

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ЭКОНОМИКИ И СЕРВИСА

КАФЕДРА МАТЕМАТИКИ И МОДЕЛИРОВАНИЯ

Методические рекомендации по выполнению лабораторных работ по дисциплине

ДИСКРЕТНАЯ МАТЕМАТИКА

Направление подготовки и направленность (профиль):

09.03.03 Прикладная информатика

Форма обучения:

очная

Греско А.А., кандидат экономических наук, доцент, Кафедра математики и моделирования, Aleksandr.Gresko@vvsu.ru

Клочкова О.И., кандидат физико-математических наук, доцент, Кафедра математики и моделирования,

Klochkova.O@vvsu.ru

Солодухин К.С., доктор экономических наук, заведующий научной лабораторией, Лаборатория стратегического планирования, Konstantin.Solodukhin@vvsu.ru

Владивосток 2022 г.

Лабораторная работа 1 по теме 2.1

Цели лабораторной работы

Сформировать у обучающихся умения и навыки построения графа и таблицы смежности в рамках теоретических и практических исследований для последующего использования при решении практических задач. Повысить навыки программирования на языке Python

Планируемые результаты обучения

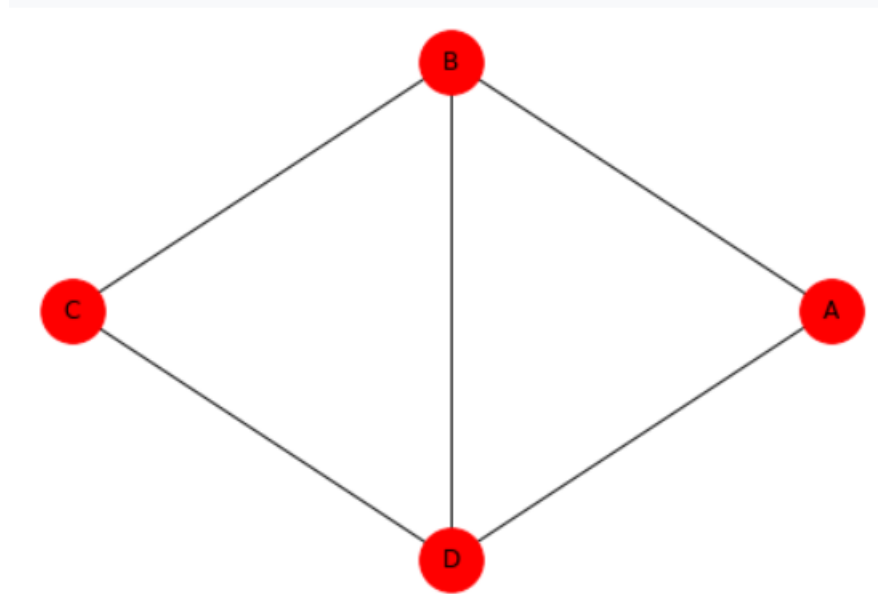
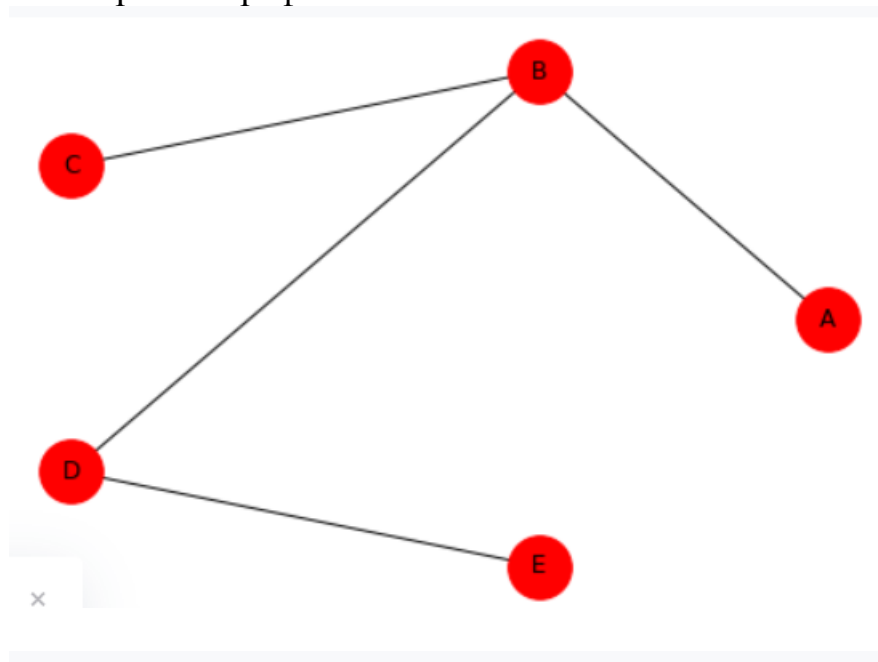
Формирование умения построить изображение заданного графа и таблицы смежности для последующего применения при решении практических задач. Формирование навыков программирования и использования методов естественнонаучных дисциплин в профессиональной деятельности для теоретического и экспериментального исследования.

Содержание лабораторной работы

Построить граф типа

Задание по теме Графы

1. Построить графы типа



Возможно, в помощь ссылка <https://shwanoff.ru/networkx-part1/>

2. Сделать из последнего полносвязный граф (т.е. каждая вершина должна быть связана с каждой).

3. Ввести расстояния - все ребра по 1, а диагонали по 10.

Вывести кратчайшие расстояния от вершины А до всех остальных вершин.

4. Возможно ли здесь реализация Алгоритма Дейкстры?

