Чащина Е.А. г. Прокопьевск

Практическая работа № 1

Тема: Графическое изображение графов.

Цель: изучить основы теоретико-множественного и графического представлений графов, простейших свойств графов, получить практический навык задания и визуализации графа на плоскости; закрепить навыки построения графов по образцу в графических средах (программы для графического представления графов).

Задачи:

1. Закрепить знания основных понятий теории графов.
2. Приобрести практические умения использования специального программного обеспечения для моделирования.

Материальное обеспечение:

Программы для графического представления графов: grin_rus, Grafoanalizator1.3.3 rus, windisc ru.

Теоретическая часть:

Графические представления в широком смысле – любые наглядные отображения исследуемой системы, процесса, явления на плоскости. К ним могут быть отнесены рисунки, чертежи, графики зависимостей характеристик, планы-карты местности, блок-схемы процессов, диаграммы и т. д. Такие изображения наглядно представляют различные взаимосвязи, взаимообусловленности: топологическое (пространственное) расположение объектов, хронологические (временные) зависимости процессов и явлений, логические, структурные, причинно-следственные и другие взаимосвязи.

Графические представления – удобный способ иллюстрации содержания различных понятий, относящихся к другим способам формализованных представлений (например, диаграммы Венна и другие графические иллюстрации основных теоретико-множественных и логических представлений).

Всё более распространенными становятся представления количественных характеристик, взаимосвязей между объектами в виде разного рода одно-, двух- и более мерных гистограмм, круговых диаграмм, других аналогичных способов представления в виде тех или иных геометрических фигур, по наглядным характеристикам которых (высоте, ширине, площади, радиусу и пр.) можно судить о количественных соотношениях сравниваемых объектов, значительно упрощая их анализ.

Мощным и наиболее исследованным классом объектов, относящихся к графическим представлениям, являются так называемые графы, изучаемые в теории графов.

Теория графов – это раздел дискретной математики, исследующий свойства конечных множеств с заданными отношениями между их элементами. Как прикладная дисциплина теория графов позволяет описывать и исследовать многие технические, экономические, биологические и социальные системы.

Основные понятия теории графов

Граф – это система, которая интуитивно может быть рассмотрена как множество кружков и множество соединяющих их линий (геометрический способ задания графа – см. рисунок 1). Кружки называются вершинами графа, линии со стрелками – дугами, без стрелок – рёбрами.

Граф, в котором направление линий не выделяется (все линии являются ребрами), называется неориентированным; граф, в котором направление линий принципиально (линии являются дугами) называется ориентированным.

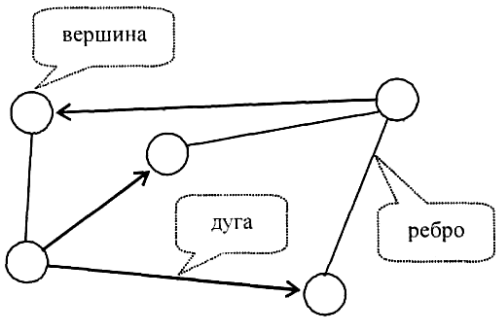


Рис. 1. Пример графа

Теория графов может рассматриваться как раздел дискретной математики (точнее – теории множеств), и тогда определение графа таково:

Граф – это конечное множество X, состоящее из n элементов $(X = \{1, 2, ... n\})$ называемых вершинами графа, и подмножество V декартова произведения $X \times X$, называемое множеством дуг.

Ориентированным графом G (орграфом) называется совокупность (X, V).

Неориентированным графом называется совокупность множеств X и множества неупорядоченных пар элементов, каждый из которых принадлежит множеству X.

Дугу между вершинами i и j, $i, j \in X$, будем обозначать (i, j). Число дуг графа будем обозначать $m(V = (v_1, v_2, ... v_m))$.

Подграфом называется часть графа, образованная подмножеством вершин вместе со всеми рёбрами (дугами), соединяющими вершины из этого множества. Если в графе удалить часть рёбер (дуг), то получим частичный граф.

Две вершины называются смежными, если они соединены ребром (дугой). Смежные вершины называются граничными вершинами соответствующего ребра (дуги), а это ребро (дуга) - инцидентным соответствующим вершинам.

Граф называется полным, если каждые две вершины его соединены одним и только одним ребром.

Граф, для которого из $(i, j) \in V$ следует $(j, i) \in V$ называется симметричным. Если из $(i, j) \in V$ следует $(j, i) \notin V$, то соответствующий граф называется антисимметричным.

Язык графов оказывается удобным для описания многих физических, технических, экономических, биологических, социальных и других систем.

Вершины в графе могут отличаться друг от друга тем, скольким рёбрам они принадлежат.

Степень вершины называется число рёбер графа, которым принадлежит эта вершина. Степень графа ещё называют его валентностью и обозначают $d(v)$. Вершина графа, для которой $d(v) = 0$, является изолированной, для которой $d(v) = 1$ – висячей.

Вершина называется нечётной, если $d(v)$ – нечётное число. Вершина называется чётной, если $d(v)$ – чётное число. Степень каждой вершины полного графа на единицу меньше числа его вершин.

В графе $G(V, E)$ сумма степеней всех его вершин – число чётное, равное удвоенному числу рёбер графа. Число нечётных вершин любого графа чётно. Во всяком графе с n вершинами, где $n \geq 2$, всегда найдутся, по меньшей мере, две вершины с одинаковыми степенями.

Если в графе с n вершинами $(n \geq 2)$ в точности две вершины имеют одинаковую степень, то в этом графе всегда найдётся либо в точности одна вершина степени 0, либо в точности одна вершина степени $n-1$.

1. Маршруты, цепи, циклы

Маршрутом в графе называется чередующаяся последовательность вершин и рёбер, в которой любые два соседних элемента инцидентны: $v_0, e_1, v_1, e_2, v_2, ..., e_k, v_k$.

Если $v_0 = v_k$, то маршрут замкнут, в противном случае открыт.

Путём называется последовательность дуг (в ориентированном графе), такая, что конец одной дуги является началом другой дуги.

Простой путь – путь, в котором ни одна дуга не встречается дважды.

Контур – путь, у которого конечная вершина совпадает с начальной вершиной.

Длиной пути (контура) называется число дуг пути (или сумма длин его дуг, если последние заданы).

Цепью называется множество рёбер (в неориентированном графе), которые можно расположить так, что конец (в этом расположении) одного ребра является началом другого. Другое определение: цепь – последовательность смежных вершин. Замкнутая цепь называется циклом. Можно определить простые и элементарные цепи.

Элементарная цепь (цикл, путь, контур), проходящая через все вершины графа называется гамильтоновой цепью.

Простая цепь (цикл, путь, контур), содержащая все рёбра (дуги) графа называется эйлеровой цепью.

Если любые две вершины графа можно соединить цепью, то граф называется связным. Если граф не является связным, то его можно разбить на связные подграфы, называемые компонентами.

Связностью графа называется минимальное число рёбер, после удаления которых граф становится несвязным.

2. Ориентированные графы

Если элементы множества E графа $G(V,E)$ – упорядоченные пары, то граф называется ориентированным или орграфом.

Ребро e графа G называется ориентированным, если одну вершину считают началом ребра, а другую – концом, на рисунке его изображают стрелкой между вершинами. Таким образом, граф, все рёбра которого ориентированы, называется ориентированным графом.

Одна и та же вершина ориентированного графа может служить началом для одних рёбер и концом для других, поэтому различают две степени вершины: степень выхода и степень входа.

Степенью выхода вершины орграфа называется число выходящих из вершины рёбер.

Степенью входа вершины орграфа называется число входящих в вершину рёбер.

В орграфах в зависимости от сочетаний степеней входа и выхода для данной вершины рассматривается три случая.

Изолированной вершиной называется вершина, у которой и степень входа и степень выхода равна 0.

Чащина Е.А.

Источником называется вершина, степень выхода которой положительна, а степень входа равна 0.

Стоком называется вершина, степень входа которой положительна, а степень выхода равна 0.

Путём в ориентированном графе называется последовательность ориентированных рёбер, т. е. для орграфов цепь называется путём.

Простым путём в ориентированном графе называется путь, в котором ни одна вершина не содержится более одного раза.

Замкнутый путь в ориентированном графе называется ориентированным циклом или контуром.

Длиной пути называется число рёбер в этом пути.

Полным ориентированным графом называется граф, каждая пара вершин которого соединена в точности одним ориентированным ребром.

Всякий полный ориентированный граф с n вершинами имеет простой ориентированный путь, проходящий через все вершины графа.

Петлёй называется ребро, у которого начальная и конечная вершины совпадают. Петля обычно считается неориентированной.

Мультиграфом называется граф, в котором пара вершин соединяется несколькими различными рёбрами. Для ориентированного мультиграфа вершины v_i и v_j могут соединяться несколькими рёбрами в каждом из направлений.

3. Изоморфизм графов

Два графа $G_1(V_1,E_1)$ и $G_2(V_2,E_2)$ называются изоморфными, если между множествами их вершин существует биективное (взаимнооднозначное) соответствие, такое, что вершины соединены рёбрами в одном из графов в том и только в том случае, когда соответствующие им вершины соединены в другом графе. Если рёбра ориентированы, то их направление в изоморфных графах должно совпадать. Изоморфизм графов есть отношение эквивалентности, так как обладает свойствами рефлексивности, симметричности, транзитивности. Для того чтобы граф G_1 был изоморфен графу G_2, необходимо и достаточно существования такой подстановки, которая бы установила взаимнооднозначное соответствие между вершинами графа, а также между их рёбрами.

При замене графа любым ему изоморфным все свойства графа сохраняются. Строго говоря, графы отличающиеся только нумерацией вершин, являются изоморфными.

Алгоритм распознания изоморфизма двух графов $G_1(X;E)$ и $G_1(Y;E)$.

1. Подсчитаем число вершин каждого графа (число вершин должно совпадать, в противном случае графы неизоморфные).

2. Выписываем все элементы обоих графов в естественной упорядоченности и определяем пары $(x_i;x_j)$ и $(y_i;y_j)$ для каждого элемента, где x_i, y_i – число исходов для каждой вершины графов G_1 и G_2, а x_j, y_j – число заходов для соответствующих графов.

3. Для каждого элемента x графа G_1 ищем такой элемент y графа G_2, что выполняется условие: число исходов x совпадает с числом исходов y, и число заходов

x совпадает с числом заходов y. Найденные элементы x и y соединяем ребром, т. е. строим граф соответствия (если соответствия нет, то графы не изоморфны).

4. Выписываем подстановку, которая переводит граф G_1 в граф G_1.

4. Плоские графы

Граф $G(V,E)$ называется плоским, если на плоскости его можно изобразить так, что все пересечения его рёбер являются вершинами графа $G(V,E)$.

В качестве характеристики плоского представления графа вводится понятие грани.

Гранью в плоском представлении графа называется часть плоскости, ограниченная простым циклом и не содержащая внутри других циклов.

5. Операции над графами

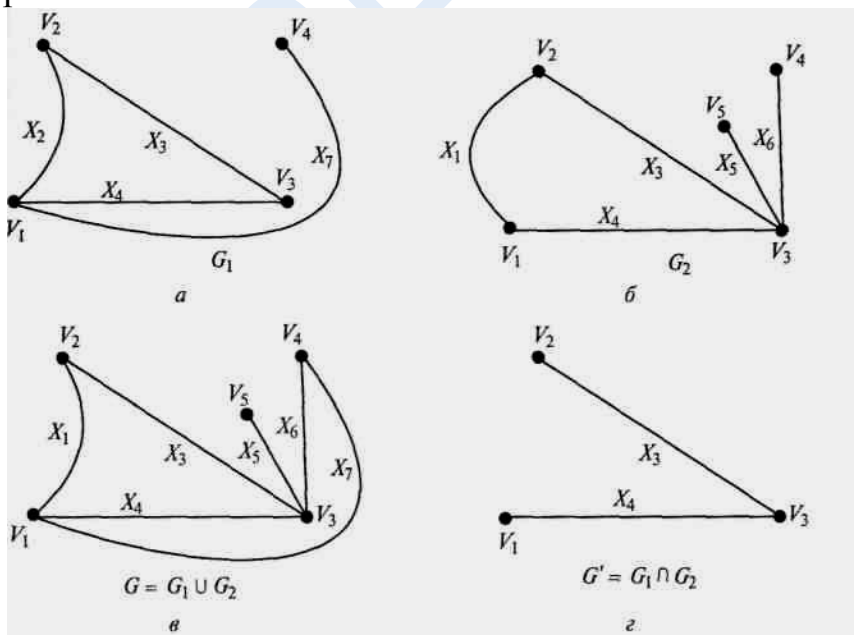
Рассмотрим графы $G_1(V_1,E_1)$ и $G_2(V_2,E_2)$.

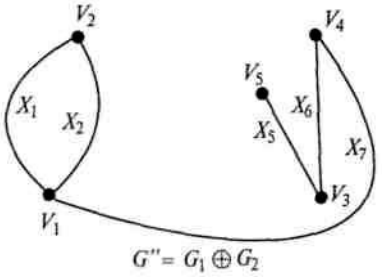
а) <u>Дополнением</u> графа $G_1(V_1,E_1)$ называется граф $\overline{G_1}(V_1,\overline{E_1})$, множеством вершин которого является множество V_1, а множеством его рёбер является множество $\overline{E_1}=\{e\in V_1\times V_2 : e\notin E_1\}$.

б) <u>Объединением</u> графов $G_1(V_1,E_1)$ и $G_2(V_2,E_2)$ при условии, что $V_1\cap V_2=\varnothing$; $E_1\cap E_2=\varnothing$, называется граф $G_1(V_1,E_1)\cup G_2(V_2,E_2)$, множеством вершин которого является множество $V_1\cup V_2$, а множеством его рёбер является множество $E_1\cup E_2$.

в) <u>Пересечением</u> графов $G_1(V_1,E_1)$ и $G_2(V_2,E_2)$ называется граф $G_1(V_1,E_1)\cap G_2(V_2,E_2)$, множеством вершин которого является множество $V_1\cap V_2$, а множеством его рёбер является множество $E_1\cap E_2$.

г) <u>Суммой по модулю два</u> графов $G_1(V_1,E_1)$ и $G_2(V_2,E_2)$ при условии, что $V_1\cap V_2=\varnothing$; $E_1\cap E_2=\varnothing$, называется граф $G_1(V_1,E_1)\oplus G_2(V_2,E_2)$, множеством вершин которого является множество $V_1\cup V_2$, а множеством его рёбер – множество $E_1\oplus E_2$. Т. е. этот граф не имеет изолированных вершин и состоит только из рёбер, присутствующих либо в первом графе, либо во втором графе, но не в обоих графах одновременно.





д

6. *Способы задания графов*

Существуют три эквивалентных способа задания графов: аналитический, геометрический и матричный. Рассмотрим каждый из них.

Аналитический способ задания графов

Граф $G(V,E)$ задан, если задано множество элементов V и отображение E множеств V в V. Отображение E может быть как однозначным, так и многозначным.

Пусть дано множество $V = \{v_1, v_2, ...v_n\}$, которое имеет мощность $|V| = n$.

Для того чтобы задать отображение E на V , необходимо каждому элементу $v_i \in V$ поставить в соответствие некоторое подмножество множества V, которому соответствует отображение E. Это подмножество обозначают через E_{v_i}. Поэтому $E_{v_i} \subset V$. Совокупность двух объектов: множества V и отображение E на V задаёт некоторый граф.

Другой формой аналитического способа задания является задание графа как совокупности множества элементов V и подмножества множества упорядоченных пар $\langle v_i, v_j \rangle \in V \times V$.

Геометрический способ задания графов

Множество элементов V графа G изображают кружками, это множество вершин. Каждую вершину $v_i \in V$ соединяют линиями с теми вершинами $v_i \in V$, для которых выполняется условие $v_i \in E_{v_i}$. Множество линий, которое соответствует множеству упорядоченных пар $\langle v_i, v_j \rangle$, есть множество рёбер.

Матричный способ задания графов

Квадратная матрица $\begin{bmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ ... & ... & ... & ... \\ a_{n1} & a_{n2} & ... & a_{nn} \end{bmatrix}$, элементами которой являются нули и единицы, а также некоторое число m, называется <u>матрицей смежности</u> графа $G(V,E)$ тогда и только тогда, когда её элементы образуются по следующему правилу: элемент a_{ij}, стоящий на пересечении v_i – й строки и v_j – го столбца, равен единице, если имеется ребро, идущее из вершины v_i в вершину v_j, и a_{ij} равен нулю в противном случае. Элемент a_{ij} равен единице, если при вершине v_i имеется петля, и равен нулю в противном случае. Элемент a_{ij} равен некоторому числу m, где m – число рёбер графа, идущее из вершины v_i в вершину v_j.

Таким образом, если граф $G(V,E)$ задан одним из указанных способов: аналитическим, геометрическим или матричным, всегда можно перейти к любому

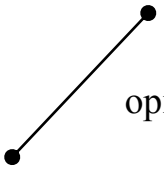
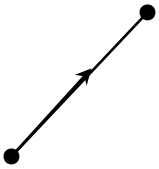
другому способу задания. Наиболее часто для задания графа используется аналитический и матричный способы, а геометрический способ служит для иллюстрации полученных результатов.

Инструкция к практической работе

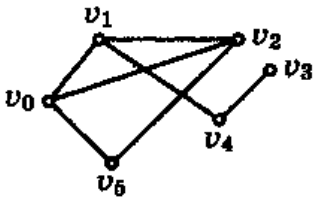
Задание 1. Изобразите графически:

1. Неориентированное и ориентированное ребро;
2. Неориентированный граф G(*V*,*E*), заданный множеством $V=\{v_0, v_1, v_2, v_3, v_4, v_5\}$ $E(v0)=\{v_1,v_2\}=\{v_0,v_2,v_4\}$; $E(v_1)=\{v_0,v_2,v_4\}$; $E(v_2)=\{v_0,v_1,v_5\}$; $E(v_3)=\{v_4\}$; $E(v_5)=\{v_2\}$;
3. Плоский граф;
4. Полный неориентированный граф на трех, четырех и пяти вершинах;
5. Неполный ориентированный граф на пяти вершинах;
6. Петлю графа;
7. Неориентированный и ориентированный мультиграф.

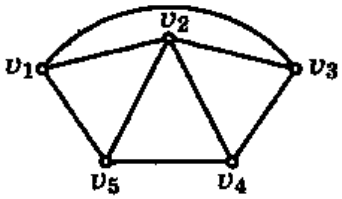
Решение:

1) Неориентированное ребро  ориентированное ребро 

2)

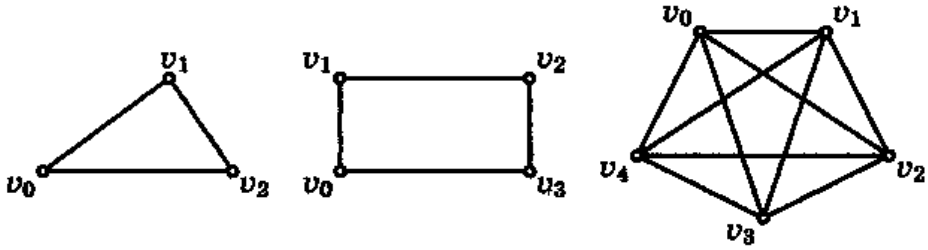


3)

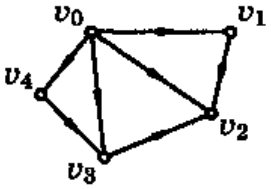


Плоский граф

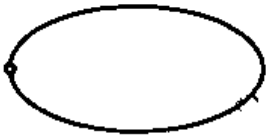
4)



5)



6)

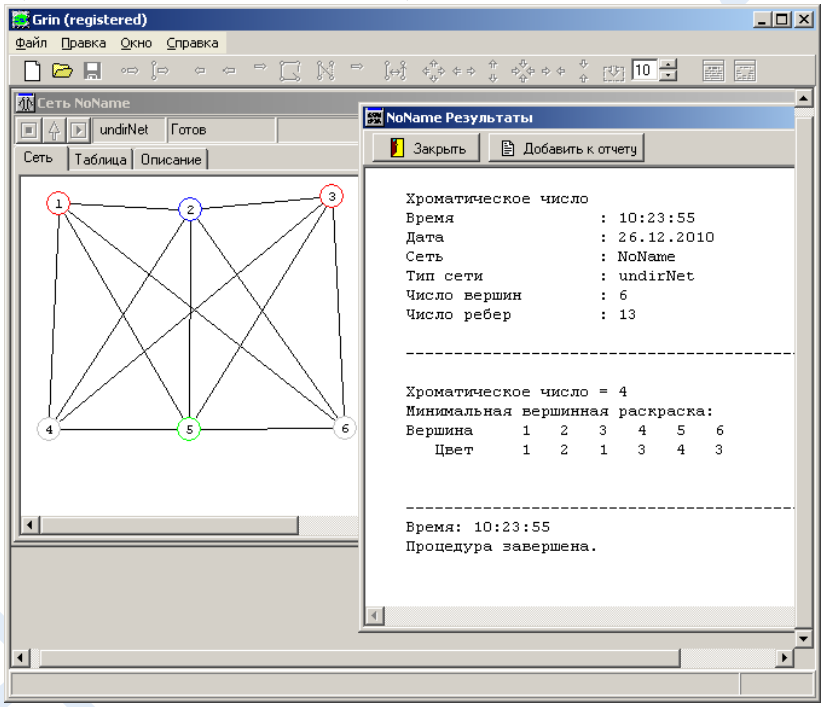


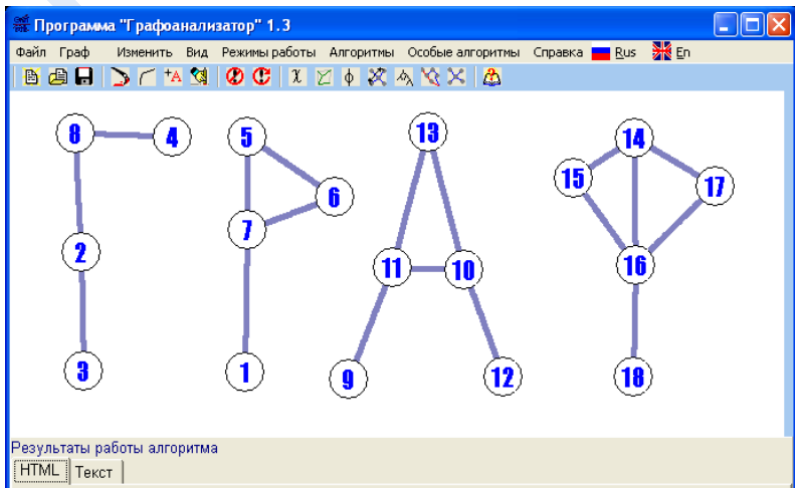
7)

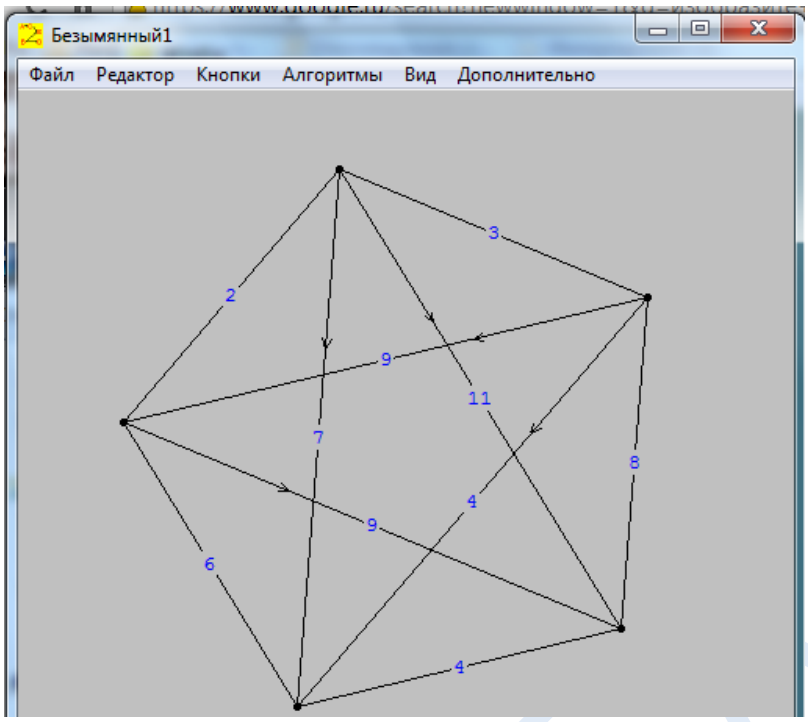
Мультиграф

Ориентированный мультиграф

Задание 2. Изобразить графы в программах:
grin_rus, Grafoanalizator1.3.3 rus, windisc ru.







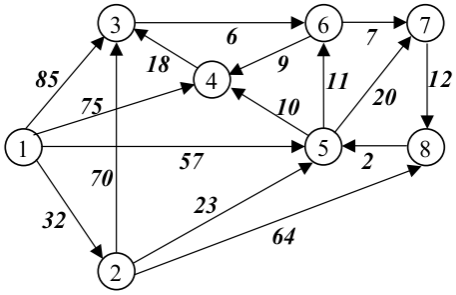
Задание:

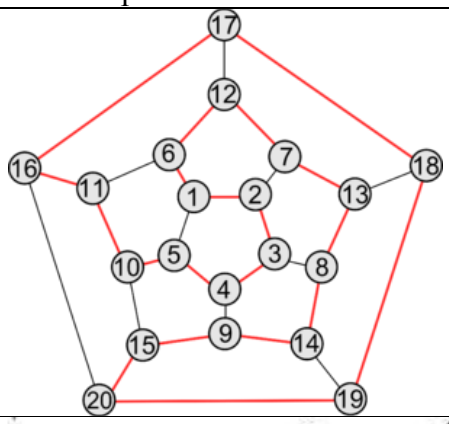
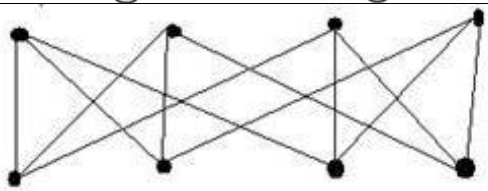
Вариант № 1

Задание 1. Выполните задание по образцу.

Изобразите графически:
***G*(*V*,*E*)** - орграф.
***V*={1,2,3,4}, *E*={(1, 2), (4, 3), (3, 4), (3, 1), (4, 1)}.**

Задание 2. Изобразите графы в соответствующих программах. Полученные графы сохранить в свои папки.

Граф	Программа
	Grafoanalizator1.3.3 rus

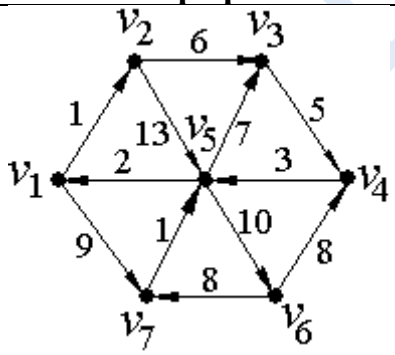
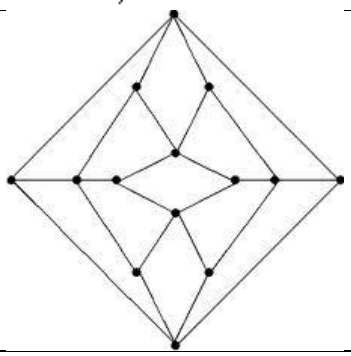
	grin_rus
	windisc ru

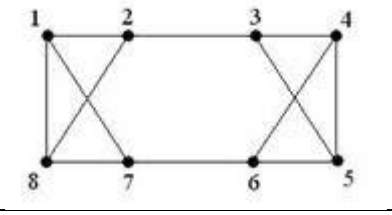
Вариант № 2

Задание 1. Выполните задание по образцу.

Изобразите графически:
***G*(*V*,*E*)** - орграф.
***V*={1,2,3,4,5}, *E*={(1, 2), (4, 3), (3, 5), (5, 1), (4, 1)}.**

Задание 2. Изобразите графы в соответствующих программах. Полученные графы сохранить в свои папки.

Граф	Программа
	Grafoanalizator1.3.3 rus
	grin_rus

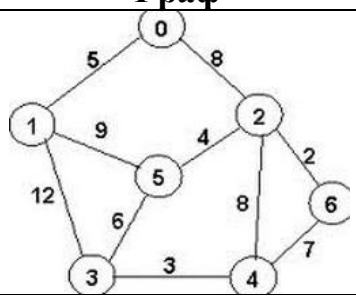
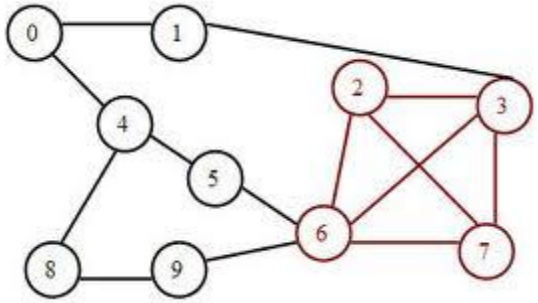
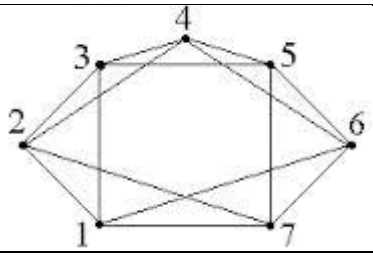
	windisc ru

Вариант № 3

Задание 1. Выполните задание по образцу.

Изобразите графически:
***G*(*V*,*E*)** - орграф.
***V*={1,2,3,4,5}, *E*={(1, 3), (2, 3), (1, 5), (2, 4), (1, 2)}.**

Задание 2. Изобразите графы в соответствующих программах. Полученные графы сохранить в свои папки.

Граф	Программа
	Grafoanalizator1.3.3 rus
	grin_rus
	windisc ru

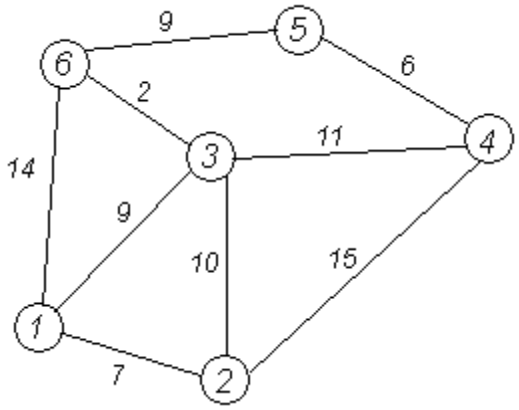
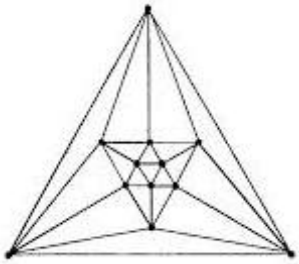
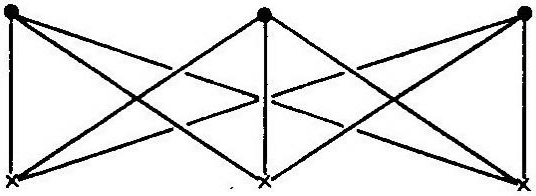
Вариант № 4

Задание 1. Выполните задание по образцу.

Изобразите графически:
***G*(*V*,*E*)** - орграф.

***V*={1,2,3,4,5,6}, *E*={(1, 6), (4, 5), (1, 2), (2, 3), (3, 6)}.**

Задание 2. Изобразите графы в соответствующих программах. Полученные графы сохранить в свои папки.

Граф	Программа
	Grafoanalizator1.3.3 rus
	grin_rus
	windisc ru

Порядок выполнения работы:

1. Изучить инструкцию к практической работе.
2. Выполнить задание.
3. Оформить отчет.

Содержание отчета:

1. Тема.
2. Цель.
3. Задачи.
4. Материальное обеспечение.
5. Практическое задание.

Вопросы для самоконтроля:

1.Что называется графом? Ориентированным графом? Приведите примеры.
2.Что такое степень вершины?
3.Перечислите основные понятия, связанные с неориентированными графами.
4.Перечислите основные понятия, связанные с орграфами.
5.В чем состоит аналитический способ задания графа?
6.В чем состоит геометрический способ задания графа?
7.В чем состоит матричный способ задания графа?
8.Что называется маршрутом, циклом и цепью графа?
9.Сформулируйте понятие связности графа. Какой граф называют связным?
10.Какие два графа называются изоморфными?
11.Сформулируйте алгоритм изоморфизма двух графов.
12.Перечислите операции над графами.

Чащина Е.А.

Практическая работа № 2,3

Тема: Решение задач по теории графов. Эйлеровы и Гамильтоновы графы.

Цель: изучить алгоритм поиска эйлерова, гамильтонова цикла (пути) в графе, рассмотреть на конкретных примерах ориентированные и неориентированные графы.

Задачи:

1. Закрепить знания основных понятий теории графов.
2. Приобрести практические умения использования специального программного обеспечения для моделирования.
3. Использовать математический аппарат теории графов

Материальное обеспечение:

Программа анализатор графов: Grafoanalizator1.3.3 rus, практическое задание.

Теоретическая часть:

Эйлеровы графы

К задачам на Эйлеровы графы относятся головоломки, в которых требуется вычертить на плоскости одним росчерком замкнутые кривые, обводя каждый участок в точности один раз. Введём следующие понятия.

Эйлеровым путём в графе называется путь, содержащий все рёбра графа.

Эйлеровым циклом или эйлеровой цепью называется цикл, содержащий все рёбра графа и притом по одному разу.

Граф, обладающий эйлеровым циклом, называется эйлеровым графом.

Замкнутую линию, если её можно начертить, не отрывая карандаша от бумаги, проходя при этом каждый участок в точности один раз, принято называть уникурсальной.

Рисунок графа, обладающий эйлеровым путём или эйлеровым циклом, является уникурсальной линией.

Докажем следующие две теоремы

Теорема 1. Если граф $G(V,E)$ обладает эйлеровым циклом, то он связный и все его вершины четные.

Доказательство. Связность графа следует из определения эйлерова цикла. Эйлеров цикл содержит каждое ребро и притом только один раз, поэтому, сколько раз эйлеров путь приведет конец карандаша в вершину, столько и выведет, причём уже по одному ребру. Следовательно, степень каждой вершины графа должна состоять из двух одинаковых слагаемых: одно – результат подсчета входов в вершину, другое – выходов.

Теорема 2. Если граф $G(V,E)$ связный и все его вершины четные, то он обладает эйлеровым циклом.

Доказательство. Если начать путь из произвольной вершины графа $G(V,E)$, то найдётся цикл, содержащий все рёбра графа. Пусть v_i - произвольная вершина. Из v_i начнём путь по l по одному из рёбер и продолжим его, проходя каждый раз по новому ребру. Все вершины графа имеют чётные степени, поэтому если l есть «выход» из v_i, то должен быть и «вход» в v_i, также как и для любой вершины другой вершины. И если есть «вход» в вершину, то должен быть и «выход». Так как число

ребер конечно, то это путь должен окончиться, причём в вершине v_i. Если путь, замкнувшийся в v_i, проходит через все рёбра графа, то мы получим искомый эйлеров цикл.

