

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПРОМЫШЛЕННЫХ ТЕХНОЛОГИЙ И ДИЗАЙНА»

ВЫСШАЯ ШКОЛА ТЕХНОЛОГИИ И ЭНЕРГЕТИКИ

В.П. Яковлев, Н.Л. Леонова

**ПРАКТИКУМ
ПО ДИСКРЕТНОЙ МАТЕМАТИКЕ**

Часть II

Учебное пособие

Санкт-Петербург
2020

УДК 519.854(075)

ББК 22.174я7

Я 474

Яковлев В.П., Леонова Н.Л. Практикум по дискретной математике: учебное пособие / ВШТЭ СПб ГУПТД. - СПб., 2020. Часть II. – 39 с.

В учебном пособии собраны задачи и минимально необходимый для их решения теоретический материал по дисциплине «Дискретная математика». Целью данного пособия является формирование у студентов практических навыков в решении задач. Рассмотрены типовые задачи теории графов и теории автоматов.

Пособие предназначено для студентов, обучающихся по направлению подготовки 01.03.02 «Прикладная математика и информатика».

Рецензенты: профессор Санкт-Петербургского государственного института кино и телевидения, д-р техн. наук, доцент В.М. Пестриков;

доцент кафедры прикладной математики и информатики ВШТЭ СПбГУПТД, канд. техн. наук П.Е. Антонюк.

Рекомендовано к изданию Редакционно-издательским советом ВШТЭ в качестве учебного пособия

© Высшая школа технологии
и энергетики СПбГУПТД, 2020

© Яковлев В.П., Леонова Н.Л., 2020

Редактор и корректор Н.П. Новикова

Техн. редактор Л.Я. Титова

Темплан 2020, поз. 137

Подп. к печати 28.12.2020 Формат 60x84/16. Бумага тип № 1.

Печать офсетная. 2,5 печ.л.; 2,5 уч.- изд.л. Тираж 30 экз. Изд. № 137.

Цена “С”. Заказ №

Ризограф Высшей школы технологии и энергетики СПбГУПТД, СПб.,
198095, ул. Ивана Черных, 4.

ВВЕДЕНИЕ

Теория графов – важнейший математический инструмент, широко используемый в информатике, химии, генетике, исследовании операций, лингвистике, проектировании, так как посредством графов можно описывать разнообразные реальные явления: организацию транспортных систем, сети передачи данных, человеческих взаимоотношений, структуру гена или молекулы. Возможность формального моделирования такого множества разных реальных структур позволяет программисту решать широкий круг прикладных задач. Разработка хорошего алгоритма «с нуля» – очень трудная задача. Часто достаточно правильно построить модель задачи и применить уже известный алгоритм, решающий задачу быстро и верно. В рамках этого пособия разбираются основные алгоритмы на графах, решающие практические задачи.

1. ОСНОВЫ ТЕОРИИ ГРАФОВ

1.1. НАЧАЛЬНЫЕ СВЕДЕНИЯ

Даже в период становления теории графов в ней возникало немало таких задач, решение которых предполагало построение некоторых алгоритмов. Достаточно вспомнить, например, задачу о кёнигсбергских мостах, которую решил в 1736 г. Л. Эйлер. В городе Кёнигсберге (нынешний Калининград) было два острова, соединенных семью мостами с берегами реки Преголя и друг с другом так, как показано на рис. 1.1.

Задача состояла в следующем: найти маршрут прохождения всех четырех частей суши, который начинался бы с любой из них, кончался бы на этой же части и ровно один раз проходил по каждому мосту. Исключительный вклад Л. Эйлера в решение этой задачи заключается в доказательстве, что такой маршрут построить невозможно.

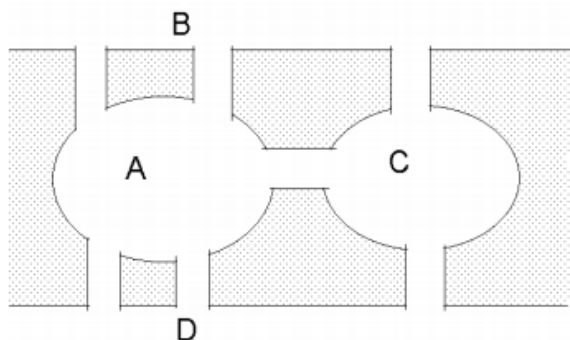


Рис. 1.1. Парк в г. Кёнигсберге

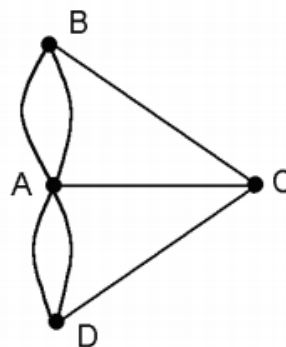


Рис.1.2. Граф к задаче о мостах

Эйлер обозначил каждую часть суши точкой (вершиной), а каждый мост линией (ребром), соединяющей соответствующие точки (см. рис. 1.2). Полученная конструкция, состоящая из точек и соединяющих их ребер, со временем получила название граф. Утверждение о том, что решения задачи о кенигсбергских мостах не существует, эквивалентно утверждению о невозможности обойти граф, представленный на рис. 1.2, проходя по каждому ребру в точности один раз. Отталкиваясь от этого частного случая, Эйлер обобщил постановку задачи и нашел критерий существования обхода у связного графа. На его основе позже был разработан алгоритм обхода связного графа.

Развитие теории графов в конце XIX - начале XX вв. было связано с распространением представлений о молекулярном строении вещества и становлением теории электрических цепей. Сейчас язык и методы теории графов используются не только в математике и традиционных приложениях в химии и электротехнике, но и в информатике, социологии, лингвистике, экономике, генетике. Очевидно, что алгоритмы для работы с графами очень важны. Внедрение вычислительной техники поставило и перед всей математикой, и перед теорией графов проблему нахождения не произвольных алгоритмов, позволяющих решать те или иные классы задач, а таких алгоритмов, которые допускали бы практическую реализацию на компьютере. Так возникла проблема практической разрешимости задач: найти эффективный или хотя бы достаточно простой в практически важных случаях алгоритм решения задачи.

1.1.1. ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ И ПОНЯТИЯ

Часто бывает полезно и наглядно изобразить некоторую ситуацию в виде рисунка, состоящего из точек (вершин), представляющих основные элементы ситуации и линий (ребер), отражающих связи между элементами. Такие рисунки называются **графами**. С помощью графа можно изобразить: сеть улиц в городе (вершины – перекрестки, улицы – ребра), блок-схемы программ (вершины – блоки, ребра – переходы), электрические цепи и т.п.

Графом называется пара $G = (V, E)$ в которой:

$V = \{v_1, v_2, \dots, v_n\}$ – конечное непустое множество;

$E = \{e_1, e_2, \dots, e_m\}$ – конечное множество, каждый элемент которого соответствует паре не обязательно различных элементов множества V .

Элементы множества V называются *вершинами* графа.

Если элемент множества E соответствует неупорядоченной паре вершин, т.е. $e_i \sim \{v_{i1}, v_{i2}\}$, то он называется *ребром*, вершины v_{i1} и v_{i2} - концами этого

ребра. Говорят, что ребро $e_i = \{v_{i1}, v_{i2}\}$ соединяет вершины v_{i1} и v_{i2} . Ребро, для которого $v_{i1} = v_{i2}$, называется петлей. Граф, содержащий ребра, называется *неориентированным графом (n-графом)*.

