

Задача А. Топологическая сортировка

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

Задан ориентированный ациклический граф с n вершинами и m ребрами. Также задана перестановка вершин графа. Необходимо проверить, является ли данная перестановка топологической сортировкой.

Формат входных данных

В первой строке даны два числа n и m — количество вершин и ребер в графе соответственно ($1 \leq n, m \leq 10^5$). В следующих m строках заданы пары чисел u_i, v_i , означающие, что в графе есть ребро из вершины u_i в вершину v_i . В последней строке задана перестановка из n элементов.

Формат выходных данных

Выведите «YES» (без кавычек), если данная перестановка является топологической сортировкой и «NO» в противном случае.

Примеры

стандартный ввод	стандартный вывод
3 3 2 3 1 3 1 2 2 1 3	NO
3 3 3 2 1 2 3 1 3 1 2	YES

Задача В. Topsort

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

Дан ориентированный невзвешенный граф. Необходимо его топологически отсортировать.

Формат входных данных

В первой строке входного файла даны два целых числа N и M ($1 \leq N \leq 100\,000, 0 \leq M \leq 100\,000$) — количества вершин и рёбер в графе соответственно. Далее в M строках перечислены рёбра графа. Каждое ребро задаётся парой чисел — номерами начальной и конечной вершин соответственно.

Формат выходных данных

Вывести любую топологическую сортировку графа в виде последовательности номеров вершин. Если граф невозможно топологически отсортировать, вывести «-1».

Пример

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

Задача С. Кратчайший путь

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

Дан ориентированный ациклический взвешенный граф. Требуется найти в нем кратчайший путь из вершины s в вершину t .

Формат входных данных

Первая строка входного файла содержит четыре целых числа n , m , s и t — количество вершин, дуг графа, начальная и конечная вершина соответственно.

Следующие m строк содержат описания дуг по одной на строке. Ребро номер i описывается тремя целыми числами b_i , e_i и w_i — началом, концом и длиной дуги соответственно ($1 \leq b_i, e_i \leq n$, $|w_i| \leq 1000$).

Входной граф не содержит циклов и петель.

$1 \leq n \leq 100\,000$, $0 \leq m \leq 200\,000$.

Формат выходных данных

Первая строка выходного файла должна содержать одно целое число — длину кратчайшего пути из s в t .

Если пути из s в t не существует, выведите «Unreachable».

Пример

стандартный ввод	стандартный вывод
2 1 1 2 1 2 -10	-10

Задача D. Конденсация графа. Light.

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	256 мегабайт

Требуется найти количество ребер в конденсации ориентированного графа. Примечание: конденсация графа не содержит кратных ребер и петель.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m – количество вершин и ребер графа соответственно ($n \leq 10\,000, m \leq 100\,000$). Следующие m строк содержат описание ребер, по одному на строке. Ребро номер i описывается двумя натуральными числами b_i, e_i – началом и концом ребра соответственно ($1 \leq b_i, e_i \leq n$). В графе могут присутствовать кратные ребра и петли.

Формат выходных данных

Первая строка выходного файла должна содержать одно число – количество ребер в конденсации графа.

Пример

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

Задача Е. Конденсация графа

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

Вам задан ориентированный граф с N вершинами и M ребрами ($1 \leq N \leq 200\,000$, $1 \leq M \leq 200\,000$). Найдите компоненты сильной связности заданного графа и топологически отсортируйте его конденсацию.

Формат входных данных

Граф задан во входном файле следующим образом: первая строка содержит числа N и M . Каждая из следующих M строк содержит описание ребра – два целых числа из диапазона от 1 до N – номера начала и конца ребра.

Формат выходных данных

На первой строке выведите число K – количество компонент сильной связности в заданном графе. На следующей строке выведите N чисел – для каждой вершины выведите номер компоненты сильной связности, которой принадлежит эта вершина. Компоненты сильной связности должны быть занумерованы таким образом, чтобы для любого ребра номер компоненты сильной связности его начала не превышал номера компоненты сильной связности его конца.

Пример

стандартный ввод	стандартный вывод
10 19	2
1 4	1 2 2 1 1 2 2 2 2 1
7 8	
5 10	
8 9	
9 6	
2 6	
6 2	
3 8	
9 2	
7 2	
9 7	
4 5	
3 6	
7 3	
6 7	
10 8	
10 1	
2 9	
2 7	

Задача F. Расстановка чисел

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

Вам даны N переменных a_1, a_2, \dots, a_N . Также про некоторые из этих переменных известно, что одна из них делится на другую.

Пусть мы присвоили этим переменным целые значения так, что все утверждения выполняются.

Требуется найти максимально возможное число **различных** значений среди этих переменных.

