

Задача А. Выбор вершин дерева

Имя входного файла: `select.in`
Имя выходного файла: `select.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 Мебибайт

Дан граф, являющийся деревом. Множество вершин графа называется *допустимым*, если никакие две вершины этого множества не соединены ребром.

Рассмотрим все допустимые множества вершин графа. Для каждого такого множества посчитаем количество вершин в нём. Каково максимальное из этих количеств?

Формат входного файла

Граф в этой задаче задан в виде *корневого дерева*. В графе выделена вершина — *корень дерева*. Для каждой вершины i , не являющейся корнем, задан номер вершины-предка p_i в корневом дереве. Дерево, заданное таким образом, состоит из рёбер $i-p_i$ для всех вершин i , кроме корня.

В первой строке входного файла записано целое число n — количество вершин в графе ($1 \leq n \leq 100$). В следующих n строках задан граф. В i -й из этих строк записано целое число p_i — номер вершины-предка i -й вершины. Для корня дерева $p_i = 0$; для всех остальных вершин $1 \leq p_i \leq n$.

Гарантируется, что заданный во входном файле граф является деревом.

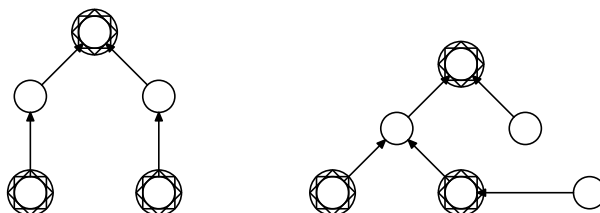
Формат выходного файла

В первой строке выходного файла выведите одно число — максимальное количество вершин в допустимом множестве.

Примеры

select.in	select.out
5 0 1 1 2 3	3
6 5 6 5 1 0 1	3

На рисунке показаны графы, заданные в примерах. В каждом графе выделено допустимое множество с максимальным количеством вершин в нём.



Задача В. Простые пути в дереве

Имя входного файла: `treenum.in`
Имя выходного файла: `treenum.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дан неориентированный связный граф из n вершин и $n - 1$ ребра. Требуется для каждого ребра посчитать количество простых путей, проходящих через это ребро.

Формат входного файла

На первой строке целое число n ($2 \leq n \leq 300\,000$). Следующие $n - 1$ строка содержат пары чисел от 1 до n — ребра графа.

Формат выходного файла

$n - 1$ строка. i -я строка должна содержать целое число — ответ для i -го ребра.

Пример

treenum.in	treenum.out
5	6
1 2	4
2 3	4
2 4	4
5 1	

Задача С. Простые пути в дереве 2

Имя входного файла: `treedp.in`
Имя выходного файла: `treedp.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дан неориентированный связный граф из n вершин и $n - 1$ ребра. Требуется для каждого ребра посчитать суммарную длину простых путей, проходящих через это ребро. Длиной пути здесь называется количество ребер в пути.

Формат входного файла

На первой строке целое число n ($2 \leq n \leq 300\,000$). Следующие $n - 1$ строка содержат пары чисел от 1 до n — ребра графа.

Формат выходного файла

$n - 1$ строка. i -я строка должна содержать целое число — ответ для i -го ребра.

Пример

treedp.in	treedp.out
5	13
1 2	8
2 3	8
2 4	9
5 1	

Задача D. Жизнь цвета индиго

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

Мальчику Севе очень нравится цвет индиго (это такой темно-синий цвет). Сева всячески старается окружить себя вещами этого цвета.

Скоро Новый год, и Сева решил изготовить гирлянду с лампочками, чтобы украсить ей свою комнату. Он купил n патронов для лампочек и соединил $(n - 1)$ -им проводом так, что в гирлянде не образовалось ни одного замкнутого контура. Его гирлянда, таким образом с точки зрения математики, представляет собой дерево. Для того, чтобы гирлянда была готова, осталось совсем немного — надо только добавить провод для подключения гирлянды в электрическую сеть и вкрутить разноцветные лампочки в патроны.

У Севы есть лампочки трех разных цветов: синего, фиолетового и индиго. При этом, в некоторые патроны из соображений красоты нельзя устанавливать лампочки определенных цветов. Также, из соображений красоты, в гирлянде не должно быть двух лампочек одного цвета, напрямую соединенных проводом. Разумеется, Сева хочет, чтобы в гирлянде было как можно больше лампочек его любимого цвета.

Помогите Севе. Напишите программу, которая найдет максимальное количество лампочек цвета индиго, которые можно установить в собранную Севой гирлянду.

