

Задача А. Кратчайший путь в невзвешенном графе

Имя входного файла: pathbge1.in
Имя выходного файла: pathbge1.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дан неориентированный невзвешенный граф. Найдите кратчайшее расстояние от первой вершины до всех вершин.

Формат входных данных

В первой строке входного файла два числа: n и m ($2 \leq n \leq 30\,000, 1 \leq m \leq 400\,000$), где n — количество вершин графа, а m — количество ребер.

Следующие m строк содержат описание ребер. Каждое ребро задается стартовой вершиной и конечной вершиной. Вершины нумеруются с единицы.

Формат выходных данных

Выведите n чисел — для каждой вершины кратчайшее расстояние до нее.

Примеры

pathbge1.in	pathbge1.out
2 1 2 1	0 1
3 2 1 3 3 2	0 2 1

Задача В. 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 С. Путь конём

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

64 мегабайта

На шахматной доске 8×8 указаны две различные клетки. Найдите кратчайший маршрут коня из первой клетки во вторую.

Формат входных данных

Во входном файле записаны координаты двух клеток. Каждая координата представлена двумя символами, где сначала указана одна строчная буква от `a` до `h`, а после буквы (без пробела) цифра от 1 до 8, например `h8`. Каждая клетка записана в отдельной строке. Гарантируется, что координаты клеток различны.

Формат выходных данных

Программа должна вывести последовательность клеток, первая из которых совпадает с первой данной, а последняя совпадает со второй данной. Две соседние клетки должны быть соединены ходом коня, при этом количество клеток в последовательности должно быть минимально возможным. Если существует несколько возможных ответов на задачу, разрешается выводить любой.

Пример

<code>knight.in</code>	<code>knight.out</code>
<code>a1</code>	<code>a1</code>
<code>b1</code>	<code>b3</code>
	<code>d2</code>
	<code>b1</code>

Замечание

При решении задачи полезно знать про функции `ord()` и `chr()`:

`ord(c)` - возвращает код символа `c`

`chr(x)` - возвращает символ с кодовым значением `x`

Например: `ord('a') = 97`, `chr(98) = 'b'`

Задача D. Только направо

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

256 мегабайт

Змей Горыныч оказался в лабиринте и хочет выбраться из него как можно скорее. К сожалению, после вчерашнего злоупотребления кефиром левая голова Змея соображает плохо. Поэтому Змей Горыныч никогда не поворачивается налево, а ещё ему после поворота направо нужно пройти минимум один шаг. Помогите Змею Горынычу определить длину кратчайшего пути до выхода из лабиринта.

Формат входных данных

В первой строке через пробел записаны числа r и c ($4 \leq r, c \leq 20$) — количество строк и столбцов в карте лабиринта. В каждой из следующих r строк записано по c символов, задающих эту карту. Символ **S** обозначает положение Змея Горыныча, символ **F** — точку выхода из лабиринта, символ **X** — стенку. Пробелами обозначены проходимые клетки. Гарантируется, что лабиринт окружен стенами. Перед началом движения Змей Горыныч может сориентироваться по любому из 4 направлений (вверх, вниз, влево или направо).

Формат выходных данных

Выведите единственное число — расстояние, которое придется пройти Змею Горынычу. Гарантируется, что он всегда сможет выйти из лабиринта.

Пример

<code>nolefts.in</code>	<code>nolefts.out</code>
10 14 XXXXXXXXXXXXXXXXX X XXX X XFXXXXX X XXX XX XX X X S X XX XXXXXX X X X X X X X X X X X XXX XX X XXXXXXXXXXXXXXXXX	29

Замечание

Путь для теста из условия: $(5, 3) \rightarrow (5, 11) \rightarrow (9, 11) \rightarrow (9, 8) \rightarrow (7, 8) \rightarrow (7, 9) \rightarrow (8, 9) \rightarrow (8, 4) \rightarrow (3, 4)$

Задача Е. Многоэтажки и пожилая черепашка

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

Пожилая черепашка Полина очень любит прямоугольное клетчатое болото Осуждаево, в левом верхнем углу которого обитает уже много-много лет. Но вот незадача: многоэтажная застройка добралась и до её родного дома. Поэтому пожилая Полина решила подать жалобу в местные управляющие болотные органы. Но, к сожалению, здание правительства находится достаточно далеко от её привычного места обитания: в правом нижнем углу.

Долгие путешествия не пугают Полину, но буквально вчера она прочитала объявление, в котором подробно описывался план застройки Осуждаево: там были перечислены клетки болота в порядке их захвата многоэтажками; план составлен так, что каждая многоэтажка занимает ровно одну клетку. Полина — очень целеустремлённая пожилая черепашка, поэтому по пути к зданию правительства она умеет двигаться только либо вниз, либо вправо, совершая шаги, которые только приближают её к своей цели, но она не может находиться в клетке, которая захвачена многоэтажкой.

Помогите Полине узнать, какое максимальное количество многоэтажек может быть построено до того, как она потеряет возможность добраться до здания правительства и подать свою жалобу и осудить застройку Осуждаева.

