

Задача А. Кратчайший путь в невзвешенном графе

Имя входного файла: pathbge1.in
Имя выходного файла: pathbge1.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дан неориентированный невзвешенный граф. Найдите кратчайшее расстояние от первой вершины до всех вершин.

Формат входного файла

В первой строке входного файла два числа: n и m ($2 \leq n \leq 30000, 1 \leq m \leq 400000$), где n — количество вершин графа, а m — количество ребер.

Следующие m строк содержат описание ребер. Каждое ребро задается стартовой вершиной и конечной вершиной. Вершины нумеруются с единицы.

Формат выходного файла

Выведите n чисел — для каждой вершины кратчайшее расстояние до нее.

Пример

pathbge1.in	pathbge1.out
2 1	0 1
2 1	

Задача В. Кратчайший путь

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

Дан ориентированный взвешенный граф. Найдите кратчайшее расстояние от одной заданной вершины до другой.

Формат входного файла

В первой строке входного файла три числа: N , S и F ($1 \leq N \leq 2000, 1 \leq S, F \leq N$), где N — количество вершин графа, S — начальная вершина, а F — конечная. В следующих N строках по N чисел — матрица смежности графа, где -1 означает отсутствие ребра между вершинами, а любое неотрицательное число — присутствие ребра данного веса. На главной диагонали матрицы всегда нули.

Формат выходного файла

Вывести искомое расстояние или -1 , если пути между указанными вершинами не существует.

Пример

pathmgep.in	pathmgep.out
3 1 2 0 -1 2 3 0 -1 -1 4 0	6

Задача С. Кратчайший путь от каждой вершины до каждой

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

Задан ориентированный взвешенный связный граф. Найдите матрицу расстояний между его вершинами.

Формат входного файла

Первая строка входного файла содержит числа n и m — количество вершин и ребер в графе соответственно ($1 \leq n \leq 200$, $0 \leq m \leq 10\,000$). Следующие m строк содержат по три числа — вершины, которые соединяет соответствующее ребро графа и его вес. Веса ребер неотрицательны и не превышают 10^4 .

Формат выходного файла

Выведите в выходной файл n строк по n чисел — для каждой пары вершин выведите расстояние между ними.

Примеры

<code>pathsg.in</code>	<code>pathsg.out</code>
3 3	0 5 7
1 2 5	10 0 2
2 3 2	8 13 0
3 1 8	

Задача D. Кратчайший путь

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

Дан неориентированный взвешенный граф. Найдите кратчайшее расстояние от первой вершины до всех вершин.

Формат входного файла

В первой строке входного файла два числа: n и m ($2 \leq n \leq 30000, 1 \leq m \leq 400000$), где n — количество вершин графа, а m — количество ребер.

Следующие m строк содержат описание ребер. Каждое ребро задается стартовой вершиной, конечной вершиной и весом ребра. Вес каждого ребра — неотрицательное целое число, не превосходящее 10^4 .

Формат выходного файла

Выведите n чисел — для каждой вершины кратчайшее расстояние до нее.

Пример

pathbgep.in	pathbgep.out
4 5 1 2 1 1 3 5 2 4 8 3 4 1 2 3 3	0 1 4 5

Задача Е. Кратчайшие пути и прочее

Имя входного файла: `path.in`
Имя выходного файла: `path.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 megabytes

Дан взвешенный ориентированный граф и вершина s в нем. Требуется для каждой вершины u найти длину кратчайшего пути из s в u .

Формат входного файла

Первая строка входного файла содержит n , m и s — количество вершин, ребер и номер выделенной вершины соответственно ($2 \leq n \leq 2\,000$, $1 \leq m \leq 5\,000$).

Следующие m строк содержат описание ребер. Каждое ребро задается стартовой вершиной, конечной вершиной и весом ребра. Вес каждого ребра — целое число, не превосходящее 10^{15} по модулю. В графе могут быть кратные ребра и петли.

Формат выходного файла

Выведите n строк — для каждой вершины u выведите длину кратчайшего пути из s в u , $*$ если не существует путь из s в u и $-$ если не существует кратчайший путь из s в u .

Пример

path.in	path.out
6 7 1	0
1 2 10	10
2 3 5	-
1 3 100	-
3 5 7	-
5 4 10	*
4 3 -18	
6 1 -1	

Задача F. Цикл отрицательного веса

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

Дан ориентированный взвешенный граф. Определить, есть ли в нем цикл отрицательного веса, и если да, то вывести его.

Формат входного файла

Во входном файле в первой строке число n ($1 \leq n \leq 250$) — количество вершин графа. В следующих n строках находится по n чисел — матрица смежности графа. Все веса ребер не превышают по модулю 10000. Если ребра нет, то соответствующее число равно 10^9 .

Формат выходного файла

В первой строке выходного файла выведите **YES**, если цикл существует или **NO** в противном случае. При его наличии выведите во второй строке количество вершин в искомом цикле (считая одинаковые первую и последнюю) и в третьей строке — вершины, входящие в этот цикл в порядке обхода.

Примеры

<code>negcycle.in</code>	<code>negcycle.out</code>
2	YES
0 -1	3
-1 0	1 2 1

Задача G. Crossroads

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

Ведущий байтландский производитель автомобилей — компания «<CDR Cars>» — планирует строительство нового завода. Завод будет состоять из n цехов. Некоторые цеха планируется соединить дорогами с двусторонним движением, причём любые два цеха планируется соединить не более, чем одной дорогой. Руководство «<CDR Cars>» интересуется, можно ли расположить цеха и дороги таким образом, чтобы любые две дороги пересекались только в случае, если они выходят из одного цеха (и только в точке, соответствующей этому цеху).

Заметим, что дороги не обязаны быть прямолинейными.

Формат входного файла

В первой строке содержится целое число t ($1 \leq t \leq 500$) — количество тестовых примеров. В каждой из следующей строк описано расположение цехов и дорог. Пусть у нас есть n цехов ($1 \leq N \leq 6$). Тогда описание состоит из $\frac{n(n-1)}{2}$ символов '0' или '1', обозначающих наличие (1) или отсутствие (0) дорог между цехами 1 и 2, 1 и 3, 2 и 3, 1 и 4 и так далее (если использовать термины теории графов, то задаётся нижний треугольник матрицы смежности, выписанный слитно построчно).

Формат выходного файла

Для каждого тестового примера выведите «YES», если цеха можно расположить так, чтобы никакие две дороги не пересекались во внутренних точках, и «NO» в противном случае.

Примеры

<code>planaritycheck.in</code>	<code>planaritycheck.out</code>
3	NO
1111111111	NO
000111111011100	YES
111111	