Для построения эйлерова цикла в связном графе со всеми вершинами чётной степени применяется следующий алгоритм:

1. Выйти из произвольной вершины v_i. Каждое пройденное ребро зачеркнуть. Если путь l_1 замыкается в v_i и проходит через все рёбра графа, то получим искомый эйлеров цикл.

2. Если остались непройденные рёбра, то должна существовать вершина v_2, принадлежащая l_1 и ребру, не вошедшему в l_1.

3. Так как v_2 – чётная, то число рёбер, которым принадлежит v_2 и которые не вошли в путь l_1, тоже чётно. Начнём новый путь l_2 из v_2 и используем только рёбра, не принадлежащие l_1. Этот путь кончится в v_2.

4. Объединим теперь оба цикла: из v_i пройдём по пути l_1 к v_2, затем по l_2 и, вернувшись в v_2, пройдём по оставшейся части l_1 обратно в v_i.

5. Если снова найдутся рёбра, которые не вошли в путь, то найдём новые циклы. Так как число рёбер и вершин конечно, то процесс закончится.

Таким образом, замкнутую фигуру, в которой все вершины чётные, можно начертить одним росчерком без повторений и начиная с любой точки.

На практике эйлеровым графом может быть план выставки; это позволяет расставить указатели маршрута, чтобы посетитель смог пройти по каждому залу в точности по одному разу.

Гамильтоновы графы

Граф, обладающий гамильтоновым циклом, называется гамильтоновым графом.

Гамильтоновым циклом, или путём в графе, называется цикл, или путь, проходящий через каждую вершину графа в точности по одному разу.

Эйлеровы и гамильтоновы пути сходны по способу задания. Первые содержат все рёбра, и притом по одному разу, вторые – все вершины по одному разу. Но, несмотря на внешнее сходство, задачи их отыскания резко отличаются по степени трудности. Для решения вопроса о существовании эйлерова цикла в графе достаточно выяснить, все ли его вершины чётные.

Критерий же существования гамильтонова цикла на произвольном графе ещё не найден.

Однако есть несколько достаточных условий существования гамильтоновых циклов в графе:

1. Всякий полный граф является гамильтоновым, так как он содержит простой цикл, которому принадлежат все вершины данного графа.

2. Если граф, помимо простого цикла, проходящего через все его вершины, содержит и другие рёбра, то он также является гамильтоновым.

3. Если граф имеет один гамильтонов цикл, то он может иметь и другие гамильтоновы циклы.

Алгоритма проверки существования эйлерова пути

Представим динамику выполнения алгоритма проверки существования эйлерова пути (цикла) из вершины 0 для представленного на рисунке 1 графа.

Цикл существует.

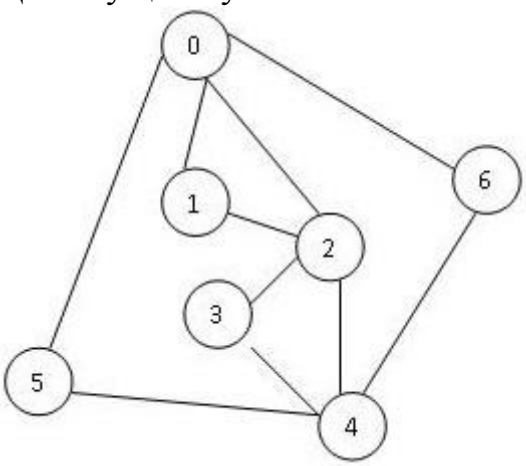


Рисунок 1

Например, один из возможных путей прохождения всех ребер графа из вершины 0 может быть следующим:

0 – 1 – 2 – 0 – 6 – 4 – 2 – 3 – 4 – 5 – 0

В приведенном списке вершин, следующих за 0, каждая вершина является одновременно концом предыдущего ребра и началом следующего.

В соответствии с алгоритмом:

0 – степень 4;
1 – 2;
2 – 4;
3 – 2;
4 – 4;
5 – 2;
6 – 2;

Степени всех вершин четные, следовательно, эйлеров цикл в данном графе существует.

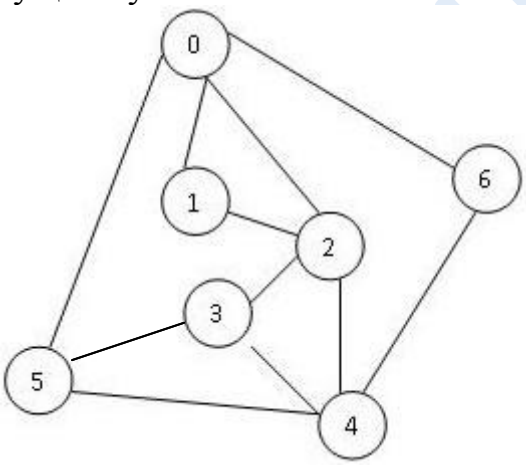


Рисунок 2

Граф, изображенный на рисунке 2 отличается от рисунка 1 только добавлением ребра (3 – 5).

При этом степени вершин 3 и 5 стали нечетными.

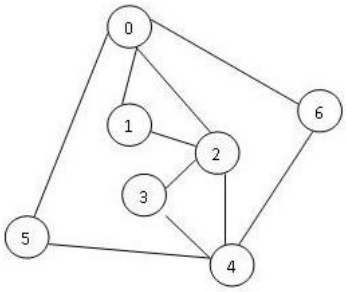
Согласно алгоритму проверки существования эйлерова цикла, основывающемуся на проверке четности степени каждой вершины, в данном графе цикла быть не может.

Однако, если учесть следствие, по которому в точности две вершины имеют нечетную степень, то и в графе, изображенном на рисунке 1 должен существовать эйлеров путь.

Пример такого пути: 3 – 2 – 4- 3 – 5 – 4 – 6 – 0 – 2 – 1 – 0 – 5.
При этом две вершины, имеющие нечетную степень, находятся на концах такого пути.

Алгоритма поиска гамильтонова пути

Представим динамику выполнения рекурсивного алгоритма поиска гамильтонова пути (цикла) из вершины 0 для графа, представленного на рисунке 3.



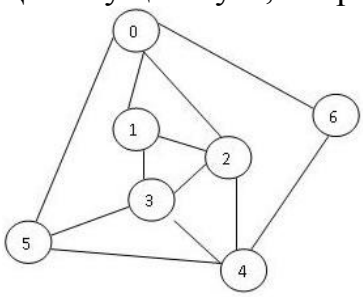
Рисунке 3.

Цикла не существует.
0-1
1-2
2-3
3-4
4-5
4-6
2-4
4-3
4-5
4-6
0-2
2-1
2-3
3-4
4-5
4-6
2-4
4-3
4-5
4-6
0-5
5-4
4-2
2-1
2-3
4-3
3-2
2-1
4-6
0-6
6-4

Чащина Е.А.

4-2
2-1
4-3
3-2
2-1
4-5
Представим динамику выполнения рекурсивного алгоритма поиска гамильтонова пути для представленного на рисунке 4 графа.

Цикл существует, например: 0 – 6 – 4 – 2 – 1 – 3 – 5 – 0.



Рисунке 4.

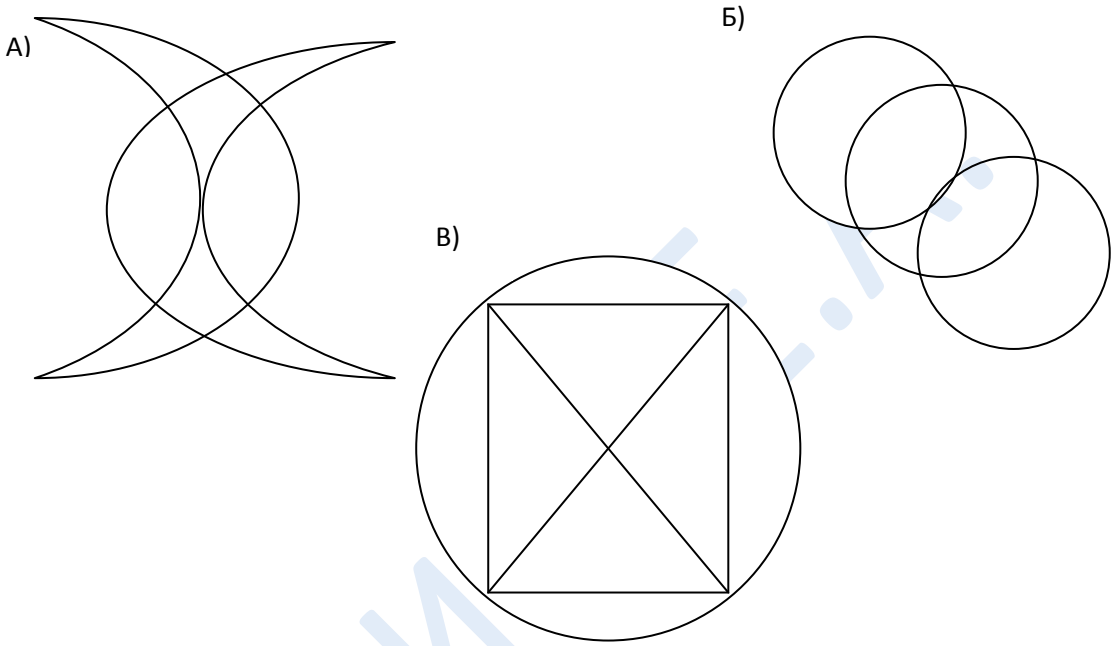
Продемонстрируем поиск цикла от вершины 1.
1-0
0-5
5-3
3-2
2-4
4-6
3-4
4-2
4-6
5-4
4-2
2-3
4-6
0-6
6-4
4-2
2-3
3-5
4-3
3-2
3-5
4-5
5-3
3-2
2-1

Искомый путь 1 – 0 – 6 – 4 – 5- 3 – 2 – 1 .

Инструкция к практической работе

Чащина Е.А.

1. Существует ли эйлеров цикл в графе G. Если существует, найдите его.



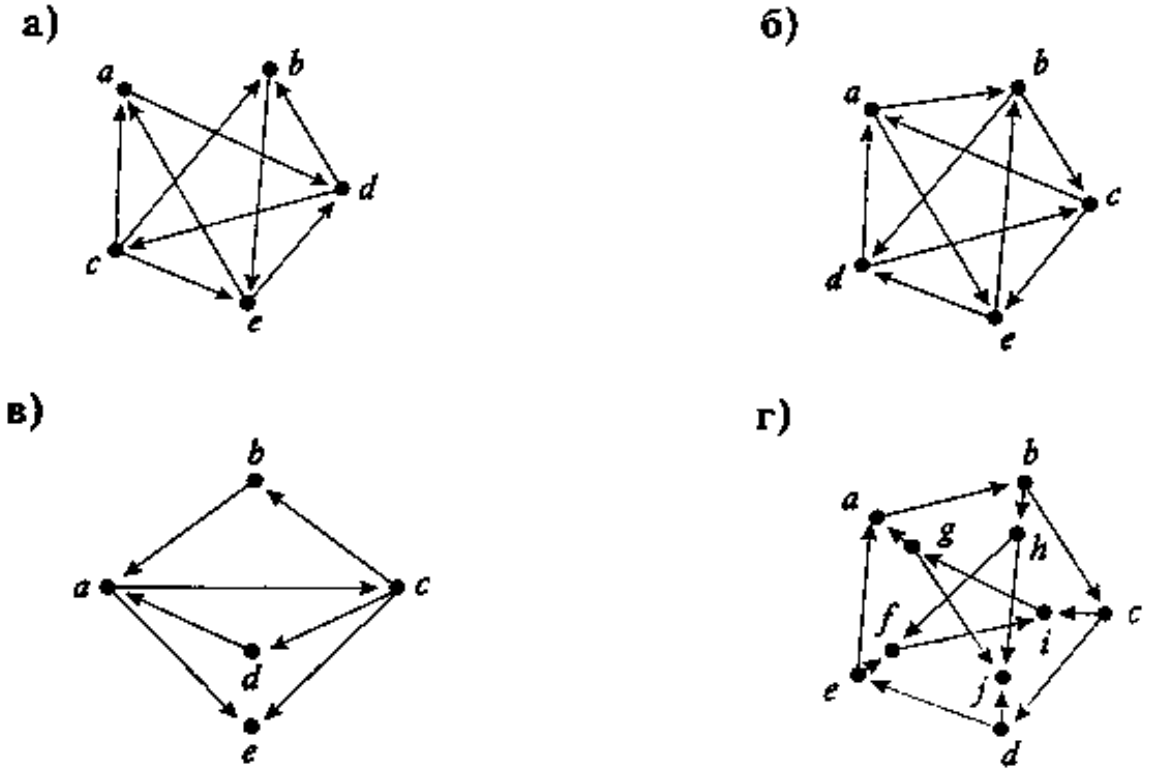
Решение:

А) Так как каждая вершина имеет чётную степень, то по критерию в этом графе существует эйлеров цикл: 1,4,6,9,10,8,5,3,2,4,7,10,11,8,6,5,2,1

Б) В этом графе также каждая вершина имеет чётную степень, значит, существует и эйлеров цикл: 1,2,3,4,5,3,1,4,5,2,1

В) Здесь каждая вершина имеет степень 5, то есть нечётную, следовательно, в этом графе (по критерию) нет эйлерова цикла.

2. Какие из следующих ориентированных графов имеют эйлеровы циклы?



Решение:

а) Граф связный, найдём степени входа и выхода вершин (по теореме 5 степени входа и выхода каждой вершины должны совпадать):

indeg(a)=2, outdeg(a)=1, то есть нашлась вершина, у которой не совпадают степени входа и выхода, значит, граф не имеет эйлерова цикла.

б) Граф связный, найдём степени вершин:

indeg(a)=2 outdeg(a)=2

indeg(b)=2 outdeg(b)=2

indeg(c)=2 outdeg(c)=2

indeg(d)=2 outdeg(d)=2

indeg(e)=2 outdeg(e)=2

Следовательно, по теореме 5, граф имеет эйлеров цикл.

в) Граф связный, найдём степени вершин:

indeg(a)=2 outdeg(a)=2

indeg(b)=1 outdeg(b)=1

indeg(c)=3 outdeg(c)=1

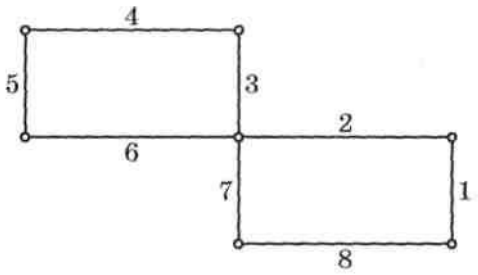
Условия теоремы 5 не выполняются, значит, граф не имеет эйлерова цикла.

г) Граф связный, найдём степени вершин:

indeg(a)=2 outdeg(a)=1

Следовательно, т.к. условия теоремы 5 не выполняются то, граф не имеет эйлерова цикла.

2. Задача. Найдите эйлеров цикл в эйлеровом графе



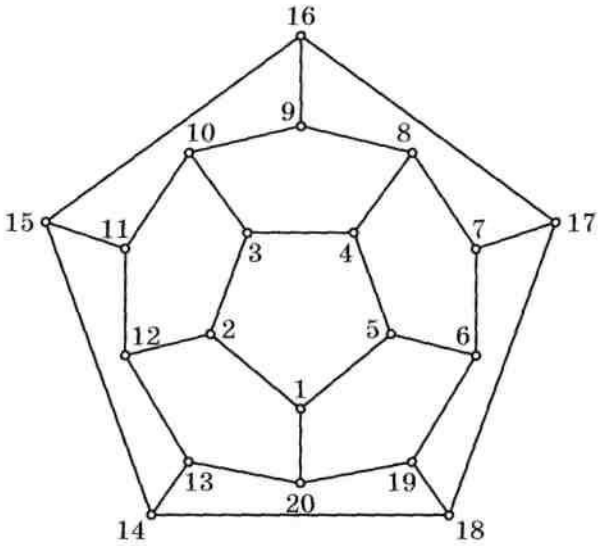
Решение. После выбора вершины а и прохождении рёбер 1 и 2 имеются три возможности выбора: рёбра 3, 6 или 7. Выбираем ребро 3 или 6. Например, ребро 3. Далее обходим оставшиеся рёбра и получаем эйлеров цикл 1, 2, 3, 4, 5, 6, 7, 8.

3.Задача. Найдите цикл, содержащий все вершины додекаэдра, причём в точности по одному разу каждую.

Решение.

Этот цикл: 1, 2, 3, 4, 5, 6, 19, 18, 14, 15, 16, 17, 7, 8, 9, 10, 11, 12, 13, 20.

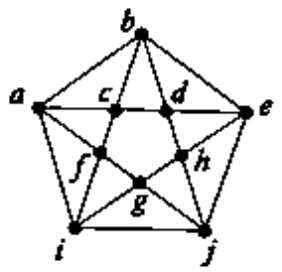
Этот цикл называется гамильтоновым циклом.



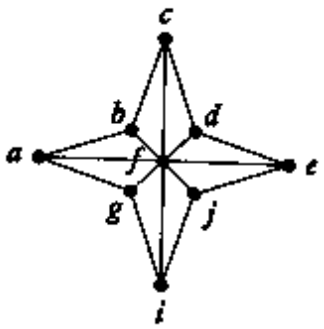
Задание:

1. Проработать алгоритм выполнения поиска эйлерова и гамильтонова пути (изобразите графы, содержащие эти пути)
2. Среди приведённых ниже графов найдите те, которые имеют эйлеров и гамильтонов цикл. Результат проверить при помощи программы Grafoanalizator1.3.3.

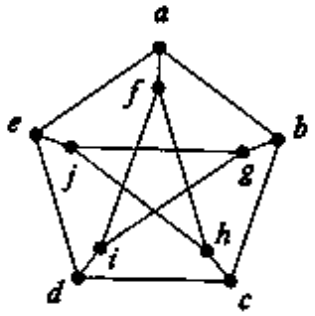
а)



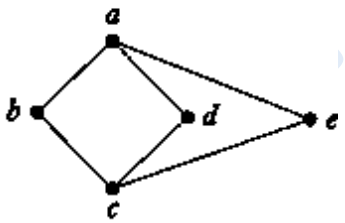
б)



в)

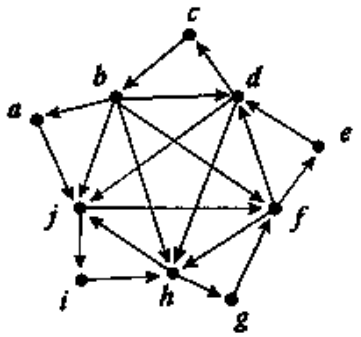


г)

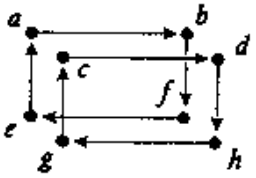


2. Какие из следующих ориентированных графов имеют эйлеровы и гамильтоновы циклы? Результат проверить при помощи программы Grafoanalizator1.3.3.

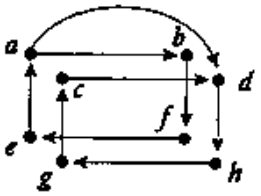
а)



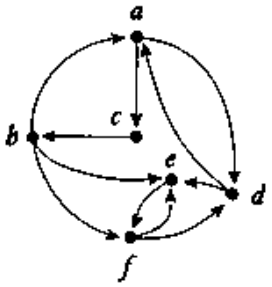
б)



в)



г)



Порядок выполнения работы:

1. Изучить инструкцию к практической работе.

2. Выполнить задание.
3. Оформить отчет.

Содержание отчета:

1. Тема.
2. Цель.
3. Задачи.
4. Материальное обеспечение.
5. Практическое задание.

Вопросы для самоконтроля:

1.Дайте определение эйлерова графа.
2.Сформулируйте алгоритм построения эйлерова цикла.
3.Какой граф называют гамильтоновым?
4.Существует ли эйлеров граф, обладающий висячей вершиной?
5.Чем отличается эйлеров путь от гамильтонова?

Чащина Е.

Практическая работа № 4

Тема: Решение задач по теории графов. Конечные графы. Бесконечные графы.

Цель: изучить теоретические основы конечных и бесконечных графов.
Задачи:
1.Закрепить знания основных понятий теории конечных и бесконечных графов.
2. Использовать математический аппарат теории графов
Материальное обеспечение:
практическое задание.

Теоретическая часть:

Конечный граф

Граф называется конечным, если число вершин и ребер в нем конечно, в противном случае – бесконечным.

Рассмотрим неориентированный граф **G**. Число ребер **ρ(v)**, инцидентных вершине **v**, назовем локальной степенью графа в вершине **v.**

Если все числа **ρ(v)** для **∀v∈V**, то граф называется локально-конечным.

Обозначим **ρ(v,u) = ρ(u,v)** – это число ребер, соединяющих вершины **v** и **u**, т.е. это кратность ребра **(v,u)**. Тогда

$$\rho(v) = \sum_{v \in V} \rho(v,u)$$

Обозначим **m(G)** – число ребер в графе **G**. Т.к. каждое ребро будет учитываться в двух локальных степенях **v** и **u**, то запишем:

$$2m(G) = \sum_{v \in V} \rho(v) = \sum_{v} \sum_{u \in V} \rho(v,u) \qquad (1)$$

Теорема. Число вершин нечетной локальной степени в графе четно.

Доказательство. Воспользуемся формулой (1), т.е. сумма всех локальных степеней четна. Тогда, удалив из суммы все четные слагаемые, получим четное число, представляющее собой сумму нечетных степеней. Теорема доказана.

Граф называется однородным в степени **k**, если локальные степени всех вершин равны **k**. Примерами однородных графов являются правильные треугольники, многоугольники, многогранники.

Для однородных графов можно записать:

$$m(G) = \frac{1}{2} \cdot n \cdot k,$$

где **n** – число вершин графа **G**.

Для куба:

$$m(G) = \frac{1}{2} \cdot 8 \cdot 3 = 12$$

Пусть **G** – ориентированный граф. Тогда введем следующие обозначения:

— **od(v)** – число дуг, выходящих из вершины;
— **id(v)** – число дуг, заходящих в вершину.

Это локальные степени графа.

Чащина Е.А.

Петли, если они есть, считаются по одному разу и в **od(v)**, и в **id(v)**.

Аналогично считаются две кратности. Обозначим как $\rho_o(v,u)$, $\rho_i(v,u)$ число дуг, выходящих из **v** в **u** и из **u** в **v** соответственно.

$$\rho_o(v,u) = \rho_i(v,u)$$

$$od(v) = \sum_{u \in V} \rho_o(v,u)$$

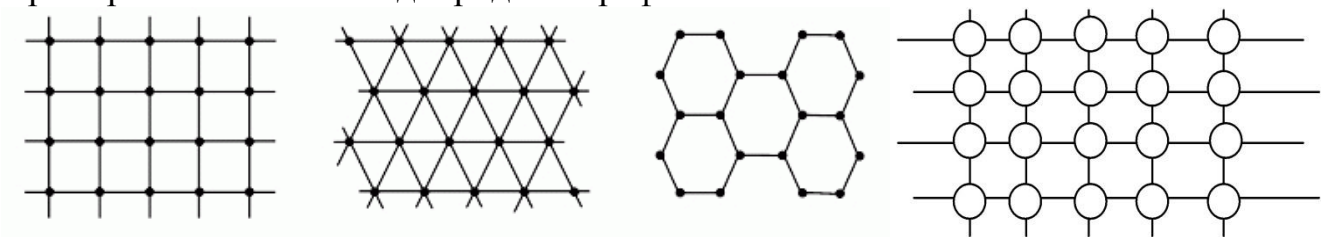
$$id(v) = \sum_{u \in V} \rho_i(v,u)$$

$$m = \sum_{v \in V} od(v) = \sum_{v \in V} id(v)$$

Ориентированный граф называется однородным, если для $\forall v \in V$ выполняется следующее: **od(v) = id(v) = k**.

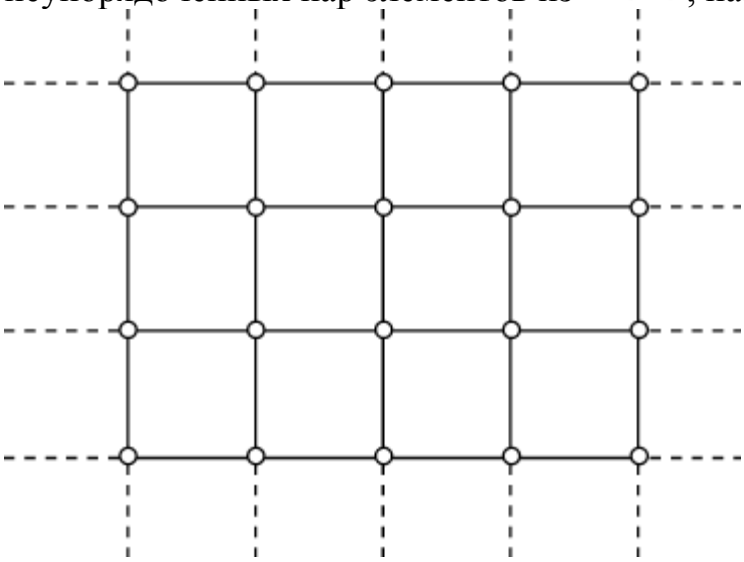
Бесконечный граф

Примеры бесконечных однородных графов.



Бесконечные однородные графы находят широкое применение в задачах трассировки печатных соединений, т.к. их использование позволяет разбивать *коммутационное поле* печатных плат на элементарные ячейки одинаковой формы.

Бесконечным графом называется пара $(V(G), E(G))$, где $V(G)$ — *бесконечное множество* элементов, называемое ***вершинами***, а $E(G)$ — бесконечное семейство неупорядоченных пар элементов из $V(G)$, называемых ***ребрами***.



Если оба *множества* $V(G)$ и $E(G)$ — счетны, то G называется ***счетным графом***. Заметим, что наши определения исключают те случаи, когда $V(G)$ — бесконечно,

а $E(G)$ — конечно. Такие объекты являются всего лишь конечными графами с бесконечным множеством *изолированных вершин*. Когда $E(G)$ — бесконечно, а $V(G)$ — конечно, такие объекты являются конечными графами с бесконечным числом петель или кратных ребер.
Некоторые определения таких понятий, как "смежный", "*инцидентный*", "изоморфный", "*подграф*", "*объединение*", "связный", "компонента" переносятся на бесконечные графы.
Степенью вершины v *бесконечного графа* называется *мощность множества* ребер, инцидентных v *Степень вершины* может быть конечной или бесконечной.
Бесконечный *граф*, все вершины которого имеют *конечные* степени, называется ***локально конечным***. Хорошо известным примером такого графа является бесконечная квадратная решетка, часть которой изображена на рисунке. ***Локально счетный бесконечный граф*** — это *граф*, все вершины которого имеют счетную степень. Под счетным множеством здесь и в дальнейшем понимается *бесконечное множество*, допускающее *взаимно однозначное отображение* на множество натуральных чисел.
Теорема Каждый связный *локально счетный бесконечный граф* является счетным.
Доказательство
Пусть v — произвольная *вершина* такого бесконечного графа, и пусть v_1 — *множество вершин*, смежных v, v_2 — множество всех вершин, смежных вершинам из v_1, и т.д. *По* условию теоремы v_1 — счетно и, следовательно, *множества* $v_2, v_3, \ldots$ тоже счетны. Здесь используется тот факт, что *объединение* не более чем счетного *множества* счетных множеств счетно.
Следовательно, $\{v\}, v_1, v_2, \ldots$ — последовательность множеств, *объединение* которых счетно. Кроме того, эта последовательность содержит каждую вершину бесконечного графа в силу его связности. Отсюда и следует нужный результат.
Следствие Каждый связный *локально конечный* бесконечный *граф* является счетным.
Помимо этого, на бесконечный *граф* G можно перенести понятие маршрута, причем тремя различными способами:

1. *Конечный маршрут* в G определяется так. Маршрутом в данном графе G называется конечная последовательность ребер вида $\{v_0, v_1\}$, $\{v_1, v_2\}, \ldots, \{v_{m-1}, v_m\}$. Маршрут можно обозначить и так: $v_0 \to v_1 \to v_2 \to \ldots \to v_m$.
2. *Бесконечным в одну сторону маршрутом* в G с *начальной вершиной* v_0 называется бесконечная последовательность ребер вида $\{v_0, v_1\}$, $\{v_1, v_2\} \ldots$.
3. *Бесконечным в обе стороны маршрутом* в графе G называется *бесконечная последовательность* ребер вида $\ldots, \{v_{-2}, v_{-1}\}, \{v_{-1}, v_0\}$, $\{v_0, v_1\}, \{v_1, v_2\}, \ldots$

Бесконечные в одну сторону и в обе стороны цепи и простые цепи определяются очевидным образом, так же как и понятия длины цепи и расстояния между вершинами. Бесконечные простые цепи не так уж трудно обнаружить.

Теорема 6.1. (Кениг, 1936) Пусть G — связный *локально конечный* бесконечный *граф*, тогда для любой вершины v существует бесконечная в одну сторону *простая цепь* с начальной вершиной v.

Доказательство

Если z — произвольная *вершина* графа G, отличная от v, то существует нетривиальная *простая цепь* от v до z, отсюда следует, что в G имеется бесконечно много простых цепей с начальной вершиной v. Поскольку степень v конечна, то *бесконечное множество* таких простых цепей должно начинаться с одного и того же ребра. Если таким ребром является $\{v, v_1\}$, то, повторяя эту процедуру для вершины v_1, получим новую вершину v_2 и соответствующее ей *ребро* $\{v_1, v_2\}$. Продолжая таким образом, получим бесконечную в одну сторону простую цепь $v \to v_1 \to v_2 \ldots$.

Важное *значение* леммы Кенига состоит в том, что она позволяет получить результаты о бесконечных графах из соответствующих результатов для конечных графов. Типичным примером является следующая теорема.

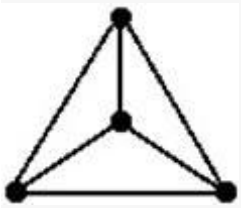
Теорема 6.2. Пусть G — *счетный граф*, каждый конечный *подграф* которого планарен, тогда и G планарен.

Доказательство Так как G — *счетный граф*, его вершины можно занумеровать в последовательность $v_1, v_2, v_3, \cdots$. Исходя из нее, построим строго возрастающую последовательность $G_1 \subset G_2 \subset G_3 \cdots$ *подграфов* графа G, выбирая в качестве G_k *подграф* с вершинами $v_1, \ldots, v_k$ и ребрами графа G, соединяющими только эти вершины между собой. Далее, примем на веру тот факт, что графы G_i могут быть уложены на плоскости конечным числом, скажем $m(i)$, топологически различных способов, и построим еще один бесконечный *граф* H. Его вершины $w_{ij} (i \geqslant 1, 1 \leqslant j \leqslant m(i))$ пусть соответствуют различным укладкам графов $\{G_i\}$, а его ребра соединяют те из вершин w_{ij} и w_{kl}, для которых $k=i+1$ и *плоская укладка*, соответствующей w_{ij}. Мы видим, что *граф* H связен и локально конечен, поэтому, как следует из леммы Кенига, он содержит бесконечную в одну сторону простую цепь. А так как *граф* G является счетным, то эта бесконечная *простая цепь* и дает требуемую *плоскую укладку* графа G.

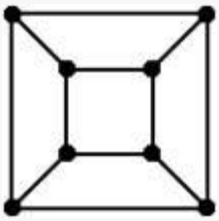
Стоит подчеркнуть, что если принять дальнейшие *аксиомы* теории множеств, в частности, аксиому выбора для несчетных множеств, то многие результаты можно перенести и на такие бесконечные графы, которые необязательно являются счетными.

Задание

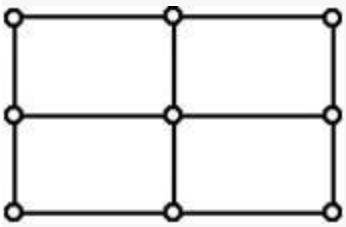
1. Изобразить конечные и бесконечные графы.
2. Определите, какие графы изображены



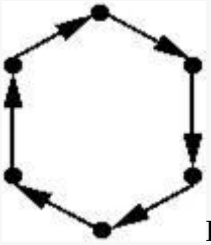
A



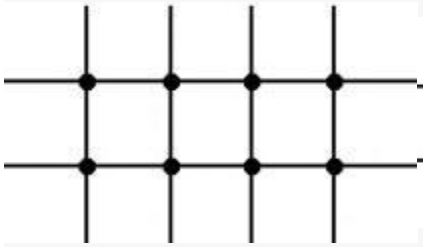
B



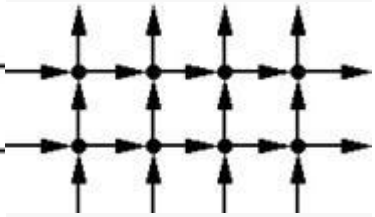
C



D



E



F

Проверьте правильность ответа: A, b – платоновы тела; c – неориентированный конечный однородный граф степени 2; d – неориентированный конечный однородный граф степени 1; e – неориентированный конечный однородный граф степени 4; f – ориентированный бесконечный однородный граф степени 2

Порядок выполнения работы:

1. Выполнить задание.
2. Оформить отчет.

Содержание отчета:

1. Тема.
2. Цель.
3. Задачи.
4. Материальное обеспечение.
5. Практическое задание.

Вопросы для самоконтроля:

1. Дайте определение конечного графа.
2. Дайте определение бесконечного графа.
3. Какой граф называется однородным?
4. Где применяются бесконечные графы?

Практическая работа № 5

Тема: Решение задач по теории графов

Цель: научиться применять математический аппарат теории графов в решении задач.
Задачи:
1.Закрепить знания основных понятий теории конечных и бесконечных графов.
2. Использовать математический аппарат теории графов
Материальное обеспечение: раздаточный материал.

Теоретическая часть:

Под графом мы будем понимать множество точек (вершин), некоторые из которых соединены отрезками (ребрами).

Степень вершины графа — это количество выходящих из нее (или, что то же самое, входящих в нее) ребер (еще говорят: количество ребер, инцидентных данной вершине). Вершина графа называется четной, если ее степень четна, и нечетной в противном случае.

