

## Лабораторная работа №7

В рамках данной лабораторной работы необходимо решить предложенные задачи на языке программирования высокого уровня из предложенного перечня. Варианты задач находятся в отдельном файле. В качестве результата выполнения практической работы необходимо приложить **архив с файлами решенных задач, а также отчет о выполнении работы**. Файлы решенных задач представляют собой файлы с исходным кодом с установленным расширением. При решении задач необходимо писать свои собственные структуры данных и методы сортировки, использование встроенных не допускается (кроме массивов).

*Отчет о выполнении практической работы* должен содержать:

1. Титульный лист
2. Содержание
3. 4 параграфа, в которых раскрыто решение каждой задачи. По каждой задаче необходимо представить следующую информацию:

3.1. *Условие задачи*. Берется из файла.

3.2. *Ход решения задачи*. Студент описывает логику решения данной задачи. Какие алгоритмы и для чего использованы, как построена программа. Данная часть является описательной. Здесь следует говорить именно о построении алгоритма, опуская процессы ввода и вывода данных (если это не является основной сутью алгоритма).

3.3. *Листинг программы с комментариями*. Копируется весь программный код.

3.4. *Тестирование программы*. Если для задачи предусмотрены автотесты, приложить скриншот их прохождения. Если нет: Составляется таблица, содержащая следующие поля: номер теста, входные данные, результат выполнения программы, корректность, время выполнения (мс), затраченная память (Мб). Составляется не менее 10-ти тестовых наборов данных согласно условию задачи. Тестовые наборы входных данных студент составляет самостоятельно. В обязательном порядке программа тестируется на граничных наборах входных данных (например, если  $N$  варьируется от 0 до  $10^9$ , то обязательно рассмотреть решение задачи при  $N=0$  и при  $N$  близком к  $10^9$ ). Если написанная программа не позволяет решить задачу при граничных входных данных, все равно включить в тест и в качестве результата написать "Не решено". В столбце "входные данные" данные впечатываются вручную, в столбце "результат..." представляется скриншот выполнения программы (если не влезает на одну страницу, делать несколько скриншотов).

Каждая лабораторная работа защищается на занятии преподавателю.

## Задачи по теме 7. Алгоритмы на графах

### Задача 1. Дорогая сеть

Ограничение по времени: 0.2 с. (Java, C# - 0.7 с., Python – 1.3 с.)

Ограничение по памяти: 64 Мб.

Тимофей решил соединить все компьютеры в своей компании в единую сеть. Для этого он придумал построить минимальное остовное дерево чтобы эффективнее использовать ресурсы.

Но от начальства пришла новость о том, что выделенный на сеть бюджет оказался очень большим и его срочно надо израсходовать. Поэтому Тимофея теперь интересуют не минимальные, а максимальные остовные деревья.

Он поручил вам найти вес такого максимального остовного дерева в неориентированном графе, который задаёт схему офиса.

#### Формат ввода

В первой строке дано количество вершин  $n$  и ребер  $m$  графа ( $1 \leq n \leq 1000, 0 \leq m \leq 100000$ ). В каждой из следующих  $m$  строк заданы рёбра в виде троек чисел  $u, v, w$ .  $u$  и  $v$  – вершины, которые соединяют это ребро.  $w$  – его вес ( $1 \leq u, v \leq n, 0 \leq w \leq 10000$ ). В графе могут быть петли и кратные рёбра. Граф может оказаться несвязным.

#### Формат вывода

Если максимальное остовное дерево существует, то выведите его вес. Иначе (если в графе несколько компонент связности) выведите фразу «Oops! I did it again».

#### Примеры

Ввод	Вывод
4 4 1 2 5 1 3 6 2 4 8 3 4 3	19
3 3 1 2 1 1 2 2 2 3 1	3
2 0	Oops! I did it again

### Задача 2. Железные дороги

Ограничение по времени: 2 с.

Ограничение по памяти: 256 мегабайт (Java, C# – 512 Mb, Python – 700 Mb)

В стране  $X$  есть  $n$  городов, которым присвоены номера от 1 до  $n$ . Столица страны имеет номер  $n$ . Между городами проложены железные дороги.

