Содержание

Вы не умеете читать/выводить данные, открывать файлы? Воспользуйтесь примерами.

В некоторых задачах большой ввод и вывод. Пользуйтесь быстрым вводом-выводом.

Обратите внимание, что ввод-вывод во всех задачах стандартный.

Задачи расположены в **произвольном порядке**!

Обязательные задачи

Задача A. Флойд [2 sec, 256 mb]

Полный ориентированный взвешенный граф задан матрицей смежности. Постройте матрицу кратчайших путей между его вершинами. Гарантируется, что в графе нет циклов отрицательного веса.

Формат входных данных

В первой строке вводится единственное число N ($1 \leqslant N \leqslant 100$) — количество вершин графа. В следующих N строках по N чисел задается матрица смежности графа (j-ое число в i-ой строке — вес ребра из вершины i в вершину j). Все числа по модулю не превышают 100. На главной диагонали матрицы — всегда нули.

Формат выходных данных

Выведите N строк по N чисел — матрицу расстояний между парами вершин, где j-ое число в i-ой строке равно весу кратчайшего пути из вершины i в j.

Пример

stdin	stdout
4 0 5 9 100 100 0 2 8 100 100 0 7 4 100 100 0	0 5 7 13 12 0 2 8 11 16 0 7 4 9 11 0

Задача B. Расстояние между вершинами [2 sec, 256 mb]

Коль Дейкстрý писать без кучи,
То тайм-лимит ты получишь...
А в совсем крутой задаче
Юзай кучу Фибоначчи!

Спектакль преподавателей
ЛКШ.июль–2007

Дан взвешенный граф. Требуется найти вес минимального пути между двумя вершинами.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m — количество вершин и ребер графа соответственно. Вторая строка входного файла содержит натуральные числа s и t — номера вершин, длину пути между которыми требуется найти ($1 \leqslant s, t \leqslant n$, $s \neq t$).

Следующие m строк содержат описание ребер по одному на строке. Ребро номер i описывается тремя натуральными числами b_i, e_i и w_i — номера концов ребра и его вес соответственно ($1 \leqslant b_i, e_i \leqslant n$, $0 \leqslant w_i \leqslant 100$).

$n \leqslant 100\,000$, $m \leqslant 200\,000$.

Формат выходных данных

Первая строка выходного файла должна содержать одно натуральное число — вес минимального пути между вершинами s и t.

Если путь из s в t не существует, выведите `-1`.

Пример

stdin	stdout
4 4 1 3 1 2 1 3 4 5 3 2 2 4 1 4	3

Обычные задачи

Задача C. Path. Кратчайший путь [2 sec, 64 mb]

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

Формат входных данных

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

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 выходных данных

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

stdin	stdout
6 7 1 1 2 10 2 3 5 1 3 100 3 5 7 5 4 10 4 3 -18 6 1 -1	0 10 - - - *

Задача D. Диаметр графа [1 sec, 32 mb]

Дан связный взвешенный неориентированный граф.

Рассмотрим пару вершин, расстояние между которыми максимально среди всех пар вершин. Расстояние между ними называется *диаметром графа*. *Эксцентриситетом вершины* v называется максимальное расстояние от вершины v до других вершин графа. *Радиусом графа* называется наименьший из эксцентриситетов вершин. Найдите диаметр и радиус графа.

Формат входных данных

В первой строке входного файла единственное число: N ($1 \leqslant N \leqslant 100$) — количество вершин графа. 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; веса рёбер не превышают 1000.

Формат выходных данных

В выходной файл выведите два числа — диаметр и радиус графа.

Пример

stdin	stdout
4 0 -1 1 2 -1 0 -1 5 1 -1 0 4 2 5 4 0	8 5

Задача E. Цикл отрицательного веса [2 sec, 256 mb]

Дан ориентированный граф. Определите, есть ли в нем цикл отрицательного веса, и если да, то выведите его.

Формат входных данных

Во входном файле в первой строке число N ($1 \leqslant N \leqslant 100$) — количество вершин графа. В следующих N строках находится по N чисел — матрица смежности графа. Все веса ребер не превышают по модулю 10 000. Если ребра нет, то соответствующее число равно 100 000.

Формат выходных данных

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 и в третьей строке — вершины входящие в этот цикл в порядке обхода.