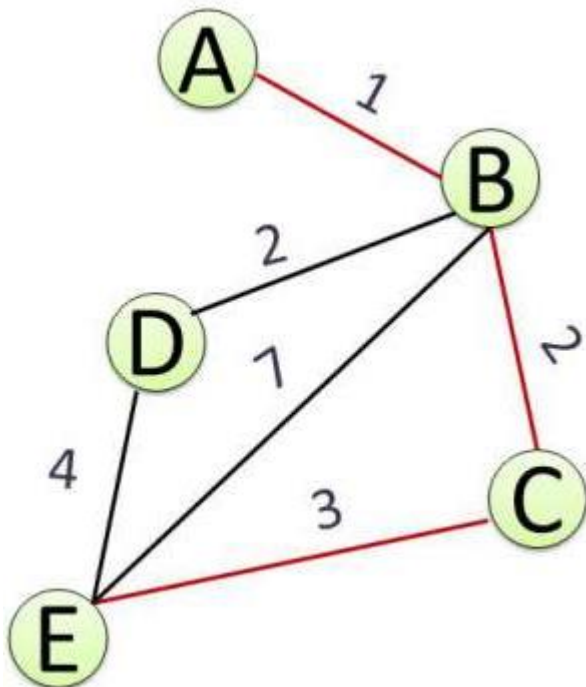
5. Задать расстояния между вершинами А, В, С, D, E с помощью таблицы смежности – определить длину кратчайшего пути между А и E. Нарисовать граф с выделенным кратчайшим путем

Тест

Между населёнными пунктами А, В, С, D, E построены дороги. Нужно определить длину кратчайшего пути между пунктами **А и E**. Передвигаться можно только по дорогам, протяжённость которых указана в таблице.

	А	В	С	D	E
А		1			
В	1		2	2	7
С		2			3
D		2			4
E		7	3	4	

Результат $1+2+3=6$



Построить таблицу смежности этого графа

Краткие методические указания

При подготовке к лабораторной работе необходимо обратить внимание на содержание основных теоретических вопросов, связанных с характеристиками графа, изложенных на лекционных занятиях.

Отчет о выполнении лабораторной работы должен быть представлен в электронном виде (на языке программирования Python).

Шкала оценки

Баллы	Описание
5	Задание выполнено полностью и абсолютно правильно. Тестовое задание выполнено -включая изображение
4	Задание выполнено полностью и правильно, но решение содержит некоторые неточности и несущественные ошибки или выполнено 1 тестовое задание, например, без изображения
3	Задание выполнено не полностью, с существенными ошибками, но подход к решению, идея решения, метод правильны.
2	Задание выполнено частично, имеет ошибки, осуществлена попытка решения на основе правильных методов и идей решения.
0	Задание не выполнено.

Лабораторная работа №2 Алгоритм связности. Тема 2.1

Введение. Граф $G(V, E)$ называется *связным*, если любую пару его вершин соединяет какой-нибудь маршрут. Минимальное число связных компонент называется числом связности графа и обозначается $c(G)$.

Алгоритм связности. Алгоритм предназначен для вычисления значения $c=c(G)$, т.е. числа компонент связности графа G . Ниже приведен код алгоритма на языке Pascal

```
begin
  V' := V;
  c := 0;
  while V' ≠ ∅ do
    begin
      Выбрать  $y \in V'$ ;
      Найти все вершины, соединенные маршрутом с  $y$ ;
      Удалить вершину  $y$  из  $V'$  и
      соответствующие ребра из  $E$ ;
      c := c + 1;
    end
end
```

Цели лабораторной работы

Сформировать у обучающихся умения и навыки числа связности заданного графа в рамках теоретических и практических исследований для последующего использования при решении практических задач. Повысить навыки программирования на языке Python

Планируемые результаты обучения

Формирование умения определить число связности заданного графа для последующего применения при решении практических задач. Формирование навыков программирования и использования методов естественнонаучных дисциплин в профессиональной деятельности для теоретического и экспериментального исследования.

Содержание лабораторной работы

ЗАДАНИЕ

1. По заданной таблице смежности или заданному изображению графа найти число связности графа и вывести изображения компонент связности

Требуется:

1. Создать программу на языке программирования Python для ввода данных согласно заданной задаче

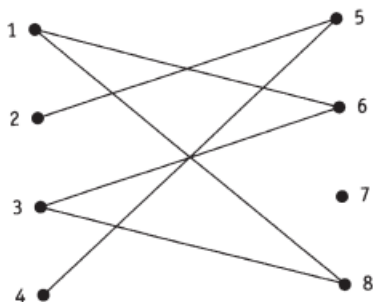
2. Вывести :

1) число связности

2) изображения компонент связности

ТЕСТОВЫЕ ДАННЫЕ

Граф имеет вид

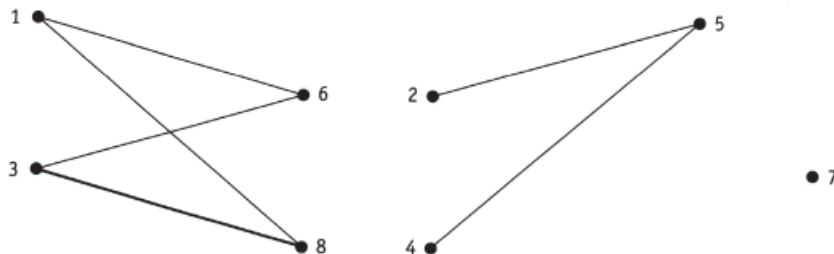


Результат

Решение (см. таблицу)

	V'	c
Исходные значения	{1, 2, 3, 4, 5, 6, 7, 8}	0
Выбор $y = 1$	{2, 4, 5, 7}	1
Выбор $y = 2$	{7}	2
Выбор $y = 7$	\emptyset	3

Таким образом, $c(G)=3$. Соответствующие компоненты связности имеют вид



Краткие методические указания

При подготовке к лабораторной работе необходимо обратить внимание на содержание основных теоретических вопросов, связанных определением связности графа, изложенных на лекционных занятиях.

Отчет о выполнении лабораторной работы должен быть представлен в электронном виде (на языке программирования Python).

Шкала оценки

Баллы	Описание
5	Задание выполнено полностью и абсолютно правильно. Тестовое задание выполнено -включая изображение
4	Задание выполнено полностью и правильно, но решение содержит некоторые неточности и несущественные ошибки или выполнено 1 тестовое задание, например, без изображения
3	Задание выполнено не полностью, с существенными ошибками, но подход к решению, идея решения, метод правильны.

2	Задание выполнено частично, имеет ошибки, осуществлена попытка решения на основе правильных методов и идей решения.
0	Задание не выполнено.

