

Задача А. Крепкий орешек против пропускных способностей

Входной файл: `thruput.in`
Выходной файл: `thruput.out`
Ограничение по времени: 3 секунды
Ограничение памяти: 64 мегабайта

Не обнаружив детонатора в гигантском холодильнике Мариарти, Уилл не сдался. В его голове уже созрел новый гениальный план — атаковать компьютерную сеть профессора, управляющую кислородной бомбой.

Сеть состоит из N компьютеров. Поскольку профессор Мариарти страдал минимализмом и недостатком финансирования, то эти компьютеры соединялись минимальным количеством каналов передачи информации, которого достаточно, чтобы сохранить ее связность — каналов было $N - 1$. Каждый канал соединяет два компьютера, способен передавать информацию в обе стороны и характеризуется *пропускной способностью*.

При планировании диверсии Уилл выбирает пару компьютеров A и B сети, после чего принимает решение атаковать канал на пути из A в B с максимальной пропускной способностью. К сожалению, во многом из-за неумелых действий участников сборов во время своего предыдущего приключения в гигантском холодильнике, Уилл слишком часто ударялся головой. Поэтому сейчас его умственных способностей уже не хватает для того, чтобы находить канал с максимальной пропускной способностью достаточно быстро. Вам придется обработать запросы Уилла и сообщить ему искомые максимальные пропускные способности.

Учтите, однако, что сеть Мариарти находится в состоянии перманентного ремонта, и пропускные способности ее каналов постоянно меняются. Поэтому вместе с запросами от Уилла вас также информируют об изменениях пропускных способностей.

Описание входного файла

В первой строке входного файла записано число N ($1 \leq N \leq 100000$). В последующих $N - 1$ строках записана информация о каналах, по одной строке на канал. Каждая из этих строк содержит тройку целых чисел U_i V_i T_i , которая означает, что канал соединяет компьютеры с номерами U_i и V_i и имеет начальную пропускную способность T_i . Сеть профессора связна.

Далее в файле следует описания запросов Крепкого орешка и сообщений о изменении пропускных способностей — по одной строке на запрос или сообщение. Строка, содержащая запрос Уилла, имеет вид “**QUERY** A B ” где A и B — номера компьютеров, между которыми супергерой хочет найти канал с максимальной пропускной способностью ($A \neq B$).

Сообщение об изменении пропускной способности имеет вид “**CHANGE** k T ” и означает, что новая пропускная способность канала с номером k равна T . Все пропускные способности сети (как начальные, так и получаемые в результате изменений) — целые положительные числа, не превосходящие 10^6 .

Последовательность запросов и сообщений заканчивается строкой, содержащей фразу **DONE**. Общее число запросов **QUERY** и сообщений **CHANGE** не превышает 100 000.

Описание выходного файла

Для каждого запроса “**QUERY** A B ” выведите в выходной файл строку, содержащую искомую максимальную пропускную способность на пути из A в B .

Пример

thruput.in	thruput.out
4	7
1 3 5	11
3 4 10	
3 2 7	
QUERY 1 2	
CHANGE 1 11	
QUERY 4 1	
DONE	

Задача В. Приятная прогулка

Входной файл: `triples.in`
Выходной файл: `triples.out`
Ограничение по времени: 5 секунд
Ограничение памяти: 64 мегабайта

В деревне Большой Костромич каждый вечер на прогулку выходят все мальчики, все девочки, а также все домашние животные (преимущественно собаки), живущие в деревне. Так оказалось, что в деревне одинаковое количество мальчиков, девочек и собак. При этом каждая девочка хочет обязательно гулять с каким-нибудь мальчиком, а мальчик — с собакой. В результате, все гуляющие жители деревни разбиваются на тройки, в каждой из которых оказывается мальчик, девочка и собака.

