1 1 2 3 3 2 1 1 2	-1
4 2 3 4 1 2 4 1 3 2 1 2 3 4	1 2 2 2