Однако дороги могут быть двух типов по ширине полотна. Любой поезд может ездить только по одному типу полотна. Условно один тип дорог помечают как  $R$ , а другой как  $B$ . То есть если маршрут от одного города до другого имеет как дороги типа  $R$ , так и

дороги типа  $B$ , то ни один поезд по этому маршруту проехать не сможет. **От одного города до другого можно проехать только по маршруту, состоящему исключительно из дорог типа  $R$  или только из дорог типа  $B$ .**

Но это ещё не всё. По дорогам страны  $X$  можно двигаться только от города с меньшим номером к городу с большим номером. Это объясняет большой приток жителей в столицу, у которой номер  $n$ .

Карта железных дорог называется оптимальной, если не существует пары городов  $A$  и  $B$  такой, что от  $A$  до  $B$  можно добраться как по дорогам типа  $R$ , так и по дорогам типа  $B$ . Иными словами, для любой пары городов верно, что от города с меньшим номером до города с большим номером можно добраться по дорогам только какого-то одного типа или же что маршрут построить вообще нельзя. Выясните, является ли данная вам карта оптимальной.

#### Формат ввода

В первой строке дано число  $n$  ( $1 \leq n \leq 5000$ ) — количество городов в стране. Далее задана карта железных дорог в следующем формате.

Карта задана  $n-1$  строкой. В  $i$ -й строке описаны дороги из города  $i$  в города  $i+1$ ,  $i+2$ , ...,  $n$ . В строке записано  $n-i$  символов, каждый из которых либо  $R$ , либо  $B$ . Если  $j$ -й символ строки  $i$  равен « $B$ », то из города  $i$  в город  $i+j$  идет дорога типа « $B$ ». Аналогично для типа « $R$ ».

#### Формат вывода

Выведите «YES», если карта оптимальна, и «NO» в противном случае.

#### Примеры

Ввод	Вывод
3 RB R	NO
4 BBB RB B	YES
5 RRRB BRR BR R	NO

### Задача 3. Зелье

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

Злой маг Крокобобр варит зелье. У него есть большая колба, которую можно ставить на огонь и две колбы поменьше, которые огня не выдержат. В большой колбе налита компонента зелья, которую нужно подогреть на огне, маленькие колбы пусты. Емкость большой колбы магу Крокобобру известна —  $N$  миллилитров, маленьких —  $M$  и  $K$  миллилитров. Смесь можно переливать из любой колбы в любую, если выполняется одно из условий: либо после переливания одна из колб становится пустой, либо одна из колб становится полной, частичные переливания недопустимы.

Требуется ровно L миллилитров смеси в большой колбе. Помогите Крокобобру определить, сколько переливаний он должен сделать для этого.

Формат входных данных

N M K L,  $1 \leq N, M, K, L \leq 2000$

Формат выходных данных

Одно число, равное количеству необходимых переливаний или слово OOPS, если это невозможно.

#### Задача 4. Скутер

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайта

У Еремея есть электросамокат и он хочет доехать от дома до института, затратив как можно меньше энергии. Весь город расположен на холмистой местности и разделен на квадраты. Для каждого перекрестка известна его высота в метрах над уровнем моря. Если ехать от перекрестка с большей высотой до смежного с ним перекрестка с меньшей высотой, то электроэнергию можно не тратить., а если наоборот, то расход энергии равен разнице высот между перекрестками.

Помогите ему спланировать маршрут, чтобы он затратил наименьшее возможное количество энергии от дома до института и вывести это количество. Дом находится в левом верхнем углу, институт – в правом нижнем.

Формат входных данных

N M

H11 H12 ... H1M

H21 H22 ... H2M

... ..

HN1 HN2 ... HNM

$1 \leq N, M \leq 1000$

$0 \leq H_{ij} \leq 1000, H \in \mathbb{Z}$

Формат выходных данных

MinConsumingEnergy

Примеры:

Стандартный ввод	Стандартный вывод
3 5 3 1 4 1 5 9 2 6 5 3 5 9 7 9 3	5

#### Задача 5. Коврики

Ограничение по времени: 2 секунды

Ограничение по памяти: 64 мегабайта

Имеется прямоугольная область, состоящая из  $6 \leq N \leq 1000$  на  $6 \leq M \leq 1000$  одинаковых квадратных клеток. Часть клеток свободна, часть закрашена. Ковриком называется максимальное множество закрашенных клеток, имеющих общую граеицу. Коврики размером 1 x 1 тоже допустимы.

