

Содержание

Must have	2
Задача 13А. Путь в случайном графе [1, 256]	2
Задачи здорового человека	3
Задача 13В. Path. Кратчайший путь [1, 256]	3
Задача 13С. Отрицательный цикл [1, 256]	4
Задача 13D. Транзитивное замыкание [1, 256]	5
Для искателей острых ощущений	6
Задача 13Е. Currency Exchange [1, 256]	6
Задача 13F. Потенциал [1, 256]	7

Обратите внимание, входные данные лежат в **стандартном потоке ввода** (он же stdin), вывести ответ нужно в **стандартный поток вывода** (он же stdout).

Обратите внимание на GNU C++ компиляторы с суффиксом inc.

Подни можно пользоваться **дополнительной библиотекой** (optimization.h).

То есть, использовать быстрый ввод-вывод: **пример про числа и строки**.

И быструю аллокацию памяти (ускоряет vector-set-map-весь-STL): **пример**.

Для тех, кто хочет разобраться, как всё это работает.

Короткая версия быстрого ввода-вывода (**тык**) и короткая версия аллокатора (**тык**).

Must have

Задача 13А. Путь в случайном графе [1, 256]

Дан случайный взвешенный ориентированный граф из n вершин, m рёбер. Гарантируется, что в графе нет отрицательных циклов. Ваша задача – найти расстояние от вершины s до вершины t . Что такое случайный граф? Рёбра равномерно случайного веса между равномерно случайными вершинами.

Формат входных данных

На первой строке $n m s t$ ($1 \leq n \leq 100\,000$, $0 \leq m \leq 200\,000$, $1 \leq s, t \leq n$). На следующих m строках тройки чисел $a_i b_i w_i$ ($1 \leq a, b \leq n$, $|w| < 10^9$), рёбра из a_i в b_i веса w_i .

Формат выходных данных

Одно число – длину кратчайшего пути.

Если пути из s в t не существует, выведите число 10^{18} .

Примеры

stdin	stdout
2 0 1 2	1000000000000000000
3 4 1 3 1 2 20 2 3 -15 1 3 10 3 1 -4	5

Подсказка по решению

Напишите Форд-Беллмана с очередью. Код почти такой же, как у bfs.

Задачи здорового человека

Задача 13В. Path. Кратчайший путь [1, 256]

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 ($2 \leq n \leq 2000$, $1 \leq m \leq 6000$).

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

Подсказка по решению

Разберите в Форд-Беллмане все случаи. Подумайте про overflow (числа переполняются выше 10^{18}) и underflow (становятся меньше -10^{18}).

Задача 13С. Отрицательный цикл [1, 256]

Дан взвешенный ориентированный граф. Требуется определить, содержит ли он цикл отрицательного веса. Гарантируется, что все вершины графа достижимы из первой.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m — количество вершин и ребер графа соответственно ($n \leq 1111$, $m \leq 11111$). Следующие m строк содержат описание ребер по одному на строке. Ребро номер i описывается тремя числами b_i , e_i и w_i — номера концов ребра и его вес соответственно ($1 \leq b_i, e_i \leq n$, $-100\,000 \leq w_i \leq 100\,000$). Обратите внимание, что в графе могут быть кратные ребра и петли.

Формат выходных данных

Первая строка выходного файла должна содержать **yes**, если граф содержит цикл отрицательного веса и **no** в противном случае.

Пример

stdin	stdout
4 4 2 1 -4 1 2 1 3 4 2 2 3 3	yes
4 6 2 1 4 1 2 1 3 4 2 2 3 3 1 1 2 1 2 2	no

Подсказка по решению

Напишите Форд-Беллмана ещё раз и получите плюстик =)

Задача 13D. Транзитивное замыкание [1, 256]

Дан ориентированный граф. Найдите его транзитивное замыкание, то есть для каждой пары вершин a, b определите, есть ли путь из a в b .

Формат входных данных

На первой строке число вершин n ($1 \leq n \leq 1000$). Следующие n строк имеют длину n , состоят из нулей и единиц и задают матрицу смежности графа. Единица в i -й строке, j -м столбце обозначает ребро из i в j .

Формат выходных данных

Выведите матрицу смежности транзитивного замыкания данного графа.