Формат входных данных

В первой строке входного файла задано два целых числа число N и M — количество переменных и количество отношений делимости ($1 \leq N \leq 10^4, 1 \leq M \leq 10^5$).

Далее следуют M пар чисел i и j ($1 \leq i, j \leq N$) задающих, что a_i делится на a_j .

При этом возможны как повторяющиеся утверждения, так и тривиальные (в которых $i = j$).

Формат выходных данных

Выведите одно число — наибольшее количество различных целых чисел в наборе a_i , удовлетворяющем всем утверждениям.

Пример

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

Замечание

В примере к задаче можно, например, взять $a_1 = 0, a_2 = -4, a_3 = 2$ и $a_4 = 3$.

Задача G. Горилла на Диком Западе

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

Однажды огромный голодный самец гориллы оказался на Диком Западе.

Дикий Запад представлен n городами. Вместо долларов на Диком Западе используются бананы. В городе i ($1 \leq i \leq n$) хранятся a_i бананов. Между некоторыми городами ходят дилижансы. Всего маршрутов дилижансов m , все маршруты **однаправленные**.

Когда горилла оказывается в каком-то городе **впервые**, она забирает **все** бананы оттуда. Горилла не получит бананы за последующие посещения этого города. Горилла может перемещаться между городами исключительно на дилижансах, поскольку на Диком Западе не нашлось лошади, способной выдержать гориллу.

Горилла хочет собрать как можно больше бананов. Помогите горилле составить оптимальный маршрут, если она может начать в любом городе и закончить в любом городе.

Формат входных данных

В первой строке даны целые числа n , m ($1 \leq n \leq 10^5$, $1 \leq m \leq 2 \cdot 10^5$) — количество городов и маршрутов дилижансов.

Во второй строке дано n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — количества бананов в городах.

В следующих m строках описаны маршруты дилижансов. Дилижанс, доставляющий **из** города v **в** город u , задаётся парой чисел v, u ($1 \leq v, u \leq n$).

Формат выходных данных

Выведите единственное целое число — максимальное количество бананов, которое может собрать горилла.

Пример

стандартный ввод	стандартный вывод
4 4 4 5 2 7 1 2 2 1 1 3 2 4	16

Замечание



Задача Н. Авиалинии

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

В Берляндии скоро появятся свои авиалинии. Комитет по разработке берляндских авиалиний уже предложил свой вариант соединения городов авиарейсами. Каждый авиарейс задается парой различных городов. Рейсы односторонние. Президенту понравился план, однако он показался ему чересчур неэкономным. Требуется разработать новый план, который содержит наименьшее количество авиарейсов и удовлетворяет условию: если из города a можно было попасть в город b (возможно, с пересадками) согласно первоначальному плану, то и в новом плане это должно быть возможным. Если же это было сделать невозможно, то и согласно новому плану это не должно быть возможным. Очевидно, что из любого города можно попасть в него самого.

Формат входных данных

В первой строке входного файла записаны целые числа N и M ($1 \leq N \leq 10^3$, $0 \leq M \leq 10^4$), где N — количество городов в стране, а M — количество авиарейсов в первоначальном плане. Города нумеруются от 1 до N . Далее записано M пар различных чисел a_i, b_i обозначающих наличие рейса из a_i в b_i в первоначальном плане ($1 \leq a_i \leq N$, $1 \leq b_i \leq N$). Пары разделяются пробелами или переводами строк. Между парой городов может быть более одного авиарейса.

Формат выходных данных

В первую строку выходного файла выведите количество рейсов в новом плане. Далее выведите авиарейсы в формате, аналогичном формату входных данных. Пары разделяйте пробелами или переводами строк. Пары выводите в любом порядке. Если существует несколько решений, выведите любое.

Пример

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

Задача I. Правильная скорая помощь

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

В связи с напряженной эпидемиологической обстановкой было решено пристраивать к уже существующим домам станции скорой помощи. Город задан в виде графа, в котором ребра — это односторонние дороги, а вершины — дома. Разумеется, во время вызова скорая может игнорировать ПДД (и даже направление движения), но вот возвращаться обратно по встрече уже не получится: больной уже под контролем врачей, да и рискованно это слишком.

Экономическая обстановка тоже не самая спокойная, поэтому требуется определить минимальное количество станций, которое нужно построить, чтобы скорая могла доехать обратно до станции от любого дома города.

Формат входных данных

В первой строке входного файла задано число n ($1 \leq n \leq 3000$) — количество домов. Во второй строке записано количество дорог m ($1 \leq m \leq 10^5$). Далее следует описание дорог в формате $a_i b_i$, означающее, что по i -й дороге разрешается движение от дома a_i к дому b_i ($1 \leq a_i, b_i \leq n$).

