

Содержание

Easy	2
Задача А. От матрицы смежности к списку ребер [1 sec, 256 mb]	2
Задача В. Связность [1 sec, 256 mb]	3
Medium	4
Задача С. Расстояние от корня [2 sec, 256 mb]	4
Задача D. Стоимость проезда [2 sec, 256 mb]	5
Задача Е. Сумма расстояний [1 sec, 256 mb]	6
Задача F. From Sasha with love [1 sec, 256 mb]	7
Hard	9
Задача G. Любители Кошек [1 sec, 256 mb]	9

Вы не умеете читать/выводить данные, открывать файлы? Воспользуйтесь **примерами**.

В некоторых задачах большой ввод и вывод. Пользуйтесь **быстрым вводом-выводом**.

Обратите внимание, что ввод-вывод во всех задачах стандартный.

Easy

Задача А. От матрицы смежности к списку ребер [1 sec, 256 mb]

Простой неориентированный граф задан матрицей смежности, выведите его представление в виде списка ребер.

Формат входных данных

Входной файл содержит число N ($1 \leq N \leq 100$) — число вершин в графе, и затем N строк по N чисел, каждое из которых равно 0 или 1 — его матрицу смежности.

Формат выходных данных

Выведите в выходной файл список ребер заданного графа. Ребра можно выводить в произвольном порядке.

Пример

stdin	stdout
3	1 2
0 1 1	2 3
1 0 1	1 3
1 1 0	

Задача В. Связность [1 sec, 256 mb]

В этой задаче требуется проверить, что граф является *связным*, то есть что из любой вершины можно по рёбрам этого графа попасть в любую другую.

Формат входных данных

В первой строке входного файла заданы числа N и M через пробел — количество вершин и рёбер в графе, соответственно ($1 \leq N \leq 100$, $0 \leq M \leq 10\,000$). Следующие M строк содержат по два числа u_i и v_i через пробел ($1 \leq u_i, v_i \leq N$); каждая такая строка означает, что в графе существует ребро между вершинами u_i и v_i .

Формат выходных данных

Выведите “YES”, если граф является связным, и “NO” в противном случае.

Примеры

stdin	stdout
3 2 1 2 3 2	YES
3 1 1 3	NO

Medium

Задача С. Расстояние от корня [2 sec, 256 mb]

В заданном корневом дереве найдите вершины, максимально удалённые от корня. Расстоянием между вершинами считается количество рёбер в пути.

Формат входных данных

В первой строке задано n — количество вершин в дереве ($1 \leq n \leq 10^5$). В следующих $n - 1$ строках заданы вершины, являющиеся предками вершин $2, 3, \dots, n$. Вершина 1 является корнем дерева.

Формат выходных данных

В первой строке выведите максимальное расстояние от корня до остальных вершин дерева. Во второй строке выведите, сколько вершин дерева находятся от корня на таком расстоянии. В третьей строке выведите номера этих вершин через пробел в порядке возрастания.

Примеры

stdin	stdout
3	1
1	2
1	2 3
3	2
1	1
2	3

Задача D. Стоимость проезда [2 sec, 256 mb]

Страна состоит из n городов и m дорог. Города пронумерованы числами от 1 до n . Город с номером s является столицей. Все дороги односторонние, проход по каждой дороге стоит ровно 1 золотой. Требуется найти минимальные стоимости проезда от каждого города до столицы.

Формат входных данных

В первой строке файла записаны три целых числа — n , s и m (количество городов, номер столичного города и количество дорог).

В следующих m строках записаны пары чисел. Пара чисел (a, b) означает, что есть дорога из города a в город b .

Ограничения: $1 \leq n \leq 10^5, 0 \leq m \leq 10^5$.

Формат выходных данных

Выведите n чисел — минимальные стоимости проезда от городов до столицы. Если от какого-то города не существует ни одного пути до столицы, выведите -1 .