Требуется по заданной раскраске области определить количество находящихся на ней ковриков.

Формат ВХОДНЫХ данных:

В первой строке содержатся N и M – размеры области. В каждой из следующих N строк находится ровно M символов, которые могут быть точкой, если поле свободно, или плюсом, если поле закрашено.

Формат выходных данных:

Общее количество ковриков на поле.

Примеры:

Стандартный ввод	Стандартный вывод
6 6 .++++. .....+ +..+.. ..++.. .+.... .+...+	6
10 10 .++..++. +.+.++.++ .....++ +....++.. +++..+++. .....+.+.. +.+.++++++ +.++.++.. ++++...++.	10
5 5 ..+.. ..+++. ..+.. +.+.+ +++++	2

### Задача 6. Передающие станции

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

Компания сотовой связи получила лицензии на установку вышек сотовой связи в одном из новых районов города. На каждую вышку нужно установить приемо-передатчик, который позволит связаться с другими вышками. Для снижения общей стоимости проекта приемо-передатчика решено закупить оптом с большой скидкой, поэтому на всех вышках они будут однотипные. Выпускается много типов передатчиков, каждый из которых имеет различную мощность, причем, чем больше мощность, тем он дороже и имеет больший радиус охватываемой территории. Наша задача — подобрать модель передатчика с наименьшей возможной мощностью, исходя из расположения вышек для того, чтобы можно было передавать сообщения между любыми вышками, возможно, ретранслируя их.

Формат входных данных:

Первая строка содержит количество вышек  $2 \leq N \leq 2000$

В остальных  $N$  строках – пары координат вышек в Декартовой системе  $-10000 \leq X_i, Y_i \leq 10000$ .

Формат выходных данных:

Одно число — наименьший требуемый радиус, достаточный для функционирования всей системы с точностью до 4-х знаков после запятой.

Примеры:

Стандартный ввод	Стандартный вывод
5 0 0 1 0 0 1 1 1 3 3	2.8284

**Задача 7. Кратчайшие пути**

Ограничение по времени: 2 секунды

Ограничение по памяти: 64 мегабайта

Взвешенный граф с  $N$  вершинами задан своими  $M$  ребрами  $E_i$ , возможно отрицательного веса. Требуется найти все кратчайшие пути от вершины  $S$  до остальных вершин. Если граф содержит отрицательные циклы, вывести IMPOSSIBLE. Если от вершины  $S$  до какой-либо из вершин нет маршрута, то в качестве длины маршрута вывести слово UNREACHABLE.

Вершины графа нумеруются, начиная с нуля.

$3 \leq N \leq 800$

$1 \leq M \leq 30000$

$-1000 \leq W_i \leq 1000$

Формат входных данных:

$N$   $M$   $S$

$S_1$   $E_1$   $W_1$

$S_2$   $E_2$   $W_2$

...

Формат выходных данных:

IMPOSSIBLE

или

$D_1$   $D_2$   $D_3$  ...  $D_N$

Где  $D_i$  может быть UNREACHABLE

Пример

стандартный ввод	стандартный вывод
4 5 0 0 1 100 1 2 100 2 0 -100 0 2 1000	0 100 200 UNREACHABLE

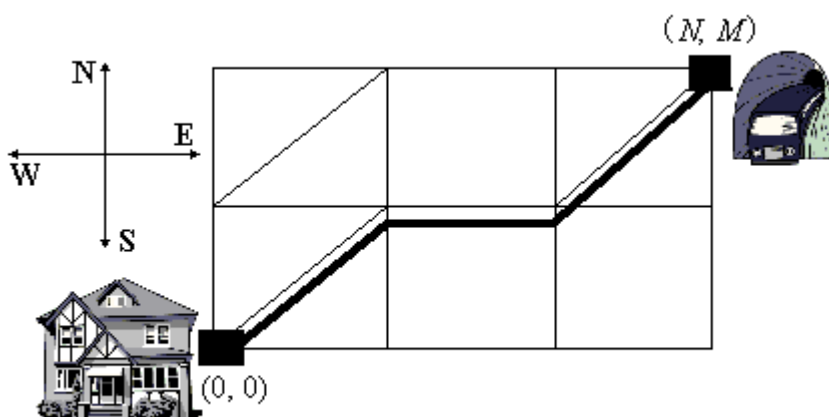
3 1 15	
3 3 0	IMPOSSIBLE
0 1 5	
1 2 8	
2 0 -20	

### Задача 8. Метро

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 мегабайта

Многие программисты СКБ Контур любят добираться до работы на метро — благо, головной офис расположен совсем недалеко от станции Уралмаш. Ну а поскольку сидячий образ жизни требует активных физических нагрузок в свободное от работы время, многие сотрудники — в том числе и Никифор — ходят от дома до метро пешком.