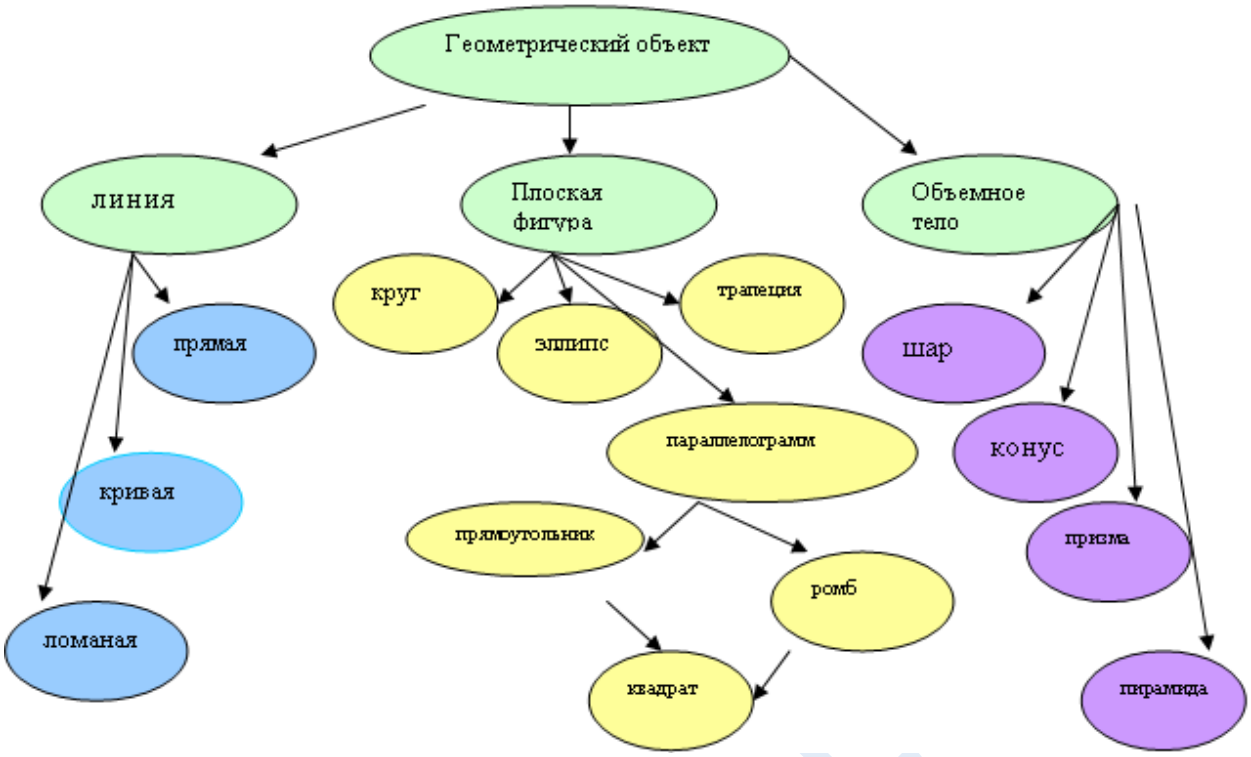
Некоторая часть вершин данного графа называется компонентой связности, если из любой ее вершины можно «дойти» до любой другой, двигаясь по ребрам.

В некоторых случаях на ребрах графа выбирается «направление движения» (например, когда на автомобильной дороге вводится одностороннее движение). При этом получается ориентированный граф. (Если направление движения по ребрам не определено, то граф называется неориентированным). В ориентированном графе различают положительную и отрицательную степень каждой вершины (то есть количество ребер, соответственно, входящих и выходящих из нее). Две вершины могут быть соединены и несколькими ребрами, направления движения по которым противоположны («дорога с двусторонним движением»). Изменяется понятие компоненты связности: теперь каждый «маршрут» от одной вершины до другой должен учитывать направление движения по ребрам.

Графовые задачи обладают рядом достоинств, позволяющих их использовать для развития воображения и улучшения логического мышления, применимы в решении многих геометрических задач.

На уроке геометрии было предложено построить граф классификации геометрических объектов. Это оказалось легко сделать с помощью понятия граф.

Чащина Е.А.



Задача о мостах

Бывший Кенигсберг (ныне Калининград) расположен на реке Прегель. В пределах города река омывает два острова. С берегов на острова были перекинуты мосты. Старые мосты не сохранились, но осталась карта города, где они изображены.

Издавна среди жителей Кёнигсберга была распространена такая загадка: как пройти по всем мостам, не проходя ни по одному из них дважды? Многие кёнигсбержцы пытались решить эту задачу как теоретически, так и практически, во время прогулок. Но никому это не удавалось, однако не удавалось и доказать, что это даже теоретически невозможно.

Этот вопрос привлек внимание ученых разных стран. В 1736 году задача о семи мостах заинтересовала выдающегося математика, члена Петербургской академии наук Леонарда Эйлера, о чём он написал в письме итальянскому математику и инженеру Мариони от 13 марта 1736 года. В этом письме Эйлер пишет о том, что он смог найти правило, пользуясь которым легко определить, можно ли пройти по всем мостам, не проходя дважды ни по одному из них (в случае семи мостов Кёнигсберга это невозможно).

Причем, он не только решил эту конкретную задачу, но придумал общий метод решения подобных задач. Эйлер поступил следующим образом: он «сжал» сушу в точки, а мосты « вытянул» в линии, как показано на рисунке 1 а, б.

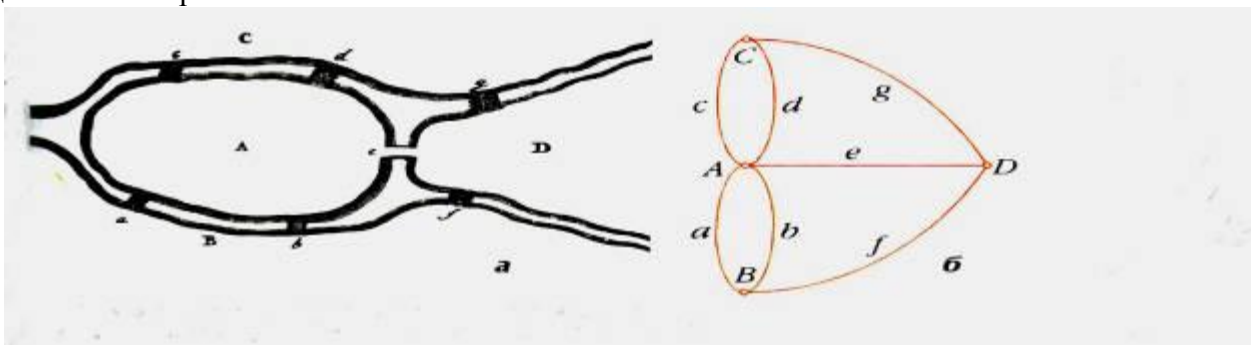


Рисунок 1

На упрощённой схеме части города (графе) мостам соответствуют линии (рёбра графа), а частям города — точки соединения линий (вершины графа). В ходе рассуждений Эйлер пришёл к следующим выводам:

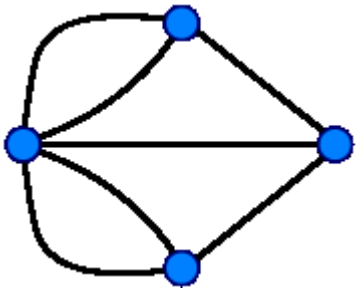
Число нечётных вершин (вершин, к которым ведёт нечётное число рёбер) графа всегда чётно. Невозможно начертить граф, который имел бы нечётное число нечётных вершин.

Если все вершины графа чётные, то можно, не отрывая карандаша от бумаги, начертить граф, при этом можно начинать с любой вершины графа и завершить его в той же вершине.

Граф с более чем двумя нечётными вершинами невозможно начертить одним росчерком.

Давайте четко сформулируем поставленную задачу. При каком условии можно обойти все ребра графа, пройдя каждое ровно один раз? Решение оказалось очень простым. Сосчитаем, сколько ребер выходит из каждой вершины. Одни из этих чисел будут четными, а другие - нечетными. Будем и сами вершины называть четными, если из них выходит четное число ребер, и нечетными в противном случае. Как мы уже знаем: количество ребер, выходящих из данной вершины, называется степенью вершины. Вершина графа, имеющая нечетную степень, называется нечетной, а четную степень – четной.

Граф кёнигсбергских мостов имел четыре нечётные вершины, следовательно, невозможно пройти по всем мостам, не проходя ни по одному из них дважды.



Решая задачу про кенигсбергские мосты, Эйлер установил, в частности, свойства графа:

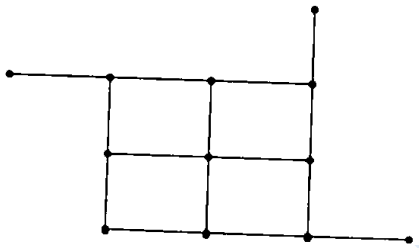
- Если все вершины графа четные, то можно одним росчерком (т.е. не отрывая карандаша от бумаги и не проводя дважды по одной и той же линии) начертить граф. При этом движение можно начать с любой вершины и окончить в той же вершине.

- Граф с двумя нечетными вершинами тоже можно начертить одним росчерком. Движение надо начинать от любой нечетной вершины, а закончить на другой нечетной вершине.
- Граф с более чем двумя нечетными вершинами невозможно начертить одним росчерком.

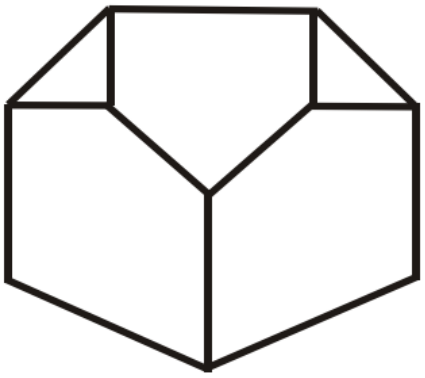
В задаче о кенигсбергских мостах все четыре вершины соответствующего графа нечетные, то есть нельзя пройти по всем мостам один раз и закончить путь там, где он был начат.

Задачи:

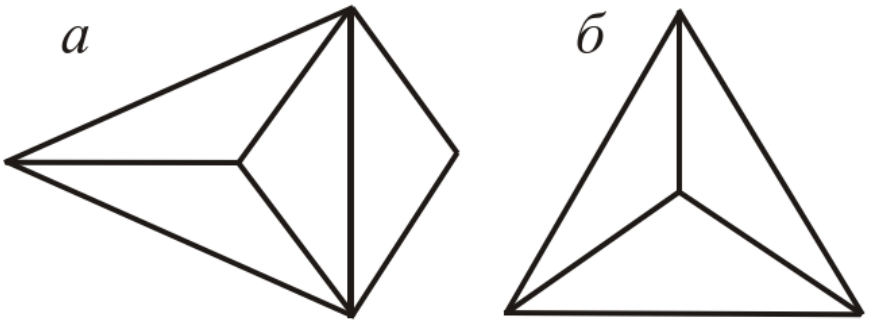
1. Между девятью планетами Солнечной системы установлено космическое сообщение. Рейсовые ракеты летают по следующим маршрутам: Земля — Меркурий, Плутон — Венера, Земля — Плутон, Плутон — Меркурий, Меркурий — Венера, Уран — Нептун, Нептун — Сатурн, Сатурн — Юпитер, Юпитер — Марс и Марс — Уран. По каждому маршруту ракеты летают в обе стороны. Можно ли долететь на рейсовых ракетах от Земли до Марса?
2. В государстве 100 городов. Из каждого города выходит 4 дороги. Сколько всего дорог в государстве?
3. Экспозиция картинной галереи представляет собой систему коридоров, на обеих стенах которых развешаны картины:
Можно ли предложить такой маршрут осмотра экспозиции, при котором посетитель проходит вдоль каждой стены ровно один раз?



4. В городе 15 телефонов. Можно ли их соединить проводами так, чтобы каждый телефон был соединён ровно с пятью другими?
5. В теннисном турнире каждый игрок команды «синих» встречается с каждым игроком команды «красных». Число игроков в командах одинаково и не больше восьми. «Синие» выиграли в четыре раза больше встреч, чем «красные». Сколько человек в каждой из команд?
6. Можно ли прогуляться по парку и его окрестностям так, чтобы при этом перелезть через каждый забор ровно один раз?



7. Можно ли нарисовать граф, изображённый на рисунках а,б



не отрывая карандаша от бумаги и проводя каждое ребро ровно один раз?
8. Дан кусок проволоки длиной 120 см.
Можно ли, не ломая проволоки, изготовить каркас куба с ребром 10 см? Какое наименьшее число раз придётся ломать проволоку, чтобы всё же изготовить требуемый каркас?
9. Можно ли так нарисовать 5 горизонтальных и 4 вертикальных отрезка, чтобы каждый горизонтальный отрезок пересекался ровно с тремя вертикальными, а каждый вертикальный ровно с тремя горизонтальными?
10. При встрече каждый из моих одноклассников пожал руку другому (каждый пожал каждому). Сколько рукопожатий было сделано, если друзей было: 1)трое; 2)четверо; 3)пятеро? (решите и изобразите граф решения)

Порядок выполнения работы:

1. Решить задачи.
2. Оформить отчет.

Содержание отчета:

1. Тема.
2. Цель.
3. Задачи.
4. Материальное обеспечение.
5. Практическое задание.

Вопросы для самоконтроля:

Чащина Е.А.

1. Как в задачах применяется ориентированный граф?
2. Какой граф можно начертить, не отрывая карандаша от бумаги, при этом можно начинать с любой вершины графа и завершить его в той же вершине?
3. Какое современное устройство решает задачу о кинесберских мостах?
4. Когда вершина графа называется четной, а когда нечетной?

Чащина Е.А.

Практическая работа № 6

Тема: Решение задач по теории графов. Алгоритм Краскаля»

Цель: изучить и отработать навыки в применении алгоритма Краскала; закрепить навыки моделирования графов в графической среде Kraskal.

Задачи:

1. Закрепить знания основных понятий теории графов.
2. Приобрести практические умения использования специального программного обеспечения для моделирования.
3. Использовать математический аппарат теории графов

Материальное обеспечение:

Программы для вычисления алгоритма краскала: Kraskal

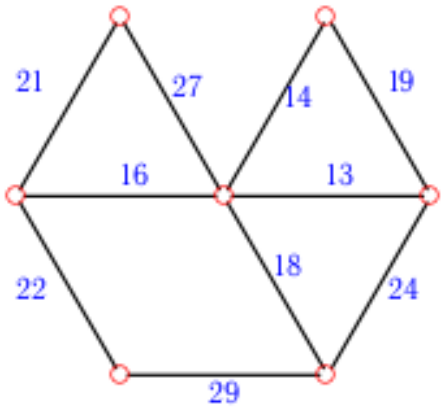
Теоретическая часть:

Остов графа (стягивающее дерево, spanning tree, ST) – дерево, полученное из графа, выбрасыванием части ребер.

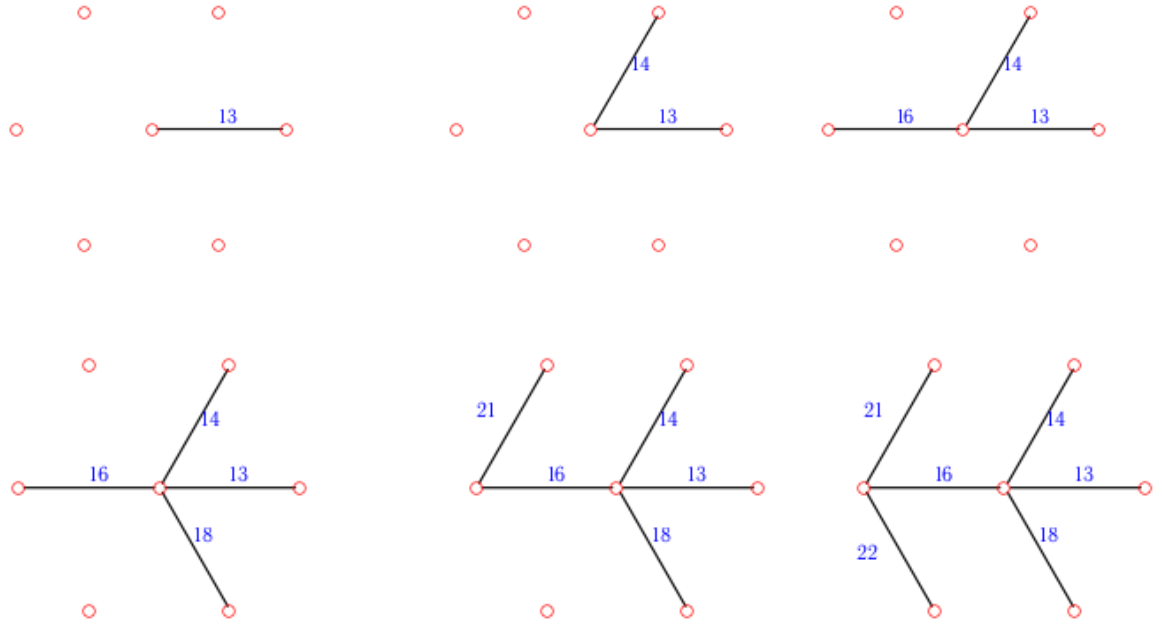
Минимальный остов графа (стягивающее дерево минимального веса, Minimal Spanning Tree, MST) – остов, с минимальной суммой весов входящих в него ребер.

Алгоритм Краскала вычисляет для заданного взвешенного неориентированного графа G остовное дерево с наименьшей суммой весов ребер— остовное дерево наименьшего веса. В новой задаче множество вершин при поиске остовного дерева наименьшего веса не меняется.

Минимальный остов графа. Дан взвешенный граф.

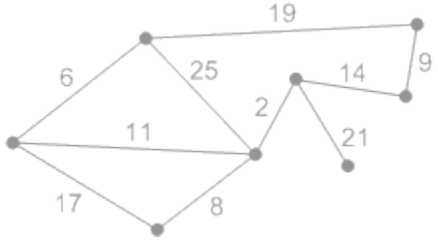


Строим граф, присоединяя к пустому графу на множестве вершин заданного графа ребро наименьшего веса. К полученному графу последовательно присоединяем остальные ребра, выбирая на каждом шаге ребро наименьшего веса, не образующее цикл с имеющимися ребрами. В нашем случае начинаем с ребра весом 13 – наименьшего в графе. На рисунках дана последовательность действий. Ребро весом 19 не включается в остов, так как оно образует цикл с ребрами весом 14 и 13.

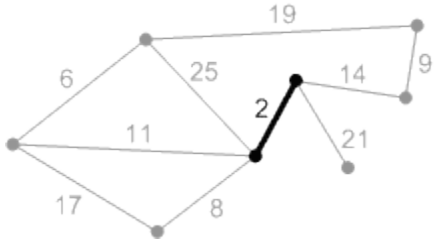


Выполнение алгоритма Краскала:

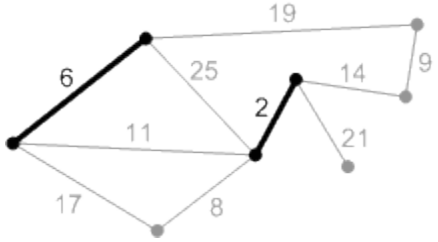
1. Начальная фаза. Минимальный покрывающий лес пуст.



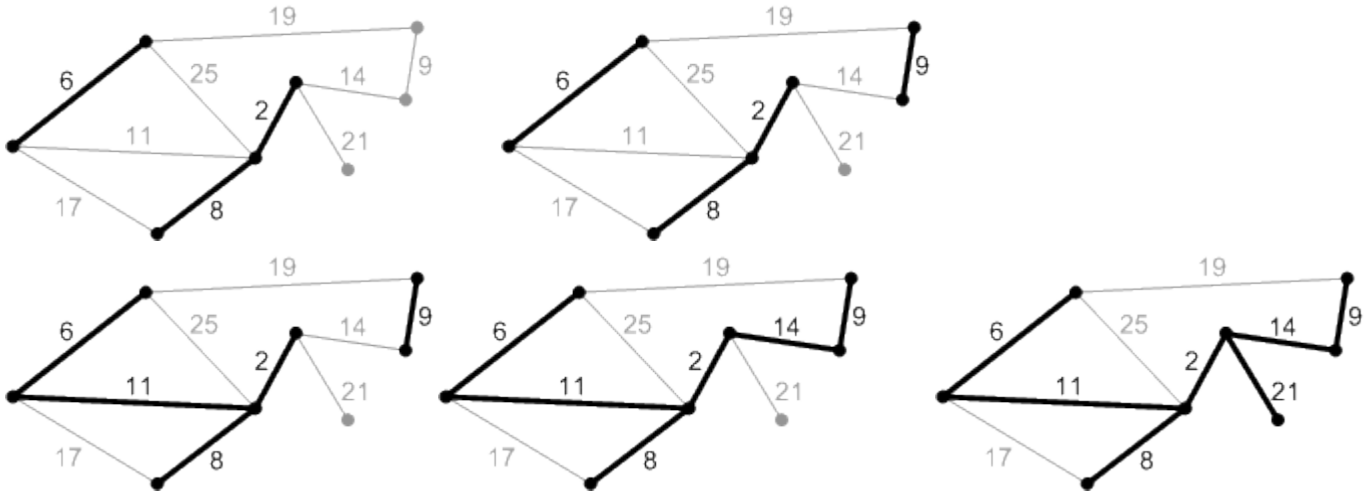
2. Перебираем ребра в порядке возрастания веса: первое ребро с весом 2. Добавляем его к А.



3. Следующее безопасное ребро с весом 6. Добавляем его.

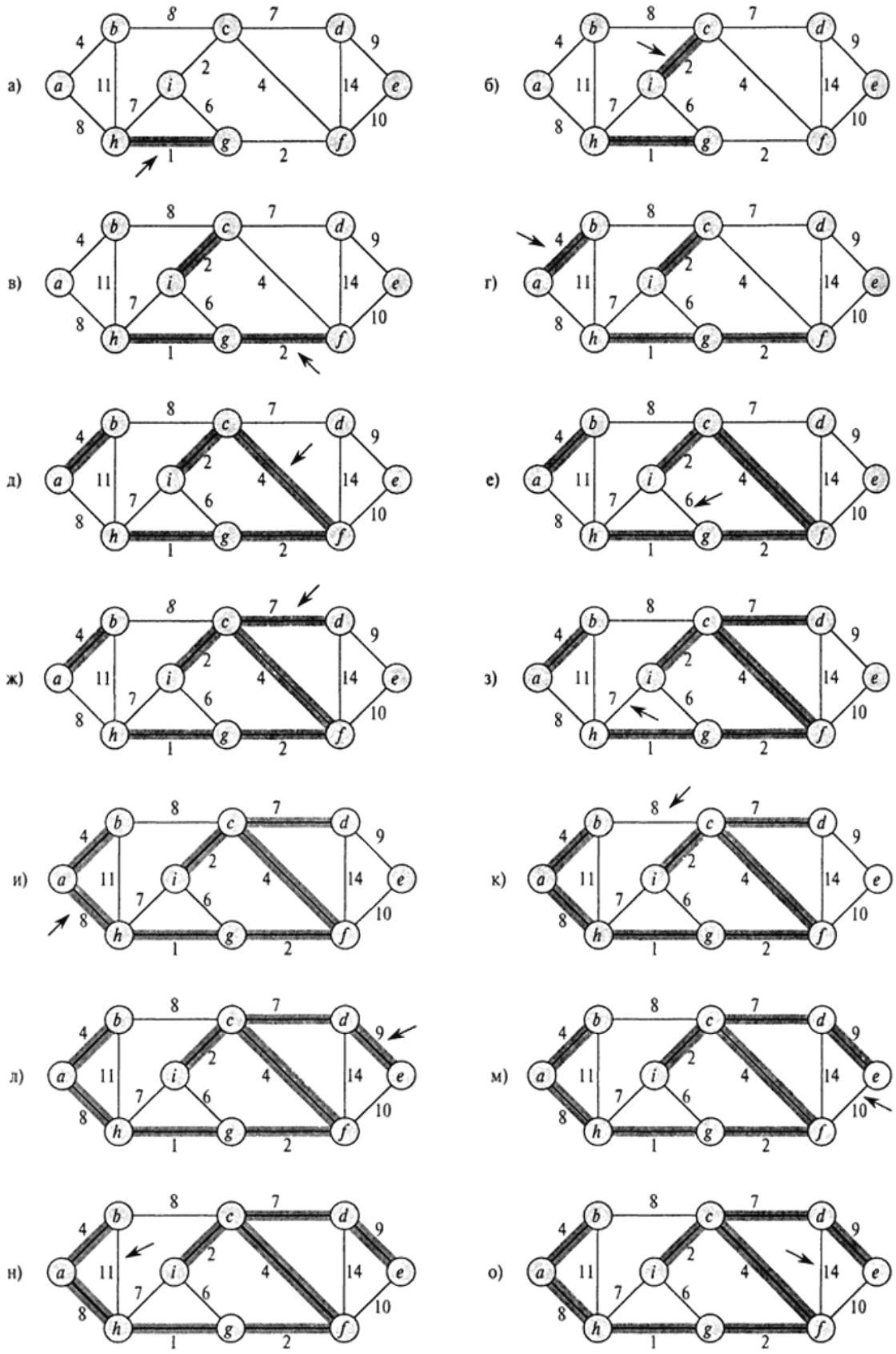


4-8. Добавляем остальные безопасные ребра.



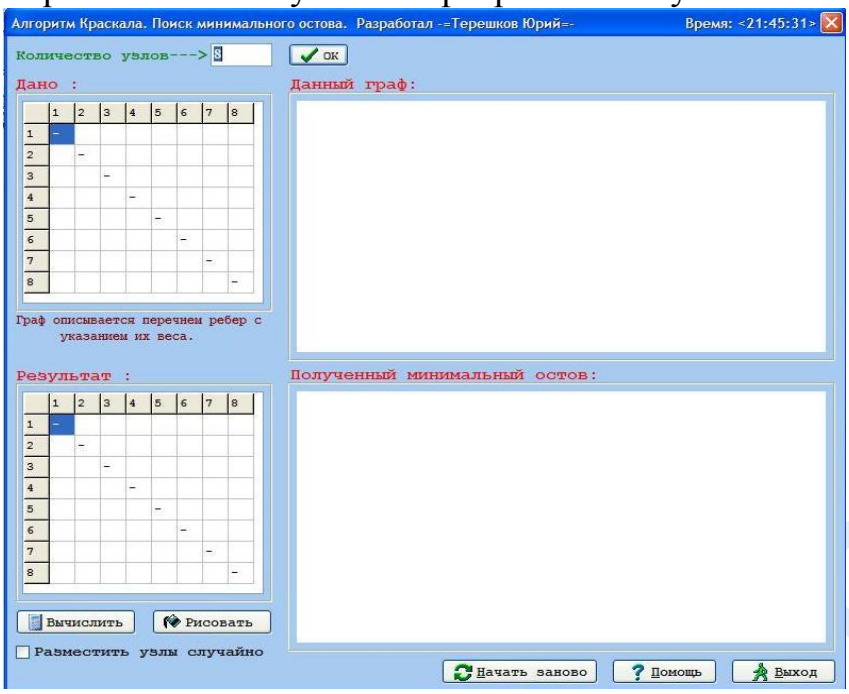
Пример выполнения алгоритма Краскаля. Заштрихованные ребра принадлежат растущему лесу.

Чащина Е.

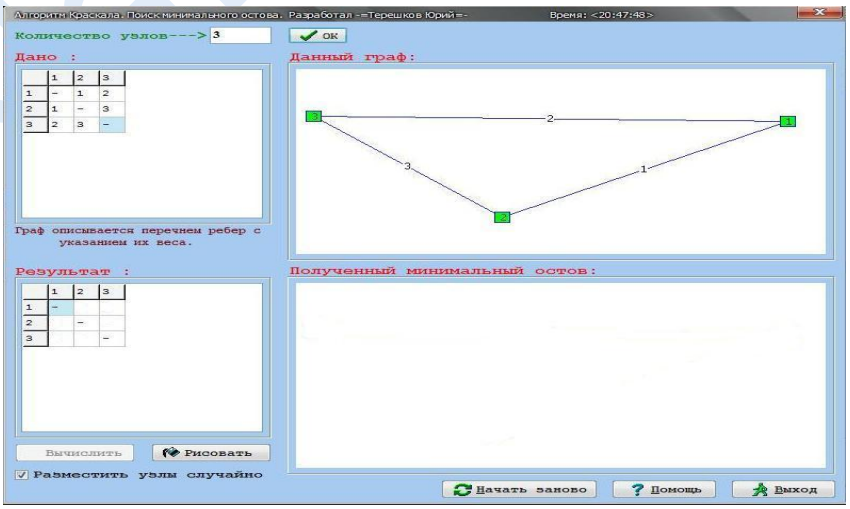

а)
б)
в)
г)
д)
е)
ж)
з)
и)
к)
л)
м)
н)
о)

Изучить инструкцию к практической работе.

Для запуска программы необходимо активировать exe – файл с названием «Краскал.exe» запустится программа. Рисунок главной формы изображен на рисунке.

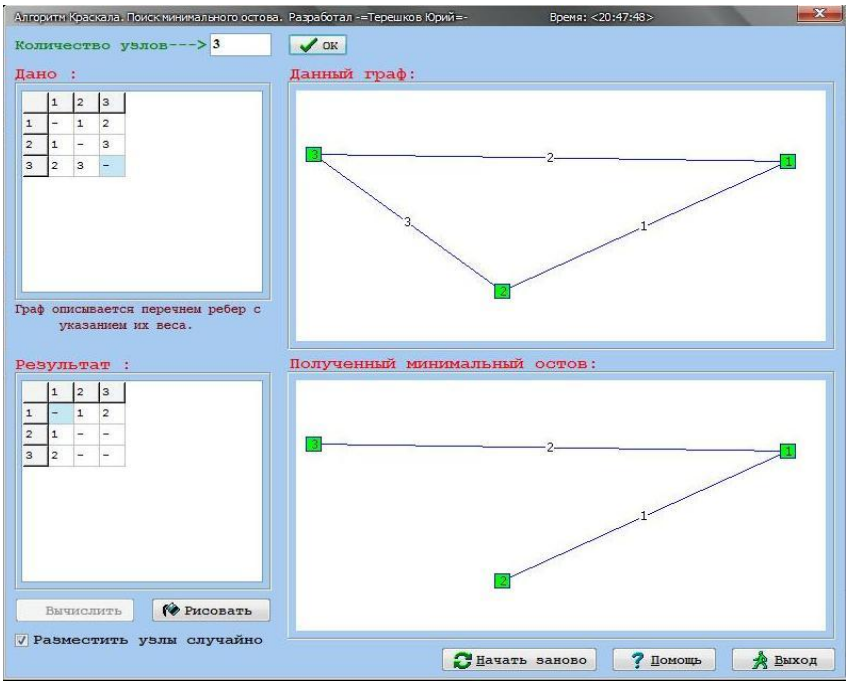


На главной форме программы изображены: текстовое поле необходимое для ввода количество узлов графа, для которого нужно будет найти остов минимального веса, затем нужно нажать кнопку «ОК». Далее нужно занести веса в матрицу весов «Дано» вводить нужно только по горизонтали, а по вертикали программа заполнит поля автоматически. Далее нужно расставить узлы нашего графа, для этого одним щелчком по полю «Данный граф» создастся узел, он будет помечен синей точкой аналогично выполнить для остальных вершин графа. Также узлы можно расставить случайным образом, для этого нужно пометить флажок «Разместить узлы случайно» и нажать кнопку «Рисовать» при каждом нажатии на кнопку вершины будут размещаться случайно.



После того, когда граф на рисован необходимо найти «Остов минимального веса» с помощью алгоритма Краскала, для этого нажимать кнопку «Вычислить». Остов

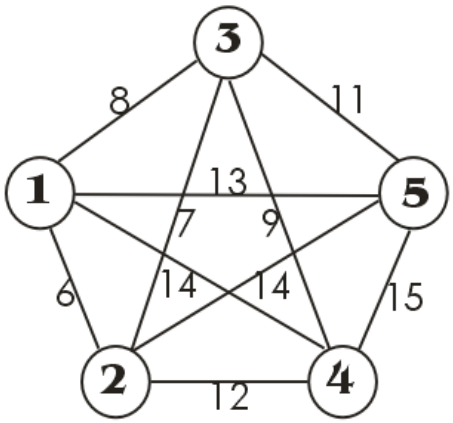
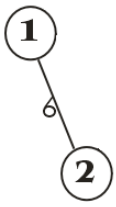
минимального веса будет изображен в поле «Полученный минимальный остов» и в поле «Результат» будет показан результат виде матрицы весов.

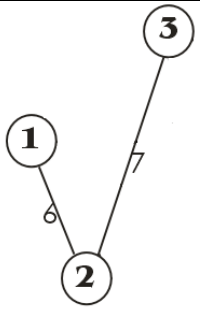
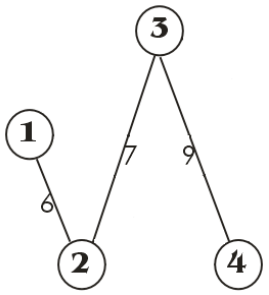
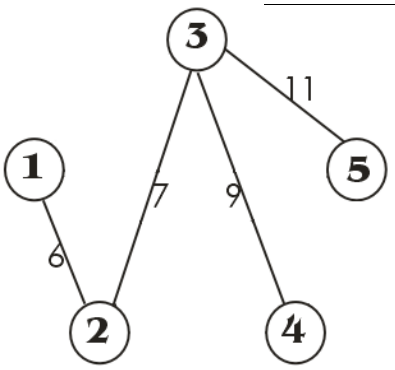


Задание:

Проработать две первые задачи. Третью и четвертую решить самостоятельно, используя программное обеспечение Kraskal .

Задача №1. Дан взвешенный связный неориентированный граф, состоящий из пяти вершин. Необходимо найти остов минимального веса с помощью алгоритма Краскала.

	Выбираем вершину начала построения остова минимального веса, например, первую вершину.
	Шаг 1. Найдено ребро минимального веса: 1-2=6. Полученный остов на рисунок.

	Шаг 2. Найдено ребро минимального веса: 2-3=7. Полученный остов на рисунок.
	Шаг 3. Найдено ребро минимального веса: 3-4=9. Полученный остов на рисунок.
	Шаг 4. Найдено ребро минимального веса: 3-5=11. Рассмотрены все вершины и инцидентные ребра этим вершинам, оставшиеся образуют цикл в полученном минимальном остове. А это не удовлетворяет условиям поставленной задачи. На четвертом шаге получили окончательный остов минимального веса, который представлен на рисунке.

При изменении вершины начала построения конфигурация остова минимального веса не измениться, а измениться лишь последовательность построения ребер остова.

