

Задача Flow-0. Максимальный поток

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

НЛЮ прилетело и написало это условие.

Формат входного файла

В первой строке входного файла содержится два числа: n и m ($2 \leq n \leq 10$, $1 \leq m \leq n * (n - 1)$). Это количество вершин и рёбер в графе, в котором вам требуется найти поток. Далее следуют описания рёбер графа, по одному в каждой строке входного файла. Описание ребра состоит из трёх чисел: a , b , c ($1 \leq a, b \leq n$, $a \neq b$, $1 \leq c \leq 100$). Эти числа означают, что из вершины a в вершину b идёт ребро пропускной способности c . Гарантируется, что в графе нет кратных рёбер.

Формат выходного файла

В единственную строку выходного файла выведите одно число — размер максимального потока из вершины 1 в вершину n .

Пример

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

Задача Flow-2. Максимальный поток

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

Вам задан ориентированный граф G . Каждое ребро имеет некоторую пропускную способность. Найдите максимальный поток между вершинами 1 и n .

Формат входного файла

Первая строка входного файла содержит n и m — число вершин и рёбер в графе ($2 \leq n \leq 500$, $1 \leq m \leq 10\,000$). Последующие строки описывают рёбра. Каждое ребро задается тремя числами: начальная вершина ребра, конечная вершина ребра и пропускная способность ребра. Пропускные способности не превосходят 10^9 .

Формат выходного файла

Выведите величину максимального потока между вершинами 1 и n .

Примеры

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

Задача Taxi. Такси

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

Управлять службой такси — совсем не простое дело. Помимо естественной необходимости централизованного управле-

ния машинами для того, чтобы обслуживать заказы по мере их поступления и как можно быстрее, нужно также планировать поездки для обслуживания тех клиентов, которые сделали заказы заранее.

В вашем распоряжении находится список заказов такси на следующий день. Вам необходимо минимизировать число машин такси, необходимых чтобы выполнить все заказы.

Для простоты будем считать, что план города представляет собой квадратную решетку. Адрес в городе будем обозначать парой целых чисел: x -координатой и y -координатой. Время, необходимое для того, чтобы добраться из точки с адресом (a, b) в точку (c, d) , равно $|a - c| + |b - d|$ минут. Машина такси может выполнить очередной заказ, либо если это первый ее заказ за день, либо она успевает приехать в начальную точку из предыдущей конечной хотя бы за минуту до указанного срока. Обратите внимание, что выполнение некоторых заказов может окончиться после полуночи.

Формат входного файла

В первой строке входного файла записано число заказов M ($0 < M < 500$). Последующие M строк описывают сами заказы, по одному в строке. Про каждый заказ указано время отправления в формате `hh:mm` (в интервале с `00:00` по `23:59`), координаты (a, b) точки отправления и координаты (c, d) точки назначения. Все координаты во входном файле неотрицательные и не превосходят 200. Заказы записаны упорядоченными по времени отправления.

Формат выходного файла

В выходной файл выведите единственное целое число — минимальное количество машин такси, необходимых для обслуживания всех заказов.

Пример

taxi.in	taxi.out
2 08:00 10 11 9 16 08:07 9 16 10 11	1
2 08:00 10 11 9 16 08:06 9 16 10 11	2