Пример

stdin	stdout
3 2 2 1 2 2 3	1 0 -1

Задача Е. Сумма расстояний [1 sec, 256 mb]

Дан связный граф. Требуется найти сумму расстояний между всеми парами вершин.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m — количество вершин и ребер графа соответственно ($1 \leq n \leq 1000$, $0 \leq m \leq 10\,000$).

Следующие m строк содержат описание ребер по одному на строке. Ребро номер i описывается двумя натуральными числами b_i , e_i — номерами концов ребра ($1 \leq b_i, e_i \leq n$).

Гарантируется, что граф связан.

Формат выходных данных

Первая строка выходного файла должна содержать одно натуральное число — сумму попарных расстояний между вершинами.

Пример

stdin	stdout
5 5 1 2 2 3 3 4 5 3 1 5	16

Задача F. From Sasha with love [1 sec, 256 mb]

Гномики Саша и Катя живут в лесу, в котором имеется N полянок, пронумерованных числами от 1 до N и соединенных $N - 1$ тропинкой. Лес устроен таким образом, что от любой полянки до любой другой можно добраться по тропинкам. Саша живет на полянке с номером A , а Катя — на полянке с номером B .

Саша захотел сделать Кате сюрприз и положил на каждой тропинке записку с буквой. Когда Катя пойдет в гости к Саше, она соберет записки, которые встретятся ей по пути. По задумке Саши из части собранных букв можно будет сложить словосочетание `love katya`. Известно, что путь, который выберет Катя, обязательно будет кратчайшим.

Девушки, как известно, весьма ветрены и непостоянны. Вот и теперь Кате нравится, когда ее называют другим именем. Поэтому вместо `love katya` ей хотелось бы сложить из записок словосочетание `love S`, где S — имя, которое ей нравится.

Помогите Саше узнать, сможет ли Катя собрать словосочетание `love S`.

Формат входных данных

В первой строке входного файла вводится строка S , состоящая из маленьких букв латинского алфавита. Длина строки S не превосходит 10^5 . Во второй строке входного файла через пробел вводятся числа N , A и B ($1 \leq N \leq 10^5$, $1 \leq A, B \leq N$) — количество полянок в лесу и номера полянок, на которых живут Саша и Катя.

В каждой из следующих $N - 1$ строк идет описание очередной тропинки. Тропинка задается тройкой a_i, b_i, c_i ($1 \leq a_i, b_i \leq N$), где a_i, b_i — номера полянок, которые соединяет i -я тропинка, а c_i — маленькая буква латинского алфавита, которую Саша положил на i -ой тропинке.

Формат выходных данных

В выходной файл в единственной строке выведите `Surprise`, если Кате удастся сложить из некоторых букв, которые она соберет по дороге к Саше, словосочетание `love S`, и `Fail` в противном случае.

Пример

stdin	stdout
katya 10 1 10 1 2 o 2 3 v 3 4 e 4 5 a 5 6 l 6 7 t 7 8 y 8 9 k 9 10 a	Surprise
katya 10 10 1 1 2 o 2 3 v 3 4 e 4 5 a 5 6 l 6 7 t 7 8 y 8 9 t 9 10 a	Fail

Hard

Задача G. Любители Кошек [1 sec, 256 mb]

В университетском клубе любителей кошек зарегистрировано n членов. Естественно, что некоторые из членов клуба знакомы друг с другом. Нужно сосчитать, сколькими способами можно выбрать из них троих, которые могли бы свободно общаться (то есть, любые два из которых знакомы между собой).

Формат входных данных

В первой строке входного файла заданы числа n и m ($1 \leq n \leq 1\,000$, $1 \leq m \leq 30\,000$), где m обозначает общее число знакомств. В последующих m строках идут пары чисел a_i b_i , обозначающие, что a_i знаком с b_i . Информация об одном знакомстве может быть записана несколько раз, причем даже в разном порядке (как (x, y) , так и (y, x)).

Формат выходных данных

В выходной файл необходимо вывести количество способов выбрать троих попарно знакомых друг с другом людей из клуба.

Пример

stdin	stdout
3 3 1 2 2 3 3 1	1