

Задача А. 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 В. От списка ребер к матрице смежности, ориентированный вариант

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

Простой ориентированный граф задан списком ребер, выведите его представление в виде матрицы смежности.

Формат входных данных

На вход программы поступают числа n ($1 \leq n \leq 100$) – количество вершин в графе и m ($1 \leq m \leq n(n-1)$) – количество ребер. Затем следует m пар чисел – ребра графа.

Формат выходных данных

Выведите матрицу смежности заданного графа.

Пример

стандартный ввод	стандартный вывод
5 3	0 0 1 0 0
1 3	0 0 1 0 0
2 3	0 0 0 0 0
5 2	0 0 0 0 0
	0 1 0 0 0

Задача С. Компоненты связности

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

Дан неориентированный невзвешенный граф. Необходимо посчитать количество его компонент связности и вывести их.

Формат входных данных

Во входном файле записано два числа N и M ($0 < N \leq 100000, 0 \leq M \leq 100000$). В следующих M строках записаны по два числа i и j ($1 \leq i, j \leq N$), которые означают, что вершины i и j соединены ребром.

Формат выходных данных

В первой строчке выходного файла выведите количество компонент связности. Далее выведите сами компоненты связности в следующем формате: в первой строке количество вершин в компоненте, во второй - сами вершины в отсортированном порядке.

Пример

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

Задача D. Отбой

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

Как всем известно, в 22:45 все ЛКШата должны быть в своих домиках. Петя слишком долго играл с друзьями в настольные игры и не успел вернуться до мягкого отбоя. Теперь он задумался о том, как пройти в свой домик так, чтобы его не заметили преподаватели.

ЛКШ проводится на базе Стёпаново, на которой есть n домиков, пронумерованных от 1 до n . Домики соединяют m дорожек. Известно, что каждая дорожка соединяет ровно два различных домика, и любые два домика соединены не более, чем одной дорожкой. Передвигаться можно только по дорожкам, при этом по любой дорожке можно двигаться в любом направлении. Известно, что из любого домика можно добраться до любого другого по дорожкам.

Петя знает, что в домике t живет Денис Павлович и, если пройти мимо, то он что-то заподозрит и поднимет тревогу. Если это произойдет, то ровно на $\lfloor \frac{m}{2} \rfloor$ случайных дорожках появятся преподаватели, жаждущие поставить кому-нибудь дырку в бейджик. По этим дорожкам Петя пройти не сможет. Петя не может заранее узнать, где будут преподаватели.

Вам нужно для k пар домиков u_i и v_i определить, сможет ли Петя гарантированно добраться из домика u_i в домик v_i , не получив дырку в бейджик.

Формат входных данных

В первой строке заданы три целых числа n , m и t ($3 \leq n, m \leq 10^5$, $1 \leq t \leq n$) – число домиков на базе, число дорожек и номер домика, где живет Денис Павлович.

В следующих m строках задано по два целых числа a_i и b_i ($1 \leq a_i, b_i \leq n$), обозначающих то, что домики a_i и b_i соединены дорожкой.

В следующей строке задано целое число k ($1 \leq k \leq 10^5$) – число запросов.

В следующих k строках задано по два целых числа u_i и v_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$) – домики, для которых нужно определить, сможет ли Петя дойти от одного до другого. Известно, что Денис Павлович не живет ни в домике u_i , ни в домике v_i .

Формат выходных данных

Для каждого из k запросов выведите на отдельной строке **Yes**, если Петя не сможет дойти до своего домика незамеченным, иначе выведите **No**.

Пример

стандартный ввод	стандартный вывод
4 3 3	Yes
4 3	No
4 2	
3 1	
2	
2 4	
1 4	

Задача Е. Авиаперелёты

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

Главного конструктора Петю попросили разработать новую модель самолёта для компании «Air Бубундия». Оказалось, что самая сложная часть заключается в подборе оптимального размера топливного бака.

Главный картограф «Air Бубундия» Вася составил подробную карту Бубундии. На этой карте он отметил расход топлива для перелёта между каждой парой городов.

Петя хочет сделать размер бака минимально возможным, для которого самолёт сможет долететь от любого города в любой другой (возможно, с дозаправками в пути).

Формат входных данных

Первая строка входного файла содержит натуральное число n ($1 \leq n \leq 1000$) — число городов в Бубундии. Далее идут n строк по n чисел каждая. j -е число в i -й строке равно расходу топлива при перелёте из i -го города в j -й. Все числа не меньше нуля и меньше 10^9 . Гарантируется, что для любого i в i -й строчке i -е число равно нулю.

Формат выходных данных

Первая строка выходного файла должна содержать одно число — оптимальный размер бака.

Пример

<code>avia.in</code>	<code>avia.out</code>
4 0 10 12 16 11 0 8 9 10 13 0 22 13 10 17 0	10

Задача F. Автостопом по планетам [C, В']

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

Артур Дент узнал, что Земля вскоре будет уничтожена. К сожалению, сообщивший ему эти известия Форд Префект куда-то запропастился, оставив лишь карту порталов.

Оказывается, если у тебя есть друг-инопланетянин, то в Солнечной системе не 8 планет, а целых n . Планеты пронумерованы от 1 до n по удаленности от Солнца. Так совпало, что планета с номером i имеет порталы на все планеты с номером, большим i , и не имеет ни одного портала на предыдущие планеты. Каждый портал контролируется одной из двух банд пришельцев.