Формат входных данных

В первой строке даны три целых числа n , m и k ($2 \leq n, m \leq 300$; $1 \leq k \leq n \cdot m - 1$) — размеры болота и количество клеток, которые будут застроены многоэтажками.

В следующих k строках даны по два целых числа x и y ($1 \leq x \leq n$; $1 \leq y \leq m$) — координаты клеток, которые будут последовательно застроены (в том порядке, в котором даны). Гарантируется, что все клетки различны. Также гарантируется, что стартовая клетка так и не будет застроена.

Формат выходных данных

В единственной строке выведите номер многоэтажки, после постройки которой пожилая Полина перестанет иметь возможность подать жалобу в правительство Осуждаево, или -1 , если этого не случится.

Примеры

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

Задача F. Эвакуация

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

Одна из Сверхсекретных организаций, чье название мы не имеем право разглашать, представляет собой сеть из N подземных бункеров, соединенных равными по длине туннелями, по которым из любого бункера можно добраться до любого другого (не обязательно напрямую). Связь с внешним миром осуществляется через специальные засекреченные выходы, которые расположены в некоторых из бункеров. Организации понадобилось составить план эвакуации персонала на случай экстренной ситуации. Для этого для каждого из бункеров необходимо узнать, сколько времени потребуется для того, чтобы добраться до ближайшего из выходов. Вам, как специалисту по таким задачам, поручено рассчитать необходимое время для каждого из бункеров по заданному описанию помещения Сверхсекретной организации. Для вашего же удобства бункеры занумерованы числами от 1 до N .

Формат входных данных

В первой строке записано число N , во второй — число K ($1 \leq N \leq 100\,000$, $1 \leq K \leq N$) — количество бункеров и количество выходов соответственно. Далее через пробел записаны K различных чисел от 1 до N , обозначающих номера бункеров, в которых расположены выходы. Потом идёт целое число M ($1 \leq M \leq 100\,000$) — количество туннелей. Далее вводятся M пар чисел — номера бункеров, соединенных туннелем. По каждому из туннелей можно двигаться в обе стороны. В организации не существует туннелей, ведущих из бункера в самого себя, зато может существовать более одного туннеля между парой бункеров.

Формат выходных данных

В первой строке выведите N чисел, разделённых пробелом — для каждого из бункеров минимальное время, необходимое чтобы добраться до выхода. Считайте, что время перемещения по одному туннелю равно 1. Во второй строке выведите N чисел — для каждого бункера номер ближайшего бункера с выходом, если таких несколько выведите бункер с наименьшим номером.

Пример

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

Задача G. Дейкстра

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

Дан ориентированный взвешенный граф.

Найдите кратчайшее расстояние от одной заданной вершины до другой.

Формат входных данных

В первой строке входного файла три числа: N , S и F ($1 \leq N \leq 2000, 1 \leq S, F \leq N$), где N — количество вершин графа, S — начальная вершина, а F — конечная. В следующих N строках по N чисел — матрица смежности графа, где -1 означает отсутствие ребра между вершинами, а любое целое неотрицательное число, не превосходящее 10 000 — присутствие ребра данного веса. На главной диагонали матрицы всегда нули.

Формат выходных данных

Вывести искомое расстояние или -1 , если пути не существует.

Пример

<code>dijkstra.in</code>	<code>dijkstra.out</code>
3 1 2 0 -1 2 3 0 -1 -1 4 0	6

Задача Н. 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 Дейкстру 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.

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 I. Игрушечный лабиринт

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

256 мегабайт

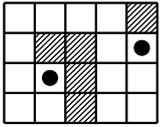
Игрушечный лабиринт представляет собой прозрачную плоскую прямоугольную коробку, внутри которой есть препятствия и перемещается шарик. Коробку можно наклонять влево, вправо, к себе или от себя, после каждого наклона шарик перемещается в заданном направлении до ближайшего препятствия или до стенки лабиринта, после чего останавливается. Целью игры является загнать шарик в одно из специальных отверстий-выходов. Шарик проваливается в отверстие, если оно встречается на его пути.

Первоначально шарик находится в левом верхнем углу лабиринта. Гарантируется, что решение существует и левый верхний угол не занят препятствием или отверстием.

Формат входных данных

В первой строке входного файла записаны числа N и M — размеры лабиринта (целые положительные числа, не превышающие 100). Затем идет N строк по M чисел в каждой — описание лабиринта. Число 0 в описании означает свободное место, число 1 — препятствие, число 2 — отверстие.

Например, лабиринту, изображенному на рисунке, будет соответствовать такое описание:

	<pre>4 5 0 0 0 0 1 0 1 1 0 2 0 2 1 0 0 0 0 1 0 0</pre>
--	--

Формат выходных данных

Выведите единственное число — минимальное количество наклонов, которые необходимо сделать, чтобы шарик покинул лабиринт через одно из отверстий.

Пример

стандартный ввод	стандартный вывод
<pre>4 5 0 0 0 0 1 0 1 1 0 2 0 2 1 0 0 0 0 1 0 0</pre>	3