Например, если в качестве начальной вершины выбрать четвертую вершину, то последовательность этапов построения остова минимального веса будет выглядеть следующим образом:

Шаг 1. 4-3=9;

Шаг 2. 3-2=7;

Шаг 3. 2-1=6;

Шаг 4. 3-5=11;

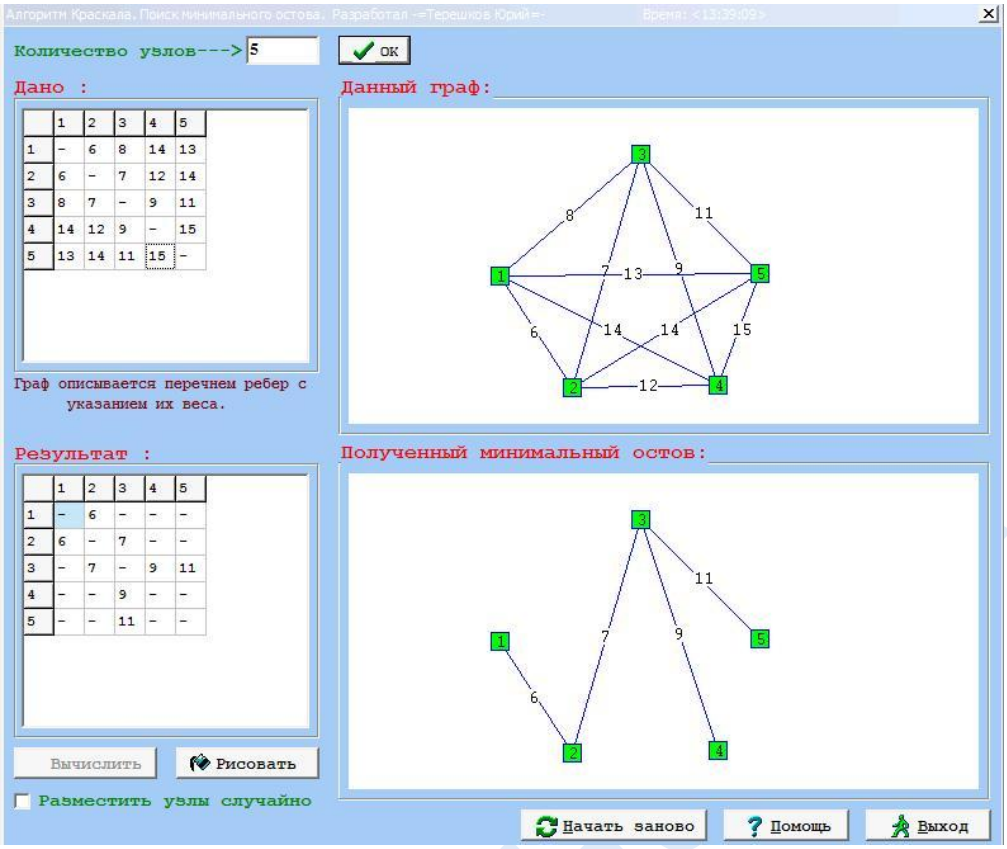
При этом конфигурация остова останется прежней. Решим данную задачу с помощью программной модели. Чтобы решить данную задачу необходимо построить матрицу весов, матрица представлена на рисунке.

	1	2	3	4	5
1	–	6	8	14	13
2	6	–	7	12	14
3	8	7	–	9	11
4	14	12	9	–	15
5	13	14	11	15	–

Чащина Е.А.

Матрица весов

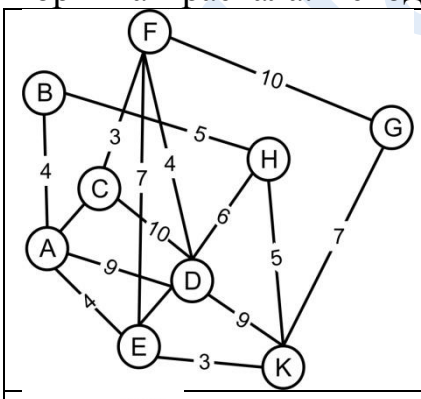
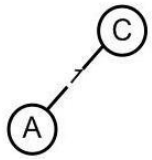
Полученный минимальный остов с помощью программной модели изображен на рисунке.

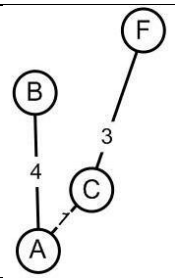
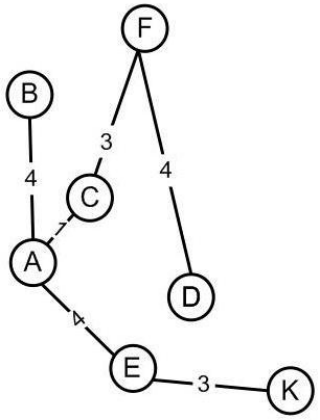
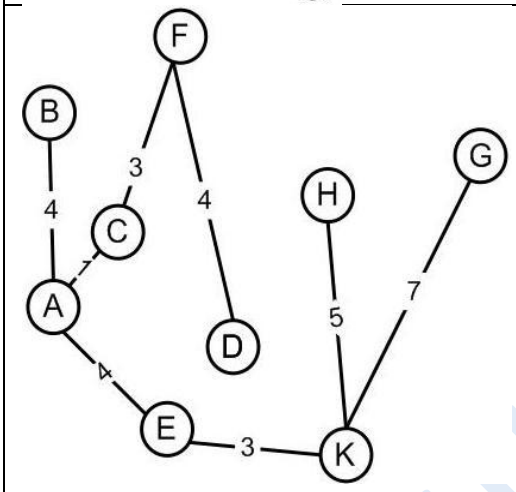


Полученный минимальный остов

После проверки вычислений вручную и программной модели результат одинаковый, это означает, что программная модель работает и функционирует верно.

Задача №2. Дан взвешенный, связный, неориентированный граф, состоящий из девяти вершин. Необходимо найти остов минимального веса с помощью алгоритма Краскала. Исходный граф на рисунке.

	Выбираем вершину начала построения остова минимального веса, например, первую вершину.
	Шаг 1. Найдено ребро минимального веса: AC=1. Полученный остов на рисунок.

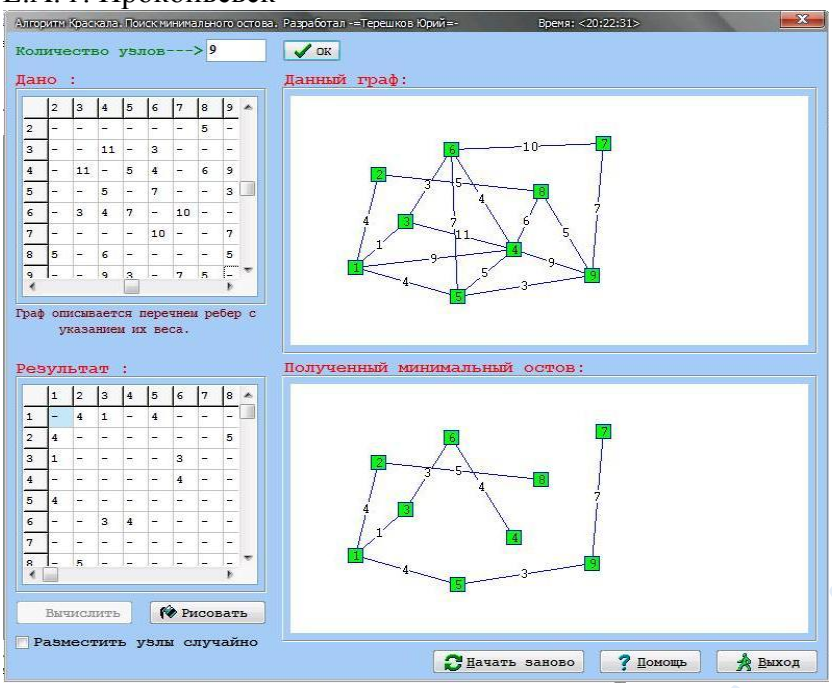
	Шаг 2. Найдено ребро минимального веса: CF=3, AB=4, AC=4. Полученный остов на рисунок.
	Шаг 3. Найдено ребро минимального веса: FD=4, EK=3, AE=4. Полученный остов на рисунок.
	Шаг 4. Найдено ребро минимального веса: KH=5, KG=4. Рассмотрены все вершины и инцидентные ребра этим вершинам, оставшиеся ребра образуют цикл в полученном минимальном остове. А это не удовлетворяет условиям поставленной задачи. На четвертом шаге получен окончательный остов минимального веса, который представлен на рисунке.

Решим данную задачу с помощью программной модели. Чтобы решить данную задачу необходимо построить матрицу весов.

Таблица 1. Матрица весов

	A	B	C	D	E	F	G	H	K
A	-	4	1	9	4	-	-	-	-
B	4	-	-	-	-	-	-	5	-
C	1	-	-	10	-	3	-	-	-
D	9	-	10	-	5	4	-	6	9
E	4	-	-	5	-	7	-	-	3
F	-	-	3	4	7	-	10	-	-
G	-	-	-	-	-	10	-	-	7
H	-	5	-	6	-	-	-	-	5
K	-	-	-	9	3	-	7	5	-

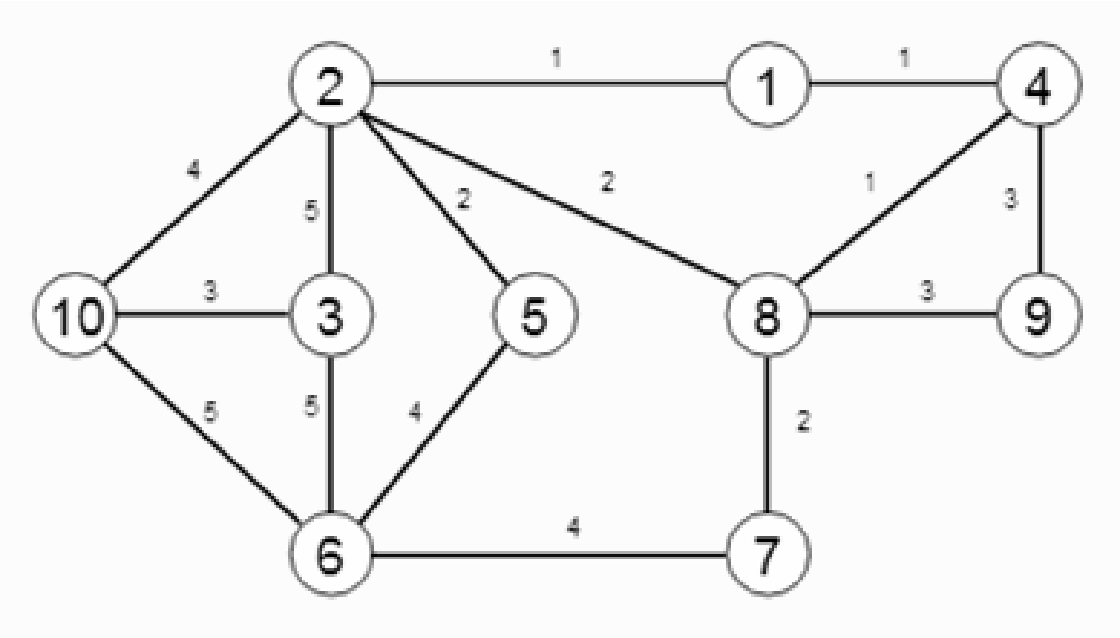
Полученный минимальный остов с помощью программной модели изображен на рисунке.



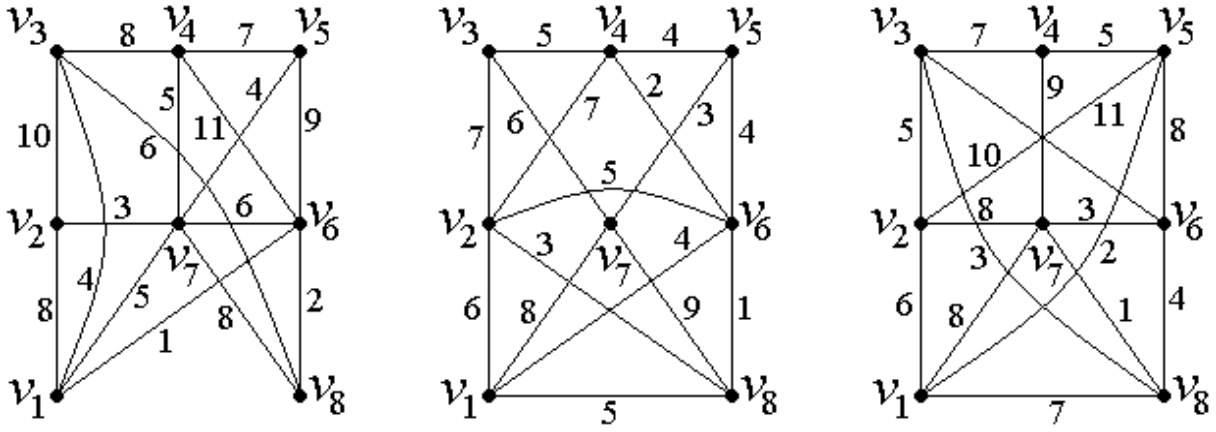
Остов минимального веса

После проверки вычислений вручную и программной модели полученные остовы минимального веса различаются, но они построены верно. Это связано с тем, что программа выбирает другую вершину начала. После решения двух контрольных задач стало ясно, что разработанная программная модель работает верно.

Задача №3. Найти остов минимального веса с помощью алгоритма Краскала., проверить программой Краскала.



Задача №4. Найти минимальное остовное дерево в неориентированном нагруженном графе. Результат проверить программным обеспечением.



а) б) в)

Порядок выполнения работы:

1. Изучить инструкцию к практической работе.
2. Выполнить задание.
3. Оформить отчет.

Содержание отчета:

1. Тема.
2. Цель.
3. Задачи.
4. Материальное обеспечение.
5. Практическое задание.

Вопросы для самоконтроля:

1. Дайте определение остов графа.
2. Дайте определение минимальный остов графа.
3. В чем смысл алгоритма Краскала?

Чащина Е.А.

Практическая работа № 7

Тема: Решение задач по теории графов. Двойственность по Уитни

Цель: изучить теоретические основы планарности, двойственности графов и теорему Уитни; закрепить навыки моделирования графов.

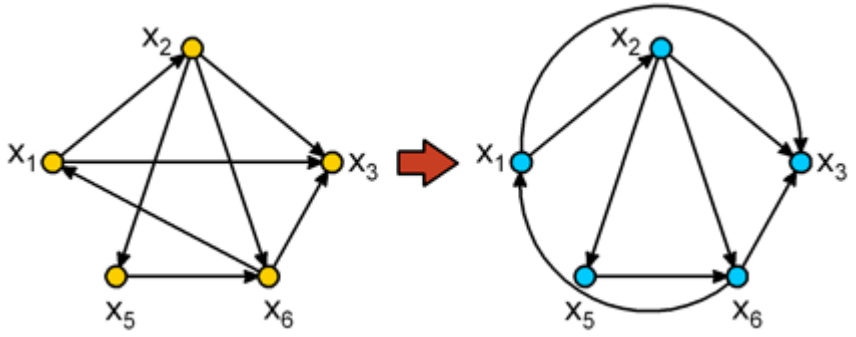
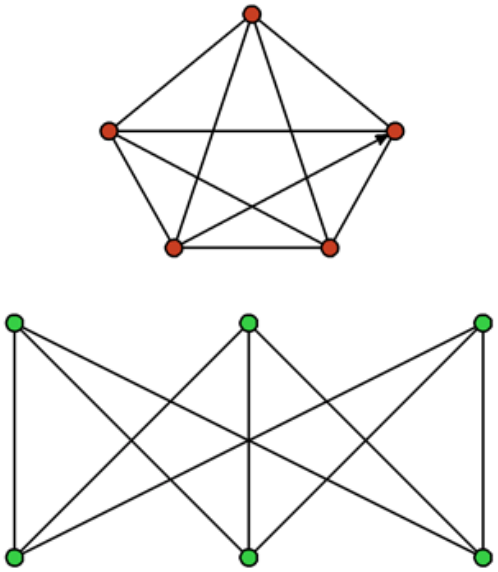
Задачи:

1. Закрепить знания основных понятий теории графов.
2. Использовать математический аппарат теории графов

Материальное обеспечение:

Раздаточный материал. Линейка карандаш.

Теоретическая часть:

	Граф G =(V, E), который может быть изображен на плоскости или сфере без пересечений называется *планарным*
	На рисунке показаны *непланарные* графы. Эти два графа играют важную роль в теории планарных графов и известны как графы Куратовского.

Неориентированный граф G = (V, E) называют *двудольным*, если множество его вершин V может быть разбито на такие два подмножества V^a и V^b, что каждое ребро имеет один конец в V^a , а другой в V^b (рисунок 1,а).

Ориентированный граф G называется *двудольным*, если его *неориентированный двойник – двудольный граф (рисунок 1,б,в).*

Двудольный граф $G=(V^a \cup V^b, E)$ называют полным, если для любых двух вершин $v_i \in V^a$ и $v_j \in V^b$ существует ребро (v_i,v_j) в G=(V,E) (рисунок 1,г).

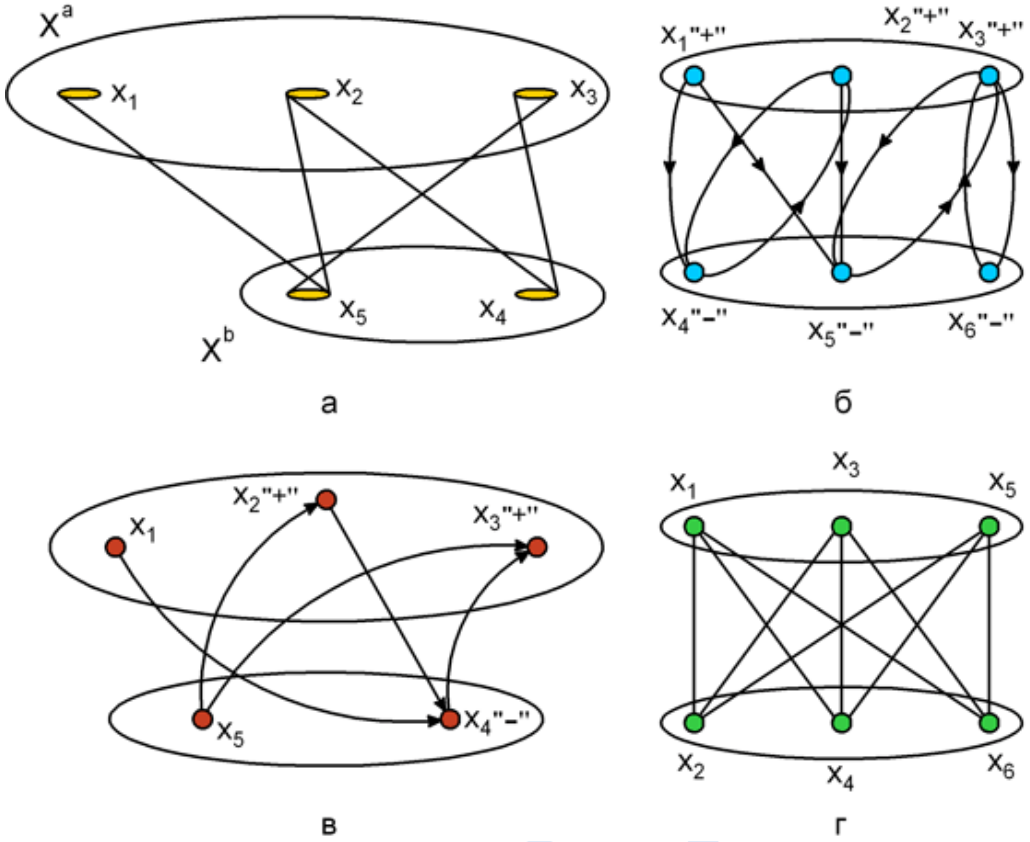


Рисунок 1

Теорема (Х. Уитни, 1932 г.). Граф планарен тогда и только тогда когда он имеет абстрактно двойственный граф.

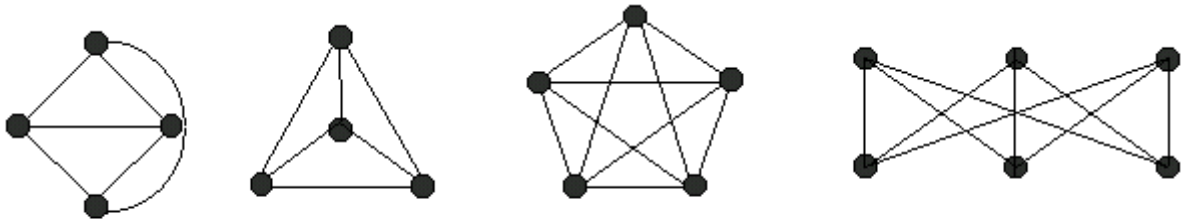
Говорят, что *граф G укладывается на поверхности S*, если его можно нарисовать на этой поверхности так, что его ребра будут пересекаться лишь в концевых точках - вершинах. Граф называется *планарным*, если его можно уложить на плоскости. Изображение планарного графа на плоскости называют *планарной укладкой*. Для каждого простого планарного графа существует планарная укладка, в которой все ребра графа будут прямыми линиями.

Если граф не укладывается на плоскости (поверхности нулевого порядка), то его можно уложить на какой - либо другой поверхности (более высокого порядка). Укладываемость графа на плоскости равносильна его укладываемости на сфере, поскольку сфера и плоскость относятся к поверхности нулевого порядка, что можно установить стереографической проекцией сферы на плоскость.

На рисунок 2, *а* и *б* показаны две планарные укладки одного и того же графа, а на рисунке *в* и *г* приведены два основных непланарных графа – K_5 и $K_{3,3}$ (*графы Куратовского*).

Графы Куратовского считаются основными непланарнами графами потому, что играют решающую роль в исследовании планарности графов.

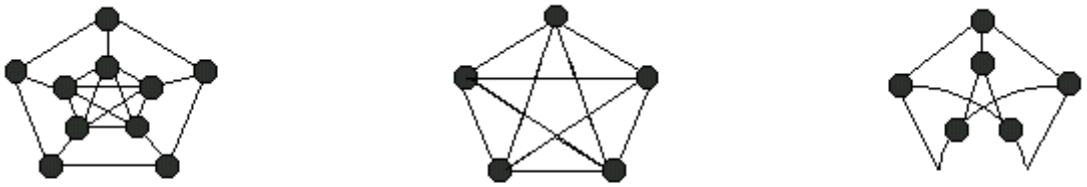
Теорема Граф планарен тогда и только тогда, когда каждый его блок планарен.



а **б** **в** **г**

Рисунок 2. Примеры планарного и непланарных графов:
а - планарная укладка с непрямолинейными ребрами;
б - планарная укладка того же графа с прямолинейными ребрами;
в - непланарный граф K_5; **г** - непланарный граф $K_{3,3}$

До появления статьи Куратовского, в которой был дан критерий планарности графов, характеризация планарных графов была труднейшей нерешенной проблемой. Доказательство приводимой ниже теоремы Куратовского можно найти в книге.



а **б** **в**

Рисунок 3. Непланарность графа Петерсена: **а** - граф Петерсена;
б - граф Петерсена, стянутый к K_5;
в - один из подграфов графа Петерсена, гомеоморфный $K_{3,3}$

Граф Петерсена не имеет подграфов, гомеоморфных K_5, но легко стягивается к K_5. Сложнее увидеть, что граф Петерсена имеет подграф, гомеоморфный $K_{3,3}$.
Уитни связал планарность графов с существованием двойственных графов.
Граф G^* называется *двойственным* к графу G, если существует такое взаимнооднозначное соответствие их ребер, что множество ребер графа G^* образует цикл тогда и только тогда, когда в графе G это же множество ребер образует разрезающее множество. Для плоского графа существует простой способ построения двойственного графа (см. пример на рисунок 4).

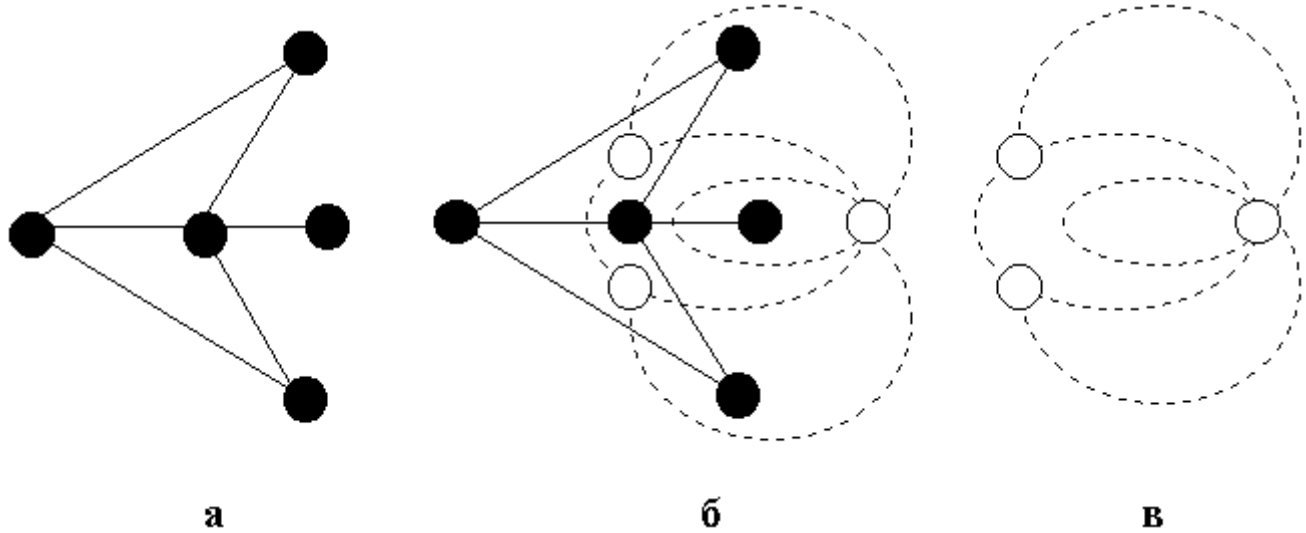


Рисунок 4. Пример: **а** – граф *G*; **б** – построение двойственного графа *G**; **в** – граф *G**, двойственный графу *G*

Суть способа построения в следующем: а) в каждой области укладки графа *G* на плоскости размещается вершина графа *G**; б) через каждое ребро *е*графа *G*, общее для областей 0_i и 0_j, проводится линия, соединяющая вершины v_i^* и v_j^* и образующая ребро *е**.

Теорема Граф имеет двойственный граф тогда и только тогда, когда он планарен.

Задание:

1. Постройте планарный граф с а) 6; б) 7; в) 8; г) 9; вершинами так, чтобы некоторые его ребра пересекались.
2. Постройте плоский граф, соответствующий графу из предыдущего задания.
3. Постройте геометрически двойственный граф к графу из задания 2.
4. Покажите, что K_5 не обладает абстрактно двойственными графами.

Порядок выполнения работы:

1. Изучить инструкцию к практической работе.
2. Выполнить задание.
3. Оформить отчет.

Содержание отчета:

1. Тема.
2. Цель.
3. Задачи.
4. Материальное обеспечение.
5. Практическое задание.

Вопросы для самоконтроля:

1. Дайте определение планарности графа.
2. Какой используется граф, в определении двудольного графа?
3. Какие два вида графа рассматриваются в теореме Х. Уитни?
4. Что означает укладка графа на поверхности?

Чащина Е.А.

Практическая работа № 8

Тема: Решение задач по теории графов.
Построение матриц смежностей и инциденций

Цель: изучение матричных способов представления графов; отработать на примерах основные понятия теории графов; научить строить графы по матрице смежности; по графу составлять матрицу смежности; закрепить навыки моделирования графов в графической среде Grafoanalizator1.3.3 rus.

Задачи:

1. Закрепить знания основных понятий теории графов.
2. Приобрести практические умения использования специального программного обеспечения для моделирования.
3. Использовать математический аппарат теории графов

Материальное обеспечение:

Программы для графического представления графов: Grafoanalizator1.3.3 rus, Просто граф, практическая работа.

Теоретическая часть:
Матрица смежности

Пусть дан граф G, его матрица смежности обозначается через $\mathbf{A}=[a_{ij}]$ и определяется следующим образом:

$a_{ij}=1$, если в G существует дуга (x_i,x_j),

$a_{ij}=0$, если в G нет дуги (x_i,x_j).

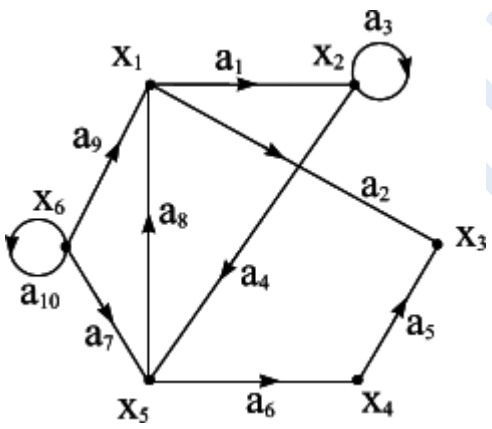


Рисунок 1.

Таким образом, матрица смежности графа, изображенного на рисунке 1, имеет вид

A=

	x_1	x_2	x_3	x_4	x_5	x_6
x_1	**0**	**1**	**1**	**0**	**0**	**0**
x_2	**0**	**1**	**0**	**0**	**1**	**0**
x_3	**0**	**0**	**0**	**0**	**0**	**0**
x_4	**0**	**0**	**1**	**0**	**0**	**0**
x_5	**1**	**0**	**0**	**1**	**0**	**0**
x_6	**1**	**0**	**0**	**0**	**1**	**1**

Матрица смежности полностью определяет структуру графа. Например, сумма всех элементов строки x_i матрицы дает полустепень исхода вершины x_i, а сумма элементов столбца x_i - полустепень захода вершины x_i. Множество столбцов, имеющих 1 в строке x_i есть множество $\Gamma(x_i)$, а множество строк, которые имеют 1 в столбце x_i совпадает с множеством $\Gamma^{-1}(x_i)$.

Петли на графе представляют собой элементы, имеющие 1 на главной диагонали матрицы, например a_{22}, a_{66} для графа, изображенного на рисунке 1.

В случае неориентированного графа матрица смежности является симметричной относительно главной диагонали (рисунок 2).

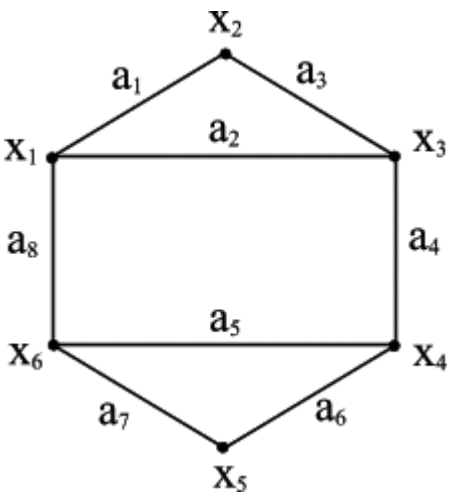


Рисунок 2.

A=

	x_1	x_2	x_3	x_4	x_5	x_6
x_1	**0**	**1**	**1**	**0**	**0**	**1**
x_2	**1**	**0**	**1**	**0**	**0**	**0**
x_3	**1**	**1**	**0**	**1**	**0**	**0**
x_4	**0**	**0**	**1**	**0**	**1**	**1**
x_5	**0**	**0**	**0**	**1**	**0**	**1**
x_6	**1**	**0**	**0**	**1**	**1**	**0**

Матрица инцидентности

Пусть дан граф *G* с *n* вершинами и *m* дугами. *Матрица инцидентности* графа *G* обозначается через **B**=[b_{ij}] и является матрицей размерности *n x m*, определяемой следующим образом:

b_{ij}=1, если x_i является начальной вершиной дуги a_j;

b_{ij}=-1, если x_i является конечной вершиной дуги a_j;

b_{ij}=0, если x_i не является концевой вершиной дуги a_j.

Для графа, приведенного на рисунке 1, матрица инцидентности имеет вид:

B=

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
x_1	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**-1**	**-1**	**0**
x_2	**-1**	**0**	**±1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**
x_3	**0**	**-1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
x_4	**0**	**0**	**0**	**0**	**-1**	**-1**	**0**	**0**	**0**	**0**
x_5	**0**	**0**	**0**	**-1**	**1**	**1**	**-1**	**1**	**0**	**0**
x_6	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**±1**

Поскольку каждая дуга инцидентна двум различным вершинам (за исключением случая, когда дуга образует петлю), то каждый столбец содержит один элемент, равный 1, и один - равный -1. Петля в матрице инцидентности не имеет адекватного математического представления (в программной реализации допустимо задание одного элемента b_{ij}=1).

Если *G* является неориентированным графом (рисунок 2), то его матрица инцидентности определяется следующим образом:

b_{ij}=1, если x_i является концевой вершиной дуги a_j;

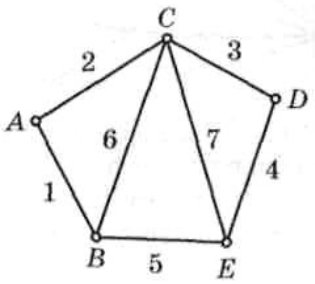
b_{ij}=0, если x_i не является концевой вершиной дуги a_j.

$$B=\begin{array}{c|cccccccc} & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 \\ \hline x_1 & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ x_2 & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ x_3 & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ x_4 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ x_5 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} \\ x_6 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} \end{array}$$

Матрица инцидентности, как способ задания графов, успешно применяется при описании мультиграфов (графов, в которых смежные вершины могут соединяться несколькими параллельными дугами).

Инструкция к практической работе

Задача. Для неориентированного графа, изображённого на рисунке, постройте матрицу смежности и матрицу инцидентности.



Решение:

Матрица смежности

$$\begin{array}{c} A \\ B \\ C \\ D \\ E \end{array}\begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$
$$\begin{array}{ccccc} A & B & C & D & E \end{array}$$

Матрица инцидентности

$$\begin{array}{c} A \\ B \\ C \\ D \\ E \end{array}\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$
$$\begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array}$$

Задача. Дана матрица $\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array}\begin{bmatrix} 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 3 & 1 & 0 & 0 \end{bmatrix}$
$$\begin{array}{cccc} 1 & 2 & 3 & 4 \end{array}$$

Постройте орграф, для которого данная матрица является матрицей смежности. Найдите матрицу инцидентности орграфа.

Решение: Для построения орграфа его вершине однозначно сопоставим точку на плоскости. Данная матрица смежности имеет четыре строки и четыре столбца, следовательно, в орграфе четыре вершины 1, 2, 3, 4.

Проанализируем элементы матрицы:

$a_{11}=0-$ при вершине 1 нет петель;

$a_{12}=2-$ из вершины 1 выходят две стрелки к вершине 2;

$a_{13}=0-$ из вершины 1 не выходит ни одной стрелки к вершине 3;

$a_{14}=0-$ из вершины 1 не выходит ни одной стрелки к вершине 4;

$a_{21}=0-$ из вершины 2 не выходит ни одной стрелки к вершине 1;

$a_{22}=0-$ при вершине 2 нет петель;

$a_{23}=1-$ из вершины 2 выходит одна стрелка к вершине 3;

$a_{24}=0-$ из вершины 2 не выходит ни одной стрелки к вершине 4;

$a_{31}=1-$ из вершины 3 выходит одна стрелка к вершине 1;

$a_{32}=0-$ из вершины 3 не выходит ни одной стрелки к вершине 2;

$a_{33}=0-$ при вершине 3 нет петель;

$a_{34}=1-$ из вершины 3 выходит одна стрелка к вершине 4;

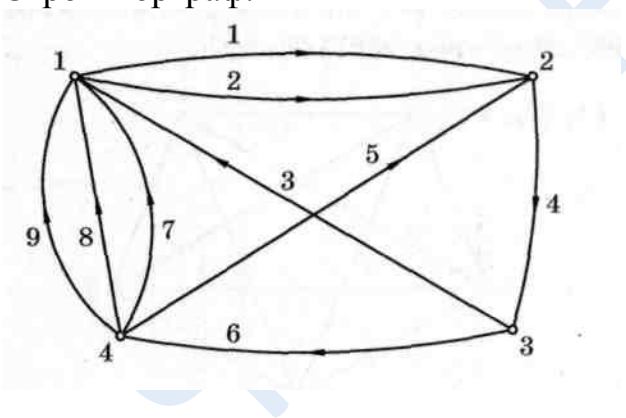
$a_{41}=3-$ из вершины 4 выходит 3 стрелки к вершине 1;

$a_{42}=1-$ из вершины 4 выходит одна стрелка к вершине 2;

$a_{43}=0-$ из вершины 4 не выходит ни одной стрелки к вершине 3;

$a_{44}=0-$ при вершине 4 нет петель.