Формат выходных данных

Выведите одно число — минимальное количество станций, которое нужно построить, чтобы скорая могла доехать от любого дома до станции, соблюдая ПДД. Если к дому пристроена станция, то от этого дома до станции, очевидно, можно доехать.

Пример

стандартный ввод	стандартный вывод
3	1
3	
1 2	
2 3	
1 3	

Задача J. Водостоки

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

Карту местности условно разбили на квадраты, и посчитали среднюю высоту над уровнем моря для каждого квадрата.

Когда идет дождь, вода равномерно выпадает на все квадраты. Если один из четырех соседних с данным квадратом квадратов имеет меньшую высоту над уровнем моря, то вода с текущего квадрата стекает туда (и, если есть возможность, то дальше), если же все соседние квадраты имеют большую высоту, то вода скапливается в этом квадрате.

Разрешается в некоторых квадратах построить водостоки. Когда на каком-то квадрате строят водосток, то вся вода, которая раньше скапливалась в этом квадрате, будет утекать в водосток.

Если есть группа квадратов, имеющих одинаковую высоту и образующих связную область, то если хотя бы рядом с одним из этих квадратов есть квадрат, имеющий меньшую высоту, то вся вода утекает туда, если же такого квадрата нет, то вода стоит во всех этих квадратах. При этом достаточно построить водосток в любом из этих квадратов, и вся вода с них будет утекать в этот водосток.

Требуется определить, какое минимальное количество водостоков нужно построить, чтобы после дождя вся вода утекала в водостоки.

Формат входных данных

Во входном файле записаны сначала числа N и M , задающие размеры карты — натуральные числа, не превышающие 100. Далее идет N строк, по M чисел в каждой, задающих высоту квадратов карты над уровнем моря. Высота задается натуральным числом, не превышающим 10000. Считается, что квадраты, расположенные за пределами карты, имеют высоту 10001 (то есть вода никогда не утекает за пределы карты).

Формат выходных данных

В выходной файл выведите минимальное количество водостоков, которое необходимо построить.

Пример

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

Задача К. Расписание электричек

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

Рассмотрим расписание движения электричек на некоторой железнодорожной линии. Нас будут интересовать только электрички, идущие в одном направлении.

Каждая электричка отправляется с некоторой станции и следует до некоторой другой станции со всеми остановками. При этом средняя маршрутная скорость у каждой электрички своя (будем считать, что весь маршрут электричка проходит с этой скоростью, временем стоянки на станциях пренебрежем). Поскольку на участке только один путь в данном направлении — электрички в процессе следования друг друга не обгоняют.

Требуется выпустить книжку-расписание электричек. Обычно такая книжка представляет собой таблицу, где в первом столбце перечислены все станции, а каждый следующий столбец соответствует электричке: если электричка проходит через станцию, то в соответствующей клетке указывается время прохождения этой электрички через эту станцию, и прочерк, если электричка через эту станцию не проходит.

Естественно, что в книжке-расписании нужно расположить электрички так, чтобы они были указаны в хронологическом порядке. А именно, если две электрички имеют хотя бы одну общую станцию (даже если она является начальной станцией для одной, и конечной — для другой электрички), электрички в расписании должны идти в том порядке, в каком они проходят через эту станцию (поскольку электрички не обгоняют друг друга, то это же будет справедливо для всех общих станций этих двух электричек). Если же электрички не имеют ни одной общей станции, то они могут быть указаны в любом порядке.

По данному расписанию движения электричек определите порядок, в котором электрички должны идти в книжке-расписании.

Формат входных данных

Сначала вводится целое число N ($1 \leq N \leq 1000$) — количество электричек.

Далее идёт описание электричек: каждая электричка задается четырьмя числами A_i, B_i, C_i, D_i ($0 \leq A_i < B_i \leq 10^6$, $1 \leq C_i \leq 100$, $0 \leq D_i \leq 10000$), которые обозначают, что данная электричка отправляется со станции « A_i -й километр» и следует до станции « B_i -й километр». Электричка отправляется с начальной станции в момент C_i . Один километр электричка проезжает за D_i секунд.

Гарантируется, что расписание можно составить корректно, в частности, никакая электричка не обгоняет другую.

Формат выходных данных

Выведите последовательность из N номеров от 1 до N — номера электричек в том порядке, в котором они должны идти в книжке-расписании. Если возможных ответов несколько, выведите любой.

Пример

стандартный ввод	стандартный вывод
3	3 2 1
1 10 3 4	
3 5 3 4	
10 11 10 1	

Замечание

Ответ 2 3 1 также будет верным.