Никифор живёт в таком районе нашего города, где улицы образуют правильную сетку кварталов; все кварталы являются квадратами с длиной стороны, равной 100 метрам. Вход на станцию метро расположен на одном из перекрёстков; Никифор начинает свой путь с другого перекрёстка, который расположен южнее и западнее входа в метро. Естественно, что выйдя из дома, Никифор всегда идет по улицам, ведущим либо на север, либо на восток. Некоторые кварталы, которые встречаются ему на пути, он может также пересечь по диагонали, ведущей из юго-западного угла квартала в северо-восточный. Таким образом, некоторые из маршрутов (ведущих всегда на север, восток или северо-восток), оказываются короче других. Никифора интересует, сколько времени понадобится ему на преодоление кратчайшего маршрута; для этого ему нужно знать его длину.

Вы должны написать программу, которая по имеющейся информации о виде сетки кварталов рассчитывает длину кратчайшего маршрута из юго-западного угла в северо-восточный.

#### Формат входных данных

В первой строке находятся два целых числа  $N$  и  $M$  ( $0 < N, M \leq 1000$ ) — размер сетки кварталов с запада на восток и с юга на север соответственно. Никифор начинает путь с перекрёстка, который находится к юго-западу от квартала с координатами (1, 1); станция метро находится к северо-востоку от квартала с координатами ( $N, M$ ). Во второй строке находится целое число  $K$  ( $0 \leq K \leq 100$ ) — количество кварталов, через которые можно пройти наискось. Далее следуют  $K$  строк с парами целых положительных чисел, разделённых пробелами — координатами таких кварталов.

### **Формат выходных данных**

Требуется вывести длину кратчайшего пути от дома Никифора до станции метро в метрах, округлённую до целых метров.

### **Примеры**

стандартный ввод	стандартный вывод
3 2 3 1 1 3 2 1 2	383

Источник: [здесь](#).

## **Задача 9. Две команды**

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 мегабайта

Группа людей состоит из  $N$  членов. У каждого члена группы есть друзья в этой группе, один или более. Напишите программу, которая разделит группу на две команды. Каждый член каждой команды должен иметь друзей в другой команде.

### **Формат входных данных**

Первая строка ввода содержит одно число  $N$  ( $N \leq 100$ ). Члены группы занумерованы от 1 до  $N$ . Вторая, третья, ...,  $(N+1)$ -я строки содержат список друзей первого, второго, ...,  $N$ -го членов группы соответственно. Списки друзей заканчиваются нулями. Помните, что отношение дружбы в группе всегда взаимное.

### **Формат выходных данных**

Первая строка вывода должна содержать количество людей в первой группе или ноль, если невозможно разделить людей на две группы. Если решение существует, выведите список первой группы во второй строке вывода. Числа должны быть разделены одиночными пробелами. Если существует более одного решения, можете вывести любое из них.

### **Примеры**

стандартный ввод	стандартный вывод
7 2 3 0 3 1 0 1 2 4 5 0 3 0 3 0 7 0 6 0	4 2 4 5 6

Источник: [здесь](#).

## **Задача 10. План электрификации**

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 мегабайта



В некоторой стране  $n$  городов. Правительство решило электрифицировать все эти города. Для начала в  $k$  различных городах были построены электростанции. Другие города должны быть связаны с электростанциями линиями электропередач. Между любой парой городов  $i$  и  $j$  можно построить линию электропередач стоимостью  $c_{ij}$  рублей. После гражданской войны страна находится в глубоком кризисе, поэтому правительство решило построить всего лишь несколько линий электропередач. Конечно, после постройки линий должен существовать путь по ним от любого города до некоторого города с электростанцией. Найдите минимальную возможную стоимость постройки всех необходимых для этого линий электропередач.

