

Содержание

07.Base [3/3]	3
1 Задача 7A. От списка ребер к матрице смежности [0.5 sec]	3
2 Задача 7B. Связность [0.5 sec (1 sec)]	4
3 Задача 7C. Дерево [0.5 sec]	5
07.Advanced [4/8]	6
4 Задача 7D. Компоненты связности [0.5 sec (1 sec)]	6
5 Задача 7E. Поиск пути на гриде [0.8 sec (1.6 sec)]	7
6 Задача 7F. TopSort. Топологическая сортировка [0.5 sec (1 sec)]	8
7 Задача 7G. Поиск цикла [0.5 sec (1 sec)]	9
8 Задача 7H. Bridges. Мосты [0.4 sec (0.8 sec)]	10
9 Задача 7I. Condense 2. Конденсация графа [0.4 sec (0.8 sec)]	11
10 Задача 7J. Unique Topsort [0.5 sec (1 sec)]	12
11 Задача 7K. Longpath. Длиннейший путь [0.4 sec (0.8 sec)]	13
07.Hard [0/1]	14
12 Задача 7L. Раскраска в три цвета [1.2 sec (2.4 sec)]	14

Общая информация:

Вход в констест: <http://contest.yandex.ru/contest/809/>

Дедлайн на задачи 12-го ноября в 23:59.

К каждой главе есть более простые задачи (base), посложнее (advanced), и сложные (hard).

В скобках к каждой главе написано сколько любых задач из этой главы нужно сдать.

Сайт курса: <http://compscicenter.ru/courses/algorithms-1/2014-autumn/>

Семинары ведет Сергей Владимирович Копелиович,
контакты: burunduk30@gmail.com, vk.com/burunduk1

В каждом условии 2 таймлимита: для C/C++ и для Java, Python.

07.Base [3/3]

1 Задача 7А. От списка ребер к матрице смежности [0.5 sec]

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

Формат входных данных

Входной файл содержит числа N ($1 \leq N \leq 100$) — число вершин в графе и M ($1 \leq M \leq \frac{n(n-1)}{2}$) — число ребер. Затем следует M пар чисел — ребра графа.

Формат выходных данных

Выведите в выходной файл матрицу смежности заданного графа.

Пример

e2m.in	e2m.out
3 3	0 1 1
1 2	1 0 1
2 3	1 1 0
1 3	

2 Задача 7В. Связность [0.5 sec (1 sec)]

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

connect.in	connect.out
3 2 1 2 3 2	YES
3 1 1 3	NO

3 Задача 7C. Дерево [0.5 sec]

Дан неориентированный граф. Проверьте, является ли он деревом.

Формат входных данных

В первой строке входного файла заданы через пробел два целых числа n и m — количество вершин и рёбер в графе, соответственно ($1 \leq n \leq 100$). В следующих m строках заданы рёбра; i -я из этих строк содержит два целых числа u_i и v_i через пробел — номера концов i -го ребра ($1 \leq u_i, v_i \leq n$). Граф не содержит петель и кратных рёбер.

Формат выходных данных

В первой строке выходного файла выведите “YES”, если граф является деревом, и “NO” в противном случае.

Примеры

tree.in	tree.out
3 2 1 2 1 3	YES
3 3 1 2 2 3 3 1	NO

07.Advanced [4/8]

4 Задача 7D. Компоненты связности [0.5 sec (1 sec)]

Вам задан неориентированный граф с N вершинами и M ребрами ($1 \leq N \leq 20\,000$, $1 \leq M \leq 200\,000$). 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 . 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 — номера компонент связности, которым принадлежат соответствующие вершины. Компоненты связности следует занумеровать от 1 до L произвольным образом.

Пример

connect.in	connect.out
4 2	2
1 2	1 1 2 2
3 4	

5 Задача 7Е. Поиск пути на гриде [0.8 sec (1.6 sec)]

Дано прямоугольное поле $W \times H$. Некоторые клетки проходимы, через некоторые ходить нельзя. Из клетки можно ходить в соседние по ребру (слева, справа, сверху, снизу).

Нужно из клетки (x_1, y_1) найти любой (не обязательно кратчайший, даже не обязательно простой) путь в клетку (x_2, y_2) .

Формат входных данных

На первой строке W, H, x_1, y_1, x_2, y_2 ($1 \leq x_1, x_2 \leq W \leq 1000, 1 \leq y_1, y_2 \leq H \leq 1000$). Далее H строк, в каждой из которых по W символов. Символ “.” означает, что клетка проходимая, а символ “*” означает, что по ней ходить нельзя.

Клетки (x_1, y_1) и (x_2, y_2) не совпадают и обе проходимы.

Формат выходных данных

Если пути не существует, выведите NO.

Иначе выведите YES и последовательность клеток (x_i, y_i) , в которой первая совпадает с клеткой (x_1, y_1) , а последняя с клеткой (x_2, y_2) .

Пример

dfsongrid.in	dfsongrid.out
4 2 1 1 4 2	YES 1 1 2 1 3 1 4 1 3 1 3 2 4 2
4 2 1 1 4 2 ..*. .*..	NO
4 2 1 1 4 2 ..*. *...	YES 1 1 2 1 2 2 3 2 4 2

6 Задача 7F. TopSort. Топологическая сортировка [0.5 sec (1 sec)]

Дан ориентированный невзвешенный граф. Необходимо его топологически отсортировать.