Пример

stdin	stdout
2 0 -1 -1 0	YES 2 1 2

Дополнительные задачи

Задача F. Экскурсия [1 sec, 256 mb]

ЛКШата собираются на экскурсию в Кострому. Кострома — красивый старый город, в котором площади соединяются друг с другом короткими улицами, каждую из которых можно пройти не более чем за десять минут. ЛКШата хотят составить интересный маршрут экскурсии. Так как они поедут на автобусах, то маршрут экскурсии должен начинаться и оканчиваться на одной и той же площади. К сожалению, у ЛКШат будет очень мало времени. Поэтому они решили выбрать наиболее короткий кольцевой маршрут, не проходящий ни по какой улице дважды.

Помогите ЛКШатам найти такой маршрут.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m — количество площадей и улиц в Костроме ($1 \leqslant n \leqslant 1\,000$, $1 \leqslant m \leqslant 10\,000$). Площади занумерованы от 1 до n.

Последующие m строк содержат описания улиц. Каждая улица описывается тремя целыми числами — номерами площадей, которые она соединяет, и количеством минут, которые требуются ЛКШатам на то, чтобы пройти по ней (от одной до десяти минут). Между двумя площадями может быть более одной улицы. Улица соединяет две различные площади.

Гарантируется, что в Костроме существует как минимум один кольцевой маршрут.

Формат выходных данных

Первая строка выходного файла должна содержать единственное число — продолжительность минимального маршрута в минутах.

Пример

stdin	stdout
5 6 1 2 1 2 3 10 1 3 1 2 4 1 3 4 1 1 5 1	4

Задача G. Лабиринт знаний [2 sec, 128 mb]

Участникам сборов подарили билеты на аттракцион "Лабиринт знаний". Лабиринт представляет собой N комнат, занумерованных от 1 до N, между некоторыми из которых есть двери. Когда человек проходит через дверь, показатель его знаний изменяется на определенную величину, фиксированную для данной двери. Вход в лабиринт находится в комнате 1, выход — в комнате N. Каждый участник сборов проходит лабиринт ровно один раз и наибрает некоторое количество знаний (при входе в лабиринт этот показатель равен нулю). Ваша задача — показать наилучший результат.

Формат входных данных

Первая строка входного файла содержит целые числа N ($1 \le N \le 2\,000$) — количество комнат и M ($1 \le M \le 10\,000$) — количество дверей. В каждой из следующих M строк содержится описание двери — номера комнат, из которой она ведет и в которую она ведет (через дверь в лабиринте можно ходить только в одну сторону), а также целое число, которое прибавляется к количеству знаний при прохождении через дверь (это число по модулю не превышает 10 000). Двери могут вести из комнаты в нее саму, между двумя комнатами может быть более одной двери.

Формат выходных данных

В выходной файл выведите "`:)`" — если можно пройти лабиринт и получить неограниченно большой запас знаний, "`:(`" — если лабиринт пройти нельзя, и максимальное количество набранных знаний в противном случае.

Пример

stdin	stdout
2 2 1 2 5 1 2 -5	5

Задача H. Кратчайший путь [1 sec, 256 mb]

Надеюсь, все вы умеете искать в ориентированном графе кратчайший путь. В этой задаче вам предлагается свое умение продемонстрировать.

Вам дан ориентированный взвешенный граф. Веса ребер — целые числа от 1000 до 2000. Нужно несколько раз (не более 1000) ответить на следующий запрос: длина кратчайшего пути из некоторой вершины s в некоторую вершину t.

Формат входных данных

На первой строке числа N и M ($1 \leqslant N \leqslant 25\,000$, $0 \leqslant M \leqslant 50\,000$) — количество вершин и ребер нашего графа, соответственно. Вершины нумеруются целыми числами от 1 до N. Далее M строк содержат информацию о ребрах графа. Каждое ребро задается тремя числами — номер начала, номер конца и вес. Все веса — целые числа от 1000 до 2000. В графе могут быть и петли, и кратные ребра. Следующая строка содержит число K ($1 \leqslant K \leqslant 1000$) — количество запросов. В следующих K строках задаются запросы. Каждый запрос описывается двумя числами — из какой вершины, и в какую должен вести путь.

Формат выходных данных

Для каждого запроса выведите на отдельной строке целое число — длину кратчайшего пути. Если кратчайшего пути не существует следует вывести -1.

Пример

stdin	stdout
5 5 1 2 2000 1 3 1000 1 4 1200 2 3 1500 3 4 1500 4 1 5 2 4 3 3 1 2	-1 3000 0 2000

Замечание

Путем в графе называется такая последовательность ребер, что конец i-го совпадает с началом $i+1$-го. Длиной пути называется суммарный вес ребер. Путь является кратчайшим, если его длина минимальна.

Подсказка по решению

Вы можете потолкать Дейкстру или Форд-Беллмана, но есть решение получше.