Для каждых двух жителей (включая собак) известно, сколько времени в течение прогулки они уделят друг другу. Таким образом, прогулка мальчика, девочки и собаки протекает примерно так: сначала мальчик и собака играют в подвижные игры, не обращая никакого внимания на девочку, потом мальчик устает, и с собакой играет девочка, потом устает собака, и некоторое время мальчик с девочкой просто прогуливаются и болтают друг с другом. Когда им это надоедает, они расходятся. Таким образом, прогулка длится столько, сколько данный мальчик готов играть с данной собакой плюс сколько с данной собакой будет играть данная девочка плюс сколько эти мальчик и девочка готовы общаться друг с другом.

Требуется найти такое разбиение мальчиков, девочек и собак на тройки, чтобы среднее время прогулки в деревне было по возможности больше.

Описание входного файла

В первой строке входного файла записано число N — количество девочек в деревне ($1 \leq N \leq 100$). Далее записана матрица $N \times N$, j -ое число в i -ой строке этой матрицы задает,

сколько времени мальчик номер i готов играть с собакой номер j . Далее записана еще одна матрица $N \times N$, j -ое число в i -ой строке этой матрицы задает, сколько времени собака номер i готова играть с девочкой номер j . Наконец, записана третья матрица $N \times N$, j -ое число в i -ой строке этой матрицы задает, сколько времени девочка номер i будет общаться с мальчиком номер j . Все числа в матрицах — натуральные, не превосходящие 10000.

Описание выходного файла

В выходной файл выведите N троек чисел. Первое число каждой тройки должно задавать номер мальчика, второе — номер собаки, третье — номер девочки.

Примечание

Будут оцениваться программы, находящие решения не с максимальным средним временем прогулки. Однако чем больше будет среднее время прогулки, тем больше баллов вы имеете шанс получить.

Пример

triples.in	triples.out
2	1 1 2
1 2	2 2 1
3 4	
5 6	
7 8	
9 10	
11 12	

Задача С. Королева и король

Входной файл: queen.in
Выходной файл: queen.out
Ограничение по времени: 2 секунды
Ограничение памяти: 16 мегабайт

Сегодня королева и король встретились на шахматной доске $2 \times n$, на которой между некоторыми полями расставлены стенки. Они ходят как соответствующие шахматные фигуры (ферзь и король), но не имеют права вставать под бой друг другу (т.е. фигура не может совершить такой ход, что противник может сходить на клетку, которая будет содержать эту фигуру после ее хода; но можно делать ход, при котором делающая его фигура нападает на фигуру соперника, а фигура соперника не имеет хода на клетку нападающей фигуры). Ваша задача — определить победителя, и число ходов обеих фигур, требуемое для выигрыша при оптимальной игре обеих сторон. Проигравшей считается та сторона, которая не может сделать ход.

Перед началом игры король находится на клетке (1,1), а королева — на (2, n). Король ходит первым.

Фигуры не имеют права ходить через стенки. Диагональные ходы запрещены в случае, если хотя бы одна из возможных стенок, касающихся соответствующего угла, присутствует на поле.

Описание входного файла

В первой строке входного файла содержатся два целых числа n и m ($2 \leq n \leq 50, m \leq 300$). Далее следуют m строк, каждая из которых содержит четыре целых числа x_1, y_1, x_2, y_2 — координаты соседних полей, разделенных стенкой ($1 \leq x_i \leq 2, 1 \leq y_i \leq n$). Стенки бывают только горизонтальными и вертикальными. Одна стенка может встречаться во входном файле более одного раза.

Описание выходного файла

Если выигрывает король, выведите в первой строке текст **king wins**, если же королева — **queen wins**. Во второй строке должно быть выведено минимальное число ходов, требуемое для победы (считаются ходы обеих сторон).

В случае ничьей выходной файл должен состоять из единственной строки **draw**.

Пример

queen.in	queen.out
2 1 1 2 2 2	queen wins 2
2 2 1 2 2 2 2 2 2 1	king wins 1
2 2 1 2 2 2 1 1 2 1	draw