Лабораторная работа №3 Алгоритм построения случайного неориентированного графа. Тема 2.2

Введение. Матрица смежности M неориентированного графа, состоящего из n вершин и m ребер будет иметь размер $n \times n$ и содержать $2m$ истинных значений I , расположенных симметрично относительно главной диагонали. Считаем что граф не содержит петель, т.е. для любого $i=1,2,\dots,n$ $M(i,i)=L$. На каждом k -ом шаге алгоритма ($k=1,2,\dots,m$) с помощью датчика случайных чисел нужно получить случайное число и записывать в этот случайный адрес (i,j) матрицы значение « I » (или 1-т.е. «истина»). Если ячейка с вычисленным адресом уже содержит истину, или $i=j$, то m нужно увеличить на 1. Ниже приведен код программы на языке Pascal

Input

n — количество вершин графа;
 m — количество ребер графа;
 M — матрица размера $n \times n$,
содержащая во всех ячейках L ;

begin

for $k = 1$ to m do

begin

получить с помощью датчика

случайное число R ;

$N := \lfloor n^2 R \rfloor + 1$;

$i := \lfloor N : n \rfloor + 1$;

$j := N - (i - 1) \cdot n$;

if $M(i, j) \neq I$ and $i \neq j$ then

begin

$M(i, j) := I$;

$M(j, i) := I$;

end

else

$m := m + 1$

end

end

Output M — матрица смежности неориентированного графа.

Цели лабораторной работы

Сформировать у обучающихся умения и навыки построения неориентированных графов в рамках теоретических и практических исследований для последующего использования при решении практических задач. Повысить навыки программирования на языке Python

Планируемые результаты обучения

Формирование умения использовать неориентированные графы для последующего нахождения соответствующих организационно-управленческих решений. Формирование навыков программирования и использования методов естественнонаучных дисциплин в профессиональной деятельности для теоретического и экспериментального исследования.

Содержание лабораторной работы

ЗАДАНИЕ

1. Записать алгоритм построения случайного неориентированного графа на языке Python обеспечить ввод количества вершин и ребер графа. Вывести матрицу смежности неориентированного графа

Требуется:

1. Создать программу на языке программирования Python для ввода данных согласно заданной задаче

2. Вывести :

1) матрицу смежности случайного неориентированного графа

2) изображение графа

ТЕСТОВЫЕ ДАННЫЕ

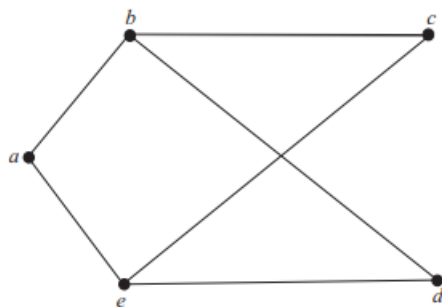
$n=5, m=6$

Результат

Матрица смежности

$$\begin{array}{c} \begin{array}{ccccc} & a & b & c & d & e \\ \begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} & \begin{bmatrix} \text{Л} & \text{И} & \text{Л} & \text{Л} & \text{И} \\ \text{И} & \text{Л} & \text{И} & \text{И} & \text{Л} \\ \text{Л} & \text{И} & \text{Л} & \text{Л} & \text{И} \\ \text{Л} & \text{И} & \text{Л} & \text{Л} & \text{И} \\ \text{И} & \text{Л} & \text{И} & \text{И} & \text{Л} \end{bmatrix} \end{array} \end{array}$$

Изображение графа по матрице смежности имеет вид



Краткие методические указания

При подготовке к лабораторной работе необходимо обратить внимание на содержание основных теоретических вопросов, связанных с неориентированными графами, изложенных на лекционных занятиях.

Отчет о выполнении лабораторной работы должен быть представлен в электронном виде (на языке программирования Python).

Шкала оценки

Баллы	Описание
5	Задание выполнено полностью и абсолютно правильно. Тестовое задание выполнено -обе части
4	Задание выполнено полностью и правильно, но решение содержит некоторые неточности и несущественные ошибки или выполнено 1 тестовое задание
3	Задание выполнено не полностью, с существенными ошибками, но подход к решению, идея решения, метод правильны.
2	Задание выполнено частично, имеет ошибки, осуществлена попытка решения на основе правильных методов и идей решения.
0	Задание не выполнено.

Введение. Граф $G(V, E)$ называется *деревом*, если он связан и ациклический, т.е. не содержит циклов. Остовное дерево строится просто: выбирается любое ребро и последовательно добавляются другие ребра, не создавая при этом циклов. Остовное дерево с наименьшим весом называется минимальным остовным деревом (МОД)

Алгоритм поиска минимального остовного дерева. Алгоритм строит МОД в графе $G(V, E)$, последовательно выбирая ребра наименьшего возможного веса до образования остовного дерева. МОД в памяти компьютера хранится в виде множества T ребер. Ниже приведен код алгоритма на языке Pascal

```
begin
  e := ребро графа G с наименьшим весом;
  T := {e};
  E' := E \ {e}
  while E' ≠ ∅
  begin
    e' := ребро из E' наименьшего веса;
    T := T ∪ {e'};
    E' := множество ребер из E' \ {T},
      чье добавление к T не ведет
      к образованию циклов;
  end
end
```

Цели лабораторной работы

Сформировать у обучающихся умения и навыки построения МОД из заданного графа в рамках теоретических и практических исследований для последующего использования при решении практических задач. Повысить навыки программирования на языке Python

Планируемые результаты обучения

Формирование умения построения МОД из заданного графа для последующего применения при решении практических задач. Формирование навыков программирования и использования методов естественнонаучных дисциплин в профессиональной деятельности для теоретического и экспериментального исследования.

Содержание лабораторной работы

ЗАДАНИЕ

1. По заданной таблице смежности найти минимальное остовное дерево в виде множества T ребер

Требуется:

1. Создать программу на языке программирования Python для ввода данных согласно заданной задаче

2. Вывести :

1) МОД в виде множества T ребер и общий вес

2) изображение графа

ТЕСТОВЫЕ ДАННЫЕ

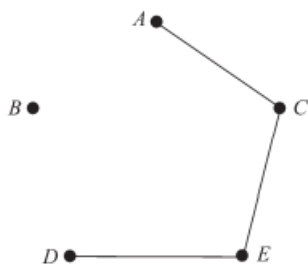
Таблица смежности имеет вид

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>
<i>A</i>	—	13	3	9	9
<i>B</i>	13	—	11	11	13
<i>C</i>	3	11	—	9	7
<i>D</i>	9	11	9	—	2
<i>E</i>	9	13	7	2	—

Результат

Решение (начало)

Ребра выбираются следующим образом: ребро DE весом 2, второе - ребро AC с весом 3, третье - ребро CE с весом 7. На этой стадии строящееся дерево выглядит так



Следующие по весу ребра AD , AE и CD , каждое из которых имеет вес 9. Однако, добавление любого из них ведет к образованию цикла. Поэтому эти ребра нужно исключить. Остаются ребра BC и BD с весом 11. Можно присоединить любое из них получив при этом два разных МОД: $\{AC, BC, CE, DE\}$ или $\{AC, BD, CE, DE\}$ общим весом 23 каждое

Краткие методические указания

При подготовке к лабораторной работе необходимо обратить внимание на содержание основных теоретических вопросов, связанных с построением остова дерева, изложенных на лекционных занятиях.