Строим орграф.



Для построения графа запишем матрицу инцидентности:

$$\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ \\ \end{array}\begin{bmatrix} 1 & 1 & -1 & 0 & 0 & 0 & -1 & -1 & -1 \\ -1 & -1 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 1 & 1 & 1 \end{bmatrix}$$
$$\begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array}$$

Здесь четыре строки по числу вершин и 9 столбцов по числу дуг.

Задание:

Для графа заданного матрицей смежности

1. найти матрицу инцидентности
2. построить граф
3. Используя программное обеспечение, изобразите граф и проверьте матрицу.

Вариант 1

	1	2	3	4	5	6
1		1	1	1		1
2	1		1			1
3	1	1		1	1	1
4	1		1			1
5			1			1
6	1	1	1	1	1	

	1	2	3	4	5	6
1		1		1		1
2			1		1	
3				1		1
4					1	
5						1
6						

	1	2	3	4	5	6
1		1	1			1
2			1	1	1	
3					1	1
4					1	
5						1
6						

Вариант 2.

	1	2	3	4	5	6
1		1	1			1
2			1		1	
3				1		1
4					1	
5						1
6						

	1	2	3	4	5	6
1		1		1	1	
2			1		1	
3				1	1	1
4					1	1
5						1
6						

	1	2	3	4	5	6
1		1	1		1	
2	1		1	1	1	
3	1	1			1	1
4		1			1	1
5	1	1	1	1		1
6			1	1	1	

Вариант 3

	1	2	3	4	5	6
1		1		1		1
2			1	1	1	1
3				1	1	1
4						1
5						1
6						

	1	2	3	4	5	6
1		1	1	1	1	
2					1	
3				1	1	1
4					1	1
5						1
6						

	1	2	3	4	5	6
1		1		1	1	
2	1		1		1	1
3		1		1		
4	1		1			1
5	1	1				1
6		1		1	1	

Вариант 4

	1	2	3	4	5	6
1		1		1	1	
2			1			1
3				1	1	
4						1
5						1
6						

	1	2	3	4	5	6
1		1		1	1	1
2			1	1		
3				1	1	
4					1	1
5						
6						

	1	2	3	4	5	6
1		1	1	1		1
2	1		1			
3	1	1		1	1	1
4	1		1		1	1
5			1	1		1
6	1		1	1	1	

Вариант 5

	1	2	3	4	5	6
1		1	1		1	1

	1	2	3	4	5	6
1		1	1	1		

	1	2	3	4	5	6
1		1		1	1	1

Чащина Е.А.

2			1	1		1
3				1	1	
4						1
5						1
6						

2			1	1		1
3				1	1	1
4					1	1
5						1
6						

2	1		1	1	1	
3		1		1	1	1
4	1	1	1		1	
5	1	1	1	1		
6	1		1			

Порядок выполнения работы:

1. Изучить инструкцию к практической работе.
2. Выполнить задание.
3. Оформить отчет.

Содержание отчета:

1. Тема.
2. Цель.
3. Задачи.
4. Материальное обеспечение.
5. Практическое задание.

Вопросы для самоконтроля:

1. Перечислите основные способы представления графов.
2. В чем отличия матричного представления ориентированных и неориентированных графов?
3. В чем особенности представления графа матрицей смежности?
4. В чем особенности представления графа матрицей инцидентности?

Чащина Е.А.

Практическая работа № 9

Тема: Решение задач по теории графов. Выделение связных компонентов. Нахождение максимального потока и минимального разреза.

Цель: научиться определять компоненты связности и находить максимальный поток и строить минимальный разрез в сети с использованием алгоритма Форда-Фалкерсона; закрепить навыки моделирования графов в графической среде Grafoanalizator1.3.3 rus.Простой граф.

Задачи:

1. Закрепить знания основных понятий теории графов.
2. Приобрести практические умения использования специального программного обеспечения для моделирования.
3. Использовать математический аппарат теории графов

Материальное обеспечение:

Программы для графического представления графов: Grafoanalizator1.3.3 rus, Прстой граф, практическая работа.

Теоретическая часть:

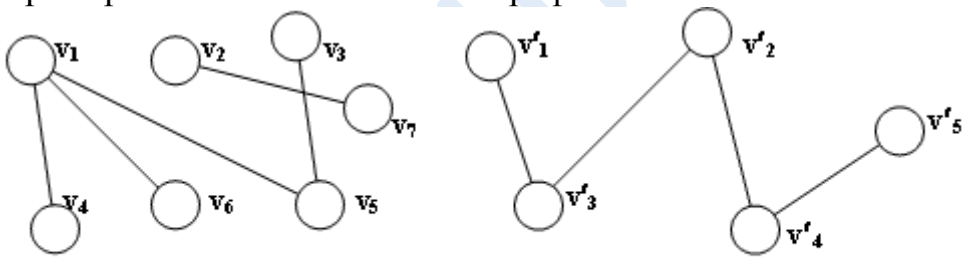
Компоненты связного и несвязного графа

В теории графов, понятие связности графа является ключевым при решении многих прикладных задач.

Определение связного и несвязного графа.

Граф G(V,E) называется связным, если для любой пары различных вершин этого графа существует цепь, соединяющая эти вершины. Если для графа G(V,E) можно указать пару различных вершин, которые не соединяются цепью, то граф называется несвязным.

Пример связного и несвязного графов.

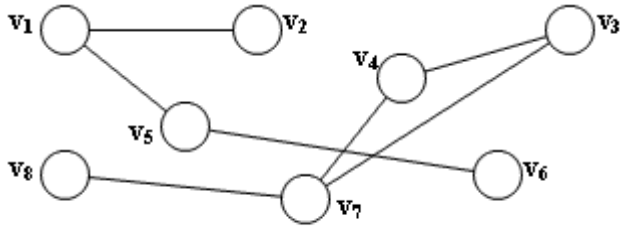


граф G=(V, E) граф G'=(V', E')

Компонента связности графа — некоторое множество вершин графа такое, что для любых двух вершин из этого множества существует путь из одной в другую, и не существует пути из вершины этого множества в вершину не из этого множества.

Для ориентированных графов определено понятие сильной компоненты связности

Пример компонент связности графа.



граф *G(V, E)*

Рассмотрим пример двусвязного графа G(V, E). Всего имеется два множества связных между собой вершин V1 = {v1, v2, v5, v6} и V2 = {v3, v4, v7, v8}.

Свойства компонент связности.

1. Каждая вершина графа входит лишь в одну компоненту связности.
2. Любой конечный граф имеет конечное число компонент связности.
3. Граф, состоящий из единственной компоненты связности, является связным.
4. Каждая компонента связности графа является его подграфом.

Теорема. Если в конечном графе G ровно две вершины u и v имеют нечетную степень, то они связаны.

Задание 1. Компоненты сильной связности ориентированного графа.

С помощью матрицы смежности найти компоненты сильной связности ориентированного графа D.

Составляем матрицу смежности $A(D)$ размерности $n \times n$ (n− количество вершин) для данного ориентированного графа: она состоит из нулей и единиц, номера строк – индексы вершин , из которых исходят дуги, номера столбцов – индексы вершин, в которые дуги входят (если есть дуга, исходящая из вершины v_i и входящая в v_j, то элемент матрицы смежности, стоящий на пересечении i-той строки и j-того столбца равен 1, иначе – 0.).

Для того, чтобы выделить компоненты сильной связности, необходимо сначала найти матрицу достижимости $T(D)$

Матрица достижимости ориентированного графа D − квадратная матрица $T(D)=[t_{ij}]$ порядка n, элементы которой равны

$$t_{ij} = \begin{cases} 1, & v_j \quad \textit{достижима} \quad \textit{из} \quad v_i, \\ 0, & \textit{в} \quad \textit{противном} \quad \textit{случае.} \end{cases}$$

ориентированного графа по (1) формуле (T(D)=sign[E+A+A2+A3+… An-1]), затем находим матрицу сильной связности $S(D)$ ориентированного графа (она должна быть симметрической) по (2) формуле ($S(D)=T(D)\&T^T(D)$ (T^T-транспонированная матрица, &- поэлементное умножение)).

Алгоритм выделения компонент сильной связности

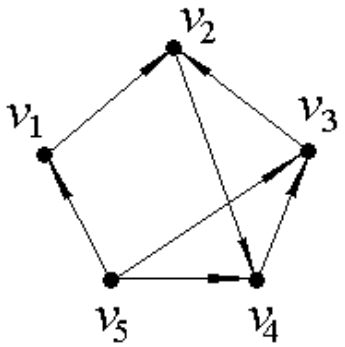
1. Присваиваем p=1 (p − количество компонент связности), $S_1 = S(D)$.

2. Включаем в множество вершин V_p компоненты сильной связности D_p вершины, соответствующие единицам первой строки матрицы S_p. В качестве матрицы $A(D_p)$ возьмем подматрицу матрицы $A(D)$, состоящую из элементов матрицы A, находящихся на пересечении строк и столбцов, соответствующих вершинам из V_p.

3. Вычеркиваем из S_p строки и столбцы, соответствующие вершинам из V_p. Если не остается ни одной строки (и столбца), то p- количество компонент сильной связности. В противном случае обозначим оставшуюся после вычеркивания срок и столбцов матрицу как S_{p+1}, присваиваем $p=p+1$ и переходим к п. 2.

Пример выполнения задания 1

Выделим компоненты связности ориентированного графа, изображенного на рисунке 1. В данной задаче количество вершин $n=5$.



Рисунке 1.

Значит, для данного ориентированного графа матрица смежности будет иметь размерность 5×5 и будет выглядеть следующим образом

$$A(D)=\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}.$$

Найдем матрицу достижимости для данного ориентированного графа по формуле (1)

$$sign\left[A^2(D)\right]=\begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix},\ sign\left[A^3(D)\right]=\begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix},$$

$$sign\left[A^4(D)\right]=\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix},$$

Следовательно,

$$T(D)=sign\left\{\begin{pmatrix}1&0&0&0&0\\0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1\end{pmatrix}+\begin{pmatrix}0&1&0&0&0\\0&0&0&1&0\\0&1&0&0&0\\0&0&1&0&0\\1&0&1&1&0\end{pmatrix}+\begin{pmatrix}0&0&0&1&0\\0&0&1&0&0\\0&0&0&1&0\\0&1&0&0&0\\0&1&1&0&0\end{pmatrix}+\right.$$

$$\left.+\begin{pmatrix}0&0&1&0&0\\0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&1&0&1&0\end{pmatrix}+\begin{pmatrix}0&1&0&0&0\\0&0&0&1&0\\0&1&0&0&0\\0&0&1&0&0\\0&0&1&1&0\end{pmatrix}=\begin{pmatrix}1&1&1&1&0\\0&1&1&1&0\\0&1&1&1&0\\0&1&1&1&0\\1&1&1&1&1\end{pmatrix}\right\}.$$

Таким образом, матрица сильной связности, полученная по формуле (2), будет следующей:

$$S(D)=\begin{pmatrix}1&1&1&1&0\\0&1&1&1&0\\0&1&1&1&0\\0&1&1&1&0\\1&1&1&1&1\end{pmatrix}\&\begin{pmatrix}1&0&0&0&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\0&0&0&0&1\end{pmatrix}=\begin{pmatrix}1&0&0&0&0\\0&1&1&1&0\\0&1&1&1&0\\0&1&1&1&0\\0&0&0&0&1\end{pmatrix}.$$

Присваиваем p=1 $S_1=S(D)$ и составляем множество вершин первой компоненты сильной связности D_1: это те вершины, которым соответствуют единицы в первой строке матрицы $S(D)$. Таким образом, первая компонента сильной связности состоит из одной вершины $V_1=\{v_1\}$.

Вычеркиваем из матрицы $S_1(D)$ строку и столбец, соответствующие вершине v_1, чтобы получить матрицу $S_2(D)$:

$$S_2(D)=\begin{pmatrix}1&1&1&0\\1&1&1&0\\1&1&1&0\\0&0&0&1\end{pmatrix}.$$

Присваиваем p=2. Множество вершин второй компоненты связности составят те вершины, которым соответствуют единицы в первой строке матрицы $S_2(D)$, то есть $V_2=\{v_2,\ v_3,\ v_4\}$. Составляем матрицу смежности для компоненты сильной связности D_2 исходного графа D − в ее качестве возьмем подматрицу матрицы $A(D)$, состоящую из элементов матрицы A, находящихся на пересечении строк и столбцов, соответствующих вершинам из V_2:

$$A(D_2)=\begin{pmatrix}0&0&1\\1&0&0\\0&1&0\end{pmatrix}.$$

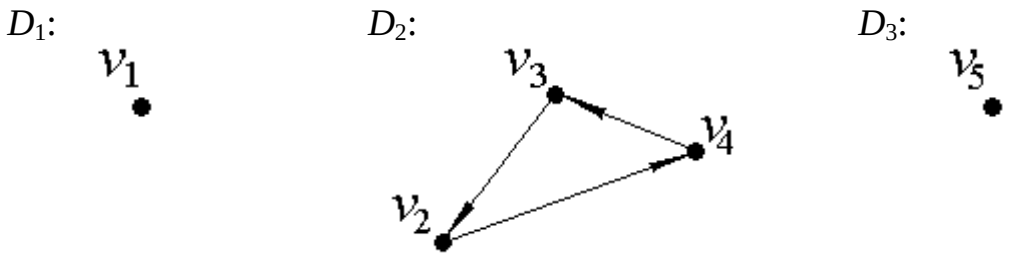
Вычеркиваем из матрицы $S_2(D)$ строки и столбцы, соответствующие вершинам из V_2 ,чтобы получить матрицу $S_3(D)$, которая состоит из одного элемента:

ЧащинаЕ.А.

$$S_3(D) = (1)$$

и присваиваем p=3. Таким образом, третья компонента сильной связности исходного графа, как и первая, состоит из одной вершины $V_3 = \{v_5\}$.

Таким образом, выделены 3 компоненты сильной связности ориентированного графа D:



Максимальный поток и минимальный разрез

Теорема Форда-Фалкерсона 1 (о максимальном потоке и минимальном разрезе).
В любой сети существует максимальный поток. Величина максимального потока равна пропускной способности минимального разреза.
Теорема Форда-Фалкерсона 2.
Поток, вычисленный с помощью алгоритма Форда-Фалкерсона имеет максимальную величину, а разрез $(X, \overline{X})$, где X -множество вершин, помеченных при последнем помечивании, имеет минимальную пропускную способность.

Нахождение максимального потока и построение минимального разреза в сети с использованием алгоритма Форда-Фалкерсона

В данной задаче основным параметром на дугах сети является c_{ij} – *пропускная способность*. Пропускная способность показывает, сколько единиц потока может быть передано по дугам сети. Таким образом, *потоком в сети* $D = [N, M]$ называется неотрицательная вещественная функция, удовлетворяющая условиям:
1. *ограниченности*: поток по любой дуге сети не превосходит пропускной способности этой дуги $f_{ij} \le c_{ij}$;
2. *сохранения*: суммарный поток, заходящий в любую вершину сети (кроме истока и стока), равен суммарному потоку, выходящему из этой вершины.
Дуга сети называется *насыщенной*, если поток по этой дуге равен пропускной способности этой дуги, т. е. $f_{ij} = c_{ij}$.
Разрезом сети называется множество дуг, удаление которых из сети приводит к тому, что исток и сток оказываются несвязанными.
Пропускной способностью разреза называется число, равное сумме пропускных способностей дуг этого разреза. Разрез называется *минимальным*, если имеет наименьшую пропускную способность.
Отыскание минимального разреза – одна из основных задач анализа транспортных сетей. В силу конечности графа минимальный разрез может быть найден перебором всех разрезов, но этот путь, конечно, неприемлем для достаточно больших графов.

Чащина Е.А.

Минимальный разрез можно отыскать при помощи теоремы Форда – Фалкерсона: в любой транспортной сети величина любого *максимального потока* равна пропускной способности любого *минимального разреза*.
Для нахождения максимального потока в сети разработан *алгоритм Форда – Фалкерсона*. Перед началом выполнения алгоритма все вершины сети нумеруются произвольным образом, кроме источника и стока (источник получает минимальный номер 1, сток – максимальный n, где n – число узлов).

Алгоритм состоит из следующих основных шагов:

1. Определить начальный поток в сети, сложив потоки по дугам, выходящим из источника.
2. Вершинам сети присвоить целочисленные метки, а дугам – знаки «+» и «–» по следующим правилам:

а) вершине-истоку присвоить метку $M_1 = 0$;

б) находим *непомеченную* вершину v, *смежную помеченной* вершине u. Если поток по соединяющей вершины $u - v$ дуге меньше пропускной способности этой дуги, то происходит *помечивание*, иначе переходим к рассмотрению следующей вершины. Если вершина v является *образом* помеченной вершины u, то происходит *прямое помечивание* (*дуга в прямом направлении допустима*): вершина v получает метку, равную номеру вершины u, соединяющая вершины $u - v$ дуга получает метку «+», переходим к рассмотрению следующей вершины. Если вершина v не имеет ни одного помеченного прообраза, поток по дуге в прямом направлении больше 0, то происходит *обратное помечивание* (*дуга допустима в обратном направлении*): вершина u получает метку, равную номеру вершины v (являющейся в данном случае ее образом), соединяющая вершины $v - u$ дуга получает метку «–», происходит переход к рассмотрению следующей вершины. Процесс помечивания продолжается до тех пор, пока все удовлетворяющие этим условиям вершины не получат метку.

3. Если в результате процедуры помечивания вершина-сток метки не получила, то текущий поток – максимальный, переход к шагу 5. В противном случае перейти к пункту 4.
4. Рассмотреть последовательность вершин *L*, метка каждой из которых равна номеру следующей за ней вершины, и множество дуг *ML*, соединяющих соседние вершины из *L*.

Построение нового потока по схеме:

а) Если дуга принадлежит множеству *ML* (смотри выше) и имеет знак «+», то *новый поток по этой дуге* = *старый поток по этой дуге* + Δ (схему нахождения смотри далее).

б) Если дуга принадлежит множеству *ML* и имеет знак «–», то *новый поток по этой дуге* = *старый поток по этой дуге* – Δ.

в) Если дуга не принадлежит множеству *ML*, то поток по дуге оставляем без изменения.

Схема нахождения Δ:

I. $\Delta = \min\{K_1, K_2\}$, *где для нахождения* K_1 рассматриваются все дуги, принадлежащие множеству *ML* и имеющие знак «+», и для каждой такой дуги вычисляется разность

между пропускной способностью дуги и потоком по этой дуге ($c_{ij} - f_{ij}$). Затем из этих значений разностей выбирается минимальное значение и присваивается K_1.

II. *Для нахождения* K_2 рассматриваются все дуги, принадлежащие множеству *ML* и имеющие знак «–». Затем из этих дуг выбирается дуга с минимальным потоком (f_{ij}), и значение потока по этой дуге присваивается K_2.

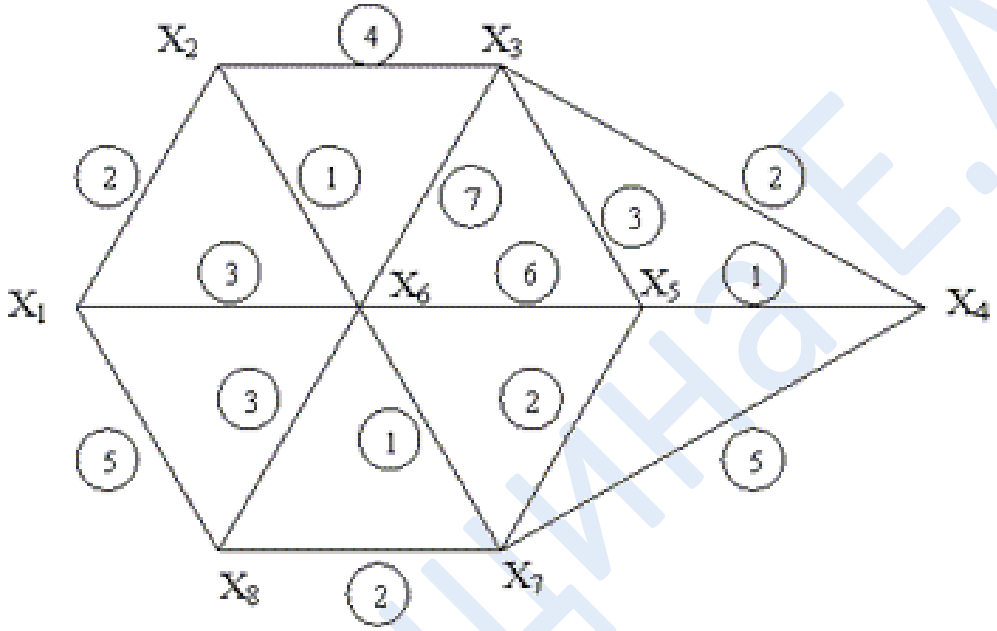
Перейти к шагу 2.

5. Определяем максимальный поток, складывая начальный поток и все полученные изменения потока.

В оптимальном решении, т. е. когда найден максимальный поток, минимальный разрез образуется насыщенными дугами.

Пример № 1 выполнения задания 2

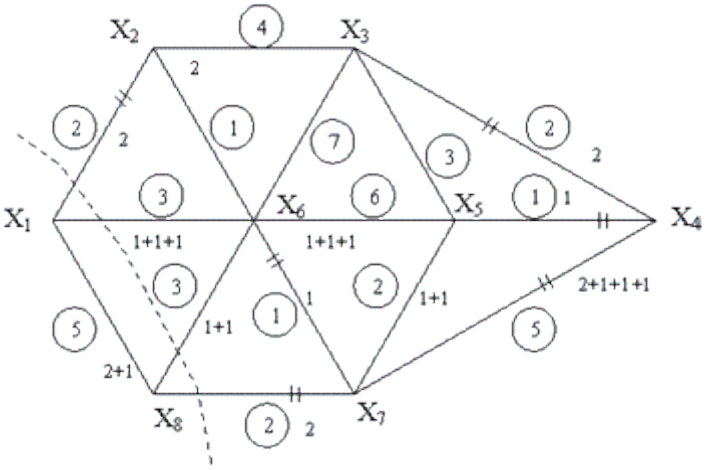
На заданной сети найти максимальный поток из X4 в X1 и минимальный разрез.



Решение

Необходимо заполнить таблицу:

	1	2	3	4	5	6	7
t X_1	2, +X_2	1, +X_6	2, +X_8	1, +X_6	1, +X_6	1,+X_8	
X_2	2, +X_3	1, +X_3			1, +X_6		
X_3	2, +X_4	1, +X_5			2, +X_5	1, +X_5	
s X_4	∞,+X_4	∞,+X_4	∞,+X_4	∞,+X_4	∞,+X_4	∞,+X_4	∞,+X_4
X_5	1, +X_4	1, +X_4	2, +X_7	2, +X_7	2, +X_7	1, +X_7	
X_6	2, X_3	1, +X_5	2, +X_7	1, +X_7	2, +X_5	1, X_5	
X_7	5, +X_4	5, +X_4	5, +X_4	3, +X_4	2, +X_4	1, +X_4	
X_8			2, +X_7			1, +X_6	
V	2	1	2	1	1	1	



После первого шага увеличиваем потоки в дугах, которые сначала были везде = 0.
Далее продолжаем наращивать поток по цепям:

X4, X3, X2, X1, V1 = 2
X4, X5, X6, X1, V2 = 1
X4, X7, X8, X1, V3 = 2
X4, X7, X6, X1, V4 = 1
X4, X7, X5, V5 = 1
X4, X7, X5, X6, X8, V6 = 1

$\sum V_i = 8$, т.е. максимальный поток = 8.

На последнем, 7-ом шаге, на котором мы не достигли (не можем пометить) вершину t = X1 находим все помеченные вершины:

Xs = {X4}

и непомеченные вершины

Xt = {X1, X2, X3, X5, X6, X7, X8}.

Минимальный разрез – ребра, один конец которых лежит в Xs, а другой в Xt.
В нашем случае это:

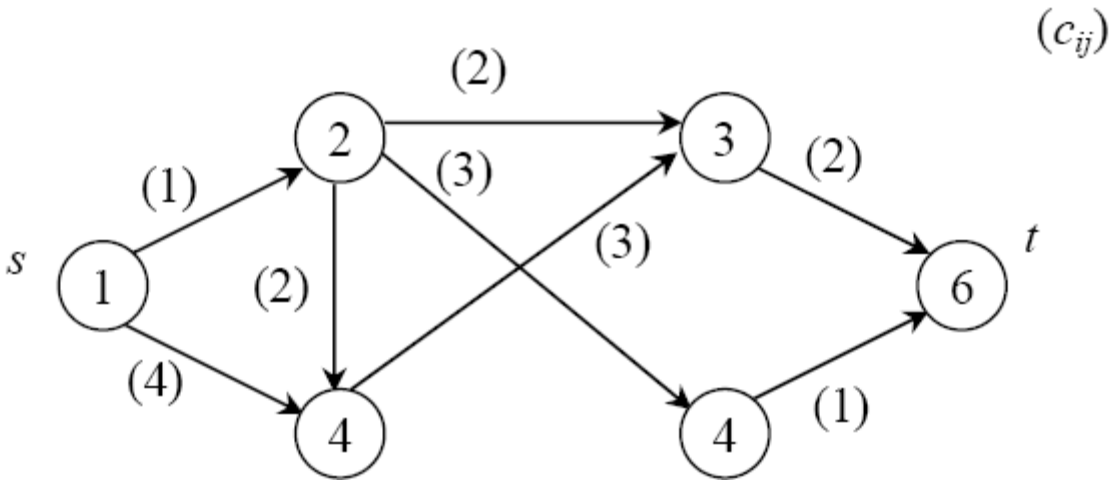
(X4, X3), V(X4, X3) = 2;
(X4, X5), V(X4, X5) = 1;
(X4, X7), V(X4, X7) = 5

$\sum V = 8$

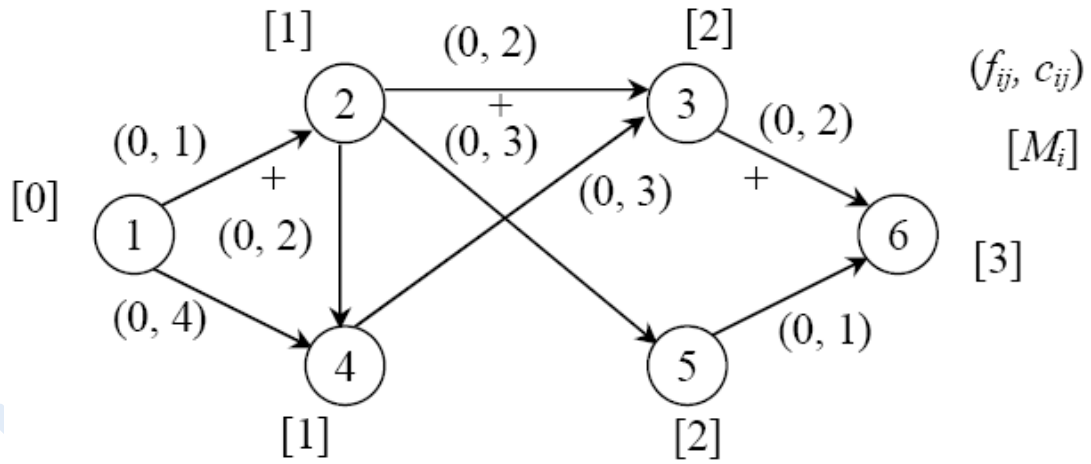
Т.е. величина минимального разреза совпадает с максимальным потоком.

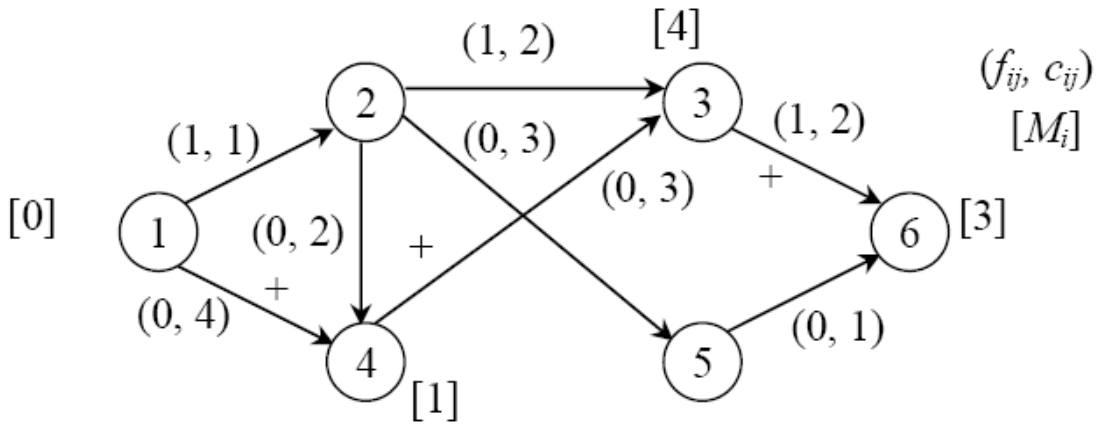
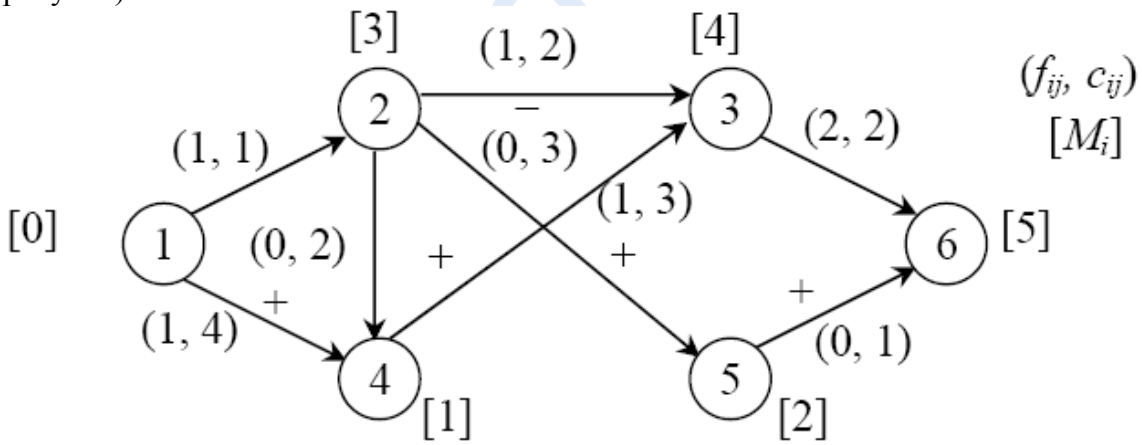
Пример № 2 выполнения задания № 2.

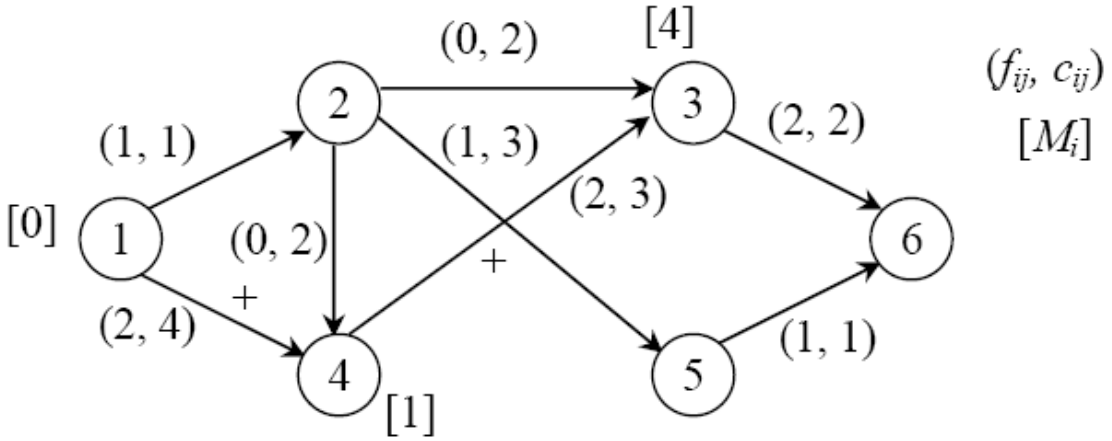
Постановка задачи поиска максимального потока: найти максимальный поток из s в t для транспортной сети (рисунок) с помощью алгоритма Форда – Фалкерсона:



Решение:

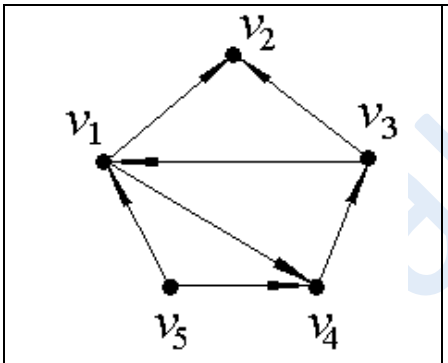
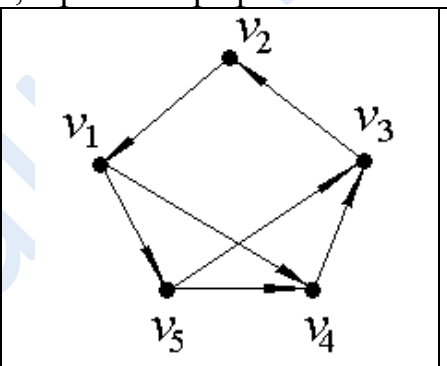
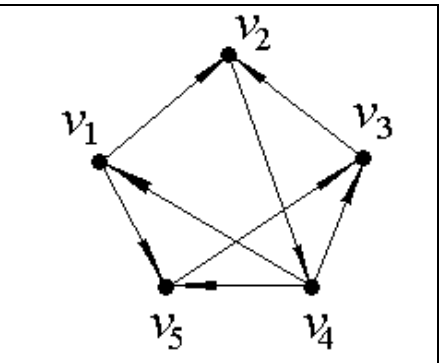
№	Алгоритм	Конкретные действия
1.	*1-я итерация*	1. $f_0 = f_{12} + f_{14} = 0$. 2.1 (Второй шаг, первая итерация – подобное обозначение идет далее для всех шагов алгоритма). Производим помечивание вершин и дуг, результат показан на рисунке. Вершина 6 получила метку $M_6 = 3$. 
2.	*2-я итерация*	3.1. $\Delta_1 = \min\{c_{12}, c_{23}, c_{36}\} = \min\{1,2,2\} = 1$. 4.1. $f'_{12} = f_{12} + \Delta_1 = 0 + 1 = 1$; $f'_{23} = f_{23} + \Delta_1 = 0 + 1 = 1$; $f'_{36} = f_{36} + \Delta_1 = 0 + 1 = 1$. 2.2. Заново осуществляется помечивание. Вершина 6 снова получает метку $M_6 = 3$ (смотри рисунок).