Считается, что система порталов конфликтная, если существует пара планет (a, b) такая, что с планеты a можно добраться до планеты b как только по порталам первой банды, так и только по порталам второй банды.

Если система порталов оказывается конфликтной, банды спорят друг с другом, пока не устроят разногласия. Так как у обеих банд бессмертие уже является нормой, споры могут длиться бесконечно долго, причем всё это время все порталы системы функционировать не будут. Если же система порталов не конфликтная, по ней можно беспрепятственно перемещаться.

Артуру хочется выяснить, сможет ли он использовать систему порталов для побега с Земли, либо ему следует срочно начать паниковать.

Формат входных данных

Первая строка содержит целое число n ($3 \leq n \leq 1000$) — количество планет.

Следующая $n - 1$ строка содержит описание порталов. В i -й из этих строк содержится $n - i$ символов, j -й из которых определяет принадлежность портала из планеты i в планету $i + j$ и равен R, если это портал первой банды, и B, если второй.

Формат выходных данных

Выведите «YES», если Артур сможет спастись, используя систему порталов. Если же использовать ее не получится, а Артуру следует начинать паниковать, выведите «NO».

Примеры

стандартный ввод	стандартный вывод
3 RB R	NO
3 RR R	YES
5 BRRB BRB RR B	YES

Замечание

В первом примере система порталов конфликтная, так как из планеты 1 можно добраться до планеты 3 такими двумя способами:

- 1 → 2 → 3 — здесь все порталы принадлежат первой банде;
- 1 → 3 — здесь все порталы принадлежат второй банде.

Задача G.

Имя входного файла: `topsort.in`
Имя выходного файла: `topsort.out`
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

<code>topsort.in</code>	<code>topsort.out</code>
6 6 1 2 3 2 4 2 2 5 6 5 4 6	4 6 3 1 2 5

Задача Н. The One with the Cop [В', В]

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

–Pivot! Pivot! Pivot!
 –Shut up!! Shut up! Shut up!
 –Okay, I don't think it's gonna pivot anymore.
 –You think?!

Росс и Рейчел купили в квартиру отвратительно длинный диван. Росс работает палеонтологом в Нью-Йоркском Музее Естественной Истории, поэтому с его зарплатой он решил не тратиться на доставку. Росс посчитал, что сможет сам донести диван до своей квартиры.

По пути до своего дома Росс попал в переулок, который из-за своей странной структуры напоминал скорее лабиринт, чем Нью-Йоркский переулок.

Переулок представляет собой поле n на m клеток. Каждая клетка либо пуста, либо в ней находится препятствие. В том магазине, в котором были Росс и Рейчел, диваны бывают разной длины, но все они занимают k подряд идущих клеток в одной линии для некоторого k . Разумеется, диван не может находиться в клетке, в которой находится препятствие. Росс очень сильный, поэтому он может перенести диван любой длины.

Изначально, Росс может зайти в переулок в любой клетке левого столбца, но он должен держать диван параллельно левой границе переулка. Чтобы выйти из переулка, он должен оказаться в какой-нибудь клетке правого столбца переулка, держа диван параллельно этой границе.

Когда Росс держит диван параллельно одной из границ переулка, он может перенести его на одну клетку вдоль этой границы, или же он может взять диван за один из его концов, поднять его в этом месте в воздух, и опустить его параллельно другой стороне переулка. Разумеется, он может сделать эти действия, только если после этих действий диван будет находиться на пустых клетках.

Теперь Россу и Рейчел интересно, какой максимальной длины диван можно перенести через переулок.

Формат входных данных

В первой строке входных данных находятся два целых числа n и m ($1 \leq n, m \leq 300$) — количество строк и столбцов в переулке, соответственно. В следующих n строках находятся m символов. j -й символ i -й строки равняется «#», если в j -й клетке i -й строки находится стена, иначе он равен «.».

Формат выходных данных

В единственной строке выведете одно число — максимальное количество клеток, которое может занимать диван такой, что Росс может перенести его через переулок. Если диван никакой длины нельзя пронести через переулок, выведите число 0.

Пример

стандартный ввод	стандартный вывод
<pre>3 5 ...## .##. ##...</pre>	2

Задача I. Долой списывание!

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

Во время теста Михаил Дмитриевич заметил, что некоторые лкшат обмениваются записками. Сначала он хотел поставить им всем двойки, но в тот день Михаил Дмитриевич был добрым, а потому решил разделить лкшат на две группы: списывающих и дающих списывать, и поставить двойки только первым.

У Михаила Дмитриевича записаны все пары лкшат, обменявшихся записками. Требуется определить, сможет ли он разделить лкшат на две группы так, чтобы любой обмен записками осуществлялся от лкшонка одной группы лкшонку другой группы.

Формат входных данных

В первой строке находятся два числа N и M — количество лкшат и количество пар лкшат, обменивающихся записками ($1 \leq N \leq 100$, $0 \leq M \leq \frac{N(N-1)}{2}$). Далее в M строках расположены описания пар лкшат: два различных числа, соответствующие номерам лкшат, обменивающихся записками (нумерация лкшат идёт с 1). Каждая пара лкшат перечислена не более одного раза.

Формат выходных данных

Необходимо вывести ответ на задачу Павла Олеговича. Если возможно разделить лкшат на две группы, выведите «YES»; иначе выведите «NO».

Пример

bipartite.in	bipartite.out
3 2 1 2 2 3	YES