Отчет о выполнении лабораторной работы должен быть представлен в электронном виде (на языке программирования Python).

Шкала оценки

Баллы	Описание
5	Задание выполнено полностью и абсолютно правильно. Тестовое задание выполнено -включая изображение
4	Задание выполнено полностью и правильно, но решение содержит некоторые неточности и несущественные ошибки или выполнено 1 тестовое задание, например, без изображения
3	Задание выполнено не полностью, с существенными ошибками, но подход к решению, идея решения, метод правильны.
2	Задание выполнено частично, имеет ошибки, осуществлена попытка решения на основе правильных методов и идей решения.
0	Задание не выполнено.

Лабораторная работа №5. Алгоритм ближайшего соседа. Тема 2.4

Введение. В отличие от *эйлеровых графов*, обладающих замкнутым маршрутом, который проходит по всем ребрам графа ровно один раз, *гамильтонов цикл* проходит через каждую вершину в точности один раз. Граф с таким циклом называется *гамильтоновым графом*. Важность задачи нахождения гамильтонова цикла связана с решением графической модели задачи коммивояжера. Эта модель состоит из гамильтонова графа, вершины которого представляют города, а ребра- связывающие их дороги. Каждое ребро снабжено весом, связанным с транспортными затратами. Для решения задачи необходимо найти гамильтонов цикл с минимальным общим весом. Эффективный алгоритм решения данной задачи пока не известен. Существует алгоритм субоптимального решения. Найденный в результате цикл имеет, как правило, значительно меньший вес, чем большинство произвольных гамильтоновых циклов

Алгоритм ближайшего соседа. Алгоритм выдает субоптимальное решение задачи коммивояжера, генерируя гамильтоновы циклы в нагруженном графе со множеством вершин V . Цикл, полученный в результате работы алгоритма, будет совпадать с конечным значением переменной *маршрут*, а его общая длина – конечное значение переменной w . Ниже приведен код алгоритма на языке Pascal

```

begin
  Выбрать  $v \in V$ ;
  маршрут :=  $v$ ;
   $w := 0$ ;
   $v' := v$ ;
  Отметить  $v'$ ;
  while остаются неотмеченные вершины do
    begin
      Выбрать неотмеченную вершину  $u$ ,
      ближайшую к  $v'$ ;
      маршрут := маршрут  $u$ ;
       $w := w + \text{вес ребра } v'u$ ;
       $v' := u$ ;
      Отметить  $v'$ ;
    end
  маршрут := маршрут  $v$ ;
   $w := w + \text{вес ребра } v'v$ ;
end

```

Цели лабораторной работы

Сформировать у обучающихся умения и навыки нахождения гамильтоновых циклов с наименьшим весом заданного графа в рамках теоретических и практических исследований для последующего использования при решении практических задач. Повысить навыки программирования на языке Python

Планируемые результаты обучения

Формирование умения определить гамильтоновы циклы с наименьшим весом заданного графа для последующего применения при решении практических задач коммивояжера. Формирование навыков программирования и использования методов естественнонаучных дисциплин в профессиональной деятельности для теоретического и экспериментального исследования.

Содержание лабораторной работы

ЗАДАНИЕ

1. По заданной таблице смежности или заданному изображению графа найти гамильтонов цикл с наименьшим общим весом, применяя алгоритм ближайшего соседа

Требуется:

1. Создать программу на языке программирования Python для ввода данных согласно заданной задаче

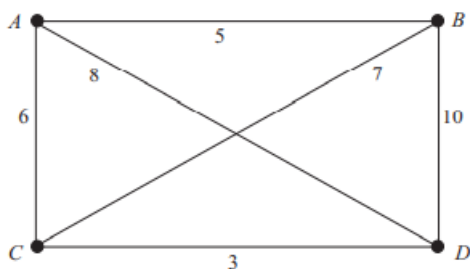
2. Вывести :

1) гамильтонов цикл и вес

2) изображения таблицы решения с промежуточным решением

ТЕСТОВЫЕ ДАННЫЕ

Взвешенный граф имеет вид



Результат

Решение (см. таблицу)- за исходную вершину – необходимо взять вершину D

	u	маршрут	w	v'
Исходные значения	—	D	0	D
	C	DC	3	C
	A	DCA	9	A
	B	$DCAB$	14	B
Последний проход	B	$DCABD$	24	B

В результате работы алгоритма был найден гамильтонов цикл $DCABD$ с общим весом 24.

Краткие методические указания

При подготовке к лабораторной работе необходимо обратить внимание на содержание основных теоретических вопросов, связанных определением гамильтоновых циклов графа, изложенных на лекционных занятиях.

Отчет о выполнении лабораторной работы должен быть представлен в электронном виде (на языке программирования Python).

Шкала оценки

Баллы	Описание
5	Задание выполнено полностью и абсолютно правильно. Тестовое задание выполнено -включая изображение
4	Задание выполнено полностью и правильно, но решение содержит некоторые неточности и несущественные ошибки или выполнено 1 тестовое задание, например, без изображения
3	Задание выполнено не полностью, с существенными ошибками, но подход к решению, идея решения, метод правильны.
2	Задание выполнено частично, имеет ошибки, осуществлена попытка решения на основе правильных методов и идей решения.
0	Задание не выполнено.

Лабораторная работа №6. Эйлеровы циклы. Тема 2.4

Введение. Граф G называется эйлеровым графом, если в нем существует маршрут, который начинается и заканчивается в одной и той же вершине и проходит по каждому ребру графа ровно один раз. Задачу построения эйлерова цикла (если он существует) можно решить с помощью алгоритма, основанного на следующем правиле. Маршрут P начинается в произвольной вершине a графа G и продолжать маршрут, насколько возможно, через новые ребра. Так как к каждой вершине подходит четное число ребер, этот процесс может закончиться только в исходной вершине a . При этом получается эйлеров цикл. Если P содержит не все ребра G , то ребра, входящие в P удаляются из G . Процесс повторяется.

