Графы

Разработать программу на языке C или C++, реализующую указанный алгоритм. Формат входных и выходных данных описан в варианте задания. Первый тест в проверяющей системе совпадает с примером.

Варианты

1. Поиск в глубину

Задан связный неориентированный граф, состоящий из n вершин и m ребер. Вершины пронумерованы целыми числами от 1 до n. Необходимо запустить из вершины с номером start поиск в глубину и вывести номера всех вершин в порядке обхода. Для обеспечения однозначности ответа списки смежности графа следует предварительно отсортировать.

Входные данные

В первой строке заданы $1 \le n \le 105$, $1 \le m \le 105$ и $1 \le start \le n$. В следующих m строках записаны ребра. Каждая строка содержит пару чисел – номера вершин, соединенных ребром.

Выходные данные

Необходимо вывести одну строку, в которой через пробел перечислены номера вершин в порядке обхода.

Пример

Входной файл	Выходной файл
6 5 1 2 1 1 5 4 5 5 3 2 6	1 2 6 5 3 4

2. Поиск в ширину

Задание и формат входных/выходных данных полностью аналогичны варианту 1, с той лишь разницей, что вместо поиска в глубину следует использовать поиск в ширину.

Пример

Входной файл	Выходной файл
6 5 1 2 1 1 5 4 5 5 3 2 6	1 2 5 6 3 4

3. Поиск компонент связности

Задан неориентированный граф, состоящий из n вершин и m ребер. Вершины пронумерованы целыми числами от 1 до n. Необходимо вывести все компоненты связности данного графа.

Входные данные

Формат аналогичен варианту 1, но без числа start.

Выходные данные

Каждую компоненту связности нужно выводить в отдельной строке, в виде списка номеров вершин через пробел. Строки при выводе должны быть отсортированы по минимальному номеру вершины в компоненте, числа в одной строке также должны быть отсортированы.

Пример

Входной файл	Выходной файл
5 4 1 2 2 3 1 3 4 5	1 2 3 4 5

4. Поиск кратчайшего пути между парой вершин алгоритмом Дейкстры

Задан взвешенный неориентированный граф, состоящий из n вершин и m ребер. Вершины пронумерованы целыми числами от 1 до n. Необходимо найти длину кратчайшего пути из вершины с номером start в вершину с номером finish при помощи алгоритма Дейкстры. Длина пути равна сумме весов ребер на этом пути. Граф не содержит петель и кратных ребер.

Входные данные

В первой строке заданы 1 ≤ n ≤ 105, 1 ≤ m ≤ 105, 1 ≤ start ≤ n и 1 ≤ finish ≤ n. В следующих m строках записаны ребра. Каждая строка содержит три числа – номера вершин, соединенных ребром, и вес данного ребра. Вес ребра – целое число от 0 до 109.

Выходные данные

Необходимо вывести одно число – длину кратчайшего пути между указанными вершинами. Если пути между указанными вершинами не существует, следует вывести строку "No solution" (без кавычек).

Пример

Входной файл	Выходной файл
5 6 1 5 1 2 2 1 3 0 3 2 10 4 2 1 3 4 4 4 5 5	8

5. Поиск кратчайшего пути между парой вершин алгоритмом Беллмана-Форда

Задан взвешенный ориентированный граф, состоящий из n вершин и m ребер. Вершины пронумерованы целыми числами от 1 до n. Необходимо найти длину кратчайшего пути из вершины с номером start в вершину с номером finish при помощи алгоритма Беллмана-Форда. Длина пути равна сумме весов ребер на этом пути. Обратите внимание, что в данном варианте веса ребер могут быть отрицательными, поскольку алгоритм умеет с ними работать. Граф не содержит петель, кратных ребер и циклов отрицательного веса.

Входные данные

В первой строке заданы 1 ≤ n ≤ 105, 1 ≤ m ≤ 3*105, 1 ≤ start ≤ n и 1 ≤ finish ≤ n. В следующих m строках записаны ребра. Каждая строка содержит три числа – номера вершин, соединенных ребром, и вес данного ребра. Вес ребра – целое число от -109 до 109.