		
3.	*3-я итерация*	3.2. $\Delta_2 = \min\{c_{14}, c_{43}, c_{36} - f_{36}\} = \min\{4,3,2-1\} = 1$. 4.2. $f'_{14} = f_{14} + \Delta_2 = 0+1 = 1$; $f'_{43} = f_{43} + \Delta_2 = 0+1 = 1$; $f'_{36} = f_{36} + \Delta_2 = 1+1 = 2$. 2.3. Осуществляется помечивание. При этом из вершины 3 прямых допустимых дуг не выходит, однако дуга 2–3 является допустимой в обратном направлении, и вершина 2 получает метку $M_2 = 3$. Вершина 6 получает метку $M_6 = 5$ (смотри рисунок). 
4.	*4-я итерация*	3.3. $\Delta_3 = \min\{c_{14} - f_{14}, c_{43} - f_{43}, f_{23}, c_{25}, c_{56}\} = \min\{3,2,1,3,1\} = 1$. 4.3. $f'_{14} = f_{14} + \Delta_3 = 1+1 = 2$; $f'_{43} = f_{43} + \Delta_3 = 1+1 = 2$; $f'_{32} = f_{32} - \Delta_3 = 1-1 = 0$; $f'_{25} = f_{25} + \Delta_3 = 0+1 = 1$; $f'_{56} = f_{56} + \Delta_3 = 0+1 = 1$. 2.4. Осуществляется помечивание. При этом из вершины 3 допустимые дуги не выходят. Вершина 6 не получает метку (смотри рисунок). Переходим к шагу 5.

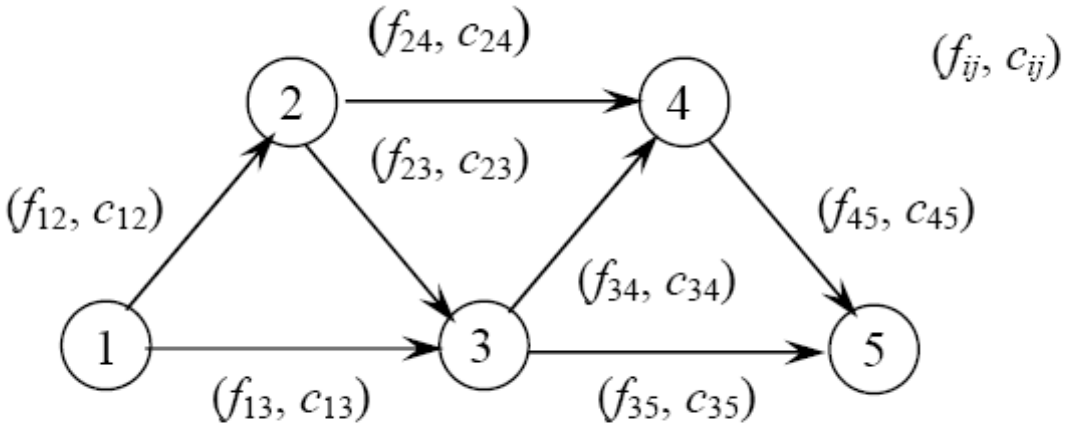
		
5.		5. $f_{\max} = f_0 + \Delta_1 + \Delta_2 + \Delta_3 = 0+1+1+1=3$. Минимальный разрез образуют насыщенные дуги 3–6 и 5–6. Пропускная способность минимального разреза $c_{\min} = c_{36} + c_{56} = 2+1=3$. Условия теоремы Форда – Фалкерсона выполняются $f_{\max} = c_{\min} \Rightarrow$ задача решена правильно. Алгоритм Форда – Фалкерсона используется при решении многих практических задач. Одна из них – *задача об источниках и потребителях*.

Задание:

1. С помощью матрицы смежности найти компоненты сильной связности ориентированного графа D. При решении использовать программу Grafoanalizator1.3.3 rus, Простой граф

		
Вариант1	Вариант 2	Вариант 3

2. Дана сеть:



Определить максимальный поток в сети при начальных значениях дуговых потоков:

$f_{12} = 2$, $f_{13} = 0$, $f_{23} = 1$, $f_{24} = 1$, $f_{34} = 1$, $f_{35} = 0$, $f_{45} = 2$.

Варианты значений пропускных способностей дуг для задания:

№ варианта	c_{12}	c_{13}	c_{23}	c_{24}	c_{34}	c_{35}	c_{45}
1	2	2	1	3	2	3	3
2	2	3	1	2	2	2	4
3	3	2	4	4	2	3	2
4	4	2	3	2	4	4	2
5	2	4	2	2	3	2	3
6	2	3	1	2	2	2	4
7	3	2	4	4	2	3	2
8	4	2	3	2	4	4	2
9	2	2	3	2	2	3	3
10	2	4	2	2	3	2	3
11	2	4	3	3	2	3	4
12	2	3	2	1	2	2	4
13	4	1	1	2	1	3	2
14	3	2	4	4	2	1	3
15	3	6	8	8	2	7	2
16	6	4	3	6	3	2	5
17	4	5	2	3	6	4	7
18	4	1	1	2	9	3	4
19	3	2	5	3	3	1	4
20	4	5	3	9	2	3	3
21	2	4	3	3	1	3	4
22	3	2	1	3	3	1	4
23	4	2	2	3	4	4	2
24	5	3	2	4	2	6	4
25	4	1	1	2	1	3	2

Порядок выполнения работы:

1. Изучить примеры выполнения заданий 1 и 2 .

2. Выполнить задание.
3. Оформить отчет.

Содержание отчета:

1. Тема.
2. Цель.
3. Задачи.
4. Материальное обеспечение.
5. Практическое задание.

Вопросы для самоконтроля:

1. Перечислите основные компоненты связности графов.
2. В чем смысл матрицы достижимости?
3. Запишите теорему Форда-Фалкерсона 1 (о максимальном потоке и минимальном разрезе).

Чащина Е.А.

Практическая работа № 10

Тема: Решение задач по теории графов. Нахождение путей в графе

Цель: реализовать алгоритмы обработки графовых структур: поиск различных путей; получение навыком при решении задач нахождения пути в графе; закрепить навыки моделирования графов в графической среде Grafoanalizator1.3.3 rus, Простой граф, Grin.

Задачи:

1. Закрепить знания основных понятий теории графов.
2. Приобрести практические умения использования специального программного обеспечения для моделирования.
3. Использовать математический аппарат теории графов.
4. Применение алгоритмов поиска кратчайшего пути.
5. Планирование структуры сети с помощью графа с оптимальным расположением узлов.

Материальное обеспечение:

Программы для графического представления графов: Grafoanalizator1.3.3 rus? Простой граф, Grin.

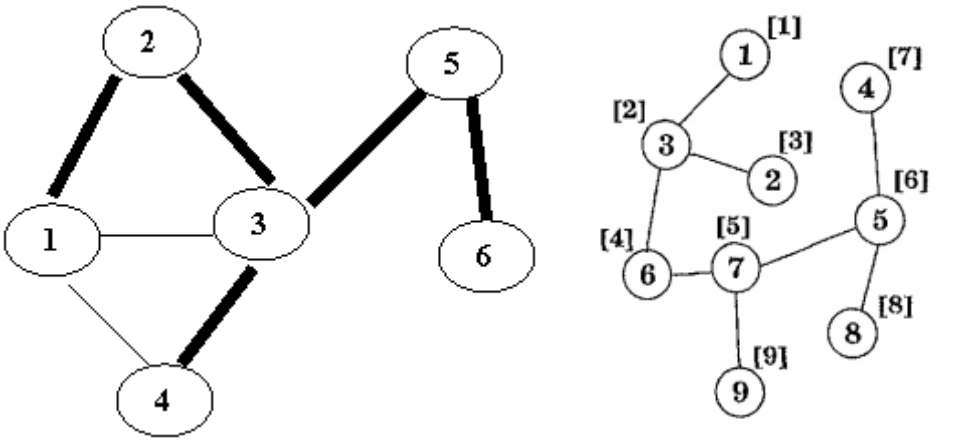
практическая работа. Интернет ссылка: http://inf.reshuege.ru/test?theme=203

Теоретическая часть:

В основе построения большинства алгоритмов на графах лежит систематический перебор вершин графа, при котором каждая вершина просматривается в точности один раз, а количество просмотров ребер графа ограничено заданной константой (лучше – не более одного раза).

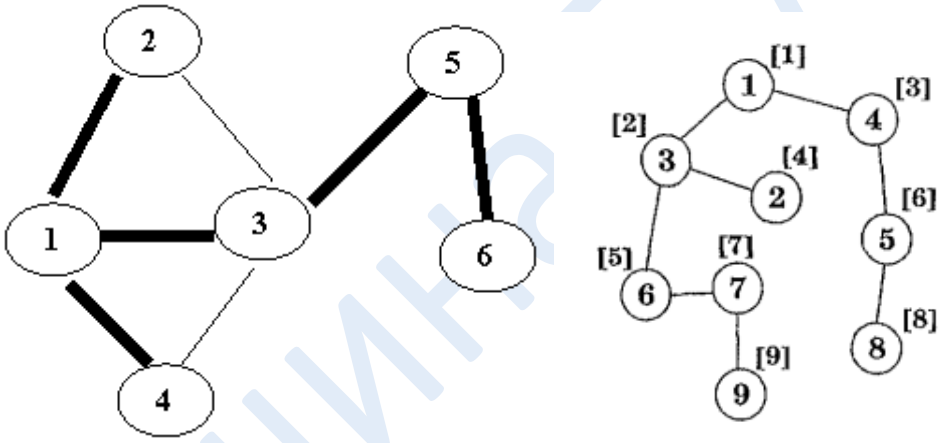
Один из основных методов проектирования графовых алгоритмов – это поиск (или обход графа) в глубину (depth first search, DFS), при котором начиная с произвольной вершины **v0**, ищется ближайшая смежная вершина **v**, для которой в свою очередь осуществляется поиск в глубину (т.е. снова ищется ближайшая, смежная с ней вершина) до тех пор, пока не встретится ранее просмотренная вершина, или не закончится список смежности вершины **v** (то есть вершина полностью обработана). Если нет новых вершин, смежных с **v**, то вершина **v** считается использованной, идет возврат в вершину, из которой попали в вершину **v,** и процесс продолжается до тех пор, пока не получим **v = v0**. Иными словами, поиск в глубину из вершины **v** основан на поиске в глубину из всех новых вершин, смежных с вершиной **v**.

Путь, полученный методом поиска в глубину, в общем случае не является кратчайшим путем из вершины **v** в вершину **u**. Это общий недостаток поиска в глубину. На рисунке показан путь, полученный обходом графа в глубину.



Указанного недостатка лишен другой метод обхода графа – поиск в ширину (breadth first search, BFS). Обработка вершины **v** осуществляется путем просмотра сразу всех новых соседей этой вершины. При этом полученный путь является кратчайшим путем из одной вершины в другую.

показан путь, найденных методом поиска в ширину



При использовании алгоритмов DFS и BFS графы обходят разными способами, получая при этом некоторые подграфы, которые имеют специфические названия: каркасы, остовы или стягивающие деревья. Все вершины исходного графа входят в полученное стягивающее дерево, а все ветви дерева – это множество ребер из все множеств ребер графа.

Произвольный путь в графе, проходящий через каждое ребро графа точно один раз, называется эйлеровым путем. При этом, если по некоторым вершинам путь проходит неоднократно, то он является непростым. Если путь замкнут, то имеем эйлеров цикл. Для существования эйлерова пути в связном графе необходимо и достаточно, чтобы граф содержал не более двух вершин нечетной степени.

Путь в графе, проходящий в точности один раз через каждую вершину графа (а не каждое ребро) и соответствующий цикл называются гамильтоновыми и существуют не для каждого графа, как и эйлеров путь. В отличие от эйлеровых путей неизвестно ни одного простого необходимого и достаточного условия для существования гамильтоновых путей. Неизвестен даже алгоритм полиномиальной сложности, проверяющий существование гамильтонова пути в произвольном графе. Проблема существования гамильтонова пути принадлежит к классу так называемых NP-полных задач.

В теории алгоритмов классом NP (от англ. *non-deterministic polynomial*) называют множество задач распознавания (*англ.*), решение которых при наличии некоторых дополнительных сведений (так называемого *сертификата решения*) можно «быстро» (за время, не превосходящее полинома от размера данных) проверить на машине Тьюринга.

NP-полная задача — задача из класса NP, к которой можно свести любую другую задачу из класса NP за полиномиальное время. Таким образом, NP-полные задачи образуют в некотором смысле подмножество «самых сложных» задач в классе NP; и если для какой-то из них будет найден «быстрый» алгоритм решения, то и любая другая задача из класса NP может быть решена так же «быстро».

При решении некоторых практических задач возникает необходимость поиска максимального пути (пути с наибольшей суммой длин дуг). Такая задача сводится к задаче нахождения минимального пути заменой знаков при длинах дуг (в матрице весов С) на противоположные . При этом необходимым является требование отсутствия в ориентированном графе контуров положительной длины .

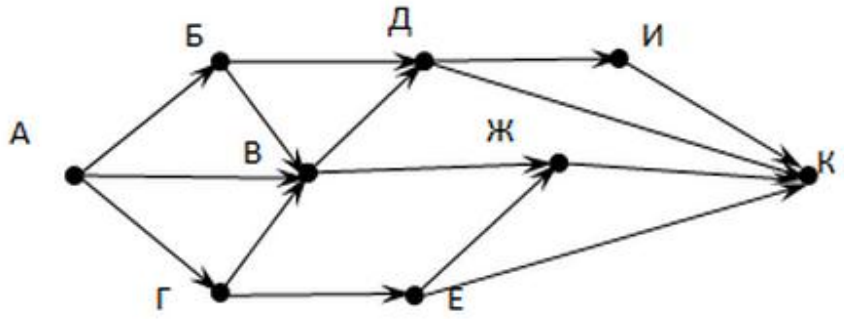
Поиск путей

знать:

- если в город R можно приехать только из городов X, Y, и Z, то число различных путей из города A в город R равно сумме числа различных путей проезда из A в X, из A в Y и из A в Z, то есть $N_R = N_X + N_Y + N_Z$, где N_Q обозначает число путей из вершины A в некоторую вершину Q
- число путей конечно, если в графе нет циклов – замкнутых путей

Пример задания:

На рисунке – схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К?



Решение (1 вариант, подстановки):

1) начнем считать количество путей с конца маршрута – с города К
2) будем обозначать через N_X количество различных путей из города А в город Х
3) общее число путей обозначим через N
4) по схеме видно, что $N_Б = N_Г = 1$
5) очевидно, что если в город Х можно приехать только из Y, Z, то $N_X = N_Y + N_Z$, то есть нужно сложить число путей, ведущих из А во все города, откуда можно приехать в город Х

6) поскольку в К можно приехать из Е, Д, Ж или И, поэтому
$N = N_К = N_Д + N_Е + N_Ж + N_И$

7) в город И можно приехать только из Д, поэтому $N_И = N_Д$

8) в город Ж можно приехать только из Е и В, поэтому
$N_Ж = N_Е + N_В$

9) подставляем результаты пп. 6 и 7 в формулу п. 5:
$N = N_В + 2N_Е + 2N_Д$

10) в город Д можно приехать только из Б и В, поэтому
$N_Д = N_Б + N_В$
так что
$N = 2N_Б + 3N_В + 2N_Е$

11) в город Е можно приехать только из Г, поэтому $N_Е = N_Г$ так что
$N = 2N_Б + 3N_В + 2N_Г$

12) по схеме видно, что $N_Б = N_Г = 1$, кроме того, $N_В = 1 + N_Б + N_Г = 3$

13) окончательно $N = 2N_Б + 3N_В + 2N_Г = 2\cdot1 + 3\cdot3 + 2\cdot1 = 13$

14) Ответ: 13.

Решение (2 вариант, удобная форма записи):

1) начнем считать количество путей с конца маршрута – с города К

2) записываем для каждой вершины, из каких вершин можно в нее попасть

вершина	откуда?
К	ИДЖЕ
И	Д
Ж	ВЕ
Е	Г
Д	БВ
Г	А
В	АБГ
Б	А

К ← ИДЖЕ
И ← Д
Ж ← ВЕ
Е ← Г
Д ← БВ
Г ← А
В ← АБГ
Б ← А

3) теперь для удобства «обратного хода» вершины можно отсортировать так[1], чтобы сначала шли все вершины, в которые можно доехать только из начальной точки А:

вершина	откуда?	N
Б	А	1
Г	А	1
В	АБГ	3
Е	Г	1
Д	БВ	4
Ж	ВЕ	4
И	Д	4
К	ИДЖЕ	13

Б ← А
Г ← А
затем на каждом шаге добавляем те вершины, в которые можно доехать из уже добавленных в список (и из исходной точки):
В ← АБГ
Е ← Г
далее добавляем все вершины, куда можно доехать из А, Б, Г, В и Е:
Д ← БВ
Ж ← ВЕ
на следующем шаге добавляем вершину И
И ←Д
и, наконец, конечную. вершину
К ←ИДЖЕ
именно в таком порядке мы и будем вычислять количество путей для каждой вершины
[1] Такая процедура называется *топологической сортировкой графа*.
4) теперь идем по полученному списку вершин, полагая, что количество вариантов попасть в вершину равно суммарному количеству вариантов попасть в ее непосредственных предшественников.
$N_Б = 1$, $N_Г = 1$
$N_В = 1+1+1 = 3$, $N_Е = 1$
$N_Д = 1+3 = 4$, $N_Ж = 3 + 1 = 4$
$N_И = 4$,
$N = N_К = 4 + 4 + 4 + 1 = 13$
5) заметим, что вершины можно и не сортировать специально, а просто выбирать возможный порядок вычисления: проверять, какие значения известны и какие можно рассчитать с их помощью на следующем шаге
6) Ответ: 13.

Возможные ловушки и проблемы:

- очень важна аккуратность и последовательность; сначала идем от конечной точки к начальной, выписывая все вершины, из которых можно приехать в данную; затем идем обратно, определяя числовые значения

- построение полного дерева маршрутов – занятие трудоемкое и достаточно бесперспективное, даже грамотные учителя информатики здесь в большинстве случаев что-то забывают и ошибаются

Решение (3 вариант, перебор вершин по алфавиту):

1) Запишем вершины в алфавитном порядке и для каждой из них определим, из каких вершин можно в нее попасть

вершина	откуда?
Б	А
В	АБГ
Г	А
Д	БВ
Е	Г
Ж	ВЕ
И	Д
К	ИДЖЕ

Б ← А
В ← АБГ
Г← А
Д ←БВ
Е ← Г
Ж ← ВЕ
И ← Д
К ← ИДЖЕ

2) теперь определяем количество путей; сначала ставим 1 для тех вершин, в которые можно проехать только из начальной (А):

вершина	откуда?	N
Б	А	1
В	АБГ	
Г	А	1
Д	БВ	
Е	Г	
Ж	ВЕ	
И	Д	
К	ИДЖЕ	

3) затем на каждом шаге добавляем те вершины, в которые можно доехать из уже добавленных в список (и из исходной точки):

вершина	откуда?	N
Б	А	1
В	АБГ	3
Г	А	1
Д	БВ	
Е	Г	1
Ж	ВЕ	
И	Д	
К	ИДЖЕ	

4) следующий шаг

вершина	откуда?	N
Б	А	1
В	АБГ	3
Г	А	1
Д	БВ	4
Е	Г	1
Ж	ВЕ	4
И	Д	
К	ИДЖЕ	

5) и последние 2 шага

вершина	откуда?	N
Б	А	1
В	АБГ	3
Г	А	1
Д	БВ	4
Е	Г	1
Ж	ВЕ	4
И	Д	4
К	ИДЖЕ	13

6) Ответ: 13.

Решение (4 вариант, перебор всех путей с начала, А. Яфарова):

1) запишем все вершины, в которые есть прямой путь из вершины А: Б, В и Г; получается три начальных отрезка:

АБ, АВ, АГ

2) рассмотрим маршрут АБ: из Б можно ехать в В и Д, поэтому получаем два маршрута:

АБ***В***, АБ***Д***

3) рассматриваем конечные точки этих маршрутов: из В можно ехать в Д и Ж, а из Д – в И и К:

АБВ***Д***, АБВ***Ж***, АБД***И***, АБД***К***

4) снова смотрим на конечные точки: из Д едем в И и К, из Ж и И – только в К:
АБВД***И***, АБВД***К***, АБВЖ***К***, АБДИ***К***, АБДК

5) из И едем только в К, таким образом, все возможные маршруты, содержащие участок АБ, доведены до конечной точки К, всего **5 таких маршрутов**:
АБВДИ***К***, АБВДК, АБВЖК, АБДИК, АБДК

6) затем аналогично рассматриваем маршруты, которые начинаются с АВ:
АВД, АВЖ
АВДИ, АВДК, АВЖК
АВДИК, АВДК, АВЖК
всего **3 маршрута**

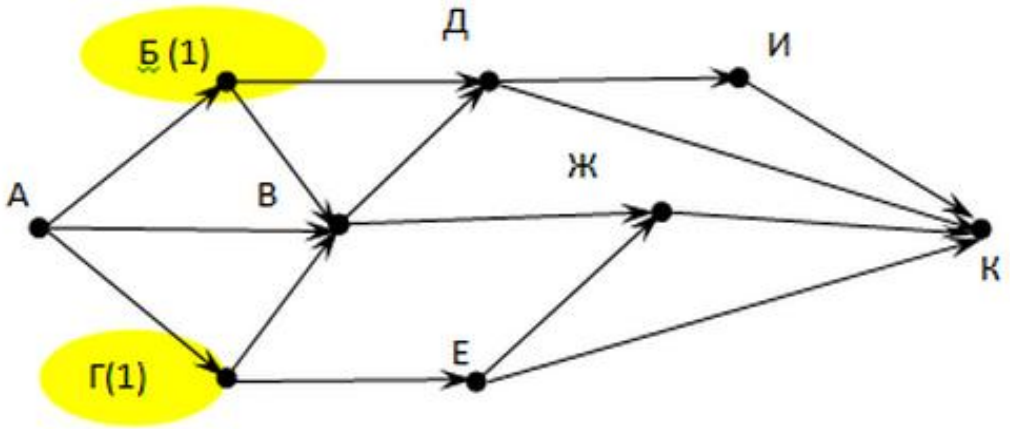
7) наконец, остается рассмотреть маршруты, которые начинаются с АГ:
АГВ, АГЕ
АГВД, АГВЖ, АГЕЖ, АГЕК
АГВДИ, АГВДК, АГВЖК, АГЕЖК, АГЕК
АГВДИК, АГВДК, АГВЖК, АГЕЖК, АГЕК
всего **5 маршрутов**

8) складываем количество маршрутов для всех начальных участков: 5 + 3 + 5 = 13

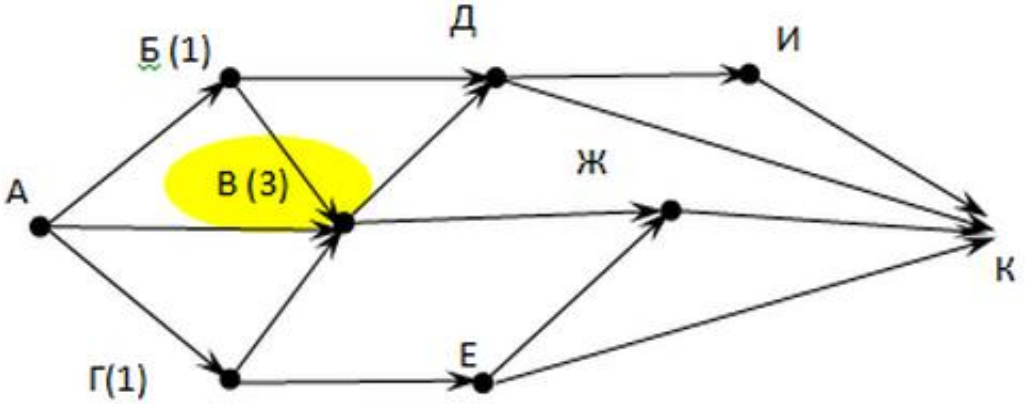
9) Ответ: 13.

Возможные проблемы:

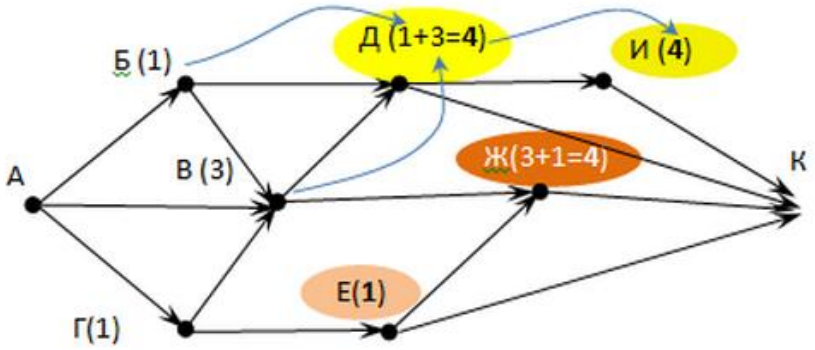
- при большом количестве маршрутов легко запутаться и что-то пропустить
- **Решение (5 вариант, графический):**
- 1) Главную идею решения: (число дорог в город N есть сумма дорог, приводящих в города, из которых есть прямой проезд в город N), отразим на самой схеме, показывая на ней ЧИСЛО ДОРОГ, приводящих в каждый город.
- 2) Последовательность очевидна: начинаем с Б и Г (городов, куда есть по 1-й дороге из А)



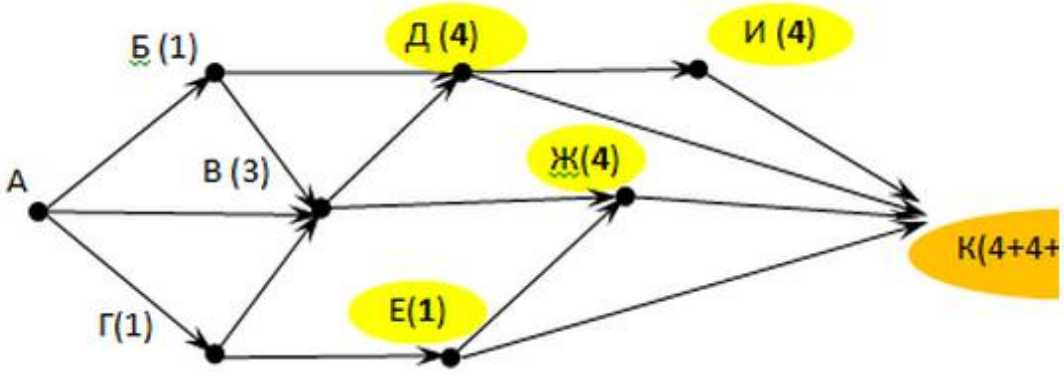
3) Посчитаем дороги в В: 1 (из А)+ 1(дороги города Б)+ 1(дороги города В)= 3



4) Аналогично посчитаем дороги в Д, И, Е, Ж:



5) Определяем число дорог в город К, как сумму дорог в города, с которыми он связан: Д, И, Ж, Е.

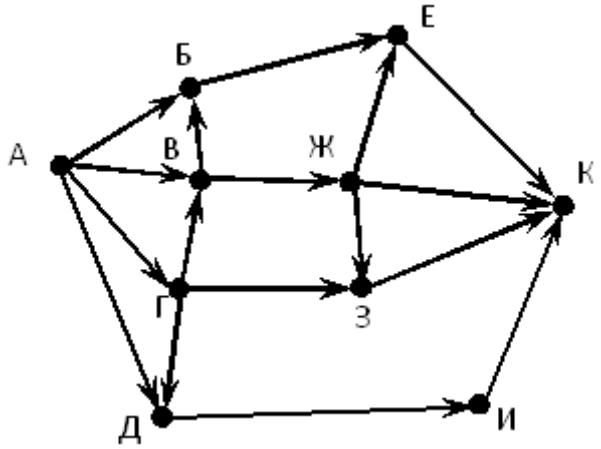


6) Ответ: 13.

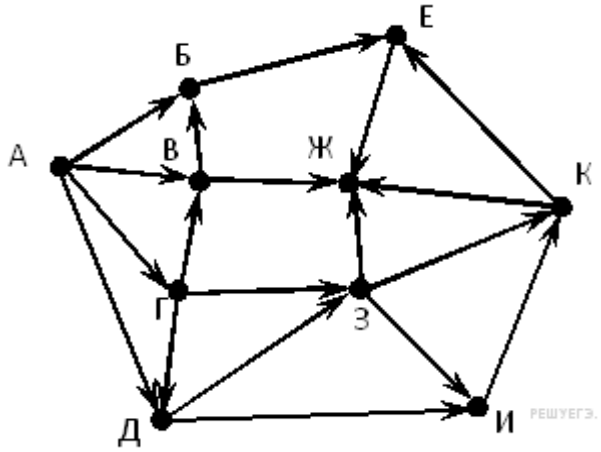
Задания:

Выполнить задание и проверить с помощью программы Grin, Простой граф.

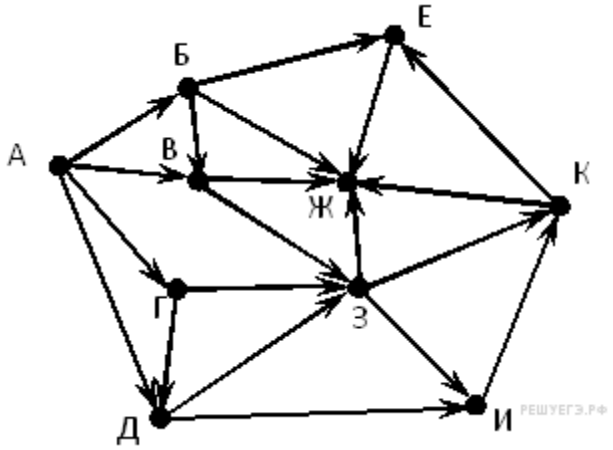
1. На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К?(ответ 12)



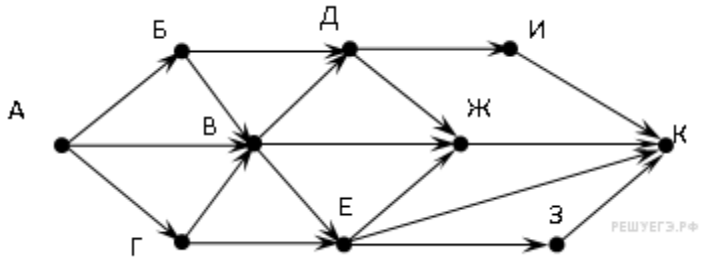
2. На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город Ж? (ответ 24)



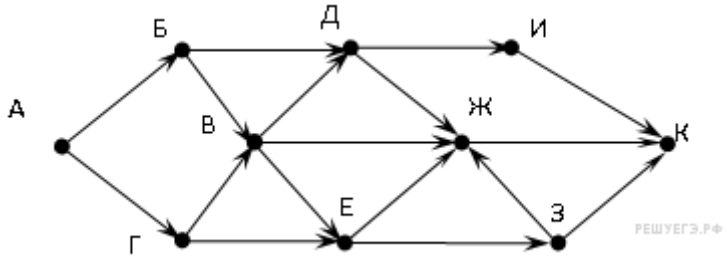
3. На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город Ж? (Ответ 33)



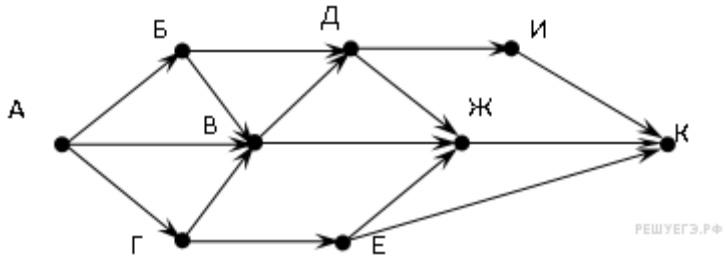
4. На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К? (Ответ: 23)



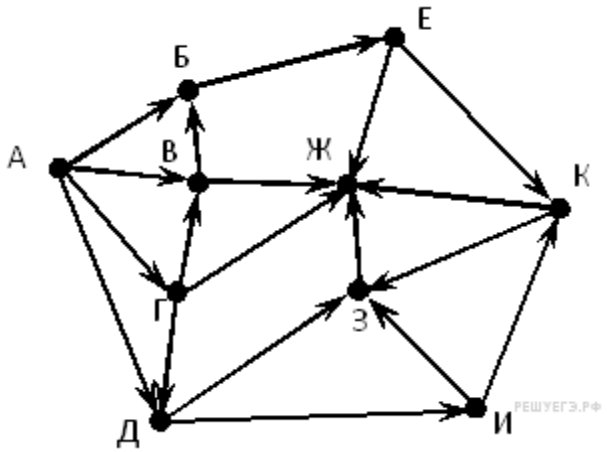
5. На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К? (ответ: 17)



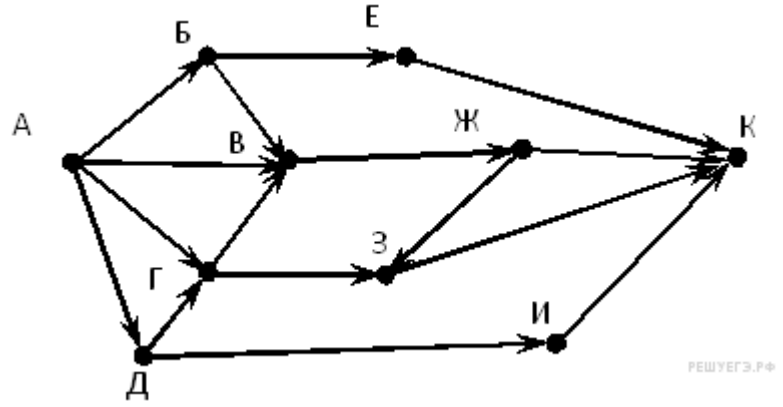
6. На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К? (Ответ: 13)



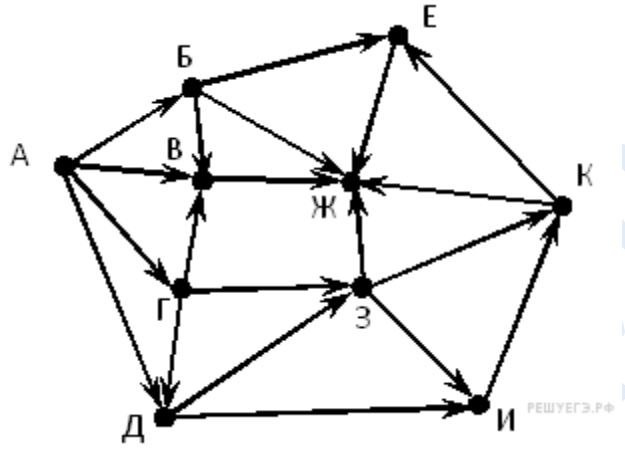
7. На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город Ж? (Ответ: 20)