Если элемент множества E соответствует упорядоченной паре вершин, т.е. $e_i \sim (v_{i1}, v_{i2})$, то он называется *дугой*, вершина v_{i1} – началом, а вершина v_{i2} – концом этой дуги. Говорят, что дуга $e_i = (v_{i1}, v_{i2})$ исходит из вершины v_{i1} и заходит в вершину v_{i2} . Дуга, для которой $v_{i1} = v_{i2}$, называется *ориентированной петлей*. Граф, содержащий только дуги, называется *ориентированным графом (орграфом)*.

Ребра (дуги), соответствующие одним и тем же парам (упорядоченным парам) вершин называются *кратными*. Граф, не содержащий кратных ребер (дуг), называется *простым графом*, в противном случае граф является *мультиграфом*.

Если $e_i \sim \{v_{i1}, v_{i2}\}$ или $e_i \sim (v_{i1}, v_{i2})$, то вершины v_{i1} и v_{i2} называются *смежными*, а ребро (дуга) e_i и вершины v_{i1} и v_{i2} – *инцидентными*. Если конец ребра e_i совпадает с концом ребра e_j , то эти ребра называются *смежными*. Если начало и конец дуги e_i совпадает с началом или концом дуги e_j , то эти дуги называют *смежными*.

Последовательность вершин и ребер (дуг), не обязательно различных, такая, что соседние вершины и ребра (дуги) инцидентны, называются *путем*. Если путь не содержит повторяющихся вершин (кроме, может быть, первой и последней) и ребер или дуг, то он называется *простым*. Если первая и последняя вершины пути совпадают, то он называется *циклом*. *Длиной пути* называется число ребер (дуг), которые он содержит.

Часть графа G' называется *подграфом* графа G , если в ней сохраняются все ребра или дуги между вершинами.

Часть графа G' – *суграфом* графа G , если все вершины графа G сохраняются в G' .

Подграф G' графа G называется *компонентой связности*, если:

- 1) между любыми двумя его вершинами G' существует путь;
- 2) G' не является подграфом ни одного связного подграфа графа G , т.е. максимальным по включению вершин связным подграфом графа G .

Если граф содержит одну компоненту связности, то он называется *связным*, если более одной – *несвязным*.

Подграф G' графа G называется *кликкой*, если:

- 1) любые две вершины G' смежны;

2) G' не является подграфом ни одного подграфа G , обладающего свойством (1), т.е. является максимальным по включению вершин сильно связным подграфом графа G .

1.1.2. ИНВАРИАНТЫ ГРАФОВ

К основным числовым характеристикам графа $G = (V, E)$, относятся:

- число вершин – n ;
- число ребер или дуг – m ;
- число компонент связности – k ;
- цикломатическое число – $c = m + k - n$.

Граф $G = (V, E)$ называется раскрашенным (помеченным), если каждой его вершине приписан цвет (или номер) так, чтобы никакие две смежные вершины не имели одного цвета (номера).

Минимальное число цветов, которыми можно раскрасить граф, называется его *хроматическим числом* – h .

Максимальная длина простого пути в графе G называется *диаметром графа* – d .

Степенью вершины $v_i \in V$ – $deg v_i$ называется число ребер или дуг, инцидентных этой вершине. При этом каждая дуга учитывается дважды. Следующим инвариантом графа является *вектор степеней вершин графа* – $deg G = (deg v_1, deg v_2, \dots, deg v_n)$.

Если граф G ориентированный, то кроме степени для вершины $v_i \in V$ определяются:

$deg^- v_i$ – *полустепень исхода* – число дуг, началом которых является вершина v_i ;

$deg^+ v_i$ – *полустепень захода* – число дуг, концом которых является вершина v_i .

Справедливо равенство $deg v_i = deg^- v_i + deg^+ v_i$.

Инвариантами всего графа G являются:

вектор полустепеней исхода – $deg^- G = (deg^- v_1, deg^- v_2, \dots, deg^- v_n)$,

вектор полустепеней захода – $deg^+ G = (deg^+ v_1, deg^+ v_2, \dots, deg^+ v_n)$.

Ни один из рассмотренных инвариантов, ни даже их полная совокупность, не могут однозначно определить граф, т.е. существуют неизоморфные графы, для которых эти инварианты совпадают. Поэтому эти инварианты называют *неполными инвариантами*.

Полные инварианты графов могут быть построены с помощью матричных представлений.

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

Матрицей смежности графа G называется матрица $A = (a_{ij})$ размерности $n \times n$ (n – число вершин), которая определяется по правилу:

$$a_{ij} = \begin{cases} n_{ij} - \text{число ребер, концами которых являются вершины } v_i \text{ и } v_j, \\ \quad \quad \quad (\text{петля учитывается дважды}), \\ 0, \quad \text{если вершины } v_i \text{ и } v_j \text{ несмежны.} \end{cases}$$

Матрица A является симметрической, т.е. симметричной относительно главной диагонали.

Матрицей инцидентий графа G называется матрица $B = (b_{ik})$ размерности $n \times m$ (n – число вершин, m – число ребер), которая определяется по правилу:

$$b_{ik} = \begin{cases} 2, & \text{если вершина } v_i \text{ инцидентна петле } e_k, \\ 1, & \text{если вершина } v_i \text{ инцидентна ребру } e_k, \text{ не являющемуся петлей,} \\ 0, & \text{если вершина } v_i \text{ неинцидентна ребру } e_k. \end{cases}$$

Числовые характеристики графа связаны с введенными матрицами следующими соотношениями: $\text{deg}v_i = \sum_{j=1}^n a_{ij} \sum_{k=1}^m b_{ik}$

Пусть $\bar{G} = (V, E)$ – ориентированный граф.

Матрицей смежности графа G называется матрица $A = (a_{ij})$ размерности $n \times n$ (n – число вершин), которая определяется по правилу:

$$a_{ij} = \begin{cases} n_{ij} - \text{число дуг, исходящих из вершины } v_i \text{ и} \\ \quad \quad \quad \text{входящих в вершину } v_j, \\ 0, \quad \text{если вершины } v_i \text{ и } v_j \text{ несмежны.} \end{cases}$$

Матрица A в общем случае уже не будет являться симметрической.

Если $\bar{G} = (V, E)$ – простой граф, то A – матрица бинарного отношения E .

Матрицей инцидентий графа G называется матрица $B = (b_{ik})$ размерности $n \times m$ (n – число вершин, m – число дуг), которая определяется по правилу:

$$b_{ik} = \begin{cases} -1, & \text{если из вершины } v_i \text{ исходит дуга } e_k, \text{ не являющаяся петлей,} \\ 1, & \text{если в вершину } v_i \text{ входит дуга } e_k, \text{ не являющаяся петлей,} \\ 0, & \text{если вершина } v_i \text{ неинцидентна дуге } e_k \text{ или } e_k \text{ является петлей.} \end{cases}$$

Если граф $G = (V, E)$ не содержит петель, то матрица B полностью его характеризует. Если есть петли, то вместо одной матрицы инцидентий рассматривают две матрицы: исходящую матрицу инцидентий $B^- = (b_{ik}^-)$ и входящую матрицу инцидентий $B^+ = (b_{ik}^+)$, которые определяются следующим образом:

$$b_{ik}^- = \begin{cases} 1, & \text{если из вершины } v_i \text{ исходит дуга } e_k, \\ 0, & \text{если вершина } v_i \text{ неинцидентна дуге } e_k. \end{cases}$$

$$b_{ik}^+ = \begin{cases} 1, & \text{если в вершину } v_i \text{ входит дуга } e_k, \\ 0, & \text{если вершина } v_i \text{ неинцидентна дуге } e_k. \end{cases}$$

Все введенные матрицы инцидентий связаны соотношением: $B = B^+ - B^-$. Для петель, в частности, получаем: $b_{ik} = b_{ik}^+ - b_{ik}^- = 1 - 1 = 0$.

