

Задача А. Коровники

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

В селе Максоярославке коровы обычно пасутся на лужайках, соединённых дорожками, на каждой лужайке пасётся хотя бы одна корова. При этом для каждой пары лужаек есть ровно один способ пройти от одной лужайки до другой. По каждой дорожке можно двигаться в обоих направлениях. Считается, что все дорожки имеют одинаковую длину.

Главный фермер села хочет построить на лужайках k коровников для своих коров. Ясно, что каждая корова вечером будет возвращаться именно в тот коровник, который ближе к её лужайке (если расстояние до коровников одинаково, то в любой из них). Поэтому возникает задача определения такого расположения коровников, при котором наибольшее из расстояний, проходимых коровами, было бы минимально.

Формат входного файла

В первой строке входного файла содержатся два числа n и k ($2 \leq n \leq 50\,000$, $1 \leq k \leq n$) — количество лужаек и планируемое число коровников, соответственно. Следующие $n - 1$ строк содержат описания дорожек. Каждая дорожка задаётся парой целых положительных чисел (a, b) , где a и b — номера лужаек, которые соединяет данная дорожка. Лужайки нумеруются с единицы.

Формат выходного файла

В первой строке входного файла выведите l — максимальное количество дорожек, по которым придётся пройти корове, чтобы попасть в коровник. Во второй строке выведите k различных целых чисел — номера лужаек, на которых следует построить коровники. Если оптимальных решений несколько, разрешается вывести любое из них.

Пример

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

Замечание. Гарантируется, что решение, работающее на всех тестах с $k \leq 2$, получает не менее 40 баллов.

Задача В. Перекрестные уги

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

Задан направленный ациклический раф G , содержащий n ершин и m уг. Параллельные уги в рафе отсутствуют. *Остовное поддерево* рафа G представляет собой ориентированное орневое ерево (уги направлены от орня), покрывающее все ершины G . Требуется выяснить, существует ли остовное поддерево рафа G , относительно которого в G нет перекрестных уг. (Напомним, что уга (x, y) рафа G называется *перекрестной* относительно остовного поддерева T , если x не достижима из y по T и наоборот.)

Формат входного файла

В первой строке входного файла заданы числа n ($2 \leq n \leq 100\,000$) и m ($1 \leq m \leq 300\,000$). В последующих m строках описаны уги рафа. Каждая уга задаётся парой чисел — номерами начальной и конечной ершины соответственно. Ершины рафа нумеруются натуральными числами, начиная с 1.

Формат выходного файла

В случае если искомого ерева не существует, то выведите в первой (и единственной) строке выходного файла число -1. Если искомое ерево T существует, то выведите в первой строке количество уг k в T . Во второй строке выведите k чисел — номера уг, образующих T (уги пронумерованы натуральными числами, начиная с 1 в соответствии с тем, как они заданы во входном файле). Номера уг необходимо выводить в порядке возрастания.

Примеры

cross.in	cross.out
3 3 1 2 1 3 2 3	2 1 3
4 5 1 2 1 3 1 4 2 4 3 4	-1

Задача С. Верификация моделей

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

Сережа работает над новым проектом по верификации недетерминированных программных моделей. Первая версия программы будет работать с ациклическими программами. Ациклическая программа в виде, пригодном для верификации, представляется в виде ориентированного ациклического взвешенного графа, в котором есть вершина s , из которой достижимы все остальные.

Результатом верификации ациклической программы является верификационное дерево. *Верификационным деревом* называется ориентированное корневое дерево с корнем в вершине s , по дугам которого можно добраться из вершины s до любой другой вершины. Изучаемой характеристикой верификационного дерева является его характеристика Бабёнок–Копелиовича (ХБК) — количество единиц в двоичной записи суммы весов дуг, из которых оно состоит. Поскольку у ациклической программы может быть несколько верификационных деревьев, Сережу интересует среднее значение ХБК по всем возможным верификационным деревьям.

Формат входного файла

Первая строка входного файла содержит два целых числа n и m — количество вершин и дуг графа, соответственно ($2 \leq n \leq 20$, $1 \leq m \leq 50$). Вершина s имеет номер 1.

Следующие m строк содержат по три целых числа a_i , b_i и c_i — номер вершины, из которой выходит дуга, номер вершины, в которую она входит и веса этой дуги. Веса дуг неотрицательны и не превышают 10^7 . В графе нет параллельных дуг. В графе нет циклов.

Формат выходного файла

Выведите в выходной файл одно вещественное число — среднее количество единиц в двоичной записи веса верификационного дерева. Ответ должен отличаться от правильного не более чем на 10^{-6} .

Примеры

model.in	model.out
4 4 1 2 1 1 3 1 2 4 1 3 4 2	1.5