#### **Формат входных данных**

В первой строке записаны целые числа  $n$  и  $k$  ( $1 \leq k \leq n \leq 100$ ). Во второй строке записаны  $k$  различных целых чисел — номера городов с электростанциями. В следующих  $n$  строках записана таблица  $\{c_{ij}\}$  размера  $n \times n$ , состоящая из целых чисел ( $0 \leq c_{ij} \leq 10^5$ ). Гарантируется, что  $c_{ij} = c_{ji}$ ,  $c_{ij} > 0$  для  $i \neq j$ ,  $c_{ii} = 0$ .

#### **Формат выходных данных**

Выведите минимальную стоимость электрификации всех городов.

#### **Примеры**

стандартный ввод	стандартный вывод
4 2 1 4 0 2 4 3 2 0 5 2 4 5 0 1 3 2 1 0	3

Источник: [здесь](#).

### **Задача 11. Генеалогическое дерево**

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 мегабайта

Система родственных отношений у марсиан достаточно запутана. Собственно говоря, марсиане почкуются когда им угодно и как им угодно, собираясь для этого разными группами, так что у марсианина может быть и один родитель, и несколько десятков, а сотней детей сложно кого-нибудь удивить. Марсиане привыкли к этому, и такой жизненный уклад кажется им естественным.

А вот в Планетарном Совете запутанная генеалогическая система создает серьёзные неудобства. Там заседают достойнейшие из марсиан, и поэтому, чтобы никого не обидеть, во всех обсуждениях слово принято предоставлять по очереди, так, чтобы сначала высказывались представители старших поколений, потом те, что помладше, и лишь затем уже самые юные и бездетные марсиане. Однако соблюдение такого порядка на деле представляет собой совсем не простую задачу. Не всегда марсианин знает всех своих родителей, что уж тут говорить про бабушек и дедушек! Но когда по ошибке сначала высказывается праправнук, а потом только молодо выглядящий прапрадед — это настоящий скандал.

Задача

Ваша цель — написать программу, которая определила бы раз и навсегда такой порядок выступлений в Планетарном Совете, который гарантировал бы, что каждый член совета получает возможность высказаться раньше любого из своих потомков.

#### **Формат входных данных**

В первой строке входных данных к этой задаче находится единственное число  $N$ ,  $1 \leq N \leq 100$  — количество членов Марсианского Планетарного Совета. По многовековой традиции все члены Совета нумеруются целыми числами от 1 до  $N$ . Далее следуют ровно  $N$  строк, причем  $i$ -тая строка содержит список детей члена Совета с порядковым номером  $i$ . Список детей представляет собой последовательность порядковых номеров детей, разделенных пробелами и следующих в произвольном порядке. Список детей может быть пустым. Список детей (даже если он пуст) оканчивается нулем.

#### **Формат выходных данных**

Выход должен содержать последовательность номеров выступающих, разделенных пробелами. Если несколько последовательностей удовлетворяют условиям задачи, то можно вывести любую из них. Гарантируется, что хотя бы одна такая последовательность существует.

#### **Примеры**

стандартный ввод	стандартный вывод
5 0 4 5 1 0 1 0 5 3 0 3 0	2 4 5 3 1

Источник: [здесь](#).

### **Задача 12. На метро или пешком?**

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 мегабайта

Представьте себя в большом городе. Вы хотите попасть из точки А в точку В. Для этого вы можете идти пешком или использовать метро. Перемещение с помощью метро быстрее, но входить в метро и выходить из него можно только на станциях. Чтобы сэкономить время, вы решили написать программу нахождения самого быстрого пути.

#### **Исходные данные**

Первая строка содержит два вещественных числа: скорость ходьбы и скорость перемещения с помощью метро. Вторая скорость всегда больше первой.

Затем следует описание метро. В первой строке описания находится целое число  $N$  — количество станций метро. Можете считать, что  $N$  не превосходит 200. Следующие  $N$  строк содержат по два вещественных числа каждая ( $i$ -я строка содержит координаты  $i$ -й станции). Затем следует описание соединений между станциями. Каждое соединение определяется парой целых чисел — номерами соединённых станций. Список соединений завершает пара нулей. Будем считать, что станции соединяются по прямой, так что время перемещения между станциями равно расстоянию между станциями, делённому на скорость перемещения с помощью метро.