Числовые характеристики графа связаны с введенными матрицами следующими соотношениями: $\deg^- v_i = \sum_{j=1}^n a_{ij} = \sum_{k=1}^m b_{ik}^-$,
 $\deg^+ v_j = \sum_{i=1}^n a_{ij} = \sum_{k=1}^m b_{jk}^+$.

Матрица инцидентий B является полным инвариантом, т.е. характеристикой, однозначно определяющей неориентированный мультиграф или ориентированный мультиграф без петель. Для ориентированного мультиграфа с петлями полным инвариантом является пара матриц B^+ и B^- .

Если $G = (V, E)$ является простым графом, то множество ребер или дуг однозначно определяется отношением E и, тем самым, полным инвариантом будет служить, кроме матриц инцидентий также и матрица смежности A , независимо от того, является ли граф ориентированным или нет.

Рассмотрим примеры вычисления основных числовых инвариантов графов и их матричных представлений.

Пример 1.1. Пусть задан неориентированный граф (Рис. 1.3):

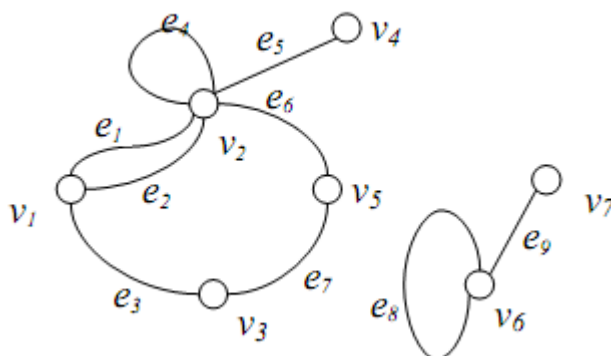


Рис. 1.3. Исходный неориентированный граф $G = (V, E)$

Находим основные инварианты:

- 1) Число вершин $|V| \quad n = 7$.
- 2) Число ребер $|E| \quad m = 9$.
- 3) Число компонент связности $k = 2$.

Одна компонента связности составлена из вершин $\{v_1, v_2, v_3, v_4, v_5\}$, другая – $\{v_6, v_7\}$.

- 4) Цикломатическое число $c = m + k - n = 9 + 2 - 7 = 4$.
- 5) Хроматическое число $h = 2$.

Минимальная раскраска графа цветами «А» и «В» представлена на рис. 1.4.

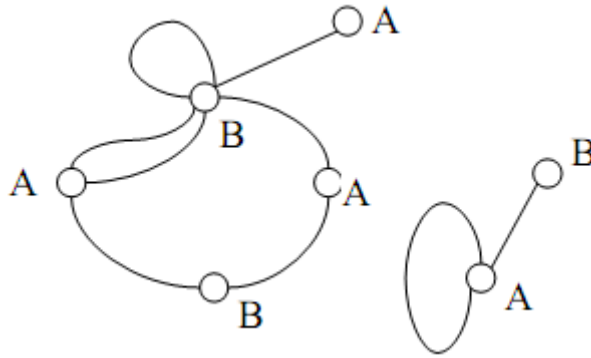


Рис. 1.4. Минимальная раскраска графа $G = (V, E)$

- 6) Плотность графа $p = 2$. Максимальную клику составляют, например, вершины v_1 и v_2 .
- 7) Диаметр графа $d = 4$. Это следует из того, что максимально длинный простой путь, например, $(v_1, e_3, v_3, e_7, v_5, e_6, v_2, e_5, v_4)$ содержит четыре ребра.
- 8) Вектор степеней вершин графа $degG = (3, 6, 2, 1, 2, 3, 1)$.
- 9) Матрица смежности графа G :

$$M = \begin{matrix} & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{matrix} & \begin{pmatrix} 0 & 2 & 1 & 0 & 0 & 0 & 0 \\ 2 & 2 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{matrix}$$

- 10) Матрица инциденций графа G :

$$N = \begin{matrix} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{matrix} & \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 2 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{matrix}$$

Рассмотрим пример нахождения основных инвариантов ориентированных графов.

Пример 1.2. Пусть задан ориентированный граф.

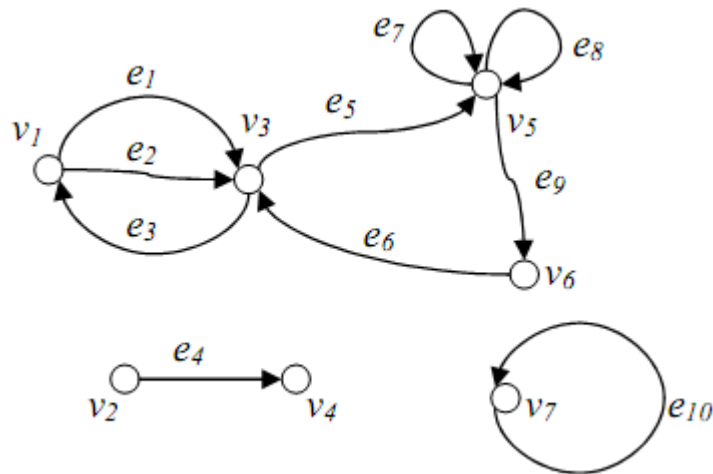


Рис. 1.5. Ориентированный граф $\bar{G} = (V, E)$

Находим основные инварианты:

1. Число вершин $|V| \quad n = 7$.
2. Число ребер $|E| \quad m = 10$.
3. Число компонент связности $k = 3$.

Одна компонента связности составлена из вершин $\{v_1, v_3, v_5, v_6\}$, другая – $\{v_2, v_4\}$, третья – $\{v_7\}$.

4. Цикломатическое число $c = m + k - n = 10 + 3 - 7 = 6$.
5. Хроматическое число $h = 3$.

Минимальная раскраска графа цветами «А», «В» и «С» представлена на рис. 1.6.

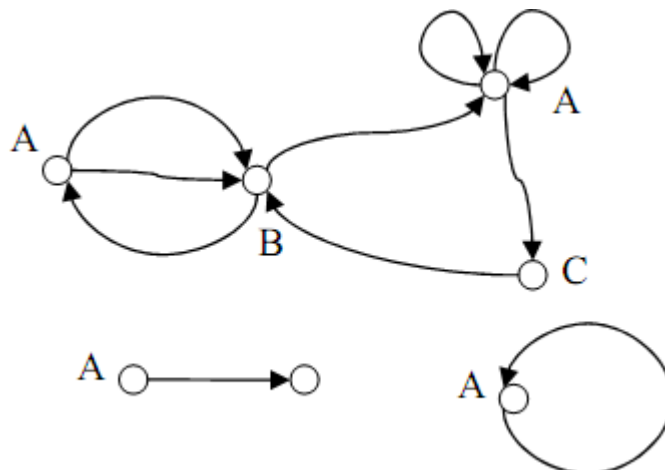


Рис. 1.6. Минимальная раскраска графа $\bar{G} = (V, E)$

6. Плотность графа $p = 3$. Максимальную клику составляют, например, вершины v_3, v_5 и v_6 .
7. Диаметр графа $d = 3$. Это следует из того, что максимально длинный простой путь, например $(v_1, e_3, v_3, e_5, v_5, e_9, v_6)$ содержит три дуги.

