

Задача А. Кратчайший путь

Имя входного файла: mindist2.in
Имя выходного файла: mindist2.out
Ограничение по времени: 3 секунды
Ограничение по памяти: 64 мегабайта

Вам дан неориентированный граф. Найдите кратчайший путь от вершины a до вершины b .

Формат входных данных

В первой строке входного файла идут целые числа n и m ($1 \leq n \leq 50\,000$, $1 \leq m \leq 100\,000$) — количества вершин и рёбер соответственно. Во второй строке идут целые числа a и b — стартовая и конечная вершины соответственно. Далее идут m строк, описывающих рёбра.

Формат выходных данных

Если пути между a и b нет, выведите единственное число -1 . Иначе выведите в первой строке число l — длину кратчайшего пути между этими двумя вершинами в рёбрах, а во второй строке выведите $l + 1$ число — вершины этого пути.

Примеры

mindist2.in	mindist2.out
4 5 1 4 1 3 3 2 2 4 2 1 2 3	2 1 2 4
4 4 2 3 2 1 2 4 4 3 1 3	2 2 1 3

Задача В. Числа

Имя входного файла: `numbers.in`
Имя выходного файла: `numbers.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

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:

1. Можно увеличить первую цифру числа на 1, если она не равна 9.
2. Можно уменьшить последнюю цифру на 1, если она не равна 1.
3. Можно циклически сдвинуть все цифры на одну вправо.
4. 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.

Примеры

<code>numbers.in</code>	<code>numbers.out</code>
9876	9876
8876	8769
	8768
	8876

Задача С. Кратчайший путь

Имя входного файла: `distance.in`
Имя выходного файла: `distance.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 64 мегабайта

Дан неориентированный взвешенный граф.

Найти кратчайший путь между двумя данными вершинами.

Формат входных данных

Первая строка входного файла содержит натуральные числа N и M ($1 \leq N \leq 2000$, $1 \leq M \leq 100\,000$) — количество вершин и ребер графа. Вторая строка входного файла содержит натуральные числа S и F ($1 \leq S, F \leq N$, $S \neq F$) — номера вершин, длину пути между которыми требуется найти. Следующие M строк по три натуральных числа b_i , e_i и w_i — номера концов i -го ребра и его вес соответственно ($1 \leq b_i, e_i \leq n$, $0 \leq w_i \leq 100\,000$).

Формат выходных данных

Первая строка должна содержать одно натуральное число — длина минимального пути между вершинами S и F . Во второй строке через пробел выведите вершины на кратчайшем пути из S в F в порядке обхода. Если путь из S в F не существует, выведите -1 .

Примеры

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

Задача D. Кратчайший путь между вершинами

Имя входного файла: `dist.in`
 Имя выходного файла: `dist.out`
 Ограничение по времени: 4 секунды
 Ограничение по памяти: 256 мегабайт

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

Спектакль преподавателей ЛКШ.июль-2007

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

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 ($1 \leq b_i, e_i \leq n$, $0 \leq w_i \leq 10\,000$).

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.

Примеры

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

Задача E. Флойд

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

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 ($1 \leq N \leq 100$) — количество вершин графа. В следующих N строках по N чисел задается матрица смежности графа (j -е число в i -й 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 -е число в i -й строке равно весу кратчайшего пути из вершины i в j .

Примеры

floyd.in	floyd.out
4	0 5 7 13
0 5 9 100	12 0 2 8
100 0 2 8	11 16 0 7
100 100 0 7	4 9 11 0
4 100 100 0	

Задача F. Цикл отрицательного веса

Имя входного файла: `negcycle.in`
Имя выходного файла: `negcycle.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайта

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 ($1 \leq N \leq 100$) — 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.

Примеры

<code>negcycle.in</code>	<code>negcycle.out</code>
2	YES
0 -1	2
-1 0	2 1

Задача G. Заправки

Имя входного файла:	gasstation.in
Имя выходного файла:	gasstation.out
Ограничение по времени:	2 секунды
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 -й, потратив как можно меньшее количество денег.

Дополнительно имеется канистра для бензина, куда входит столько же бензина, сколько входит в бак. В каждом городе можно заправить бак, залить бензин в канистру, залить и туда и туда, или же перелить бензин из канистры в бак.

Формат входных данных

В первой строке дано число N ($1 \leq N \leq 100$). Во второй строке входного файла записано N чисел, i -е из которых задает стоимость бензина в i -м городе (все числа целые в диапазоне от 0 до 100).

