

Задача А. Кратчайший и быстрый путь между вершинами

Имя входного файла: `dist.in`
Имя выходного файла: `dist.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дан неориентированный взвешенный граф. Требуется найти минимальный путь между двумя вершинами.

Обратите внимание на файловый ввод/вывод.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m — количества вершин и рёбер графа соответственно ($1 \leq n \leq 100\,000$, $1 \leq m \leq 200\,000$). Вторая строка входного файла содержит натуральные числа s и t — номера вершин, длину пути между которыми требуется найти ($1 \leq s, t \leq n$, $s \neq t$).

Следующие m строк содержат описание рёбер по одному на строке. Ребро номер i описывается тремя натуральными числами b_i , e_i и w_i — номерами концов ребра и его вес соответственно ($1 \leq b_i, e_i \leq n$, $0 \leq w_i \leq 10000$).

Формат выходных данных

Первая строка выходного файла должна содержать одно натуральное число — вес минимального пути между вершинами s и t , или -1 , если такого пути нет. Если путь есть, то вторая строка должна содержать одно целое неотрицательное число k — количество вершин в кратчайшем пути от s до t . В третьей строчке выведите k чисел — сам кратчайший путь. Если кратчайших путей несколько, выведите любой.

Пример

<code>dist.in</code>	<code>dist.out</code>
4 4	3
1 3	3
1 2 1	1 2 3
2 3 2	
3 4 5	
4 1 4	

Задача В. Числа

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Витя хочет придумать новую игру с числами. В этой игре от игроков требуется преобразовывать четырехзначные числа не содержащие нулей при помощи следующего разрешенного набора действий:

- Можно увеличить первую цифру числа на 1, если она не равна 9.
- Можно уменьшить последнюю цифру на 1, если она не равна 1.
- Можно циклически сдвинуть все цифры на одну вправо.
- Можно циклически сдвинуть все цифры на одну влево.

Например, применяя эти правила к числу 1234 можно получить числа 2234, 1233, 4123 и 2341 соответственно. Точные правила игры Витя пока не придумал, но пока его интересует вопрос, как получить из одного числа другое за минимальное количество операций.

Формат входных данных

Во входном файле содержится два различных четырехзначных числа, каждое из которых не содержит нулей.

Формат выходных данных

Программа должна вывести последовательность четырехзначных чисел, не содержащих нулей. Последовательность должна начинаться первым из данных чисел и заканчиваться вторым из данных чисел, каждое последующее число в последовательности должно быть получено из предыдущего числа применением одного из правил. Количество чисел в последовательности должно быть минимально возможным.

Пример

стандартный ввод	стандартный вывод
1234	1234
4321	2234
	3234
	4323
	4322
	4321

Задача С. Найти короткий путь

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Однажды Юрий решил снарядить очередной поход в горы и собрался взять с собой несколько друзей. Горы можно представить как n полян, между которыми есть тропы. Юрий знает тропы двух типов — короткие и длинные. По каждой тропе можно ходить только в одном направлении. Правда, длинные тропы не всегда длиннее коротких.

Для того, чтобы друзья не считали своего проводника скучным человеком, Юрий всегда чередует тропы. Строго говоря, Юрий никогда не ходит по двум тропам одного типа подряд. Юрий рассматривает несколько маршрутов через горы, каждый из которых заканчивается на поляне с номером n , от которой можно добраться до города. Не смотря на длину троп, каждую из них Юрий проходит за одну минуту. Для каждого варианта начальной поляны посчитайте минимальное время прогулки от текущей поляны до поляны с номером n , если Юрий верен своим принципам и никогда не ходит по двум тропам одного типа подряд.

Формат входных данных

В первой строке через пробел заданы два числа n и m — количество полян и троп соответственно ($2 \leq n \leq 5 \cdot 10^5, 1 \leq m \leq 5 \cdot 10^5$).

В следующих m строках записаны три числа: a_i, b_i и $type_i$, означающие, что с поляны a_i на поляну b_i ведет тропа типа $type_i$, где $type_i \in \{1, 2\}$ ($type_i$ означает тип тропы).

Обратите внимание, что тропа может соединять поляну саму с собой, а также между двумя полянами может быть несколько троп.

Формат выходных данных

Выведите через пробел $n - 1$ число — минимальное время прогулки от вершины с номером i до вершины с номером n . Если от какой-то вершины нельзя добраться до вершины с номером n , выполняя принципы Юрия, выведите вместо минимального времени число -1 .