8. Вектор степеней вершин графа \bar{G} :

$$\text{deg}\bar{G} = (3, 1, 5, 1, 6, 2, 2).$$

9. Вектор полустепеней исхода вершин графа \bar{G} :

$$\text{deg}^-\bar{G} = (2, 1, 2, 0, 3, 1, 1).$$

10. Вектор полустепеней захода вершин графа \bar{G} :

$$\text{deg}^+\bar{G} = (1, 0, 3, 1, 3, 1, 1).$$

11. Матрица смежности графа \bar{G} :

$$M = \begin{matrix} & \begin{matrix} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{matrix} & \begin{pmatrix} 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{matrix}.$$

12. Матрица инцидентий графа \bar{G} :

$$N = \begin{matrix} & \begin{matrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 & e_{10} \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{matrix} & \begin{pmatrix} -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & -1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix};$$

Исходящая матрица инцидентий:

$$N^- = \begin{matrix} & \begin{matrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 & e_{10} \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{matrix} & \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{matrix};$$

Входящая матрица инциденций:

$$N^+ = \begin{matrix} & \begin{matrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 & e_{10} \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{matrix} & \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{matrix}.$$

1.1.3. ЭЙЛЕРОВЫ И ГАМИЛЬТОНОВЫ ПУТИ И ЦИКЛЫ

Эйлеровым циклом называется цикл в графе G , который содержит все ребра и каждое только один раз.

Эйлеровым графом называется граф, содержащий эйлеров цикл.

Эйлеровым путем называется путь в графе G , не являющийся циклом, который содержит все ребра в графе и каждое только один раз.

Полуэйлеровым графом называется граф, содержащий эйлеров путь.

Граф G является эйлеровым тогда и только тогда, когда все вершины графа имеют четную степень.

Граф G является полуэйлеровым тогда и только тогда, когда все вершины графа, кроме двух, имеют четную степень. При этом путь начинается в одной вершине с нечетной степенью и заканчивается в другой.

Гамильтоновым циклом называется простой цикл в графе G , который содержит все вершины графа и каждую только один раз.

Гамильтоновым графом называется граф, содержащий гамильтонов цикл.

Гамильтоновым путем называется простой путь в графе G , не являющийся циклом, который содержит все вершины в графе и каждую только один раз.

Очевидно, что если граф содержит гамильтонов цикл, то он содержит и гамильтонов путь.

Полугамильтоновым графом называется граф, содержащий гамильтонов путь, но не содержащий гамильтонов цикл.

Одним из наиболее простых достаточных условий гамильтоновости графа является *теорема Дирака*.

Если для всех вершин $v_i \in V$ графа G , имеющего n вершин, выполняется условие $deg v_i \geq \frac{n}{2}$, то граф является гамильтоновым.

Историческая справка: Слово «гамильтонов» в этих определениях связано с именем известного ирландского математика У. Гамильтона, которому в 1859 г. была предложена игра «Кругосветное путешествие». Каждой из двадцати вершин додекаэдра приписано название одного из крупных городов мира. Требуется, переходя от одного города к другому по ребрам додекаэдра, посетить каждый город в точности один раз и вернуться в исходный город. Эта задача, очевидно, сводится к отысканию в графе додекаэдра простого цикла, проходящего через каждую вершину этого графа (рис.1.7).

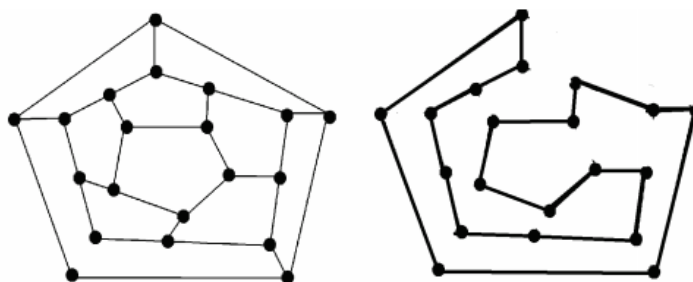


Рис. 1.7. Граф додекаэдра и его гамильтонов цикл

Несмотря на внешнее сходство постановок, задачи распознавания эйлеровости и гамильтоновости графа принципиально различны. Легко узнать, является ли граф эйлеровым и, в случае положительного ответа, алгоритм Флёрри позволяет достаточно быстро построить один из эйлеровых циклов. Ответить на вопрос, является ли данный граф гамильтоновым, как правило, очень трудно.

Рассмотрим алгоритмы построения эйлерова пути или цикла в связном неориентированном эйлеровом мультиграфе (алгоритм Х. Туя).

Пусть $G = (V, E)$ – полуэйлеров граф, а u и w – вершины, между которыми будет строиться эйлеров путь, т.е. вершины с нечетными степенями.

Алгоритм построения эйлерова пути.

Шаг 1. Находим простой путь, соединяющий вершины u и w .

Шаг 2. Всем ребрам найденного пути присвоим метки $h = 0$.

Шаг 3. Если в графе G помечены все ребра, то переход к шагу 8.

Шаг 4. Пусть максимальная пометка ребер равна h . Найти непомеченное ребро e , смежное с каким-либо уже помеченным.

Шаг 5. Построить простой цикл из непомеченных ребер, содержащий ребро e .

Шаг 6. Пусть ранее достигнутая максимальная пометка ребер равна h . Тогда всем ребрам вновь построенного цикла присвоить пометку $h + 1$.

Шаг 7. Переход к шагу 3.

Шаг 8. Построить эйлеров путь

$$u = x_1, e_1, x_2, e_2, x_3, e_3, \dots, x_m, e_m, w = x_{m+1},$$

начинающийся в вершине u и заканчивающийся в вершине w .

При этом, если путь был определен до вершины x_i , то последующее ребро e_{i+1} должно обладать следующими свойствами:

- 1) быть инцидентным x_i ;
- 2) не быть уже включенным в путь;
- 3) иметь максимальную пометку.

Пусть теперь $G = (V, E)$ – эйлеров граф.

Обозначим u и w – две произвольные смежные вершины, соединенные ребром e_0 .

Алгоритм построения эйлерова цикла отличается от приведенного алгоритма следующими шагами.

Шаг 1. Находим простой цикл, содержащий вершины u и w .

Шаг 8. Построить эйлеров цикл

$$u = x_1, e_1, x_2, e_2, x_3, e_3, \dots, x_m, e_m, w = x_{m+1}, e_0, u.$$

Для нахождения эйлеровых путей и циклов может быть использован алгоритм Туя. Рассмотрим использование этого алгоритма на примере.

Пример 1.3. Пусть задан неориентированный мультиграф, рис 1.8.