Формат входного файла

Первая строка входного файла содержит целое число n — количество патронов для лампочек в Севиней гирлянде ($1 \leq n \leq 50000$).

Последующие $(n - 1)$ строка содержат каждая по два числа: u_i и v_i ($1 \leq u_i, v_i \leq n$) — номера патронов, соединенных соответствующим проводом.

Далее следуют n строк с описанием патронов. Каждая из них не пуста и состоит из не более, чем трех различных символов «I», «B» или «V». i -ая из этих строк описывает i -ый патрон, а именно: если она содержит символ «I», то в i -ый патрон можно устанавливать лампочку цвета индиго, «B» — можно устанавливать лампочку синего цвета, «V» — можно устанавливать лампочку фиолетового цвета.

Формат выходного файла

В выходной файл выведите единственное число: ответ на задачу. Если же при наложенных ограничениях Севе вообще не удастся собрать гирлянду, выведите число -1 .

Примеры

<code>indigo.in</code>	<code>indigo.out</code>
3 1 2 2 3 IB IV IB	2

Задача Е. Редукция дерева

Имя входного файла: `tree.in`
Имя выходного файла: `tree.out`
Ограничение по времени: 0.5 секунд
Ограничение по памяти: 256 мегабайт

Задано неориентированное дерево, содержащее n вершин. Можно выбрать некоторое ребро и удалить его, при этом инцидентные ему вершины не удаляются. Таким образом можно удалить из дерева некоторый набор рёбер. В результате дерево распадается на некоторое количество меньших деревьев. Требуется, удалив наименьшее количество рёбер, получить в качестве хотя бы одной из компонент связности дерево, содержащее ровно p вершин.

Формат входного файла

Первая строка входного файла содержит пару натуральных чисел n и p ($1 \leq p \leq n \leq 1000$). Далее в $n - 1$ строке содержатся описания рёбер дерева. Каждое описание состоит из пары натуральных чисел a_i, b_i ($1 \leq a_i, b_i \leq n$) — номеров соединяемых ребром вершин.

Формат выходного файла

В первую строку выведите наименьшее количество рёбер q в искомом наборе. Во вторую строку выведите номера удаляемых рёбер. Номера рёбер определяются порядком их задания по входном файле. Рёбра нумеруются с единицы. Если оптимальных решений несколько, разрешается выводить любое.

Пример

tree.in	tree.out
11 6	2
1 2	3 6
1 3	
1 4	
2 6	
2 7	
1 5	
2 8	
4 9	
4 10	
4 11	

Задача F. Максимальные пути

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дано взвешенное дерево из n вершин. Постройте на нем несколько непересекающихся по вершинам путей с максимальным суммарным весом.

Формат входного файла

В первой строке входных данных находится целое число n ($1 \leq n \leq 100000$) — количество вершин дерева.

В следующих $n - 1$ строках находятся описания ребер. Каждое ребро описано тройкой чисел x, y, w ($1 \leq x, y \leq n, 1 \leq w \leq 1000$) — номера вершин, соединенных ребром и его вес.

Формат выходного файла

Выведите одно число: максимальный суммарный вес путей.

Пример

стандартный ввод	стандартный вывод
5 5 1 7 2 4 6 5 2 4 3 5 5	18

Комментарий

Комментарий : в тесте из примера оптимальное покрытие - это пути 3-5-1 (вес пути равен $5+7=12$) и 2-4 (вес пути равен 6). Эти пути вершинно не пересекаются и имеют суммарный вес $12+6 = 18$

Задача G. Суммарное расстояние

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Байтландия представляет собой n городов, соединенных $n - 1$ дорогой так, что от каждого города можно добраться до любого другого. Байтландцы часто спорят, какой из городов более центральный, а какой — более провинциальный. Чтобы решить эти споры был принят закон, что город A является более центральным по сравнению с городом B , если суммарное расстояние от A до всех остальных городов меньше, чем суммарное расстояние от B до всех остальных городов. Чтобы помочь Байтландцам решить старый спор посчитайте для каждого города суммарное расстояние до остальных городов.

Формат входного файла

В первой строке входных данных находится целое число n ($1 \leq n \leq 100000$) — количество городов дерева.

В следующих $n - 1$ строках находятся описания дорог. Каждая дорога описана тройкой чисел x, y, w ($1 \leq x, y \leq n, 1 \leq w \leq 1000$) — номера городов, соединенных дорогой и ее длина.

Формат выходного файла

Выведите n чисел. Для каждого города выведите суммарное расстояние от него до остальных городов.