Формат входных данных

В первой строке входного файла даны два натуральных числа N и M ($1 \leq N \leq 100\,000, M \leq 100\,000$) — количество 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

topsort.in	topsort.out
6 6 1 2 3 2 4 2 2 5 6 5 4 6	4 6 3 1 2 5
3 3 1 2 2 3 3 1	-1

7 Задача 7G. Поиск цикла [0.5 sec (1 sec)]

Дан ориентированный невзвешенный граф. Необходимо определить есть ли в нём циклы, и если есть, то вывести любой из них.

Формат входных данных

В первой строке входного файла находятся два натуральных числа N и M ($1 \leq N \leq 100\,000$, $M \leq 100\,000$) — количество 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

Если в графе нет цикла, то вывести «NO», иначе — «YES» и затем перечислить все вершины в порядке обхода цикла.

Примеры

cycle.in	cycle.out
2 2 1 2 2 1	YES 1 2
2 2 1 2 1 2	NO

8 Задача 7Н. Bridges. Мосты [0.4 sec (0.8 sec)]

Дан неориентированный граф. Требуется найти все мосты в нем.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m — количество вершин и ребер графа соответственно ($n \leq 20\,000$, $m \leq 200\,000$).

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

Формат выходных данных

Первая строка выходного файла должна содержать одно натуральное число b — количество мостов в заданном графе. На следующей строке выведите b целых чисел — номера ребер, которые являются мостами, в возрастающем порядке. Ребра нумеруются с единицы в том порядке, в котором они заданы во входном файле.

Пример

bridges.in	bridges.out
6 7	1
1 2	3
2 3	
3 4	
1 3	
4 5	
4 6	
5 6	

9 Задача 71. Condense 2. Конденсация графа [0.4 sec (0.8 sec)]

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

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

condense2.in	condense2.out
4 4 2 1 3 2 2 3 4 3	2

10 Задача 7J. Unique Topsort [0.5 sec (1 sec)]

Дан ориентированный ациклический граф G . Проверить, что существует единственный топологический порядок вершин графа.

Формат входных данных

Первая строка входных данных содержит число вершин графа n ($1 \leq n \leq 100\,000$) и число ребер графа m ($0 \leq m \leq 100\,000$). Следующие m строк содержат пары чисел от 1 до n , задающие начало и конец соответствующего ребра. Гарантируется, что граф не содержит циклов.

Формат выходных данных

Если топологический порядок единственный, выведите на первой строке YES, а на второй номера вершин в топологическом порядке, иначе выведите NO.

Примеры

unitopsort.in	unitopsort.out
1 0	YES 1
2 1 2 1	YES 2 1
4 2 1 2 4 3	NO

11 Задача 7К. Longpath. Длиннейший путь [0.4 sec (0.8 sec)]

Дан ориентированный граф без циклов. Требуется найти в нем длиннейший путь.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m — количество вершин и дуг графа соответственно. Следующие m строк содержат описания дуг по одной на строке. Ребро номер i описывается двумя натуральными числами b_i и e_i — началом и концом дуги соответственно ($1 \leq b_i, e_i \leq n$).

Входной граф не содержит циклов и петель.

$n \leq 10\,000$, $m \leq 100\,000$.

Формат выходных данных

Первая строка выходного файла должна содержать одно натуральное число — количество дуг в длиннейшем пути.

Пример

longpath.in	longpath.out
5 5 1 2 2 3 3 4 3 5 1 5	3

07.Hard [0/1]

12 Задача 7L. Раскраска в три цвета [1.2 sec (2.4 sec)]

Петя нарисовал на бумаге n кружков и соединил некоторые пары кружков линиями. После этого он раскрасил каждый кружок в один из трех цветов — красный, синий или зеленый.

Теперь Петя хочет изменить их раскраску. А именно — он хочет перекрасить каждый кружок в некоторый другой цвет так, чтобы никакие два кружка одного цвета не были соединены линией. При этом он хочет обязательно перекрасить каждый кружок, а перекрашивать кружок в тот же цвет, в который он был раскрашен исходно, не разрешается.

Помогите Пете решить, в какие цвета следует перекрасить кружки, чтобы выполнялось указанное условие.

Формат входных данных

Первая строка содержит два целых числа n и m — количество кружков и количество линий, которые нарисовал Петя, соответственно ($1 \leq n \leq 1\,000$, $0 \leq m \leq 20\,000$).

Следующая строка содержит n символов из множества $\{\text{'R'}, \text{'G'}, \text{'B'}\}$ — i -й из этих символов означает цвет, в который раскрашен i -й кружок ('R' — красный, 'G' — зеленый, 'B' — синий).

Следующие m строк содержат по два целых числа — пары кружков, соединенных отрезками.

Формат выходных данных

Выведите в выходной файл одну строку, состоящую из n символов из множества $\{\text{'R'}, \text{'G'}, \text{'B'}\}$ — цвета кружков после перекраски. Если решений несколько, выведите любое.

Если решения не существует, выведите в выходной файл слово "Impossible".

Примеры

color.in	color.out
4 5 RRRG 1 3 1 4 3 4 2 4 2 3	BBGR
4 5 RGRR 1 3 1 4 3 4 2 4 2 3	Impossible