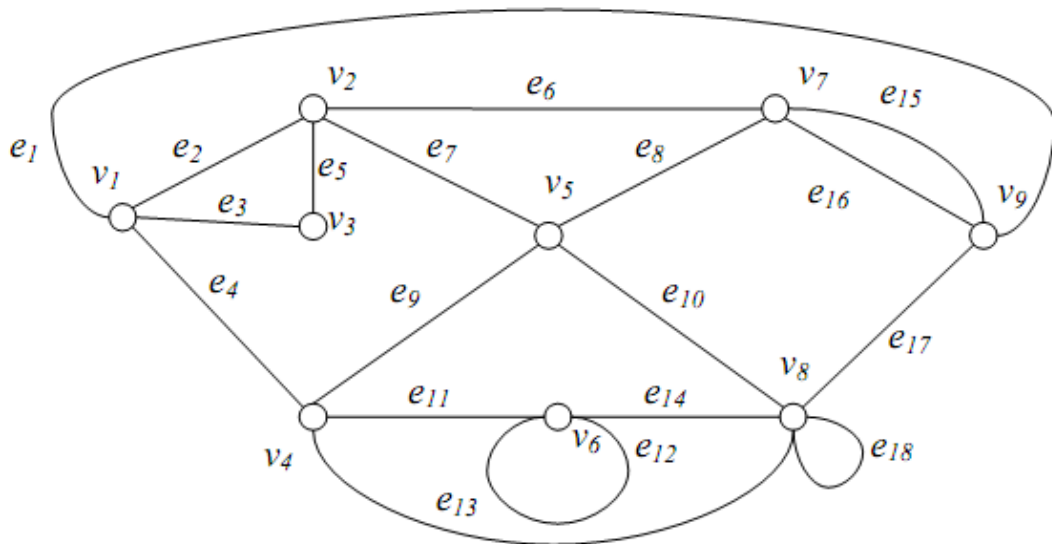


Рис. 1.8. Исходный неориентированный граф $G = (V, E)$

Находим степени вершин графа G .

$$\text{deg}G = (4, 4, 2, 4, 4, 4, 4, 6, 4).$$

Все степени вершин являются четными, поэтому по теореме Эйлера граф является эйлеровым, т.е. содержит эйлеров цикл.

Для того чтобы найти эйлеров цикл, выделяем ребро e_1 , соединяющее различные вершины v_1 и v_9 , а затем осуществим пометку ребер в соответствии с алгоритмом Туя:

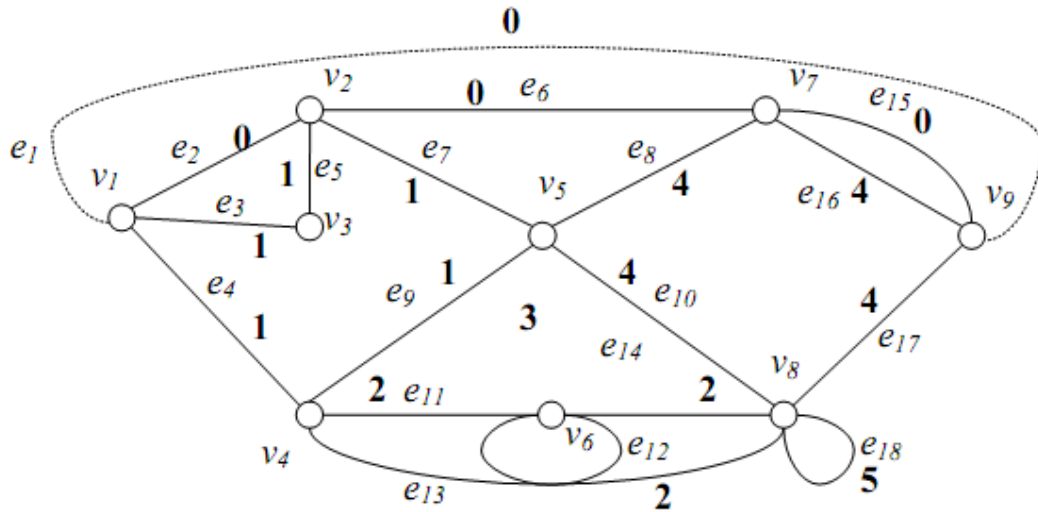


Рис. 1.9. Граф $G = (V, E)$, помеченный в соответствии с алгоритмом Туя. Используя полученные пометки вершин, находим эйлеров цикл:

$v_1, e_4, v_4, e_{11}, v_6, e_{14}, v_6, e_{12}, v_8, e_{18}, v_8, e_{17}, v_9, e_{16}, v_7, e_8, v_5, e_{10}, v_8, e_{13}, v_4, e_9, v_5, e_7, v_2, e_5, v_3, e_3, v_1, e_2, v_2, e_6, v_7, e_{15}, v_9, e_1, v_1$.

Если граф скажется полуэйлеровым, т.е. будет содержать эйлеров путь, то пометки «0» ставятся на ребра любого простого пути, соединяющего обе вершины с нечетными степенями.

Условие теоремы Дирака для рассматриваемого графа не выполняется. Поэтому сделать вывод о гамильтоновости графа, используя эту теорему, нельзя. Находим гамильтонов путь с помощью *перебора вариантов*:

$v_1, e_3, v_3, e_5, v_2, e_7, v_5, e_8, v_7, e_{16}, v_9, e_{17}, v_8, e_{14}, v_6, e_{11}, v_4$;

и гамильтонов цикл:

$v_1, e_3, v_3, e_5, v_2, e_7, v_5, e_8, v_7, e_{16}, v_9, e_{17}, v_8, e_{14}, v_6, e_{11}, v_4, e_4, v_1$.

Таким образом, эмпирически доказано, что данный граф является гамильтоновым.

1.1.4. КРАТЧАЙШИЙ ПУТЬ В ГРАФЕ

Пусть $\bar{G} = (V, E)$ – ориентированный взвешенный связный граф. При этом дугам приписаны веса. Сумму весов дуг в пути называется длиной пути. Путь называется кратчайшим, если его длина минимальна.

Будем считать, что \bar{G} не содержит кратные дуги и петли. Если это не так, то его необходимо преобразовать: удалить все петли и кратные дуги, кроме одной дуги, имеющей наименьший вес.

Если $(v_i, v_j) \in E$, то вес этой дуги – p_{ij} , в противном случае $p_{ij} = \infty$.

В процессе реализации алгоритма вершины получают метки, равные длине пути от начальной вершины v_0 до данной вершины v_i . Метки могут быть временными и окончательными.

Временная метка вершины v_i – q_i (кратчайший путь от вершины v_0 до вершины v_i , при условии, что учтены, возможно, не все пути от вершины v_0 до вершины v_i).

Окончательная метка вершины v_i – Q_i (кратчайший путь от вершины v_0 до вершины v_i , при условии, что учтены все пути от вершины v_0 до вершины v_i).

Рассмотрим общую схему реализации алгоритма.

Пусть требуется найти кратчайший путь от вершины v_0 до вершины v_n .

Шаг 1. Вершине v_0 присваивается окончательная метка $Q_i = 0$ (что означает нулевой кратчайший путь от вершины до самой себя). Всем остальным вершинам графа приписываются временные метки $q_i = \infty$.

Шаг 2. Пусть v_j – последняя вершина, которой присвоена окончательная метка. Каждой вершине v_i , не имеющей окончательной метки, присваивается новая временная метка $q = \min(q_i, Q_j + p_{ij})$.

Шаг 3. Определяется наименьшая из всех временных меток и объявляется окончательной (если таких меток несколько, выбирается любая).

Шаг 4. Если вершина v_n не получила окончательной метки, то переход к шагу 2. Иначе величина Q_n – искомая длина кратчайшего пути.

Шаг 5. Величина Q_n – искомая длина кратчайшего пути. Найти кратчайший путь, двигаясь обратно от v_n к v_0 только через те вершины, из которых последующие получали окончательные метки.

Рассмотрим пример нахождения кратчайшего пути в заданном графе из вершины v_0 в вершину u (рис.1.10).