Пример

стандартный ввод	стандартный вывод
5	47
5 1 7	30
2 4 6	41
5 2 4	48
3 5 5	26

Задача Н. Гонка

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

В регионе Паттайя-Чонбури находятся n городов, пронумерованных от 0 до $n - 1$ соединённых сетью из $n - 1$ магистралей. Каждая магистраль — двусторонняя, соединяет два различных города, и для неё известна длина в километрах — целое число. Известно, что между каждой парой городов существует ровно один возможный путь, соединяющий эти города. Таким образом, для любой пары городов существует ровно одна последовательность различных магистралей, по которой можно проехать из одного города в другой, не посещая никакой город дважды.

По требованиям организаторов МОГ трасса должна являться путём суммарной длины ровно k километров, начинающимся и заканчивающимся в различных городах. Естественно, никакая магистраль и, поэтому, никакой город не могут быть использованы дважды при выборе трассы, иначе возможны столкновения. Чтобы минимизировать влияние гонок на трафик движения в регионе, необходимо выбрать для трассы путь из наименьшего возможного количества магистралей.

Формат входного файла

Первая строка ввода содержит число n и число k ($1 \leq n \leq 200\,000$, $1 \leq k \leq 10^6$). Следующие $n - 1$ строк содержат по три целых числа: u_i , v_i и l_i — номера городов, соединенных магистралью и ее длина в километрах. Гарантируется, что между каждой парой городов существует ровно один возможный путь, соединяющий эти города.

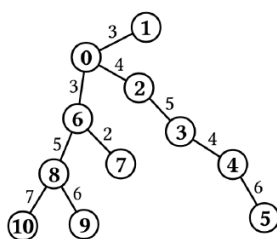
Формат выходного файла

Выведите одно целое число: минимальное число магистралей, которое можно задействовать для организации трассы. Если организовать трассу невозможно, выведите -1 .

Примеры

стандартный ввод	стандартный вывод
4 3 0 1 1 1 2 2 1 3 4	2
3 3 0 1 1 1 2 1	-1
11 12 0 1 3 0 2 4 2 3 5 3 4 4 4 5 6 0 6 3 6 7 2 6 8 5 8 9 6 8 10 7	2

Комментарий



В третьем примере оптимальный способ организации трассы: 6-8-10 (смотрите рисунок).

Задача I. Большой огромный коллайдер

Имя входного файла: `collider.in`
Имя выходного файла: `collider.out`
Ограничение по времени: 5 секунд
Ограничение по памяти: 256 мегабайта

Приехав в Хоббитанию, белый маг Гэндальф принялся рассказывать Бильбо последние новости из Средиземья. Больше всего впечатлительного хоббита поразил рассказ о Большом огромном коллайдере — только представить себе гигантских размеров кольцо, зарытое под землей!

Вдохновленный рассказом мага, Бильбо решил смастерить собственный коллайдер прямо у себя в норе. До начала строительства нора представляет собой множество комнат, причем некоторые пары комнат соединены коридорами. Как это принято у хоббитов, из любой комнаты в любую другую можно добраться по коридорам ровно одним способом.

Бильбо хочет прокопать новые коридоры в норе, но так как копать будут только Фродо и сам Бильбо, есть возможность прокопать только один или два новых коридора.

После этого последовательность из нескольких комнат будет названа коллайдером. При этом должны быть выполнены следующие условия: первая комната в этой последовательности соединена коридором со второй, вторая — с третьей, и так далее, наконец, последняя комната последовательности должна быть соединена с первой. Кроме того, каждая комната может входить в эту последовательность не более одного раза.

Помогите Бильбо выбрать такие один или два новых коридора, чтобы коллайдер имел максимальную возможную длину, то есть состоял из максимального возможного числа комнат.

Формат входного файла

В первой строке входного файла содержится целое число n ($3 \leq n \leq 100\,000$) — число комнат в норе Бильбо.

В следующих $n - 1$ строках содержатся по два целых числа — номера комнат, соединенных коридорами. Комнаты нумеруются от 1 до n .

Формат выходного файла

В первую строку выходного файла выведите максимально возможное число комнат в коллайдере.

На следующих одной или двух строках выведите пары номеров комнат, между которыми следует прокопать новые коридоры.

Если существует несколько возможных планов строительства коллайдера максимальной длины, выведите любой из них.

Примеры

<code>collider.in</code>	<code>collider.out</code>
4 1 2 2 3 3 4	4 1 4
4 1 2 1 3 1 4	4 2 3 2 4

В первом примере коллайдер состоит из комнат с номерами 1, 2, 3 и 4 (именно в этом порядке), во втором примере — 1, 3, 2, 4.