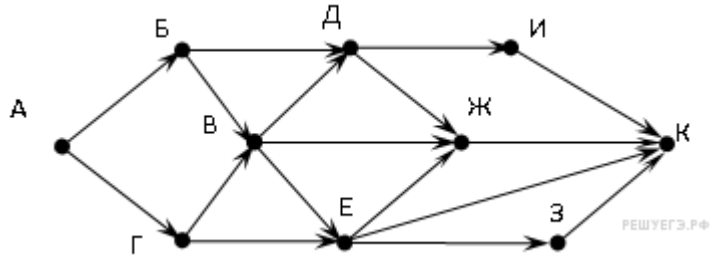
8. На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К? (Ответ: 12)



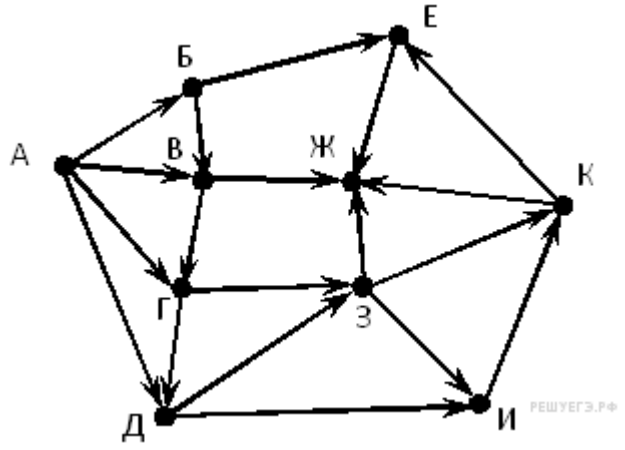
9. На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город Ж? (Ответ:24)



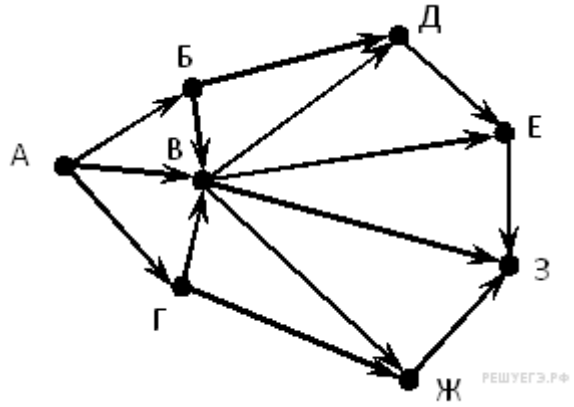
10. На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К? (Ответ: 17)



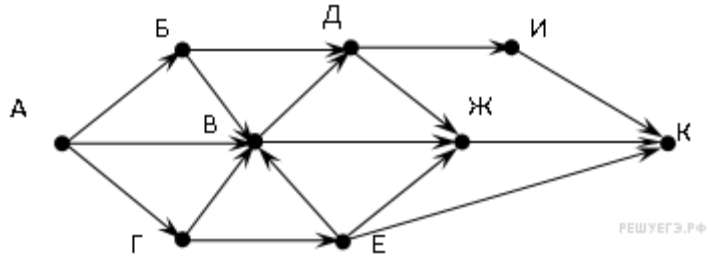
11. На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город Ж? (Ответ: 46)



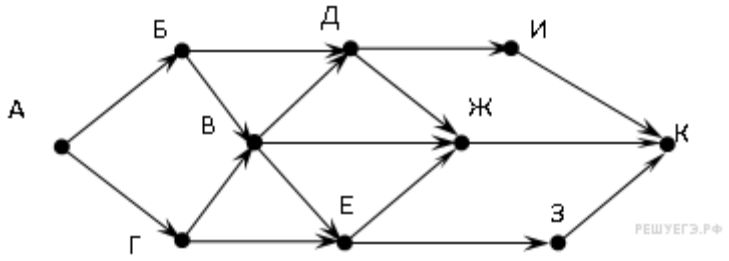
12. На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город З? (Ответ 14)



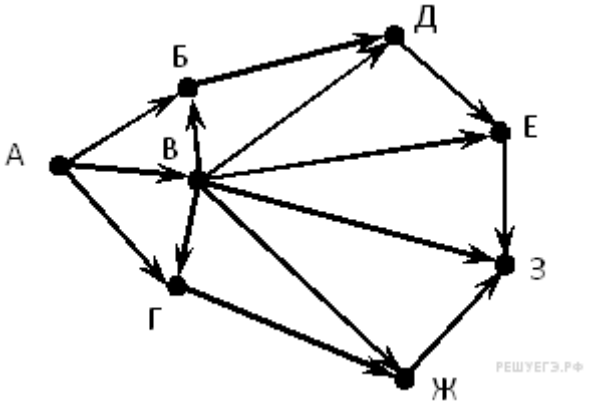
13. На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К? (ответ: 16)



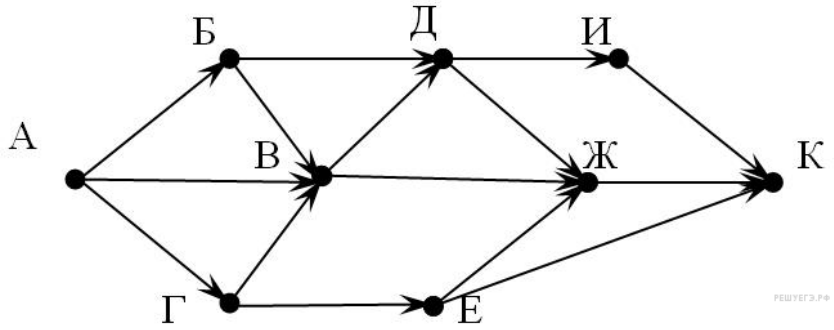
14. На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К? (Ответ: 14)



15. На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город З? (Ответ: 8)

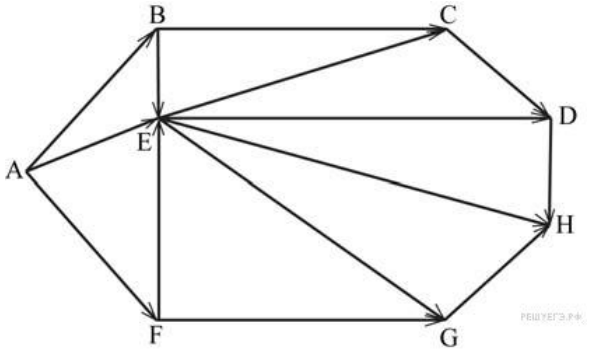


16. На рисунке – схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город Ж? (Ответ:14)

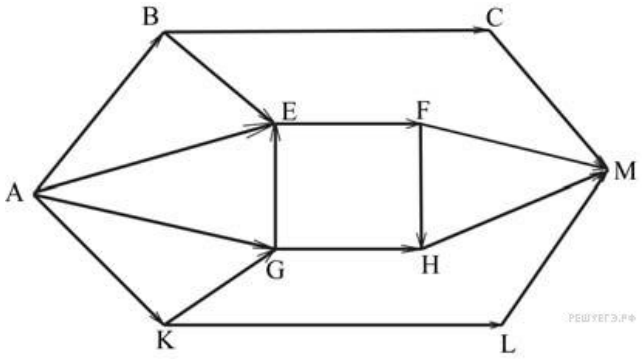


17. На рисунке – схема дорог, связывающих города A, B, C, D, E, F, G H. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город Н? (Ответ: 14)

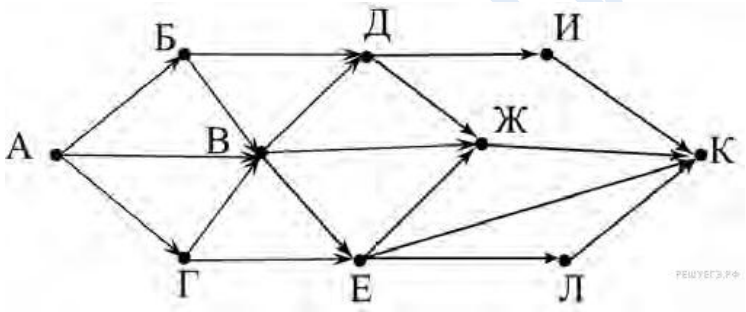
Чащина Е.А. г. Прокопьевск



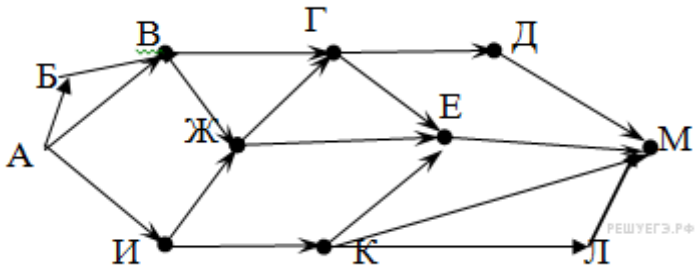
18. На рисунке – схема дорог, связывающих города A, B, C, D, E, F, G, H, K, L, M. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города A в город M? (Ответ:12)



19. На рисунке – схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К, Л. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К? (Ответ:23)

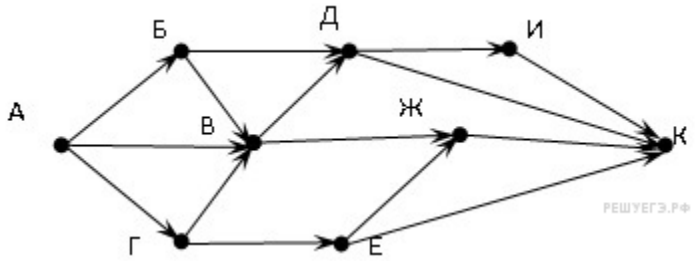


20. На рисунке – схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К, Л, М. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город М?(Ответ:16)

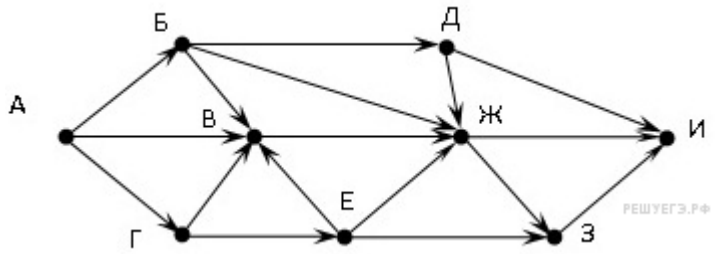


Чащина г. Прокопьвск

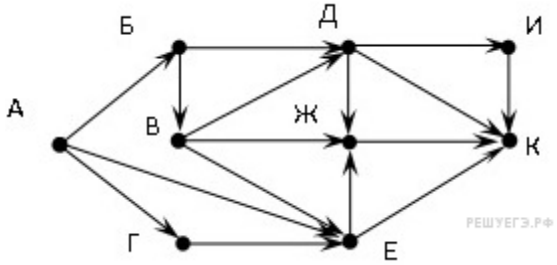
21. На рисунке – схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К? (ответ:13)



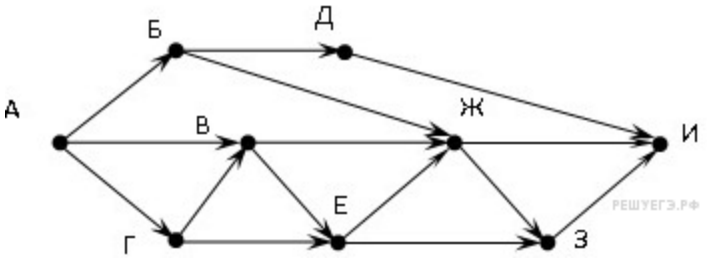
22. На рисунке – схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город И? (ответ:16)



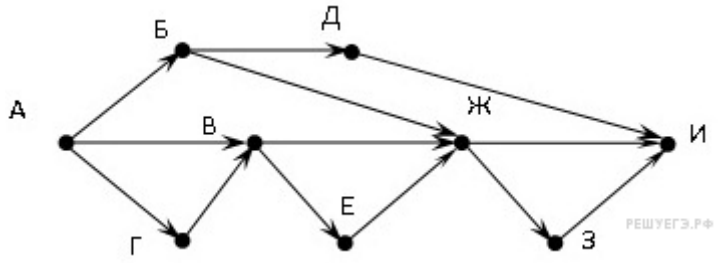
23. На рисунке – схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К? (ответ:13)



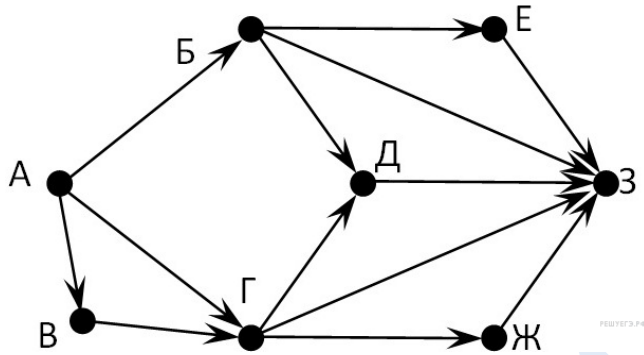
24. На рисунке – схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город И? (ответ:16)



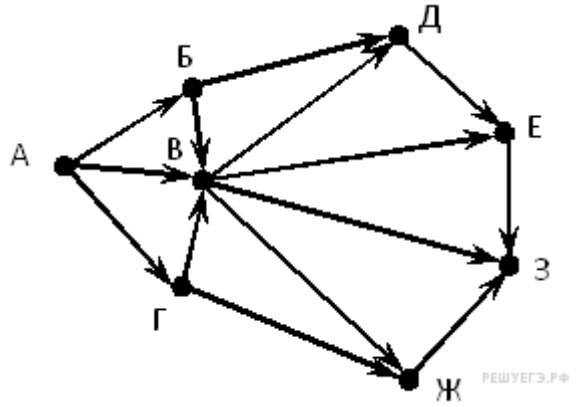
25. На рисунке – схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город И? (Ответ: 11)



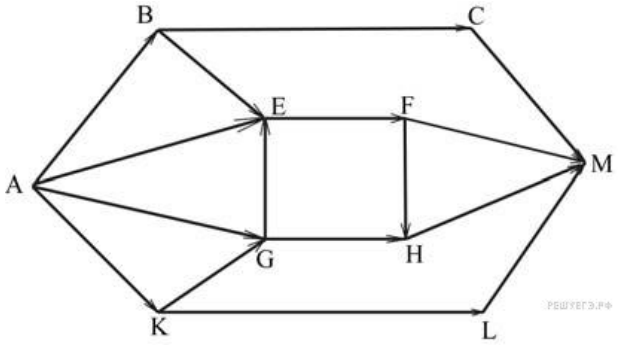
26. На рисунке – схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город З? (Ответ: 9)



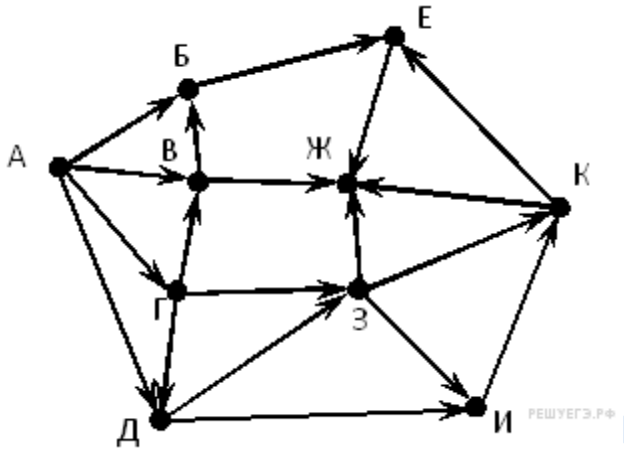
27. На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город Е? (Ответ:7)



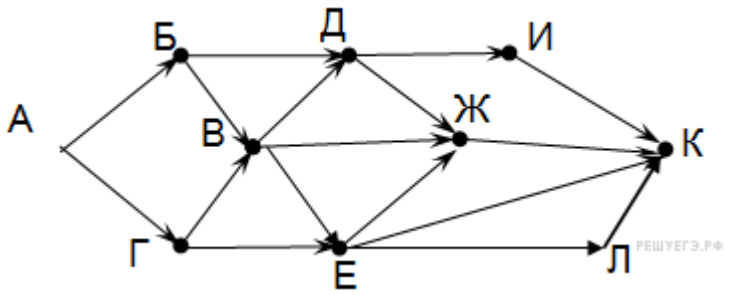
28. На рисунке – схема дорог, связывающих города A, B, C, D, E, F, G, H, K, L, M. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город Е? (ответ:4)



29. На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город З? (ответ:3)



30. На рисунке – схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К, Л. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К? (Ответ: 17)



Порядок выполнения работы:

1. Изучить примеры выполнения заданий.
2. Выполнить задание.
3. Оформить отчет.

Содержание отчета:

1. Тема.
2. Цель.

3. Задачи.
4. Материальное обеспечение.
5. Практическое задание.

Вопросы для самоконтроля:

1. В поиск в графе иначе называется?
2. Что означает DFS и BFS?
3. Дайте определение NP-полная задача?

Чащина Е.А.

Практическая работа № 11

***Тема*:** Решение задач по теории графов. Нахождение минимально доминирующих множеств (МДМ). Нахождение максимально независимых множеств (МНМ)

Цель: реализовать алгоритмы нахождения МДМ и МНМ; закрепить навыки моделирования графов в графической среде Grafoanalizator1.3.3 rus, Простой граф

Задачи:

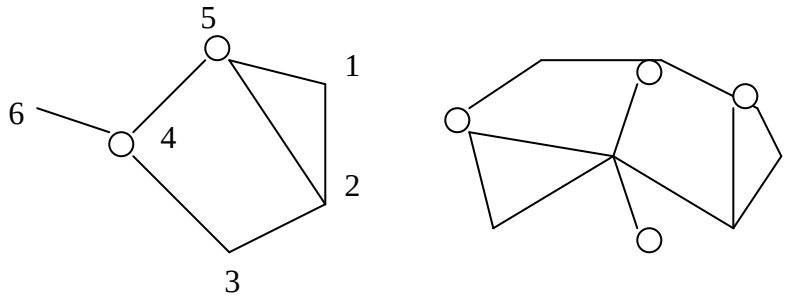
1. Закрепить знания основных понятий теории графов.
2. Приобрести практические умения использования специального программного обеспечения для моделирования.
3. Использовать математический аппарат теории графов.
4. Планирование структуры сети с помощью графа с оптимальным расположением узлов.

Материальное обеспечение:

Программы для графического представления графов: Grafoanalizator1.3.3 rus, Простой граф, практическая работа.

Теоретическая часть:

Подмножество вершин *S* графа *G = (V,U)* называется *доминирующим (внешне устойчивым)*, если каждая вершина из *V — S* смежна с некоторой вершиной из *S*. Другими словами, каждая вершина графа находится на расстоянии не более единицы от доминирующего множества. Доминирующее множество называется *минимальным*, нет другого доминирующего множества, содержащегося в нем. Минимальных доминирующих множеств в графе может быть несколько, и они не обязательно содержат одинаковое количество вершин. Минимальное доминирующее множество, имеющее наименьшую мощность называется *наименьшим*.



○ - вершины доминирующего множества

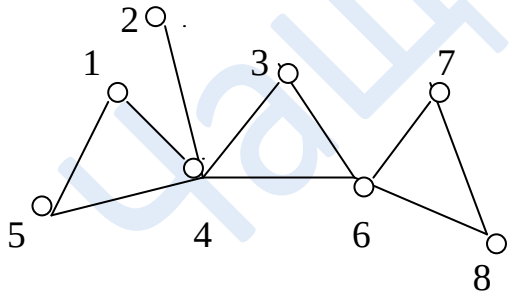
Подмножество вершин графа, являющееся одновременно независимым и доминирующим, называется *ядром графа*.

Понятие доминирующего множества переносится и на случай ориентированных графов. Подмножество *S* вершин орграфа называется доминирующим, если для любой вершины $v_j \in V - S$ существует такая вершина $v_j \in S$, что $(v_i, v_j) \in A$ где A - множество дуг орграфа Подмножество вершин *S*, являющееся одновременно и независимым, и доминирующим называется *ядром орграфа*.

Множество вершин графа называется **независимым**, если никакие две вершины этого множества не соединены ребром. Другими словами, индуцированный этим множеством подграф состоит из изолированных вершин. Иногда также говорят, что каждое ребро графа инцидентно не более чем одной вершине из независимого множества. Задача разрешения выглядит так: существует ли в заданном графе *G* независимое множество размера *k*? Соответствующая ей оптимизационная задача, она же **задача о независимом множестве**, формулируется следующим образом: в заданном графе *G* требуется найти независимое множество максимального размера.

Иногда эту задачу называют поиском *независимого множества максимального размера* или *максимального (по размеру) независимого множества*. Не стоит путать это понятие с *максимальным (по включению) независимым множеством*, которое определяется как такое независимое множество вершин, что при добавлении к нему еще одной (любой) вершины исходного графа оно перестает быть независимым. Понятно, что таких множеств, вообще говоря, может быть несколько и разных размеров. Максимальное по включению независимое множество отнюдь не всегда является максимальным по размеру. В то же время, каждое независимое множество максимального размера по определению является также и максимальным по включению. Для нахождения (какого-то) максимального по включению независимого множества можно воспользоваться жадным алгоритмом, работающим за полиномиальное время, тогда как задача о независимом множестве максимального размера принадлежит к классу NP-полных задач.

Например, для графа G, изображенного на рисунке, $\alpha(G)=4$, множества вершин {1,2,3,7}, {1,2,3,8}, {2,3,5,7}, {2,3,5,8} являются наибольшими независимыми, а {4,7} – максимальное независимое множество, не являющееся наибольшим.



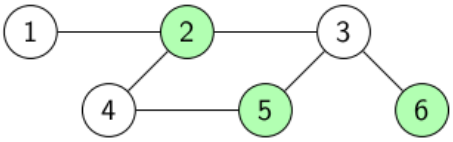
Определение независимого множества

Пусть $G = (V, E)$ — неориентированный граф.

Множество $M \subset V$ называют *независимым множеством* в графе G, если $\forall u, v \in M \quad \{u, v\} \notin E$.

Другими словами, если $M \cap \Gamma(M) = \varnothing$.

Множество всех независимых множеств в графе G

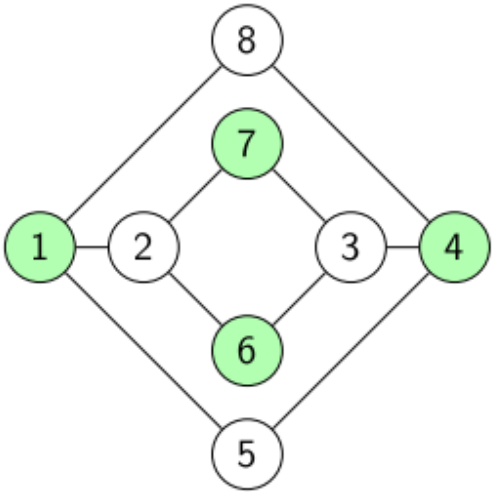


$\{1, 3, 4\}$
$\{2, 5\}$
$\{2, 5, 6\}$
$\ldots\ldots$

Независимые множества можно сравнивать по числу элементов: $\#\{2, 5\} < \#\{1, 3, 4\}$ и по включению: $\{2, 5\} \subset \{2, 5, 6\}$.

Дано: неориентированный граф $G(V, E)$.

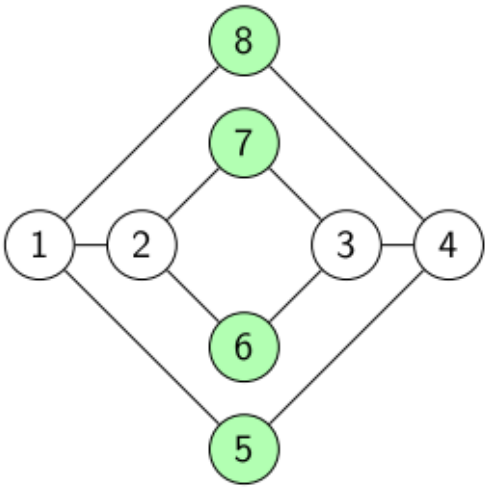
Задача: найти максимальное по числу элементов независимое множество в графе G.



Максимальное независимое множество: $\{1, 4, 6, 7\}$

Дано: неориентированный граф $G(V, E)$.

Задача: найти максимальное по числу элементов независимое множество в графе G.

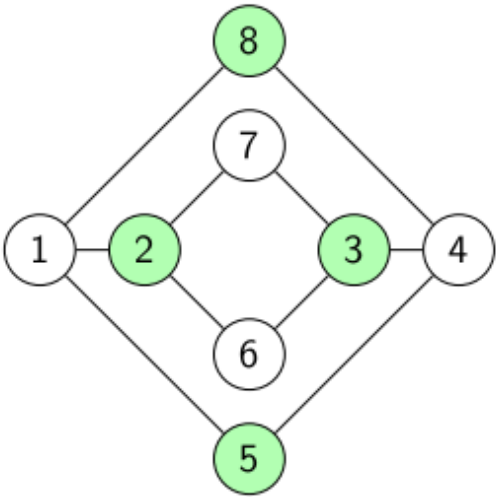


Максимальное независимое множество: $\{1, 4, 6, 7\}$

Есть и другие максимальные независимые множества:
$\{5, 6, 7, 8\}$

Дано: неориентированный граф $G(V, E)$.

Задача: найти максимальное по числу элементов независимое множество в графе G.



Максимальное независимое множество: $\{1, 4, 6, 7\}$

Есть и другие максимальные независимые множества:
$\{5, 6, 7, 8\}$
$\{2, 3, 5, 8\}$

Инструкция по выполнению задания

1. В заданном графе G = (X, V) (рисунок 1) удалить указанные ребра (дуги) и новом графе G' найти все минимальные доминирующие множества (МДМ).

Ребро (дуга), которую необходимо удалить (X4,X5), (X1,X2)

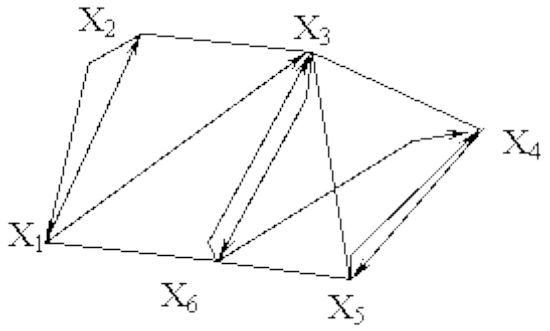
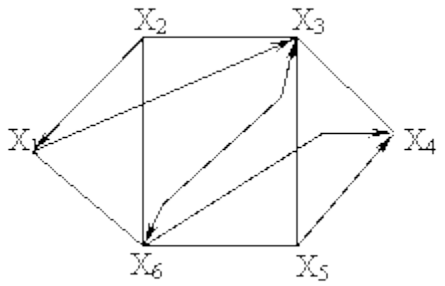


Рисунок 1

Решение



Описываем каждую вершину с помощью дизъюнкции самой вершины и вершин, из которых она достижима. Например для $X_1 : (X_1 \cup X_2 \cup X_3 \cup X_6)$ и составим конъюнкцию таких выражений для всех вершин.

$$(X_1 \cup X_2 \cup X_3 \cup X_6)(X_2 \cup X_3 \cup X_6)(X_3 \cup X_2 \cup X_4 \cup X_5 \cup X_6 \cup X_1) \cap$$
$$\cap (X_4 \cup X_2 \cup X_6 \cup X_5)(X_5 \cup X_3 \cup X_6)(X_6 \cup X_1 \cup X_2 \cup X_3 \cup X_4 \cup X_5) =$$
$$= (X_2 \cup X_3 \cup X_6)(X_5 \cup X_3 \cup X_6 \cup) = X_2 X_5 \cup X_3 \cup X_6$$

Таким образом МДМ1 = {X2, X5}, МДМ2 = {X3}, МДМ3 = {X6}.

2. В заданном графе G = (X, V) (рисунок 2) удалить указанные ребра (дуги) и новом графе G' найти все максимальные независимые множества (МНМ).

Ребро (дуга), которую необходимо удалить (X4,X5), (X1,X2)

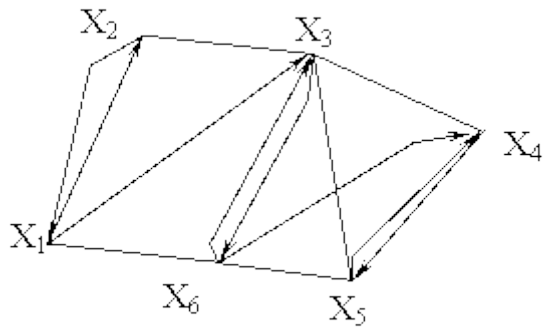
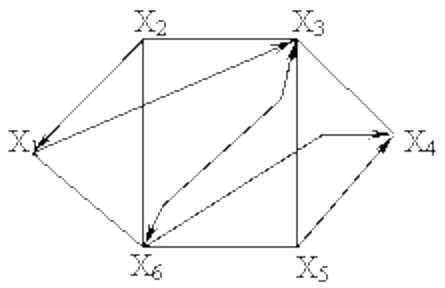


Рисунок 2

Решение



Описываем каждое ребро (дугу) дизъюнкцией отрицаний и производим конъюнкцию всех дизъюнкций. Например ребро (X2, X1) описываем как $\left(\overline{X}_2 \cup \overline{X}_1\right)$ и т.д.
Получим

$$\left(\overline{X}_2 \cup \overline{X}_1\right)\left(\overline{X}_1 \cup \overline{X}_3\right)\left(\overline{X}_1 \cup \overline{X}_6\right)\left(\overline{X}_2 \cup \overline{X}_3\right)\left(\overline{X}_2 \cup \overline{X}_6\right)\cdot$$
$$\cdot\left(\overline{X}_6 \cup \overline{X}_5\right)\left(\overline{X}_6 \cup \overline{X}_3\right)\left(\overline{X}_6 \cup \overline{X}_4\right)\left(\overline{X}_3 \cup \overline{X}_5\right)\left(\overline{X}_4 \cup \overline{X}_5\right)$$

Производим упрощение.

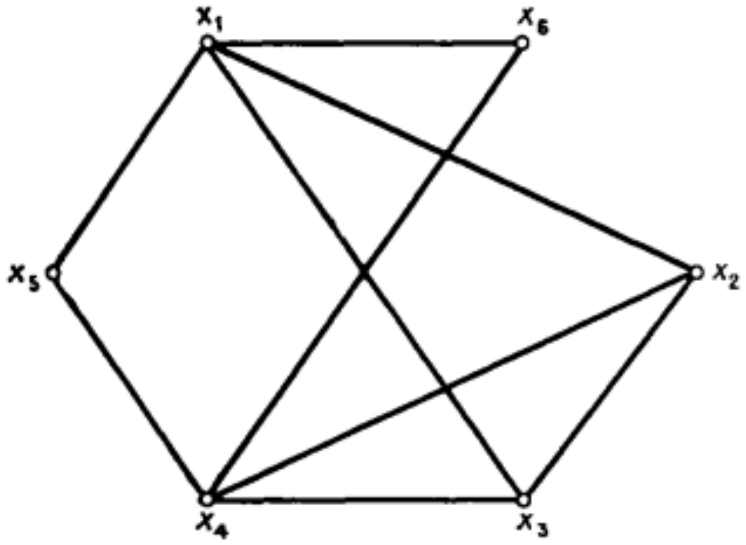
$$\left(\overline{X}_1 \cup \overline{X}_2\overline{X}_3\overline{X}_6\right)\left(\overline{X}_2 \cup \overline{X}_3\overline{X}_6\right)\left(\overline{X}_6 \cup \overline{X}_3\overline{X}_4\overline{X}_5\right)$$
$$\left(\overline{X}_3 \cup \overline{X}_4\overline{X}_5\right)\left(\overline{X}_4 \cup \overline{X}_5\right)=$$
$$=\left(\overline{X}_1\overline{X}_2 \cup \overline{X}_1\overline{X}_3\overline{X}_6 \cup \overline{X}_2\overline{X}_3\overline{X}_6\right)\left(\overline{X}_6 \cup \overline{X}_3\overline{X}_4\overline{X}_5\right)$$
$$\left(\overline{X}_3\overline{X}_4 \cup \overline{X}_3\overline{X}_5 \cup \overline{X}_4\overline{X}_5 \cup \overline{X}_4\overline{X}_5\right)=$$
$$=\left(\overline{X}_1\overline{X}_2 \cup \overline{X}_1\overline{X}_3\overline{X}_6 \cup \overline{X}_2\overline{X}_3\overline{X}_6\right)\cdot$$
$$\cdot\left(\overline{X}_6\overline{X}_3\overline{X}_4 \cup \overline{X}_6\overline{X}_3\overline{X}_5 \cup \overline{X}_6\overline{X}_4\overline{X}_5 \cup \overline{X}_3\overline{X}_4\overline{X}_5 \cup\right.$$
$$\left.\cup\, \overline{X}_3\overline{X}_4\overline{X}_5 \cup \overline{X}_3\overline{X}_4\overline{X}_5\right)=$$
$$=\left(\overline{X}_1\overline{X}_2 \cup \overline{X}_1\overline{X}_3\overline{X}_6 \cup \overline{X}_2\overline{X}_3\overline{X}_6\right)\cap$$
$$\cap\left(\overline{X}_6\overline{X}_3\overline{X}_4 \cup \overline{X}_6\overline{X}_3\overline{X}_5 \cup \overline{X}_6\overline{X}_4\overline{X}_5 \cup \overline{X}_3\overline{X}_4\overline{X}_5\right)=$$
$$=\overline{X}_1\overline{X}_2\overline{X}_3\overline{X}_4\overline{X}_6 \cup \overline{X}_1\overline{X}_2\overline{X}_3\overline{X}_5\overline{X}_6 \cup \overline{X}_1\overline{X}_2\overline{X}_4\overline{X}_5\overline{X}_6 \cup$$
$$\cup\, \overline{X}_1\overline{X}_2\overline{X}_3\overline{X}_4\overline{X}_5 \cup \overline{X}_1\overline{X}_3\overline{X}_4\overline{X}_6 \cup \overline{X}_1\overline{X}_3\overline{X}_5\overline{X}_6 \cup$$
$$\cup\, \overline{X}_1\overline{X}_3\overline{X}_4\overline{X}_5\overline{X}_6 \cup \overline{X}_1\overline{X}_3\overline{X}_4\overline{X}_5\overline{X}_6 \cup \overline{X}_2\overline{X}_3\overline{X}_4\overline{X}_6 \cup$$
$$\cup\, \overline{X}_2\overline{X}_3\overline{X}_5\overline{X}_6 \cup \overline{X}_2\overline{X}_3\overline{X}_5\overline{X}_6 \cup \overline{X}_2\overline{X}_3\overline{X}_4\overline{X}_5\overline{X}_6 \cup$$
$$\cup\, \overline{X}_2\overline{X}_3\overline{X}_4\overline{X}_5\overline{X}_6 = \overline{X}_1\overline{X}_2\overline{X}_4\overline{X}_5\overline{X}_6 \cup \overline{X}_1\overline{X}_2\overline{X}_3\overline{X}_4\overline{X}_5 \cup$$
$$\cup\, \overline{X}_2\overline{X}_3\overline{X}_4\overline{X}_6 \cup \overline{X}_2\overline{X}_3\overline{X}_5\overline{X}_6.$$

Откуда из первой группы следует:
МНМ1 = {X3}, МНМ2 = {X6}, МНМ3 = {X1, X5}, МНМ4 = { X1, X4}.