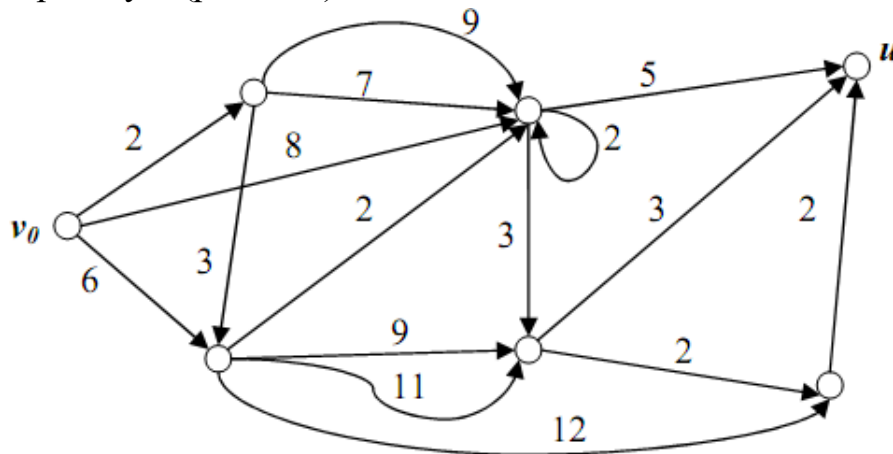


Рис. 1.10. Исходный ориентированный граф

1. Исключаем петли и кратные дуги большего веса из графа, после чего обозначаем вершины графа (рис.1.11).

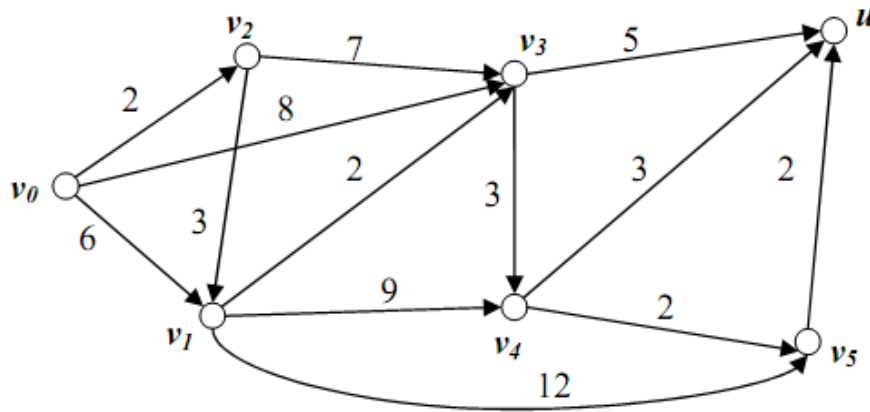


Рис. 1.11. Означенный простой граф

1. Строим таблицу, отображающую шаги реализации метода поиска в ширину, табл. 1.1.

Таблица 1.1

вершины \ шаги	v_0	v_1	v_2	v_3	v_4	v_5	u
0	0	∞	∞	∞	∞	∞	∞
1		6	2	8	∞	∞	∞
2		5		8	∞	∞	∞
3				7	14	17	∞
4					10	17	12
5						12	12

Анализ табл.1.1 показывает, что кратчайший путь проходит через вершины: v_0, v_2, v_1, v_3, u . Длина пути равна 12.

1.1.5. ПОСТРОЕНИЕ МИНИМАЛЬНОГО ОСТОВА

Пусть $G = (V, E)$ – неориентированный связный взвешенный граф. При этом веса приписаны ребрам. Граф G может содержать несколько остовных деревьев, т.е. связных суграфов $T = (V, U), U \subseteq E$, не содержащих циклов. Сумму весов ребер остовного дерева будем называть весом этого дерева. Дерево называется минимальным, если его вес минимален.

Наиболее просто описывается построение остовного дерева с помощью жадного алгоритма (алгоритма Краскала).

Пусть граф G имеет n вершин. Тогда число ребер в остовном дереве будет содержать $n - 1$ ребро.

Алгоритм Краскала имеет следующий вид.

Шаг 1. Строим 0-граф $T = (V, \emptyset)$.

Шаг 2. Составляем список ребер графа, упорядоченных по возрастанию весов.

Шаг 3. Добавляем очередное ребро e_i из списка в граф T .

Шаг 4. Проверяем, не появился ли в графе T цикл.

Шаг 5. Если цикл появляется, то ребро e_i из графа T исключается.

Шаг 6. Если число ребер в графе T равно $n - 1$, то T – искомое минимальное остовное дерево. В противном случае перейти к шагу 3.

Покажем способ нахождения минимального остовного дерева с помощью алгоритма Краскала на примере.

Пусть задан неориентированный мультиграф, ребрам которого приписаны веса (рис.1.12).

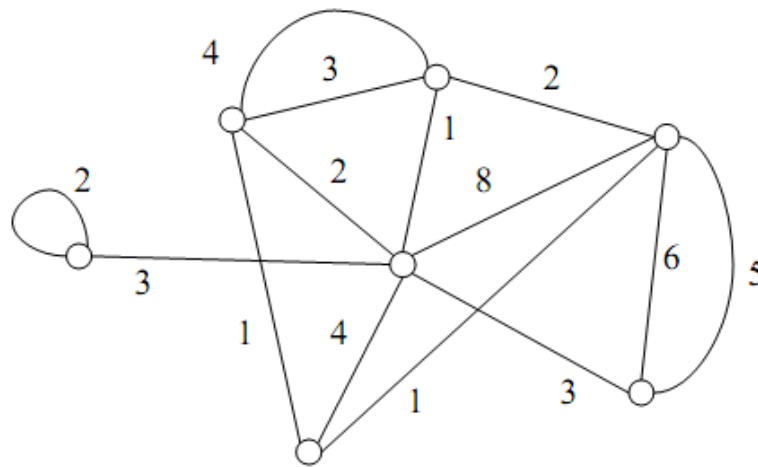


Рис. 1.12. Исходный неориентированный взвешенный граф
Обозначим вершины и ребра графа (рис.1.13):

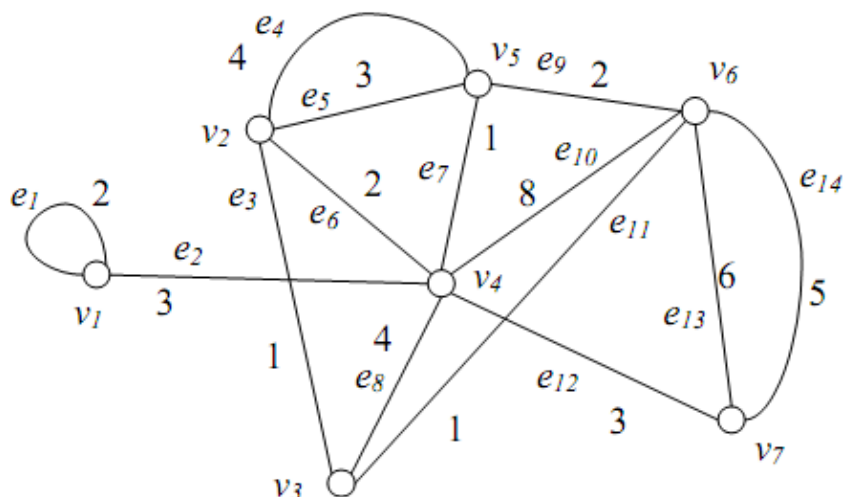


Рис. 1.13. Означенный граф $G = (V, E)$

Составим табл.1.2 на основе матрицы инциденций графа $G = (V, E)$, в которой:

- 1) все ребра упорядочены по возрастанию весов;

- 2) указаны веса ребер;
- 3) отмечен признак «+» образования цикла при добавлении данного ребра.