Во следующих строках описаны все дороги (по одной в строке). Каждая дорога задается двумя числами — номерами городов, которые она соединяет. Все дороги двухсторонние, между двумя городами существует не более одной дороги, не существует дорог, ведущих из города в себя.

Формат выходных данных

В выходной файл выведите одно число — суммарную стоимость маршрута или -1 , если добраться до нужного города невозможно.

Примеры

gasstation.in	gasstation.out
4 1 10 2 15 1 2 1 3 4 2 4 3	2

Задача Н. Егор и граф

Имя входного файла:	<code>stdin</code>
Имя выходного файла:	<code>stdout</code>
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

У Егора есть взвешенный ориентированный граф, состоящий из n вершин. В этом графе между любой парой различных вершин есть ребро в обоих направлениях. Егор любит играть с графом, и сейчас он придумал новую игру:

- Игра состоит из n шагов.
- На i -м шаге Егор удаляет из графа вершину номер x_i . Удаляя вершину, Егор удаляет все ребра, которые входили в данную вершину и которые выходили из нее.
- Перед выполнением каждого шага, Егор хочет знать сумму длин кратчайших путей между всеми парами оставшихся вершин. Кратчайший путь может проходить через любую оставшуюся вершину. Другими словами, если обозначить как $d(i, v, u)$ кратчайший путь между вершинами v и u в графе, который получился до удаления вершины x_i , то Егор хочет знать значение следующей суммы: $\sum_{v, u, v \neq u} d(i, v, u)$.

Помогите Егору, выведите значение искомой суммы перед каждым шагом.

Формат входных данных

В первой строке содержится целое число n ($1 \leq n \leq 500$) — количество вершин в графе.

В следующих n строках содержится по n целых чисел — матрица смежности графа: j -е число в i -й строке a_{ij} ($1 \leq a_{ij} \leq 10^5$, $a_{ii} = 0$) обозначает вес ребра, ведущего из вершины i в вершину j .

В следующей строке содержится n различных целых чисел: x_1, x_2, \dots, x_n ($1 \leq x_i \leq n$) — вершины, которые удаляет Егор.

Формат выходных данных

Выведите n целых чисел — i -е число равно искомой сумме перед i -м шагом.

Примеры

<code>stdin</code>	<code>stdout</code>
2 0 5 4 0 1 2	9 0

Задача I. Кратчайшие пути

Имя входного файла: `path.in`
Имя выходного файла: `path.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 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 ($2 \leq n \leq 2\,000$, $1 \leq m \leq 5\,000$).

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 «-».

Примеры

<code>path.in</code>	<code>path.out</code>
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 J. Авиаперелёты-2

Имя входного файла: avia2.in
 Имя выходного файла: avia2.out
 Ограничение по времени: 1 секунда
 Ограничение по памяти: 256 мегабайт

Профессору Форду необходимо попасть на международную конференцию. Он хочет потратить на дорогу наименьшее количество денег, поэтому решил, что будет путешествовать исключительно ночными авиарейсами (чтобы не тратиться на ночевку в отелях), а днем будет осматривать достопримечательности тех городов, через которые он будет проезжать транзитом. Он внимательно изучил расписание авиаперелетов и выбрал подходящие ему авиарейсы, выяснив, что перелеты на выбранных направлениях совершаются каждую ночь и за одну ночь он не сможет совершить два перелета.

Теперь профессор хочет найти путь наименьшей стоимости, учитывая что до конференции осталось K ночей (то есть профессор может совершить не более K перелетов).

Формат входных данных

В первой строке находятся числа N (количество городов), M (количество авиарейсов), K (количество оставшихся ночей), S (номер города, в котором живет профессор), F (номер города, в котором проводится конференция). Ограничения: $2 \leq N \leq 100$, $1 \leq M \leq 10^5$, $1 \leq K \leq 100$, $1 \leq S \leq N$, $1 \leq F \leq N$.

Далее идет M строк, задающих расписание авиарейсов. i -я строка содержит три натуральных числа: S_i , F_i и P_i , где S_i — номер города, из которого вылетает i -й рейс, F_i — номер города, в который прилетает i -й рейс, P_i — стоимость перелета i -м рейсом. $1 \leq S_i \leq N$, $1 \leq F_i \leq N$, $1 \leq P_i \leq 10^6$.

Формат выходных данных

Выведите одно число — минимальную стоимость пути, подходящего для профессора. Если профессор не сможет за K ночей добраться до конференции, выведите число -1.

Примеры

avia2.in	avia2.out
4 5 2 1 4 1 2 1 2 3 1 3 4 1 1 3 3 1 4 5	4