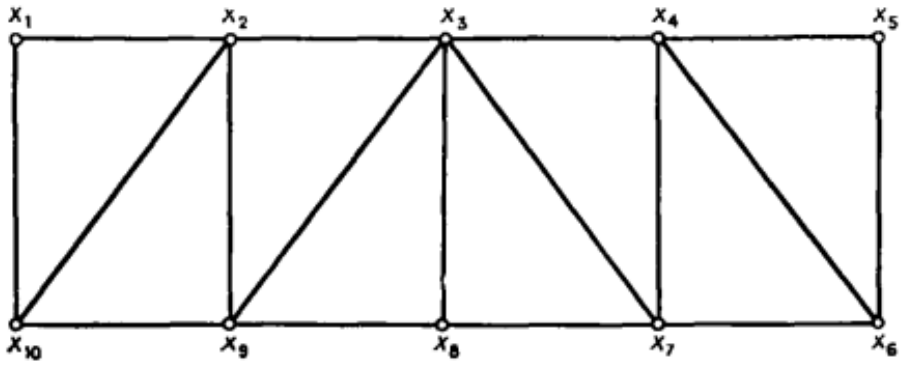
Задания:

1. Перечислите все максимальные независимые множества графа G, показанного на рисунке, а также найти число независимости.

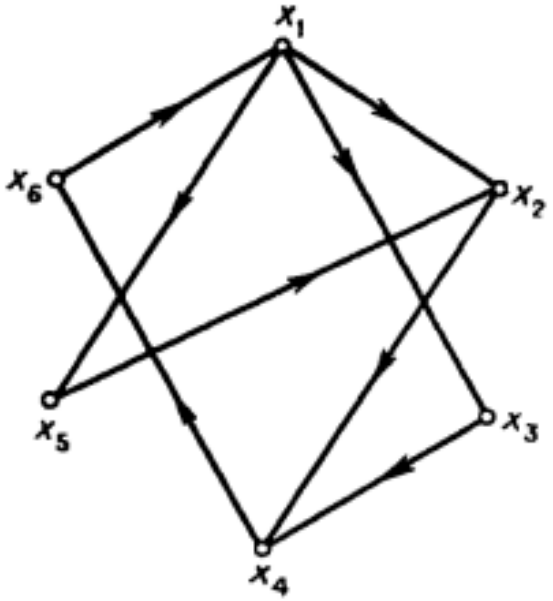
Чащина Е.А.



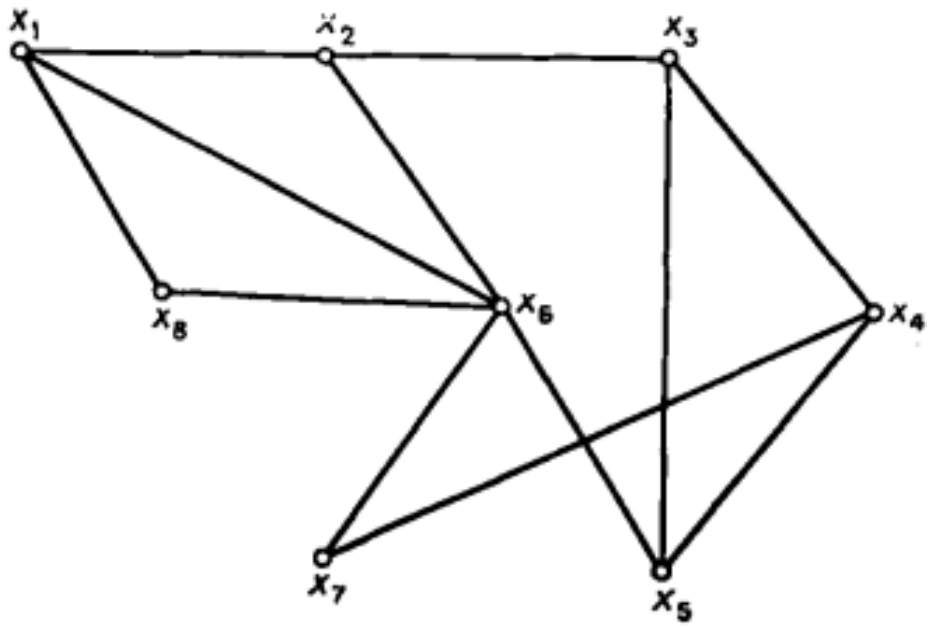
2. Составить список всех максимальных независимых множеств графа приведенного на рисунке. (Обратить внимание на симметрию)



3. Найти доминирующие множества по рисунку.



4. Найти максимально независимые множества по рисунку.



Порядок выполнения работы:

1. Изучить инструкцию по выполнению задания
2. Выполнить задание.
3. Оформить отчет.

Содержание отчета:

1. Тема.
2. Цель.
3. Задачи.
4. Материальное обеспечение.
5. Практическое задание.

Вопросы для самоконтроля:

1. Какое множество вершин графа называется независимым?
2. Дайте определение доминирующего множества?
3. Дайте определение понятию ядро графа?

Практическая работа № 12

Тема: Решение задач по теории графов. Нахождение кратчайшего пути

Цель: научиться находить кратчайшие пути в графе при помощи алгоритма Дейкстры; получение навыком при решении задач нахождения пути в графе; закрепить навыки моделирования графов в графической среде Grafoanalizator1.3.3 rus, Простой граф, Grin.

Задачи:

1. Закрепить знания основных понятий теории графов.
2. Приобрести практические умения использования специального программного обеспечения для моделирования.
3. Использовать математический аппарат теории графов.
4. Применение алгоритмов поиска кратчайшего пути.
5. Планирование структуры сети с помощью графа с оптимальным расположением узлов.

Материальное обеспечение:

Программы для графического представления графов: Grafoanalizator1.3.3 rus, Простой граф, Grin, практическая работа.

Теоретическая часть:

Нахождение *кратчайшего пути* на сегодняшний день является жизненно необходимой задачей и используется практически везде, начиная от нахождения оптимального маршрута между двумя объектами на местности (например, *кратчайший путь* от дома до университета), в системах автопилота, для нахождения оптимального маршрута при перевозках, *коммутации* информационного пакета в сетях и т.п.

Кратчайший путь рассматривается при помощи некоторого математического объекта, называемого графом. *Поиск кратчайшего пути* ведется между двумя заданными вершинами в графе. Результатом является *путь*, то есть последовательность вершин и ребер, *инцидентных* двум соседним вершинам, и его *длина*.

Описываемый в данном разделе алгоритм позволяет находить в графе кратчайший путь между двумя выделенными вершинами s и t при положительных длинах дуг. Этот алгоритм предложенный в 1959 г. Дейкстрой, считается одним из наиболее эффективных алгоритмов решения задачи.

Главная идея, лежащая в основе алгоритма Дейкстры, предельно проста. Предположим, что нам известны m вершин, ближайших к вершине s (близость любой вершины x к вершине s определяется длиной кратчайшего пути, ведущего из s в x). Пусть также известны сами кратчайшие пути, соединяющие вершину s с выделенными m вершинами). Покажем теперь, как может быть определена $(m+1)$-я ближайшая к s вершина.

Окрасим вершину s и m ближайших к ней вершин. Построим для каждой неокрашенной вершины y пути, непосредственно соединяющие с помощью дуг (x, y) каждую окрашенную вершину x с y. Выберем из этих путей кратчайший, и будем считать его условно кратчайшим путем из вершины s в вершину y.

Какая же из неокрашенных вершин является $(m+1)$-й ближайшей к s вершиной? Та, для которой условно кратчайший путь имеет наименьшую длину. Это обусловливается тем, что кратчайший путь из вершины s в $(m+1)$-ю ближайшую вершину при положительном значении длин всех дуг должен содержать в качестве промежуточных лишь окрашенные вершины, т. е. вершины, входящие в число m вершин, ближайших к вершине s.

Итак, если известны m ближайших к s вершин, то $(m+1)$-я ближайшая к s вершина может быть найдена так, как это описано выше. Начиная с $m = 0$, описанная процедура может повторяться до тех пор, пока не будет получен кратчайший путь, ведущий из вершины s к вершине t.

Алгоритм работает только для графов без рёбер отрицательного веса.

Любая задача,требующая нахождения оптимальных маршрутов может быть выполнена с помощью алгоритма Дейкстры. Это касается и сетей, и транспортных потоков, и обработка графов. Очень часто используется не сам алгоритм в чистом виде, а его модификация.

Примеры:

1. При эвакуации населения из очагов бедствия оптимальные маршруты до пунктов сбора транспорта для каждой группы людей(дом,улица,школа и т.д) в штабе МЧС рассчитывает программа на основе алгоритма Дейкстры.
2. Компьютерная игра. Указываете точку назначения для персонажа и он движется туда по кратчайшему маршруту. Это тоже алгоритм Дейкстры.
3. Дана сеть автомобильных дорог, соединяющих города Гомельской области. Некоторые дороги односторонние. Найти кратчайшие пути от города Гомеля до каждого города области (если двигаться можно только по дорогам).
4. Имеется некоторое количество авиарейсов между городами мира, для каждого известна стоимость. Стоимость перелёта из А в В может быть не равна стоимости перелёта из В в А. Найти маршрут минимальной стоимости (возможно, с пересадками) от Милана до Минска.

Алгоритм широко применяется в программировании и технологиях, например, его использует протокол OSPF для устранения кольцевых маршрутов.

OSPF разбивает процесс построения таблицы маршрутизации на 2 этапа.

Второй этап состоит в нахождении оптимальных маршрутов с помощью полученного графа. Задача нахождения оптимального пути на графе является достаточно сложной и ёмкой. В протоколе OSPF для её решения используется итеративный алгоритм Дейкстры. Каждый маршрутизатор считает себя центром сети и ищет оптимальный маршрут до каждой известной ему сети. В каждом найденном таким образом маршруте запоминается только один шаг - до следующего маршрутизатора, в соответствии с принципом одношаговой маршрутизации. Данные об этом шаге и попадают в таблицу маршрутизации. Если несколько маршрутов имеют одинаковую метрику до сети назначения, то в таблице маршрутизации запоминаются первые шаги всех этих маршрутов".

Алгоритм

1. Каждой вершине x в ходе алгоритма присваивается число $d(x)$, равное длине кратчайшего пути из вершины s в вершину x и включающем только окрашенные вершины. Положить $d(s)=0$ и $d(x)=\infty$ для всех остальных вершин графа. Окрашиваем вершину s и полагаем $y=s$, где y – последняя окрашенная вершина.

2. Для каждой неокрашенной вершины x пересчитывается величина $d(x)$ по следующей формуле:

$$d(x)=\min\{d(x);d(y)+a(y,x)\} \quad (1)$$

3. Если $d(x)=\infty$ для всех неокрашенных вершин, то алгоритм заканчивается т. к. отсутствуют пути из вершины s в неокрашенные вершины. Иначе окрашивается та вершина, для которой величина $d(x)$ является минимальной. Окрашивается и дуга, ведущая в эту вершину в соответствии с выражением (1) и полагаем $y=x$.

4. Если $y=t$, кратчайший путь из s в t найден. Иначе переходим к шагу 2.

Каждый раз окрашивается вершина и дуга, заходящая в эту вершину. Окрашенные дуги не могут образовывать цикл, а образуют в исходном графе дерево с корнем (началом) в вершине s. Это дерево называют ориентированным деревом кратчайших путей. Путь из s в t принадлежит этому дереву. При поиске одного кратчайшего пути процедура наращивания завершается при достижении конечной вершины этого пути. Нам же необходимо получить все кратчайшие пути начинающиеся в вершине №1. Для этого процедуру наращивания ориентированного дерева продолжается до тех пор, пока все вершины не будут включены. Таким образом, мы получаем ориентированное дерево кратчайших путей, которое является покрывающим деревом графа.

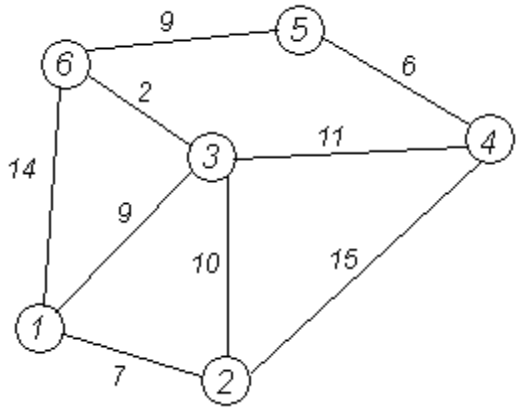
Иногда в графе имеются несколько кратчайших путей. Кратчайший путь будет единственным, если в алгоритме ни разу не возникает неоднозначность при окрашивании дуги.

Отметим, что главным условием успешного применения алгоритма Дейкстры к задаче на графе является неотрицательность длин дуг этого графа

Для графа с отрицательными весами применяется более общий алгоритм — Алгоритм Дейкстры с потенциалами.

Пример выполнения задания.

Требуется найти кратчайшие расстояния от 1-й вершины до всех остальных для графа, представленного на рисунке:



<table>
<tr><th>№</th><th>Алгоритм</th><th>Конкретные действия</th></tr>
<tr><td>1.</td><td>Инициализация.</td><td>Стартовая вершина, от которой строится дерево кратчайших путей - вершина № 1.
Задаем стартовые условия: d(1)=0, d(x)=∞.
Окрашиваем вершину № 1, y=1.
Находим ближайшую вершину к окрашенной нами, используя формулу: $d(x)=\min\{d(x);d(y)+a(y,x)\}$.
Составим матрицу длин кратчайших дуг для данного графа.
<table>
<tr><th>№/№</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th></tr>
<tr><td>1</td><td>∞</td><td>7</td><td>9</td><td>∞</td><td>∞</td><td>14</td></tr>
<tr><td>2</td><td>7</td><td>∞</td><td>10</td><td>15</td><td>∞</td><td>∞</td></tr>
<tr><td>3</td><td>9</td><td>10</td><td>∞</td><td>11</td><td>∞</td><td>2</td></tr>
<tr><td>4</td><td>∞</td><td>15</td><td>11</td><td>∞</td><td>6</td><td>∞</td></tr>
<tr><td>5</td><td>∞</td><td>∞</td><td>∞</td><td>6</td><td>∞</td><td>9</td></tr>
<tr><td>6</td><td>14</td><td>∞</td><td>2</td><td>∞</td><td>9</td><td>∞</td></tr>
</table>
или</td></tr>
</table>

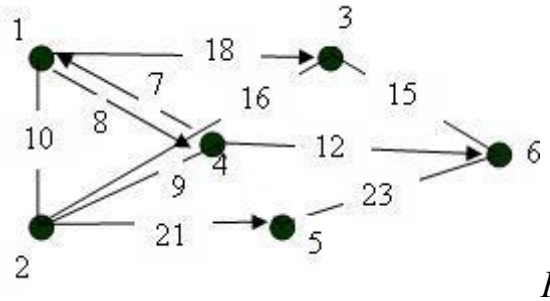
		$\begin{pmatrix} \infty & 7 & 9 & \infty & \infty & 14 \\ 7 & \infty & 10 & 15 & \infty & \infty \\ 9 & 10 & \infty & 11 & \infty & 2 \\ \infty & 15 & 11 & \infty & 6 & \infty \\ \infty & \infty & \infty & 6 & \infty & 9 \\ 14 & \infty & 2 & \infty & 9 & \infty \end{pmatrix}$
2.	Первый шаг. Рассмотрим шаг алгоритма Дейкстры.	d(2)=min{d(2) ; d(1)+a(1,2)}=min{∞; 0+7}=7 d(3)=min{d(3) ; d(1)+a(1,3)}=min{∞; 0+9}=9 d(4)=min{d(4) ; d(1)+a(1,4)}=min{∞; 0+∞}=∞ d(5)=min{d(5) ; d(1)+a(1,5)}=min{∞; 0+∞}=∞ d(6)=min{d(6) ; d(1)+a(1,6)}=min{∞; 0+14}=14 Минимальную длину имеет путь от вершины № 1 до вершины № 2 d(2)=7. Включаем вершину № 2 в текущее ориентированное дерево, а так же дугу ведущую в эту вершину. Согласно выражению это дуга (1,2).
3.	Второй шаг.	d(3)=min{d(3) ; d(2)+a(2,3)}=min{9; 7+10}=9 d(4)=min{d(4) ; d(2)+a(2,4)}=min{∞; 7+15}=22 d(5)=min{d(5) ; d(2)+a(2,5)}=min{∞; 7+∞}=∞ d(6)=min{d(6) ; d(2)+a(2,6)}=min{14; 7+∞}=14 Минимальную длину имеет путь от вершины № 1 до вершины № 3 d(3)=9. Включаем вершину № 3 в текущее ориентированное дерево, а так же дугу ведущую в эту вершину. Согласно выражению это дуга (1,3)
4.	Третий шаг.	d(4)=min{d(4) ; d(3)+a(3,4)}=min{22; 9+11}=20 d(5)=min{d(5) ; d(3)+a(3,5)}=min{∞; 9+∞}=∞ d(6)=min{d(6) ; d(3)+a(3,6)}=min{14; 9+2}=11 Минимальную длину имеет путь от вершины № 1 до вершины № 6 d(6)=11. Включаем вершину № 6 в текущее ориентированное дерево, а так же дугу ведущую в эту вершину. Согласно выражению это

Чащина Е.А.

		дуга (1,6)=(1,3)+(3,6)
5.	Четвертый шаг.	d(4)=min{d(4) ; d(6)+a(6,4)}=min{20; 11+∞}=20 d(5)=min{d(5) ; d(6)+a(6,5)}=min{∞; 11+9}=20 Минимальную длину имеет путь от вершины № 1 до вершины № 4 и № 5 d(4)=d(6)=20. Включаем вершину № 4 и № 5 в текущее ориентированное дерево, а так же дугу ведущую в эту вершину. Согласно выражению это дуга (1,4)=(1,3)+(3,4) и (1,5)=(1,3)+(3,6)+(6,5).
6.	Заключение.	Мы получили ориентированное дерево кратчайших путей начинающихся в вершине №1 для исходного графа. d(1)=1 Длина маршрута L=0 d(2)=1-2 Длина маршрута L=7 d(3)=1-3 Длина маршрута L=9 d(4)=1-3-4 Длина маршрута L=20 d(5)=1-3-6-5 Длина маршрута L=20 d(6)=1-3-6 Длина маршрута L=11

Пример выполнения задание № 2

Требуется найти кратчайшие расстояния от 1-й вершины до всех остальных для графа, представленного на рисунке:



Решение:

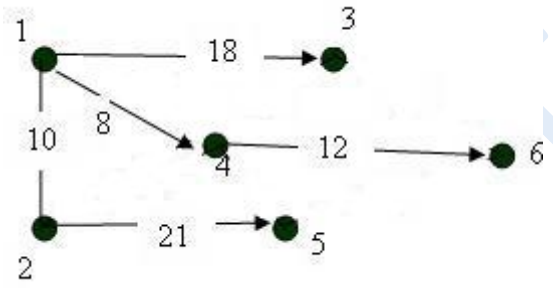
№	Алгоритм	Конкретные действия

<table>
<tr>
<td>1.</td>
<td>Инициализация.</td>
<td>
Стартовая вершина, от которой строится дерево кратчайших путей - вершина № 1.
Задаем стартовые условия: d(1)=0, d(x)=∞.
Окрашиваем вершину № 1, y=1.
Находим ближайшую вершину к окрашенной нами, используя формулу: $d(x)=\min\{d(x);d(y)+a(y,x)\}$.
Составим матрицу длин кратчайших дуг для данного графа.
<table>
<tr><td>№/№</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr>
<tr><td>1</td><td>∞</td><td>10</td><td>18</td><td>8</td><td>∞</td><td>∞</td></tr>
<tr><td>2</td><td>10</td><td>∞</td><td>16</td><td>9</td><td>21</td><td>∞</td></tr>
<tr><td>3</td><td>∞</td><td>16</td><td>∞</td><td>∞</td><td>∞</td><td>15</td></tr>
<tr><td>4</td><td>7</td><td>9</td><td>∞</td><td>∞</td><td>∞</td><td>12</td></tr>
<tr><td>5</td><td>∞</td><td>∞</td><td>∞</td><td>∞</td><td>∞</td><td>23</td></tr>
<tr><td>6</td><td>∞</td><td>∞</td><td>15</td><td>∞</td><td>23</td><td>∞</td></tr>
</table>
или $\begin{pmatrix} \infty & 10 & 18 & 8 & \infty & \infty \\ 10 & \infty & 16 & 9 & 21 & \infty \\ \infty & 16 & \infty & \infty & \infty & 15 \\ 7 & 9 & \infty & \infty & \infty & 12 \\ \infty & \infty & \infty & \infty & \infty & 23 \\ \infty & \infty & 15 & \infty & 23 & \infty \end{pmatrix}$
</td>
</tr>
<tr>
<td>2.</td>
<td>Первый шаг.
Рассмотрим шаг алгоритма Дейкстры.</td>
<td>
d(2)=min{d(2) ; d(1)+a(1,2)}=min{∞; 0+10}=10
d(3)=min{d(3) ; d(1)+a(1,3)}=min{∞; 0+18}=18
d(4)=min{d(4) ; d(1)+a(1,4)}=min{∞; 0+8}=8
d(5)=min{d(5) ; d(1)+a(1,5)}=min{∞; 0+∞}=∞
d(6)=min{d(6) ; d(1)+a(1,6)}=min{∞; 0+∞}=∞
Минимальную длину имеет путь от вершины № 1 до вершины № 4 d(4)=8.
Включаем вершину № 4 в текущее ориентированное дерево, а так же дугу ведущую в эту вершину.
Согласно выражению это дуга (1,4)
</td>
</tr>
</table>

Чащина Е.А.

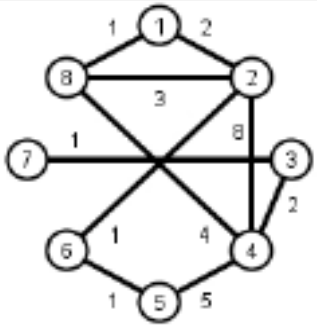
3.	Второй шаг.	d(2)=min{d(2) ; d(4)+a(4,2)}=min{10; 8+9}=10 d(3)=min{d(3) ; d(4)+a(4,3)}=min{18; 8+∞}=18 d(5)=min{d(5) ; d(4)+a(4,5)}=min{∞; 8+∞}=∞ d(6)=min{d(6) ; d(4)+a(4,6)}=min{∞; 8+12}=20 Минимальную длину имеет путь от вершины № 1 до вершины № 2 d(2)=10. Включаем вершину № 2 в текущее ориентированное дерево, а так же дугу ведущую в эту вершину. Согласно выражению это дуга (1,2)
4.	Третий шаг.	d(3)=min{d(3) ; d(2)+a(2,3)}=min{18; 10+16}=18 d(5)=min{d(5) ; d(2)+a(2,5)}=min{∞; 10+21}=31 d(6)=min{d(6) ; d(2)+a(2,6)}=min{20; 10+∞}=20 Минимальную длину имеет путь от вершины № 1 до вершины № 3 d(3)=18. Включаем вершину № 3 в текущее ориентированное дерево, а так же дугу ведущую в эту вершину. Согласно выражению это дуга (1,3)
5.	Четвертый шаг.	d(5)=min{d(5) ; d(3)+a(3,5)}=min{31; 18+∞}=31 d(6)=min{d(6) ; d(3)+a(3,6)}=min{20; 18+15}=20 Минимальную длину имеет путь от вершины № 1 до вершины № 6 d(6)=20. Включаем вершину № 6 в текущее ориентированное дерево, а так же дугу ведущую в эту вершину. Согласно выражению это дуга (4,6)
6.	Пятый шаг.	d(5)=min{d(5) ; d(6)+a(6,5)}=min{31; 20+23}=31 Минимальную длину имеет путь от вершины № 1 до вершины № 5 d(5)=31. Включаем вершину № 5 в текущее ориентированное дерево, а так же дугу ведущую в эту вершину. Согласно выражению это дуга (2,5)

Чащина Е.А.

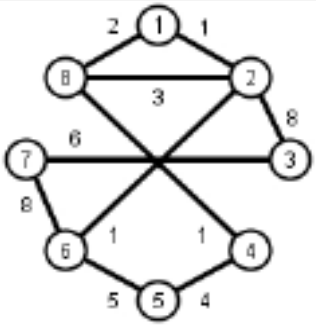
7.	Заключение.	Мы получили ориентированное дерево кратчайших путей начинающихся в вершине №1 для исходного графа. d(1)=1 Длина маршрута L=0 d(2)=1-2 Длина маршрута L=10 d(3)=1-3 Длина маршрута L=18 d(4)=1-4 Длина маршрута L=8 d(5)=1-2-5 Длина маршрута L=31 d(6)=1-4-6 Длина маршрута L=20 Ориентированное дерево с корнем в вершине №1: 

Задание.

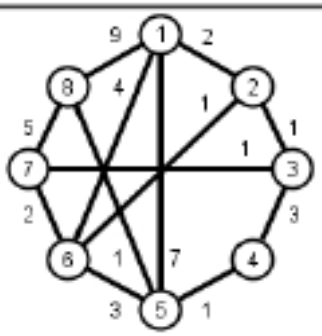
1. Найти кратчайшие расстояния от 1-й вершины до всех остальных для графа, представленного на рисунке. Используя программное обеспечение, проверьте другие алгоритмы вычисления кратчайшего пути.



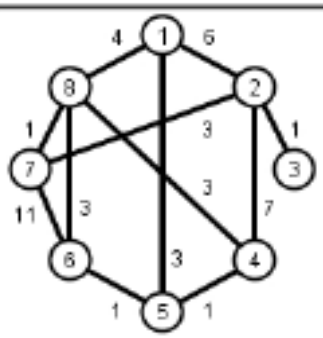
11.22



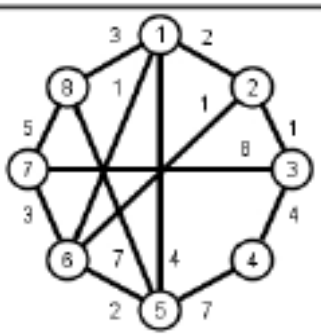
11.23



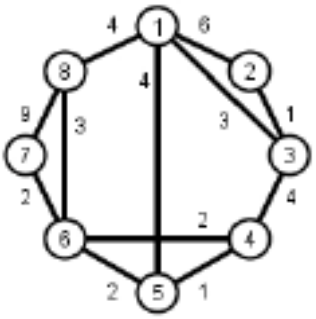
11.24



11.25

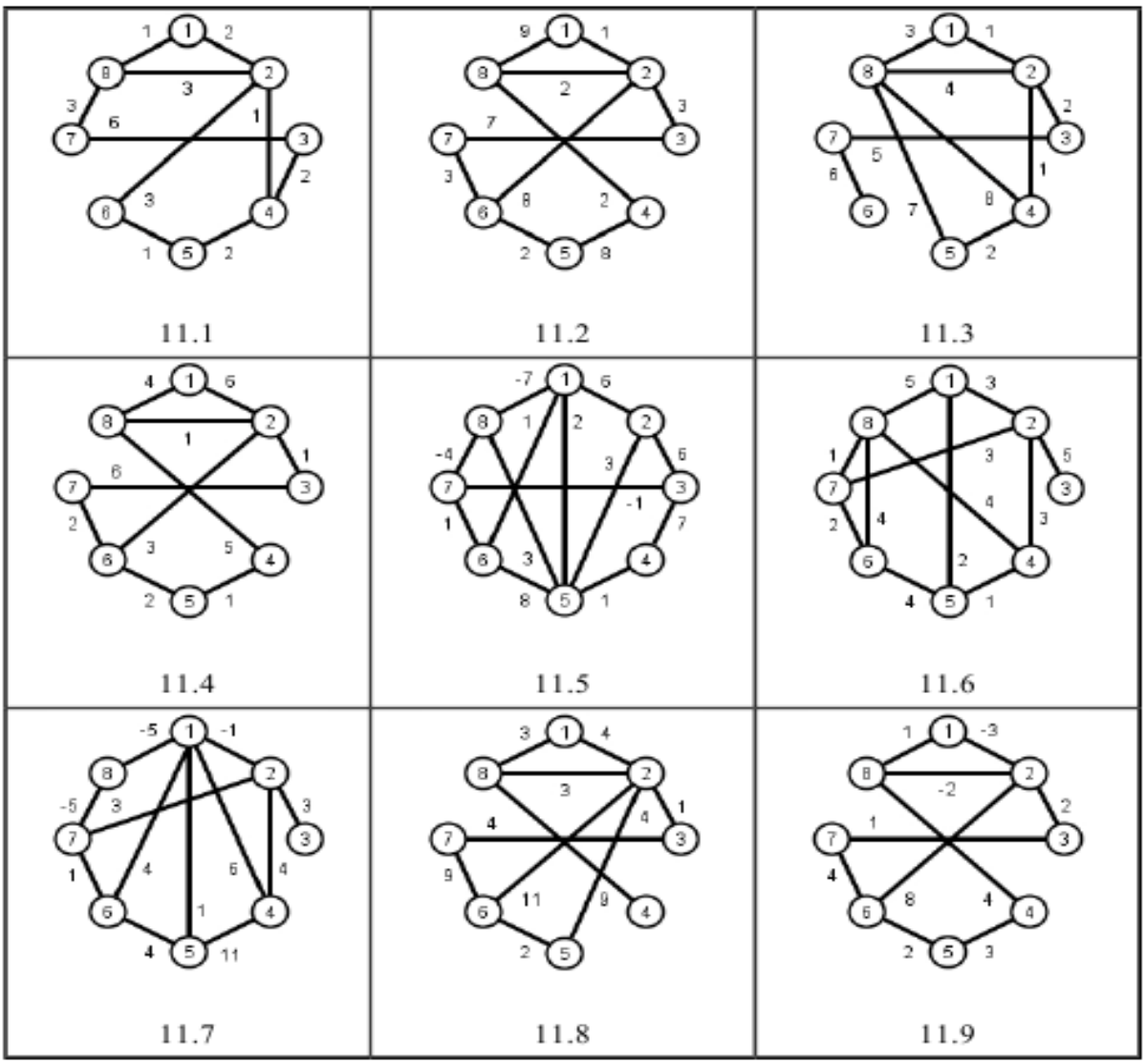


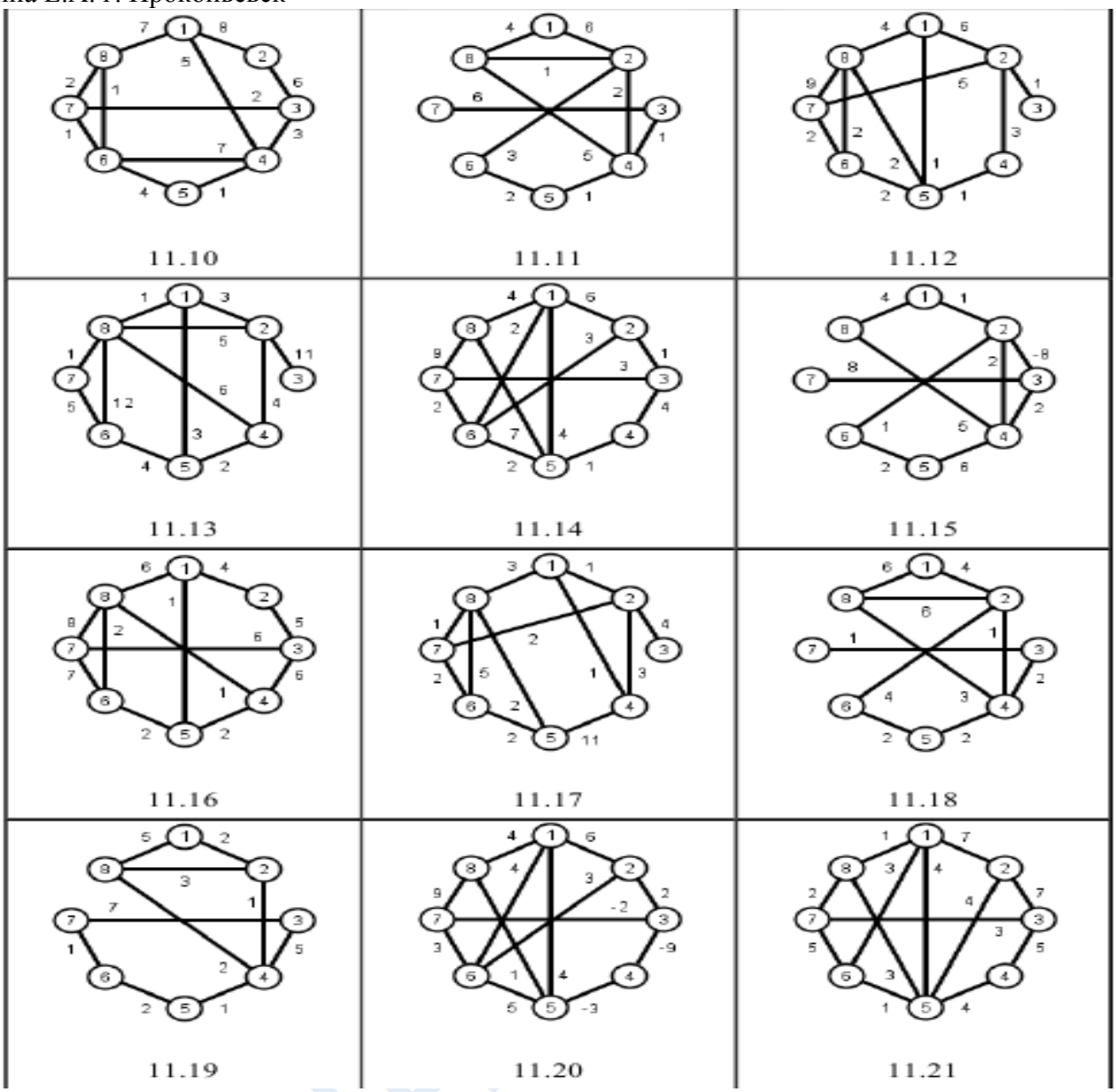
11.26



11.27

Чащина


11.1
11.2
11.3
11.4
11.5
11.6
11.7
11.8
11.9



Порядок выполнения работы:

1. Изучить примеры выполнения заданий
2. Выполнить задание.
3. Оформить отчет.

Содержание отчета:

1. Тема.
2. Цель.
3. Задачи.
4. Материальное обеспечение.
5. Практическое задание.

Вопросы для самоконтроля:

1.Чем отличается путь от маршрута?
2. Дайте определение понятию поиск кратчайшего пути?

3. Перечислите, какие алгоритмы по поиску кратчайшего пути вы знаете?
4. Какое программное обеспечение лучше подходит для вычисления пути?

Чащина Е.А.