

Задача 1. Встреча шаманов

Источник: базовая
Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

В далекой стране живут дикие племена. Каждое племя имеет своего покровителя — шамана. Шаманы обладают удивительной способностью общаться между собой на астральном уровне. Но однажды над страной нависла угроза, поэтому шаманам понадобилось собраться всем вместе, чтобы провести некий обряд, известный только им.

Племена обитают в непроходимом лесу, в котором есть полянки, на которых кто-нибудь живет — шаман или же обычный дикарь. Некоторые полянки соединены между собой тропами, по которым можно свободно перемещаться.

Вам необходимо определить, смогут ли шаманы собраться вместе и предотвратить угрозу.

Формат входных данных

В первой строке входного файла записано два целых числа N — количество полянок в стране и M — количество тропинок ($1 < N \leq 50, 1 < M \leq 100$).

Следующие M строк содержат по два числа — номера полянок, между которыми есть тропа. Следующая строка содержит число шаманов в стране S ($1 \leq S \leq 10$).

За ней идет строка, которая содержит S различных чисел, записанных через пробел — номера полянок, на которых живут шаманы.

Полянки нумеруются подряд целыми числами, начиная с 1.

Формат выходных данных

В случае положительного ответа на вопрос, поставленный в условии задачи, в выходной файл необходимо вывести строку `YES`, в противном случае — `NO`.

Примеры

input.txt	output.txt
3 2 1 2 2 3 2 1 3	YES
4 2 1 2 3 4 2 1 3	NO

Задача 2. Знакомства

Источник: базовая
Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

В компании из N человек составили список попарно знакомых. Так как дело происходило на востоке, а восток — дело тонкое, то при этом, если A знаком с B и B знаком с C , то A и C считаются знакомыми (хотя и шапочными) и т.д. После этого возник вопрос — определить минимальное возможное число людей K , такое, чтобы при любом выборе K человек из N имеющихся среди них гарантированно встретились хотя бы двое хоть как-то знакомых.

Формат входных данных

В первой строке входного файла записано число $N \leq 100$. В последующих строках — непустой список пар знакомых, по одной паре знакомых, отделенных пробелом, в каждой строке.

Формат выходных данных

В выходной файл нужно записать одно число K .

Пример

input.txt	output.txt
6 1 2 2 3 4 5	4

Задача 3. Компоненты связности

Источник: основная
Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Вам задан неориентированный граф с N вершинами и M ребрами. В графе отсутствуют петли и кратные ребра.

Определите компоненты связности заданного графа.

Формат входных данных

Граф задан во входном файле следующим образом: первая строка содержит числа N и M ($1 \leq N \leq 20\,000$, $1 \leq M \leq 200\,000$). Каждая из следующих M строк содержит описание ребра — два целых числа из диапазона от 1 до N — номера концов ребра.

Формат выходных данных

На первой строке выходного файла выведите число L — количество компонент связности заданного графа. На следующей строке выведите N чисел из диапазона от 1 до L — номер компонент связности, которым принадлежат соответствующие вершины. Компоненты связности следует занумеровать от 1 до L произвольным образом.

Пример

input.txt	output.txt
4 2 1 2 3 4	2 1 1 2 2

Задача 4. Лабиринт

Источник: основная
Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Как известно, из любого лабиринта можно заведомо найти выход, если начать его “инспекцию” от входа, а затем двигаться, все время касаясь левой (или правой — на выбор) стены рукой.

Однако со времен фараоновых гробниц лабиринты строили весьма умные люди, это правило было им известно, и они старались сделать так, чтобы при подобном обходе лабиринта “посетитель” заведомо не попал в самые интересные места, — скажем, в те комнаты, где хранятся главные сокровища.

В этой задаче по имеющемуся плану лабиринта нужно определить, есть ли в нем недоступные области. При этом нужно полагать, что посетитель **может гулять по лабиринту как ему вздумается**, даже не используя правило левой руки.

Формат входных данных

В первой строке входного файла задано одно целое число N — размер лабиринта (лабиринт у нас квадратный, $3 \leq N \leq 1\,000$). Далее N строк содержат по N символов, описывающих сам лабиринт: стенки изображаются звездочками, пустоты — пробелами.

Вход в лабиринт — это единственная пустая клетка в первой строке.

Формат выходных данных

В выходной файл нужно вывести одно число — количество недоступных областей лабиринта. Недоступные области считаются различными, если из одной нельзя попасть в другую (проходы “по диагонали” не учитываются).

Пример

input.txt	output.txt
7 *** ** * * * *** * * * * * * *** * * * *****	1

Задача 5. Топологическая сортировка

Источник: основная
Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дан ориентированный граф. Необходимо каждой вершине присвоить номер так, чтобы все ребра шли из вершины с меньшим номером в вершину с большим. Номера должны быть разными для всех вершин.

Формат входных данных

В первой строке содержатся два числа N и M , где N — количество вершин графа, M — количество ребер ($1 \leq N \leq 10^4$, $1 \leq M \leq 10^5$).

В следующих M строках содержится по два числа: u, v , означающих, что в графе есть ребро из вершины u в вершину v ($1 \leq u, v \leq N$).

Формат выходных данных

В выходной файл необходимо вывести присвоенные номера вершин в том порядке, в котором они пронумерованы. Если решений несколько, выведите любое. Если решений нет, выведите “-1”. Номера вершин должны быть от 1 до N .

Примеры

input.txt	output.txt
6 5 1 2 3 2 1 3 6 5 5 4	1 3 2 6 5 4
6 5 1 2 2 3 3 1 6 5 5 4	-1
3 2 2 3 3 1	3 1 2

Задача 6. Корректность онтологии

Источник: повышенной сложности
Имя входного файла: Имя выходного файла: output.txt
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Онтология — эксплицитная спецификация концептуализации (Грубер).

Для автоматического анализа предметных областей или моделей, составления баз знаний, доказательства корректности и т.п. часто используются онтологии. В нашем случае под онтологией мы будем понимать словарь понятий с заданными связями между этими понятиями. На описание понятий накладываются следующие правила:

- отсутствие множественных определений: каждое понятие определяется только один раз;
- отсутствие циклических зависимостей.

Ваша задача состоит в том, чтобы проверить, является ли заданная онтология корректной, то есть удовлетворяет ли она указанным правилам описания.

Формат входных данных

В первой строке входного файла записано натуральное число N — количество определений. Далее в N строках приводится описание определений в формате: <определяемое понятие> <количество определяющих понятий> <определяющее понятие> <определяющее понятие> <определяющее понятие> ... Общее количество зависимостей между понятиями не превосходит 10^5 . Каждое понятие состоит из набора строчных букв латинского алфавита или знака подчёркивания '_'. Длина понятия не превышает 20 символов, суммарное число понятий не больше 10^5 . Могут встречаться также неопределяемые понятия, для которых описания не даны.

Формат выходных данных

В выходном файле должна находиться одна из следующих строк:

- “correct”, если онтология является корректной;
- “multiple definition”, если в описании есть хотя бы одно множественное определение;
- “loop detected”, если множественных определений нет, но есть циклические зависимости.

Примеры

input.txt	output.txt
3 woman 2 person female man 2 person male parent 3 woman man child	correct
7 woman 2 person female man 2 person male parent 3 woman man child left_leg 2 great_toe right woman 2 human female sipulka 2 breather sipulkaria sipulkaria 2 place sipulka	multiple definition
6 woman 2 person female man 2 person male parent 3 woman man child left_leg 2 great_toe right sipulka 2 breather sipulkaria sipulkaria 2 place sipulka	loop detected