Алгоритм построения эйлерова цикла в графе. Алгоритм строит эйлеров цикл, проходящий через каждое ребро по одному разу. Ниже приведен код алгоритма на языке Pascal

Input

$G(V, E)$ — конечный связный граф с четными степенями всех вершин;

begin

выбрать любую вершину $d \in V$;

$P := \emptyset$;

$a := d$;

while $E \neq \emptyset$ do

begin

выбрать вершину $d \in P$, для которой существует ребро $(d, x) \in E$;

$c := d$;

$P' := \emptyset$;

repeat

begin

выбрать ребро $(d, y) \in E$ с условием:

$(d, y) \notin P'$;

$P' := P' \cup \{(d, y)\}$;

удалить ребро (d, y) из E ;

$d := y$;

end;

until $y \neq c$

if $P \neq \emptyset$ then

$P := P(a, c) \cup P' \cup P(c, a)$;

else

$P := P'$;

end

end

Output P — эйлеров цикл графа G .

Цели лабораторной работы

Сформировать у обучающихся умения и навыки нахождения эйлеровых циклов заданного графа в рамках теоретических и практических исследований для последующего использования при решении практических задач. Повысить навыки программирования на языке Python

Планируемые результаты обучения

Формирование умения определить эйлеровы циклы заданного графа для последующего применения при решении практических задач составления маршрутов движения по заданной схеме улиц. Формирование навыков программирования и использования методов естественнонаучных дисциплин в профессиональной деятельности для теоретического и экспериментального исследования.

Содержание лабораторной работы

ЗАДАНИЕ

1. Построить эйлеров цикл в графе $G(V, E)$

Требуется:

1. Создать программу на языке программирования Python для ввода данных согласно заданной задаче

2. Вывести :

1) эйлеров цикл

2) изображение графа с циклом

ТЕСТОВЫЕ ДАННЫЕ

Граф задан в виде

$V = \{1, 2, 3, 4, 5, 6\}$ и

$E = \{(1, 2), (1, 4), (2, 3), (2, 4), (2, 5), (3, 4), (4, 6), (5, 6)\}$.

Решение (пояснение алгоритма Pascal). Элементы просматриваются слева направо. Начальной вершиной выбирается вершина 1

- Положим $d := 1$;
- $P := \emptyset$;
- $a := d = 1$.

Выполняем операторы внешнего цикла **while** $E \neq \emptyset$ **do** ... **end**. Так как $P = \emptyset$, новую вершину выбрать невозможно и d сохраняет значение, равное 1.

- $c := d = 1$;
- $P' := \emptyset$.

Выполняем операторы внутреннего цикла **repeat** ... **until** $y \neq c$.

- Выбираем ребро $(d, y) \in E$ с условием: $(d, y) \notin P'$, где $d = 1$ (это ребро $(1, 2)$);
- включаем его в маршрут: $P' := 1\ 2$;
- удаляем ребро $(1, 2)$ из E
($E := \{(1, 4), (2, 3), (2, 4), (2, 5), (3, 4), (4, 6), (5, 6)\}$);
- $d := y = 2$.

Выбираем ребро $(d, y) \in E$ с условием: $(d, y) \notin P'$, где $d = 2$ (это ребро $(2, 3)$);

включаем его в маршрут: $P' := 1\ 2\ 3$;

удаляем ребро $(2, 3)$ из E

($E := \{(1, 4), (2, 4), (2, 5), (3, 4), (4, 6), (5, 6)\}$);

$d := y = 3$.

Выбираем такое ребро $(d, y) \in E$, что $(d, y) \notin P'$, где $d = 3$ (это ребро $(3, 4)$);

включаем его в маршрут: $P' := 1\ 2\ 3\ 4$;

- удаляем ребро $(3, 4)$ из E
($E := \{(1, 4), (2, 4), (2, 5), (4, 6), (5, 6)\}$);
- $d := y = 4$.
- Выбираем такое ребро $(d, y) \in E$, что $(d, y) \notin P'$, где $d = 4$
(это ребро $(4, 1)$);
- включаем его в маршрут: $P' := 1\ 2\ 3\ 4\ 1$;
- удаляем ребро $(1, 4)$ из E ($E := \{(2, 4), (2, 5), (4, 6), (5, 6)\}$);
- $d := y = 1$.

Выполнилось условие окончания внутреннего цикла, поскольку y стал равен c . Так как $P = \emptyset$, полагаем $P := P' = 1\ 2\ 3\ 4\ 1$. Поскольку $E \neq \emptyset$, повторяем внешний цикл.

- Выбираем вершину $d \in P$, для которой существует ребро (d, x) , принадлежащее E (это вершина $d = 2$);
- $c := d = 2$;
- $P := \emptyset$.

Снова переходим к выполнению операторов внутреннего цикла **repeat . . . until** $y \neq c$.

- Выбираем такое ребро $(d, y) \in E$, что $(d, y) \notin P'$, где $d = 2$
(это ребро $(2, 4)$);
- включаем его в маршрут: $P' := 2\ 4$;
- удаляем ребро $(2, 4)$ из E ($E := \{(2, 5), (4, 6), (5, 6)\}$);
- $d := y = 4$.
- Выбираем такое ребро $(d, y) \in E$, что $(d, y) \notin P'$, где $d = 4$
(это ребро $(4, 6)$);
- включаем его в маршрут: $P' := 2\ 4\ 6$;
- удаляем ребро $(4, 6)$ из E ($E := \{(2, 5), (5, 6)\}$);
- $d := y = 6$.
 - Выбираем такое ребро $(d, y) \in E$, что $(d, y) \notin P'$, где $d = 6$
(это ребро $(6, 5)$);
 - включаем его в маршрут: $P' := 2\ 4\ 6\ 5$;
 - удаляем ребро $(6, 5)$ из E ($E := \{(2, 5)\}$);
 - $d := y = 5$.

- Выбираем такое ребро $(d, y) \in E$, что $(d, y) \notin P'$, где $d = 5$ (это ребро $(5, 2)$);
- включаем его в маршрут: $P' := 24652$;
- удаляем ребро $(5, 2)$ из E ($E := \emptyset$);
- $d := y = 2$.