Пример

стандартный ввод	стандартный вывод
6 6 1 2 1 2 3 2 3 1 1 1 6 2 4 3 1 5 1 2	1 3 2 -1 5

Задача D. Кратчайшие пути

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3.5 секунд
Ограничение по памяти: 256 мегабайт

256 мегабайт

Вам дан взвешенный ориентированный граф и вершина s в нём. Для каждой вершины графа u выведите длину кратчайшего пути от вершины s до вершины u .

Формат входных данных

Первая строка входного файла содержит три целых числа n , m , s — количество вершин и рёбер в графе и номер начальной вершины соответственно ($2 \leq n \leq 1\,000$, $1 \leq m \leq 2\,000$).

Следующие m строчек описывают рёбра графа. Каждое ребро задаётся тремя числами — начальной вершиной, конечной вершиной и весом ребра соответственно. Вес ребра — целое число, не превосходящее 10^{15} по абсолютной величине. В графе могут быть кратные рёбра и петли.

Формат выходных данных

Выведите n строчек — для каждой вершины u выведите длину кратчайшего пути из s в u . Если не существует пути между s и u , выведите «*». Если не существует кратчайшего пути между s и u , выведите «-».

Пример

стандартный ввод	стандартный вывод
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 Е. Полнейший Флойд

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Вам дан **ориентированный** граф. Если в нём есть отрицательные циклы, то выведите хотя бы один из них. В противном случае выведите матрицу попарных кратчайших расстояний между вершинами.

Формат входных данных

На первой строке даны $1 \leq V \leq 500$ — число вершин в графе и $0 \leq E \leq \min(2.5 \times 10^4, |V| \times |V|)$ — число рёбер в графе. Далее записаны рёбра графа, по одному в строчке. Каждое ребро записано в виде трёх целых чисел: $1 \leq from, to \leq |V|, |w| \leq 10^5$.

Формат выходных данных

На первой строке выведите LOOP или NO LOOP в зависимости от наличия отрицательного цикла.

Далее, если в графе есть отрицательный цикл, выведите на второй строке его длину, а на третьей строке выведите его вершины в любом порядке, соблюдая направление рёбер графа. В противном случае выведите V строк по V чисел, где j -й элемент на i -й строке означает кратчайшее расстояние между вершинами i и j и INF означает, что пути между вершинами нет.

Примеры

стандартный ввод	стандартный вывод
3 3 1 2 1 2 3 2 1 3 2	NO LOOP 0 1 2 INF 0 2 INF INF 0
3 3 1 2 -1 2 3 -1 3 1 -1	LOOP 3 3 1 2
7 7 2 1 -4 2 7 -17 3 1 -14 4 5 16 4 5 -19 4 5 14 7 6 -15	NO LOOP 0 INF INF INF INF INF INF -4 0 INF INF INF -32 -17 -14 INF 0 INF INF INF INF INF INF INF 0 -19 INF INF INF INF INF INF 0 INF INF INF INF INF INF INF 0 INF INF INF INF INF INF -15 0

Задача F. Пифагоров экспресс (для C++)

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2.5 секунд
Ограничение по памяти: 256 мегабайт

В стране Флатландии есть n городов, расположенных в целочисленных точках плоскости. Транспортная система Флатландии настолько развита, что между любыми двумя городами ходит экспресс имени Пифагора. С помощью него можно добраться от города с координатами x_1, y_1 до города с координатами x_2, y_2 за время $(x_1 - x_2)^2 + (y_1 - y_2)^2$. По каждой линии ходит достаточно поездов, и временем на пересадки можно пренебречь.

Сообщение с внешним миром во Флатландии продумано несколько хуже, и единственный аэропорт международного сообщения находится в городе с номером s . Вам же хочется попасть в город с номером t . Определите, за какое наименьшее время это можно сделать.

Формат входных данных

В первой строке задано число городов n ($1 \leq n \leq 2 \cdot 10^4$). В следующих n строках заданы координаты городов x_i, y_i ($|x_i|, |y_i| \leq 10^4$). В последней строке даны s и t — номера начального и конечного города в пути ($1 \leq s, t \leq n$).

Формат выходных данных

Выведите одно число — минимальное время, за которое можно добраться из s в t .