Заметим, что вход и выход из метро, а также смена поезда возможны только на станциях и не требуют дополнительного времени.

Последними даны координаты точек А и В по паре координат в строке.

#### Результат

Первая строка должна содержать минимальное время, необходимое, чтобы попасть из точки А в точку В. Время должно быть выведено с точностью  $10^{-6}$ . Вторая строка описывает использование метро при передвижении. Она начинается количеством станций, которые нужно посетить, затем следует список номеров станций в порядке, в котором их нужно посетить.

#### Пример

стандартный ввод	стандартный вывод
1 100	2.6346295
4	4 4 2 1 3
0 0	
1 0	
9 0	
9 9	
1 2	
1 3	
2 4	
0 0	
10 10	
10 0	

Источник: [здесь](#).

### **Задача 13. Российские газопроводы**

Ограничение по времени: 0,5 секунды

Ограничение по памяти: 64 мегабайта

Вступление

Большими неприятностями обернулся прошедший год для Государства Российского. То неурожай, то птичий грипп, то вечные споры хозяйствующих субъектов... А тут ещё и Президент задумал, наконец, собрать средства на покупку новой балалайки и ручного медведя для своего двоюродного племянника. Все эти факторы (в особенности, конечно, последний) сильно ударили по экономике государства. Посовещавшись со своими друзьями в валенках и ушанках, Президент решил воспользоваться традиционным методом укрепления национального бюджета - увеличением налога на транспортировку газа.

Задача

Сеть российских газопроводов представляет собой  $N$  перекачивающих станций, некоторые из которых соединены газопроводами. Для каждого из  $M$  газопроводов известны номера станций  $A[i]$  и  $B[i]$ , которые он соединяет, и его прибыльность  $C[i]$ , т.е. то количество долларов, которое будет ежедневно приносить в виде налогов перекачка газа по этому газопроводу. Каждая пара станций соединена не более чем одним газопроводом.

Сеть была построена советскими инженерами, которые точно знали, что газ поставляется из месторождений Украины в Сибирь, а не наоборот. Поэтому все газопроводы являются однонаправленными, т.е. для каждого газопровода перекачка газа

возможна только в направлении из станции с номером  $A[i]$  на станцию с номером  $B[i]$ . Более того, для любых двух станций  $X$  и  $Y$  верно, что если возможна перекачка газа из  $X$  на  $Y$  (возможно, через промежуточные станции), то обратная перекачка из  $Y$  на  $X$  невозможна. Известно, что газ поступает на начальную станцию с номером  $S$  и отгружается потребителям на конечной станции с номером  $F$ .

Президент потребовал от Правительства указать маршрут (т.е. линейную последовательность попарно соединённых газопроводами станций) перекачки газа из начальной станции на конечную, причём прибыльность этого маршрута должна быть максимальной. Под прибыльностью маршрута понимается суммарная прибыльность входящих в него газопроводов.

К сожалению, Президент не учёл того факта, что многие газопроводы изначальной сети уже давно прекратили существование, в результате чего может оказаться, что перекачка газа из начальной станции на конечную вообще невозможна...

#### Исходные данные

Первая строка содержит целые числа  $N$  ( $2 \leq N \leq 500$ ) и  $M$  ( $0 \leq M \leq 124750$ ). Каждая из следующих  $M$  строк содержит целые числа  $A[i]$ ,  $B[i]$  ( $1 \leq A[i], B[i] \leq N$ ) и  $C[i]$  ( $1 \leq C[i] \leq 10000$ ) для соответствующего газопровода. Последняя строка содержит целые числа  $S$  и  $F$  ( $1 \leq S, F \leq N$ ;  $S \neq F$ ).

#### Результат

Если искомый маршрут существует, выведите его прибыльность. Иначе выведите "No solution".

#### Пример

стандартный ввод	стандартный вывод
6 7 6 5 10 1 4 11 1 2 4 3 1 5 2 4 5 6 3 1 6 1 3 6 4	17

#### Замечания

В примере искомым маршрутом является маршрут  $6 > 3 > 1 > 4$ .

Источник: [здесь](#).