Поскольку $y = c = 2$, внутренний цикл закончен. Так как $P \neq \emptyset$, полагаем

$$P := P(a, c) \cup P' \cup P(c, a) = P(1, 2) \cup P' \cup P(2, 1) = \\ = 124652341.$$

Кроме того, к настоящему моменту $E = \emptyset$. Следовательно, закончен внешний цикл и весь алгоритм. В результате построен эйлеров цикл: $P = 24652341$.

Краткие методические указания

При подготовке к лабораторной работе необходимо обратить внимание на содержание основных теоретических вопросов, связанных определением эйлеровых циклов графа, изложенных на лекционных занятиях.

Отчет о выполнении лабораторной работы должен быть представлен в электронном виде (на языке программирования Python).

Шкала оценки

Баллы	Описание
5	Задание выполнено полностью и абсолютно правильно. Тестовое задание выполнено -включая изображение графа и эйлерова цикла в нем
4	Задание выполнено полностью и правильно, но решение содержит некоторые неточности и несущественные ошибки или выполнено 1 тестовое задание, например, без изображения
3	Задание выполнено не полностью, с существенными ошибками, но подход к решению, идея решения, метод правильны.
2	Задание выполнено частично, имеет ошибки, осуществлена попытка решения на основе правильных методов и идей решения.
0	Задание не выполнено.

Лабораторная работа №7. Алгоритм Терри. Тема 2.5

Введение. Если граф G не является эйлеровым графом, а что-то вроде эйлерова цикла построить необходимо, можно рассмотреть задачу построения нескольких маршрутов, покрывающих все его ребра. Для нахождения цикла, проходящего через каждое ребро по одному разу в каждом из двух направлений, можно использовать правило Терри. Начиная с произвольной вершины a_0 , необходимо идти по какому-нибудь маршруту P , отмечая на каждом ребре направление, в котором оно было пройдено. При прохождении некоторой вершины g в первый раз отмечается первое входящее ребро. Из вершины g нужно следовать всегда по ребру (g, r) , которое или вообще еще не было пройдено или было пройдено в противоположном направлении. При этом по первому входящему в g ребру можно идти только тогда, когда других возможностей не остается. Цепь P должна покрывать все ребра графа по одному разу в каждом из двух направлений.

Алгоритм Терри. Алгоритм строит ориентированный цикл, проходящий через каждое ребро по одному разу в каждом направлении. Ниже приведен код алгоритма на языке Pascal

Input

n — число вершин графа;
 V — множество вершин графа;
 m — число ребер графа;
 E — множество ребер графа;
 M — матрица смежности графа;

begin

выбрать произвольную вершину $a \in V$;

$P := \emptyset$;

$k := 2m$;

$V' := V$;

while $k \neq 0$ do

begin

выбрать ребро (a, b) , для которого
 $M(a, b) = I$, причем в последнюю
очередь выбирать ребро (a, b) , для
которого $M(b, a) = J$;

if $b \in V'$ then

begin

удалить b из V' ;

положить $M(a, b) := J$;

end

else

положить $M(a, b) := J$;

$P := P \cup \{(a, b)\}$;

$k := k - 1$;

$a := b$;

end

end

Output P — ориентированный цикл, проходящий через
каждое ребро графа по одному разу в каждом из двух
направлений.

Цели лабораторной работы

Сформировать у обучающихся умения и навыки нахождения циклов Терри заданного графа в рамках теоретических и практических исследований для последующего использования при решении практических задач. Повысить навыки программирования на языке Python

Планируемые результаты обучения

Формирование умения определить эйлеровы циклы с помощью алгоритма Терри заданного графа для последующего применения при решении практических задач составления маршрутов движения по заданной схеме улиц. Формирование навыков программирования и использования методов естественнонаучных дисциплин в профессиональной деятельности для теоретического и экспериментального исследования.

Содержание лабораторной работы

ЗАДАНИЕ

1. По заданной таблице смежности или заданному изображению графа найти цикл по алгоритму Терри

Требуется:

1. Создать программу на языке программирования Python для ввода данных согласно заданной задаче

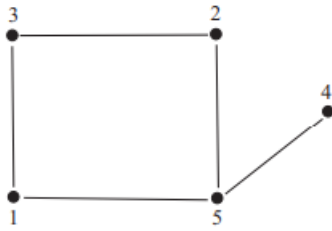
2. Вывести :

1) маршрут по алгоритму Терри

2) изображения таблицы решения с промежуточным решением

ТЕСТОВЫЕ ДАННЫЕ

Граф имеет вид



Построить ориентированный граф, проходящий через каждое ребро по одному разу в каждом из двух направлений

Результат

Решение. Взять первой вершину 1. Первая строка таблицы соответствует начальному состоянию массивов и переменных

k	a	(a, b)	V'	P	M
12	1		{1, 2, 3, 4, 5}	\emptyset	$\begin{bmatrix} Л & И & И & Л & И \\ И & Л & И & Л & И \\ И & И & Л & Л & Л \\ Л & Л & Л & Л & И \\ И & И & Л & И & Л \end{bmatrix}$
11	2	(1, 2)	{1, 3, 4, 5}	{(1, 2)}	$\begin{bmatrix} Л & Л & И & Л & И \\ И & Л & И & Л & И \\ И & И & Л & Л & Л \\ Л & Л & Л & Л & И \\ И & И & Л & И & Л \end{bmatrix}$
10	3	(2, 3)	{1, 4, 5}	{(1, 2), (2, 3)}	$\begin{bmatrix} Л & Л & И & Л & И \\ И & Л & Л & Л & И \\ И & И & Л & Л & Л \\ Л & Л & Л & Л & И \\ И & И & Л & И & Л \end{bmatrix}$