Пример

стандартный ввод	стандартный вывод
3 0 1 0 0 1 3 2 3	6

Задача G. Олимпиада по алхимии

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В государстве алхимиков есть N населённых пунктов, пронумерованных числами от 1 до N , и M дорог. Населённые пункты бывают двух типов: деревни и города. Кроме того, в государстве есть одна столица (она может располагаться как в городе, так и в деревне). Каждая дорога соединяет два населённых пункта, и для проезда по ней требуется T_i минут. В столице было решено провести 1-ю государственную командную олимпиаду по алхимии. Для этого во все города из столицы были отправлены гонцы (по одному гонцу на город) с информацией про олимпиаду.

Напишите программу, которая посчитает, в каком порядке и через какое время каждый из гонцов доберётся до своего города. Считается, что гонец во время пути не спит и нигде не задерживается.

Формат входных данных

Во входном файле сначала записаны 3 числа N , M , K — количество населённых пунктов, количество дорог и количество городов ($2 \leq N \leq 1000$, $1 \leq M \leq 10000$, $1 \leq K \leq N$). Далее записан номер столицы C ($1 \leq C \leq N$).

Следующие K чисел задают номера городов.

Далее следуют M троек чисел S_i , E_i , T_i , описывающих дороги: S_i и E_i — номера населённых пунктов, которые соединяет данная дорога, а T_i — время для проезда по ней ($1 \leq T_i \leq 100$).

Гарантируется, что до каждого города из столицы можно добраться по дорогам (возможно, через другие населённые пункты).

Формат выходных данных

Выведите в выходной файл K пар чисел: для каждого города должен быть выведен его номер и минимальное время, когда гонец может в нем оказаться (время измеряется в минутах с того момента, как гонцы выехали из столицы). Пары в выходном файле должны быть упорядочены по времени прибытия гонца.

Примеры

стандартный ввод	стандартный вывод
5 4 5 1 1 2 3 4 5 1 2 1 2 3 10 3 4 100 4 5 100	1 0 2 1 3 11 4 111 5 211
5 5 3 1 2 4 5 2 1 1 2 3 10 3 4 100 4 5 100 1 5 1	2 1 5 1 4 101

Задача Н. Грандиозное ограбление

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Датч ван дер Линде планирует ограбить нефтеперерабатывающую станцию. Такое большое дело надо тщательно спланировать, поэтому он раздобыл карту.

На карте изображены n городов, соединённых m **односторонними** дорогами. Лагерь вашей банды находится в городе 1, нефтеперерабатывающая станция находится в городе n . Каждая дорога имеет уровень *опасности*. *Опасностью* маршрута называется сумма *опасностей* дорог в нём.

Для ограбления вам нужно составить маршрут от лагеря до нефтеперерабатывающей станции. Гарантируется, что это можно сделать.

Чтобы всё прошло гладко, Датчу необходимо узнать:

1. **Минимальную** возможную *опасность* маршрута;
2. Количество маршрутов с **минимальной** *опасностью* (по модулю $10^9 + 7$);
3. **Минимальное** количество дорог в маршруте с **минимальной** *опасностью*;
4. **Максимальное** количество дорог в маршруте с **минимальной** *опасностью*.

Помогите ему с планированием вашего последнего дела.

Формат входных данных

В первой строке даны целые числа n, m ($1 \leq n \leq 10^5, 1 \leq m \leq 2 \cdot 10^5$) — количество городов и количество дорог между ними.

В следующих m строках описаны дороги. Каждая строка содержит целые числа v, u, w ($1 \leq v, u \leq n, 1 \leq w \leq 10^9$) — города, соединяемые дорогой и *опасность* проезда по ней.

Формат выходных данных

Выведите через пробел четыре значения, интересующие Датча.

Пример

стандартный ввод	стандартный вывод
4 5 1 4 5 1 2 4 2 4 5 1 3 2 3 4 3	5 2 1 2

Замечание



Задача I. Пятница, тринадцатое

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Девочка Маша очень суеверная, поэтому, когда наступает пятница, тринадцатое, она начинает вести себя очень неадекватно. К сожалению, ей нужно ходить в школу, и ей приходится ездить на автобусах. В городе есть N автобусных остановок и M автобусных маршрутов. Если в пятницу, тринадцатое она проедет от какой-то остановки до какой-то другой (возможно, используя несколько маршрутов) за время T , причём T делится на 13, то она начинает истошно орать и бегать по автобусу. Помогите ей добраться до школы и остаться в здравом уме.