Выходные данные

Необходимо вывести одно число – длину кратчайшего пути между

указанными вершинами. Если пути между указанными вершинами не существует, следует вывести строку "No solution" (без кавычек).

Пример

Входной файл	Выходной файл
5 6 1 5 1 2 2 1 3 0 3 2 -1 2 4 1 3 4 4 4 5 5	5

6. Поиск кратчайших путей между всеми парами вершин алгоритмом Джонсона

Задан взвешенный ориентированный граф, состоящий из n вершин и m ребер. Вершины пронумерованы целыми числами от 1 до n. Необходимо найти длины кратчайших путей между всеми парами вершин при помощи алгоритма Джонсона. Длина пути равна сумме весов ребер на этом пути. Обратите внимание, что в данном варианте веса ребер могут быть отрицательными, поскольку алгоритм умеет с ними работать. Граф не содержит петель и кратных ребер.

Входные данные

В первой строке заданы $1 \le n \le 2000$ и $1 \le m \le 4000$. В следующих m строках записаны ребра. Каждая строка содержит три числа – номера вершин, соединенных ребром, и вес данного ребра. Вес ребра – целое число от -109 до 109.

Выходные данные

Если граф содержит цикл отрицательного веса, следует вывести строку "Negative cycle" (без кавычек). В противном случае следует вывести матрицу из n строк и n столбцов, где j-е число в i-й строке равно длине кратчайшего пути из вершины i в вершину j. Если такого пути не существует, на соответствующей позиции должно стоять слово "inf" (без кавычек). Элементы матрицы в одной строке разделяются пробелом.

Пример

Входной файл	Выходной файл

5 4 1 2 -1 2 3 2 1 4 -5 3 1 1	0 -1 1 -5 inf 3 0 2 -2 inf 1 0 0 -4 inf inf inf inf 0 inf inf inf inf inf 0

7. Поиск максимального потока алгоритмом Форда-Фалкерсона

Задан взвешенный ориентированный граф, состоящий из n вершин и m ребер. Вершины пронумерованы целыми числами от 1 до n. Необходимо найти величину максимального потока в графе при помощи алгоритма Форда-Фалкерсона. Для достижения приемлемой производительности в алгоритме рекомендуется использовать поиск в ширину, а не в глубину. Истоком является вершина с номером 1, стоком – вершина с номером n. Вес ребра равен его пропускной способности. Граф не содержит петель и кратных ребер.

Входные данные

В первой строке заданы $1 \le n \le 2000$ и $1 \le m \le 10000$. В следующих m строках записаны ребра. Каждая строка содержит три числа – номера вершин, соединенных ребром, и вес данного ребра. Вес ребра – целое число от 0 до 109.

Выходные данные

Необходимо вывести одно число – искомую величину максимального потока. Если пути из истока в сток не существует, данная величина равна нулю.

Пример

Входной файл	Выходной файл
5 6 1 2 4 1 3 3 1 4 1 2 5 3 3 5 3 4 5 10	7

8. Поиск максимального паросочетания алгоритмом Куна

Задан неориентированный двудольный граф, состоящий из n вершин и m

ребер. Вершины пронумерованы целыми числами от 1 до n. Необходимо найти максимальное паросочетание в графе алгоритмом Куна. Для обеспечения однозначности ответа списки смежности графа следует предварительно отсортировать. Граф не содержит петель и кратных ребер.

Входные данные

В первой строке заданы $1 \le n \le 110000$ и $1 \le m \le 40000$. В следующих m строках записаны ребра. Каждая строка содержит пару чисел – номера вершин, соединенных ребром.

Выходные данные

В первой строке следует вывести число ребер в найденном паросочетании. В следующих строках нужно вывести сами ребра, по одному в строке. Каждое ребро представляется парой чисел – номерами соответствующих вершин. Строки должны быть отсортированы по минимальному номеру вершины на ребре. Пары чисел в одной строке также должны быть отсортированы.

Пример

Входной файл	Выходной файл
4 3 1 2 2 3 3 4	2 1 2 3 4