Конец таблицы

4	2	(3, 2)	{4}	{(1, 2), (2, 3), (3, 1), (1, 5), (5, 2) (2, 1), (1, 3), (3, 2)}	$\begin{bmatrix} Л & Л & Л & Л & Л \\ Л & Л & Л & Л & И \\ Л & Л & Л & Л & Л \\ Л & Л & Л & Л & И \\ И & Л & Л & И & Л \end{bmatrix}$
3	5	(2, 5)	{4}	{(1, 2), (2, 3), (3, 1), (1, 5), (5, 2), (2, 1), (1, 3), (3, 2), (2, 5)}	$\begin{bmatrix} Л & Л & Л & Л & Л \\ Л & Л & Л & Л & Л \\ Л & Л & Л & Л & Л \\ Л & Л & Л & Л & И \\ И & Л & Л & И & Л \end{bmatrix}$
2	4	(5, 4)	∅	{(1, 2), (2, 3), (3, 1), (1, 5), (5, 2), (2, 1), (1, 3), (3, 2), (2, 5), (5, 4)}	$\begin{bmatrix} Л & Л & Л & Л & Л \\ Л & Л & Л & Л & Л \\ Л & Л & Л & Л & Л \\ Л & Л & Л & Л & И \\ И & Л & Л & Л & Л \end{bmatrix}$
1	5	(4, 5)	∅	{(1, 2), (2, 3), (3, 1), (1, 5), (5, 2) (2, 1), (1, 3), (3, 2), (2, 5), (5, 4), (4, 5)}	$\begin{bmatrix} Л & Л & Л & Л & Л \\ Л & Л & Л & Л & Л \\ Л & Л & Л & Л & Л \\ Л & Л & Л & Л & Л \\ И & Л & Л & Л & Л \end{bmatrix}$
0	4	(5, 1)	∅	{(1, 2), (2, 3), (3, 1), (1, 5), (5, 2) (2, 1), (1, 3), (3, 2), (2, 5), (5, 4), (4, 5), (5, 1)}	$\begin{bmatrix} Л & Л & Л & Л & Л \\ Л & Л & Л & Л & Л \\ Л & Л & Л & Л & Л \\ Л & Л & Л & Л & Л \\ Л & Л & Л & Л & Л \end{bmatrix}$

Таким образом, в результат работы алгоритма построен цикл

$$P = \{(1, 2), (2, 3), (3, 1), (1, 5), (5, 2), (2, 1), (1, 3), (3, 2), (2, 5), (5, 4), (4, 5), (5, 1)\}$$

Краткие методические указания

При подготовке к лабораторной работе необходимо обратить внимание на содержание основных теоретических вопросов, связанных определением эйлеровых циклов графа, изложенных на лекционных занятиях.

Отчет о выполнении лабораторной работы должен быть представлен в электронном виде (на языке программирования Python).

Шкала оценки

Баллы	Описание
5	Задание выполнено полностью и абсолютно правильно. Тестовое задание выполнено -включая изображение конца таблицы
4	Задание выполнено полностью и правильно, но решение содержит некоторые неточности и несущественные ошибки или выполнено 1 тестовое задание, например, без изображения
3	Задание выполнено не полностью, с существенными ошибками, но подход к решению, идея решения, метод правильны.
2	Задание выполнено частично, имеет ошибки, осуществлена попытка решения на основе правильных методов и идей решения.
0	Задание не выполнено.

Лабораторная работа № 8. Паросочетания. Венгерский алгоритм. Тема 2.6

Введение. Появление венгерского метода (венгерского алгоритма) связано с решением задачи о назначениях.

Суть: имеется 5 работ (столбцы) и столько же работников (строки), которые имеют разную оплату за работы (см матрицу) Нужно так распределить работников по работам, чтобы суммарная оплата была минимальной:

	1	2	3	4	5
A	32	28	4	26	4
B	17	19	4	17	4
C	4	4	5	4	4
D	17	14	4	14	4
E	21	16	4	13	4

Если выбирать минимальные значения наоборот, с учетом того, что каждый работник может занять только одну работу, можно получить (начиная с пятой строки) общую суммарную зарплату $4+4+4+17+28= 57$.

Венгерский алгоритм позволяет уменьшить это число до 39.

Определенным способом сводят матрицу, приведенную выше к матрице, где в каждой строке будет нулевое значение. Позиция нулей и указывает – какие цифры в вышеприведенной матрице взять:

18	14	0	12	0
3	5	0	3	0
0	0	11	0	10
3	0	0	0	0
8	3	1	0	1

из пятой строки нужно взять предпоследний элемент 13

из четвертой 14, из третьей первый элемент 4 и два минимальных элемента из первой и второй строки (по 4). Итого $13+14+4+4+4= 39$

Венгерский алгоритм. Программируя венгерский алгоритм, нужно выделить 5 шагов

Шаг 1. Вычтите минимальный элемент в каждой строке из своей строки (для всех строк)

Шаг 2. Вычтите минимальный элемент в каждом столбце из своего столбца (для всех столбцов)

Шаг 3. Используйте наименьшее количество горизонтальных или вертикальных линий, чтобы покрыть все нули, например,

15	0	0	0
0	50	20	25
35	5	0	10
0	65	50	65

Шаг 4. Поскольку общее количество горизонтальных и вертикальных линий равно 3, что меньше 4, введите Шаг 5

Шаг 5 Минимальное значение, которое не покрывается, равно 20, минимальное значение 20 вычитается из каждой строки, которая не покрывается, и минимальное значение 20 добавляется к каждому покрытому столбцу, а затем переходите к шагу 3.

20	0	5	0
0	45	20	20
35	0	0	5
0	60	50	60

 \rightarrow

20	0	5	0
- 20	25	0	0
35	0	0	5
- 20	40	30	40

 \rightarrow

40	0	5	0
0	25	0	0
55	0	0	5
0	40	30	40

Шаг 3. Используйте наименьшее количество горизонтальных или вертикальных линий, чтобы покрыть все нули

$$\begin{bmatrix} 40 & 0 & 5 & 0 \\ 0 & 25 & 0 & 0 \\ 55 & 0 & 0 & 5 \\ 0 & 40 & 30 & 40 \end{bmatrix}$$

Шаг 4. Поскольку общее количество горизонтальных и вертикальных линий равно 4, алгоритм завершается, и результат распределения показан на рисунке ниже.

$$\begin{bmatrix} 40 & 0 & 5 & 0 \\ 0 & 25 & 0 & 0 \\ 55 & 0 & 0 & 5 \\ 0 & 40 & 30 & 40 \end{bmatrix} \rightarrow \begin{bmatrix} 90 & 75 & 75 & 80 \\ 35 & 85 & 55 & 65 \\ 125 & 95 & 90 & 105 \\ 45 & 110 & 95 & 115 \end{bmatrix}$$

В этом примере исходной матрицей была крайняя правая матрица. Нули левой матрицы указывают на адреса значений, которые нужно выбрать в исходной матрице. В данном случае минимальная суммарная оплата составляет $45+95+55+80=275$.