Формат входных данных

В первой строке входного файла задано одно число T ($1 \leq T \leq 10$) — число тестов. В первой строке теста заданы два числа N и M ($1 \leq N \leq 50$, $1 \leq M \leq 2500$) — число остановок и маршрутов соответственно. Следующие M строк описывают маршруты в формате From To Time ($1 \leq \text{From}, \text{To} \leq N$, $1 \leq \text{Time} \leq 100$) — откуда и куда едет автобус и время в пути. На последней строчке теста будет указано, является ли сегодняшний день пятницей, тринадцатым (True) или нет (False).

Формат выходных данных

Для каждого теста выведите на отдельной строчке минимальное время, за которое Маша сможет доехать от дома (остановка 1) до школы (остановка N), или -1 , если она этого сделать не сможет.

Пример

стандартный ввод	стандартный вывод
3	16
5 5	-1
1 2 1	42
1 3 2	
2 4 1	
3 4 3	
4 5 11	
True	
2 1	
1 2 26	
True	
3 3	
1 1 7	
1 2 26	
2 3 16	
False	

Задача J. Заправки

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

В стране N городов, некоторые из которых соединены между собой дорогами. Для того, чтобы проехать по одной дороге, требуется один бак бензина. В каждом городе бак бензина имеет разную стоимость. Вам требуется добраться из первого города в N -ый, потратив как можно меньше денег. Покупать бензин впрок нельзя.

Формат входных данных

В первой строке вводится число N ($1 \leq N \leq 100$), в следующей строке идет N чисел, i -ое из которых задает стоимость бензина в i -ом городе (всё это целые числа из диапазона от 0 до 100).

Затем идет число M — количество дорог в стране, далее идет описание самих дорог. Каждая дорога задается двумя числами — номерами городов, которые она соединяет. Все дороги двухсторонние (то есть по ним можно ездить как в одну, так и в другую сторону), между двумя городами всегда существует не более одной дороги, не существует дорог, ведущих из города в себя.

Формат выходных данных

Требуется вывести одно число — суммарную стоимость маршрута или -1 , если добраться невозможно.

Примеры

стандартный ввод	стандартный вывод
5 3 6 1 7 6 8 1 2 5 4 5 1 3 4 5 2 2 4 2 3 3 1	3
5 3 7 2 9 4 4 1 2 1 3 2 3 4 5	-1

Задача К. Заповедник

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд
Ограничение по памяти:	1024 мегабайта

Заповедник состоит из n островов, заросших лесом, и m канатных дорог с односторонним движением, их соединяющих. На k островах растут редкие деревья, вызывающие интерес у экотуристов. Перемещаться между островами можно только по канатным дорогам. Для каждой канатной дороги известна её длина.

Всего планируется q маршрутов для туристов. i -й маршрут начинается на острове s_i , заканчивается на острове t_i и проходит через p_i островов с редкими деревьями $v_1 \dots v_{p_i}$. Маршрут может проходить требуемые острова в любом порядке, в том числе посещать какой-то из них неоднократно и при этом может проходить через любые другие острова (включая и другие острова с редкими деревьями).

Для каждого запланированного маршрута определите, возможен ли он, и, если возможен, вычислите длину кратчайшего пути на этом маршруте.

Формат входных данных

Первая строка входных данных содержит три целых числа n, m, k ($1 \leq n \leq 10^5, n-1 \leq m \leq 2 \cdot 10^5, 1 \leq k \leq \min(n, 50)$) — количество островов, количество канатных дорог и количество островов с редкими деревьями, соответственно.

Вторая строка содержит k попарно различных целых чисел r_i ($1 \leq r_i \leq n$) — острова с редкими деревьями.

Каждая из последующих m строк содержит по три целых числа a_i, b_i и l_i ($1 \leq a_i \leq n, 1 \leq b_i \leq n, 1 \leq l_i \leq 10^9, a_i \neq b_i$) и описывает канатную дорогу из a_i в b_i длиной l_i .

Следующая строка содержит одно целое число q ($1 \leq q \leq 5 \cdot 10^3$) — количество маршрутов. Каждая из последующих q строк задаёт один маршрут и начинается с двух целых чисел s_i и t_i ($1 \leq s_i, t_i \leq n$) — начальной и конечной точек маршрута, затем идёт целое число p_i — количество островов с редкими деревьями на маршруте ($1 \leq p_i \leq \min(k, 10)$), далее идёт p_i попарно различных целых чисел v_i — номера островов с редкими деревьями. Гарантируется, что каждое из v_i присутствует во второй строке (то есть совпадает с каким-то из r_i).

Формат выходных данных

Для каждого запроса в новой строке выведите единственное целое число — минимальную длину маршрута или -1 , если такого маршрута не существует.