Примеры

stdin	stdout
3	011
010	001
001	000
000	

Подсказка по решению

А вот тут пригодится Флойд. Вообще с практик вы эту задачу умеете решать за $\mathcal{O}(nm/w)$ конденсацией + динамикой, но Флойд + `bitset` гораздо более просто в реализации, просто 2 цикла, 1 `if`. Удачи.

Для искателей острых ощущений

Задача 13Е. Currency Exchange [1, 256]

Several currency exchange points are working in our city. Let us suppose that each point specializes in two particular currencies and performs exchange operations only with these currencies. There can be several points specializing in the same pair of currencies. Each point has its own exchange rates, exchange rate of A to B is the quantity of B you get for $1A$. Also each exchange point has some commission, the sum you have to pay for your exchange operation. Commission is always collected in source currency.

For example, if you want to exchange 100 US Dollars into Russian Rubles at the exchange point, where the exchange rate is 29.75, and the commission is 0.39 you will get $(100 - 0.39) \cdot 29.75 = 2963.3975$ RUR.

You surely know that there are N different currencies you can deal with in our city. Let us assign unique integer number from 1 to N to each currency. Then each exchange point can be described with 6 numbers: integer A and B – numbers of currencies it exchanges, and real RAB , CAB , RBA and CBA – exchange rates and commissions when exchanging A to B and B to A respectively.

Nick has some money in currency S and wonders if he can somehow, after some exchange operations, increase his capital. Of course, he wants to have his money in currency S in the end. Help him to answer this difficult question. Nick must always have non-negative sum of money while making his operations.

Формат входных данных

The first line contains four numbers: N – the number of currencies, M – the number of exchange points, S – the number of currency Nick has and V – the quantity of currency units he has. The following M lines contain 6 numbers each – the description of the corresponding exchange point – in specified above order. Numbers are separated by one or more spaces. $1 \leq S \leq N \leq 100$, $1 \leq M \leq 100$, V is real number, $0 \leq V \leq 10^3$.

For each point exchange rates and commissions are real, given with at most two digits after the decimal point, $10^{-2} \leq rate \leq 10^2$, $0 \leq commission \leq 10^2$.

Let us call some sequence of the exchange operations simple if no exchange point is used more than once in this sequence. You may assume that ratio of the numeric values of the sums at the end and at the beginning of any simple sequence of the exchange operations will be less than 10^4 .

Формат выходных данных

If Nick can increase his wealth, output YES, in other case output NO.

Примеры

stdin	stdout
3 2 1 10.0 1 2 1.0 1.0 1.0 1.0 2 3 1.1 1.0 1.1 1.0	NO
3 2 1 20.0 1 2 1.0 1.0 1.0 1.0 2 3 1.1 1.0 1.1 1.0	YES

Подсказка по решению

Разобрана на практике. Сложная версия обменников.

Задача 13F. Потенциал [1, 256]

Дан взвешенный ориентированный граф. Пусть у каждой вершины есть потенциал Φ_i . Тогда к весу каждого ребра прибавляется потенциал начала и вычитается потенциал конца.

Требуется найти такие целые Φ_i , чтобы веса у всех рёбер были одинаковыми.

Формат входных данных

В первой строке задано целое число t — количество тестовых случаев.

В первой строке каждого тестового случая заданы целые числа n и m ($1 \leq n \leq 300\,000$, $0 \leq m \leq 300\,000$) — количество вершин и рёбер в графе. В следующих m строках задано по три целых числа x_i, y_i и w_i ($1 \leq x_i, y_i \leq n$, $-10^9 \leq w_i \leq 10^9$) — начало, конец и вес ребра. Гарантируется, что граф не содержит кратных рёбер и петель.

Также гарантируется, что сумма всех n и m по всем тестовым случаям не превосходит 600 000.

Формат выходных данных

Для каждого тестового случая выведите «YES», если существует целочисленное решение, и «NO» в противном случае.

Если ответ положительный, то в следующей строке выведите n целых чисел — потенциалы вершин. Все выведенные числа должны быть не больше 10^{18} по абсолютной величине. Гарантируется, что если ответ существует, то существует и ответ, удовлетворяющий этому ограничению.

Если возможных ответов несколько, выведите любой из них.

Пример

stdin	stdout
2	YES
5 4	0 -1 1 2 181
1 2 -1	YES
2 3 2	0 0 0 0 -1
3 4 1	
4 5 179	
5 5	
1 2 1	
2 3 1	
3 4 1	
4 5 0	
5 1 2	

Подсказка по решению

На практике вы обсудили потенциалы. Примените их.