Код алгоритма на языке Python можно посмотреть по ссылке <https://russianblogs.com/article/5547740509/>

Цели лабораторной работы

Сформировать у обучающихся умения и использования венгерского алгоритма для нахождения оптимального решения в рамках теоретических и практических исследований для последующего использования при решении задачи о назначениях. Повысить навыки программирования на языке Python

Планируемые результаты обучения

Формирование умения применения венгерского алгоритма для нахождения оптимального решения задачи и последующего применения при решении задач о назначениях. Формирование навыков программирования и использования методов естественнонаучных дисциплин в профессиональной деятельности для теоретического и экспериментального исследования.

Содержание лабораторной работы

ЗАДАНИЕ

1. По заданной таблице о назначениях найти оптимальное решение, применить венгерский алгоритм, распечатать суммарное значение заработной платы

Требуется:

1.Использовать программу на языке программирования Python для ввода данных согласно заданной задаче

2.Вывести :

1)минимальную стоимость и лучшие значения

2)изображения таблицы с промежуточным решением

ТЕСТОВЫЕ ДАННЫЕ

Начальная таблица

```
cost matrix =  
[[12 53 31 40 47]  
 [ 2 21 72 61 17]  
 [70 72 85 54 39]  
 [93 34 62 75 51]  
 [76 14 15  7 72]]
```

Результат работы алгоритма

```
min cost = 113  
best solution = [31, 2, 39, 34, 7]
```

Краткие методические указания

При подготовке к лабораторной работе необходимо обратить внимание на содержание основных теоретических вопросов, связанных определением парасочетаний, венгерского алгоритма, изложенных на лекционных занятиях.

Отчет о выполнении лабораторной работы должен быть представлен в электронном виде (на языке программирования Python).

Шкала оценки

Баллы	Описание
5	Задание выполнено полностью и абсолютно правильно. Тестовое задание выполнено -включая изображение конца таблицы
4	Задание выполнено полностью и правильно, но решение содержит некоторые неточности и несущественные ошибки или выполнено 1 тестовое задание, например, без изображения
3	Задание выполнено не полностью, с существенными ошибками, но подход к решению, идея решения, метод правильны.
2	Задание выполнено частично, имеет ошибки, осуществлена попытка решения на основе правильных методов и идей решения.
0	Задание не выполнено.

Лабораторная работа №9 «Потоки в сетях» тема 2.9

Цели лабораторной работы

Сформировать у обучающихся умения и навыки использования сетевых моделей в рамках теоретических и практических исследований для последующего принятия организационно-управленческих решений. Повысить навыки программирования на языке Python

Планируемые результаты обучения

Формирование умения использовать сетевые модели для последующего нахождения соответствующих организационно-управленческих решений. Формирование навыков программирования и использования методов естественнонаучных дисциплин в профессиональной деятельности для теоретического и экспериментального исследования.

Содержание лабораторной работы

1. Каждый обучающийся получает задание, содержащее таблично заданную информацию о сети с единственными источником и стоком.

Требуется:

1. Создать программу на языке программирования Python для ввода данных согласно типовому варианту

2. Вывести :

1) изображение сети в виде взвешенного орграфа;

2) максимальный поток в сети.

Типовой вариант.

Задание 1. Найти максимальный поток в сети с единственным источником и стоком.

a	s	s	s	2	2	4	4	5	1	6	6	3	7	t	t
b	2	4	5	4	1	5	t	6	3	8	7	t	t	8	7
c(a,b)	3	3	1	4	2	3	3	2	1	5	3	2	2	1	3

Задание 2. Найти максимальный поток в сети с несколькими источниками и стоками.

a	s ₁	s ₂	s ₂	s ₂	2	2	3	4	5	1	1	5	6	t ₁	t ₁	t ₂
b	2	2	4	3	1	5	6	6	6	t ₁	t ₂	t ₂	t ₃	t ₂	t	t ₃
c(a,b)	4	2	3	4	2	3	2	2	2	1	2	2	3	1	1	2

$$a(s_1)=8 \quad b(t_1)=3$$

$$a(s_2)=6 \quad b(t_2)=3$$

$$b(t_3)=4$$

Задание 4. Найти максимальный поток в сети с заданными ограничениями потока в источниках и стоках.

a	s ₁	s ₂	1	3	5	t ₂	s ₁	s ₂	3	t ₁	t ₂	3	1	1
b	3	5	s ₁	t ₂	2	t ₁	s ₃	2	4	5	4	5	4	t ₁
c(a,b)	2	2	0	4	1	3	2	3	2	4	0	4	4	2

$$a(s_1)=1 \quad a'(s_1)=2 \quad a(s_2)=1 \quad a'(s_2)=3 \quad a(s_3)=2 \quad a'(s_3)=4$$

$$b(t_1)=2 \quad b'(t_1)=4 \quad b(t_2)=1 \quad b'(t_2)=2$$

Задание 5. Построить допустимую циркуляцию или показать, что ее не существует (начальная циркуляция нулевая).

a	1	2	2	3	3	3	3	4	4	5	5	6
b	2	4	1	1	2	4	5	6	5	3	4	5
l(a,b)	2	1	0	2	1	1	2	2	1	1	1	1
c(a,b)	3	3	2	5	3	3	4	4	3	4	4	5

Задание 6. Построить допустимую циркуляцию или показать, что ее не существует (начальная циркуляция ненулевая).

a	1	1	1	1	2	3	3	4	5	5	5	5
b	2	3	4	5	3	4	2	5	2	3	1	4
l(a,b)	1	2	1	1	3	2	1	1	0	1	2	1
f(a,b)	2	3	1	2	2	3	2	4	1	2	2	1
c(a,b)	3	4	2	3	4	4	4	3	1	4	4	2

При подготовке к лабораторной работе необходимо обратить внимание на содержание основных теоретических вопросов, связанных с сетевыми моделями, изложенных на лекционных занятиях.

Отчет о выполнении лабораторной работы должен быть представлен в электронном виде (на языке программирования Python).

Шкала оценки

Баллы	Описание
15	Задание выполнено полностью и абсолютно правильно.
10-14	Задание выполнено полностью и правильно, но решение содержит некоторые неточности и несущественные ошибки.
5-9	Задание выполнено не полностью, с существенными ошибками, но подход к решению, идея решения, метод правильны.
1-4	Задание выполнено частично, имеет ошибки, осуществлена попытка решения на основе правильных методов и идей решения.
0	Задание не выполнено.