Пример

стандартный ввод	стандартный вывод
8 7 7	100515
1 2 8 3 6 4 5	-1
1 7 100500	
7 2 9	
2 3 11	
3 2 8	
7 2 4	
7 5 3	
5 6 1	
2	
1 3 2 1 2	
1 8 1 8	

Задача L. Настя и неожиданный гость

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Если девушка не идет к Денису, то Денис идет к девушке. Руководствуясь таким принципом, молодой человек вышел из дома, купил цветы и направился к Насте.

По пути от его дома до дома его возлюбленной есть дорога из n полос, перейти которую за один зеленый свет не всегда возможно. Предвидя это, добрый мэр распорядился поставить в некоторых точках этой дороги островки безопасности. Каждый островок расположен после какой-то полосы, а также в начале и в конце дороги. На них пешеходы могут отдохнуть, набраться сил и дожждаться зеленого света.

Денис подошел к левому краю дороги ровно в тот момент, когда загорелся зеленый свет. Мальчик знает, что светофор сначала горит g секунд зеленым, а потом r секунд красным, потом опять g секунд зеленым и так далее.

Формально, дорогу можно представить как отрезок $[0, n]$. Изначально Денис стоит в точке 0. Его задача - попасть в точку n за минимальное возможное время.

Он знает множество различных целых чисел d_1, d_2, \dots, d_m , где $0 \leq d_i \leq n$ — координаты точек, в которых расположены островки безопасности. Только в одной из этих точек мальчик может находиться в момент, когда горит красный свет.

К сожалению, из-за волнения Денис не всегда может контролировать себя, поэтому сейчас на его передвижения наложены некоторые ограничения:

- Он обязан двигаться всегда, пока горит зеленый, ведь сложно стоять, когда тебя ждет такая девушка. Денис за 1 секунду может изменить свое положение на ± 1 . При этом он должен всегда оставаться внутри отрезка $[0, n]$.
- Он может менять направление движения только на островках безопасности (ведь это безопасно). Это означает, что если за предыдущую секунду мальчик изменил свое положение на $+1$, то если он находится на островке безопасности, он может изменить свое положение на ± 1 , иначе он может изменить свое положение вновь только на $+1$. Аналогично, если за предыдущую секунду он изменил свое положение на -1 , на островке безопасности он может изменить свое положение на ± 1 , а в любой другой точке вновь только на -1 .
- В момент, когда загорается красный, мальчик должен оказаться на каком-то островке безопасности. Он сможет продолжить движение в любом направлении, когда вновь загорится зеленый.

Считается, что Денис перешел дорогу как только его координата станет равна n .

Эта задача оказалась не так проста, ведь возможно, таким способом даже невозможно перейти дорогу. Так как у Дениса все мысли о любви, он не справился решить ее и попросил нас помочь ему в этом. Найдите минимальное возможное время, за которое он сможет перейти дорогу по таким правилам, либо установите, что это сделать невозможно.

Формат входных данных

В первой строке находится два целых числа n и m ($1 \leq n \leq 10^6, 2 \leq m \leq \min(n + 1, 10^4)$) — ширина дороги и количество островков безопасности.

Во второй строке находится m различных целых чисел d_1, d_2, \dots, d_m ($0 \leq d_i \leq n$) — точки, в которых расположены островки безопасности. Гарантируется, что среди них есть 0 и n .

В третьей строке находится два целых числа g, r ($1 \leq g, r \leq 1000$) — время, которое на светофоре горит зеленый свет и время, которое на светофоре горит красный свет.

Формат выходных данных

Выведите одно целое число — минимальное время, за которое Денис может перейти дорогу по всем правилам.

Если перейти дорогу по всем правилам невозможно выведите -1 .

Примеры

стандартный ввод	стандартный вывод
15 5 0 3 7 14 15 11 11	45
13 4 0 3 7 13 9 9	-1

Замечание

В первом тесте оптимальный маршрут такой:

- за первый зеленый свет дойти до 7 и вернуться на 3. В этом случае, мы сменим направление движения в точке 7, что разрешено, поскольку в этой точке есть островок безопасности. В конце мы окажемся в точке 3, где также есть островок безопасности. Следующие 11 секунд мы должны подождать красный свет.
- за второй зеленый свет дойти до 14. Снова подождать красный свет.
- за 1 секунду перейти в 15. В итоге Денис оказывается в конце дороги.

Всего получается 45 секунд.

Во втором тесте невозможно перейти дорогу по всем правилам.