Таблица 1.2

	e_3	e_7	e_{11}	e_1	e_9	e_6	e_2	e_{12}	e_5	e_8	e_4	e_{14}	e_{13}	e_{10}
v_1	0	0	0	2	0	0	1	0	0	0	0	0	0	0
v_2	1	0	0	0	0	1	0	0	1	0	1	0	0	0
v_3	1	0	1	0	0	0	0	0	0	1	0	0	0	0
v_4	0	1	0	0	0	1	1	1	0	1	0	0	0	1
v_5	0	1	0	0	1	0	0	0	1	0	1	0	0	0
v_6	0	0	1	0	1	0	0	0	0	0	0	1	1	1
v_7	0	0	0	0	0	0	0	1	0	0	0	1	1	0

Вес	1	1	1	2	2	2	3	3	3	4	4	5	6	8
Образ. цикла	-	-	-	+	-	+	-	-						

По результатам анализа табл.1.2 строим минимальное остовное дерево (рис.1.14).

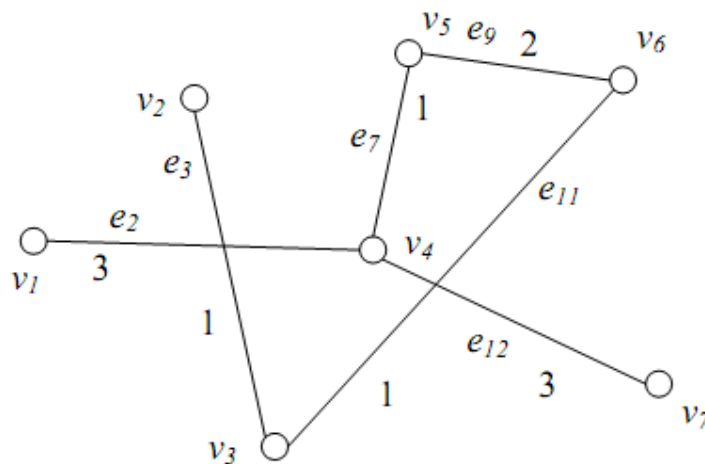


Рис. 1.14. Минимальное остовное дерево графа $G = (V, E)$

Меньшую вычислительную сложность имеет **алгоритм Прима**, построенный по схеме поиска в ширину. Алгоритм имеет следующий вид.

Шаг 1. Выбирается произвольная вершина v_i и строится граф из одной изолированной вершины $T = (v_i, \emptyset)$.

Шаг 2. Среди всех ребер, не включенных в граф T и инцидентных вершинам графа T , выбирается ребро $e = (v_j, v_k)$, где вершина v_j ,

предполагается включенной в граф T . Если таких ребер несколько, то выбирается любое.

Шаг 3. Ребро e и вершина v_k включаются в граф T .

Шаг 4. Если число ребер в графе T равно $n - 1$, то T – искомое минимальное остовное дерево. В противном случае - переход к шагу 2.

Покажем способ нахождения минимального остовного дерева с помощью алгоритма Прима на примере.

Пусть задан неориентированный мультиграф (рис.1.15), ребрам которого приписаны веса и означенный граф (рис.1.16).

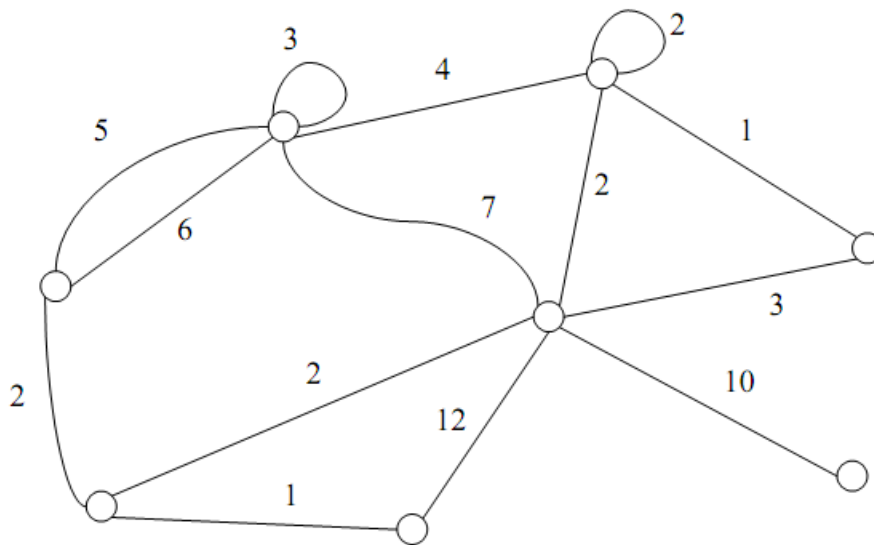


Рис. 1.15. Исходный неориентированный взвешенный граф

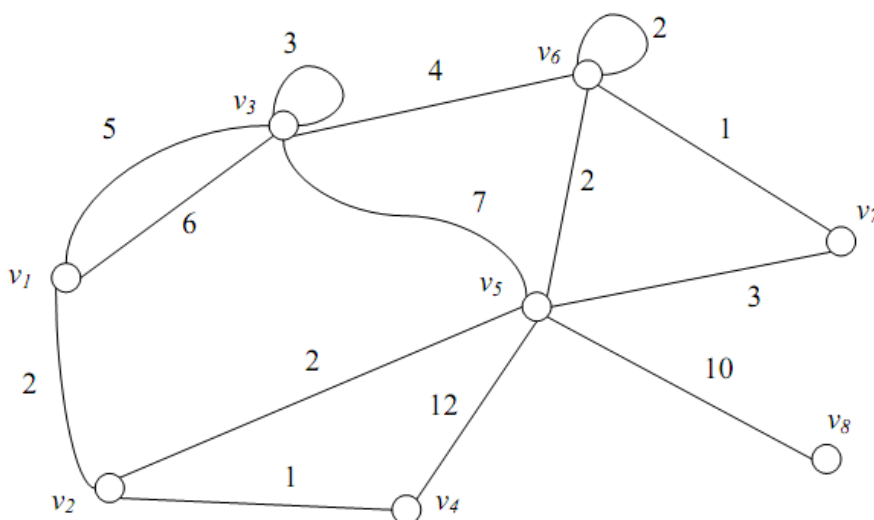


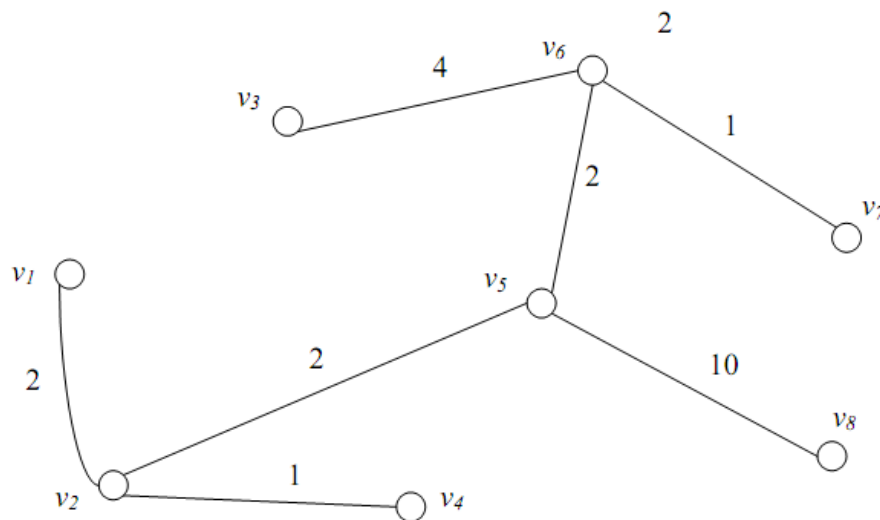
Рис. 1.16. Означенный граф $G = (V, E)$

Построим табл.1.3, описывающую шаги реализации метода поиска в ширину.

Таблица 1.3

Шаг \ Вершина	1	2	3	4	5	6	7
v_1	-	-	-	-	-	-	-
v_2	$(v_1, 2)^*$	-	-	-	-	-	-
v_3	$(v_1, 5)$	$(v_1, 5)$	$(v_1, 5)$	$(v_1, 5)$	$(v_6, 4)$	$(v_6, 4)^*$	-
v_4	∞	$(v_2, 1)^*$	-	-	-	-	-
v_5	∞	$(v_2, 2)$	$(v_2, 2)^*$	-	-	-	-
v_6	∞	∞	∞	$(v_5, 2)^*$	-	-	-
v_7	∞	∞	∞	$(v_5, 3)$	$(v_6, 1)^*$	-	-
v_8	∞	∞	∞	$(v_5, 10)$	$(v_5, 10)$	$(v_5, 10)$	$(v_5, 10)$

По результатам анализа табл.1.3 строим минимальное остовное дерево (рис.1.17).

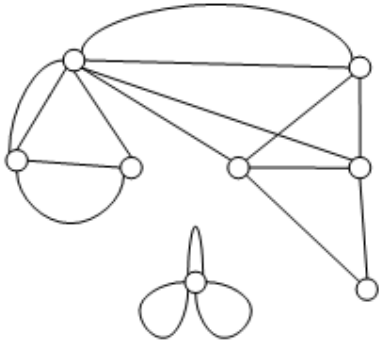


1.17. Минимальное остовное дерево графа $G = (V, E)$

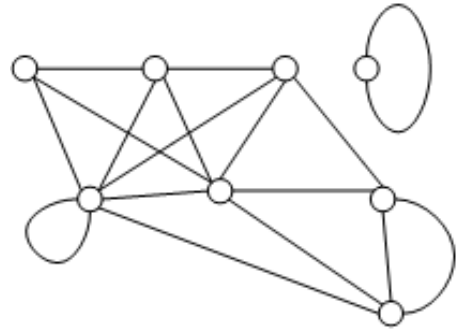
1.2. ЗАДАЧИ ДЛЯ САМОСТОЯТЕЛЬНОГО РЕШЕНИЯ

Задание 1.2.1

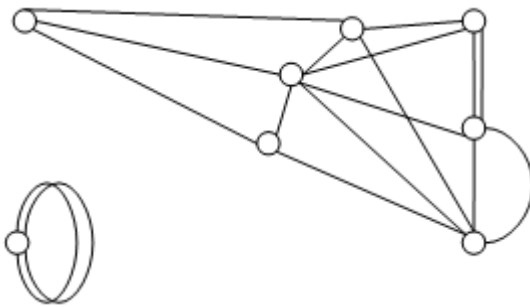
Найти инварианты неориентированных графов (число вершин, число ребер, число компонент связности, цикломатическое число, хроматическое число, плотность графа, вектор степеней вершин, матрицу смежности, матрицу инцидентий).



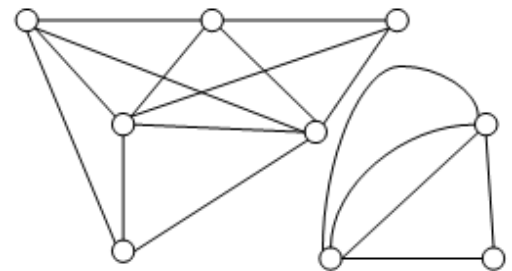
1



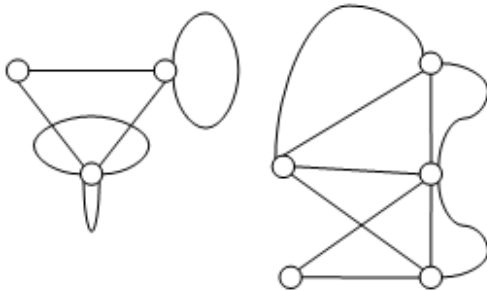
2



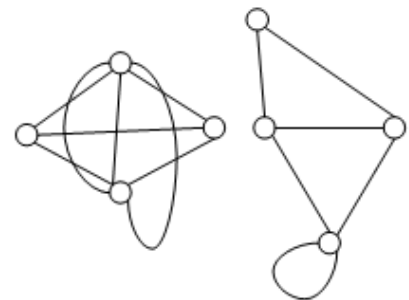
3



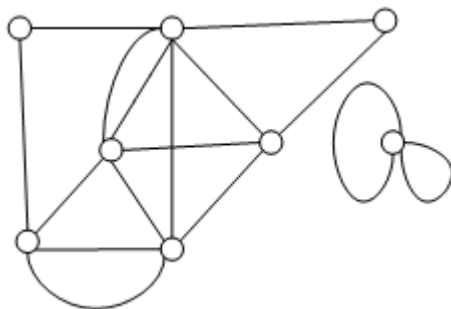
4



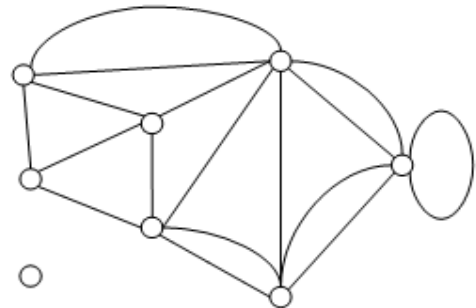
5



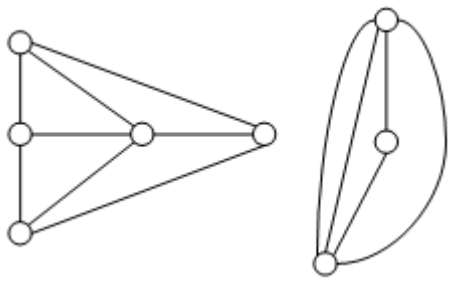
6



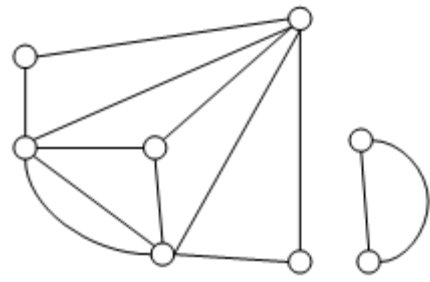
7



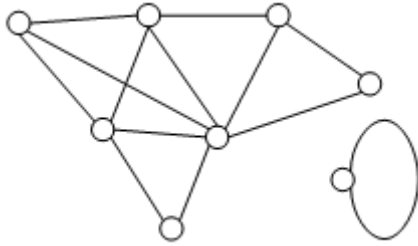
8



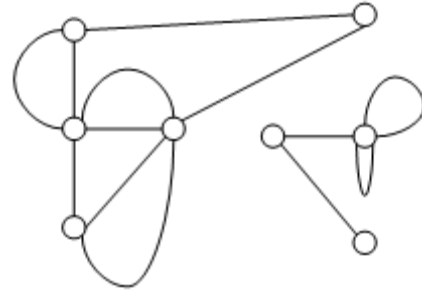
9



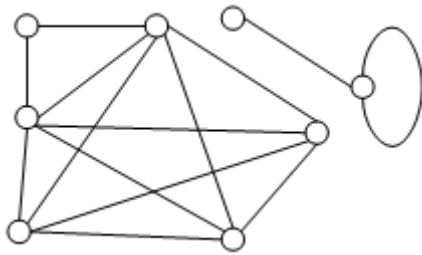
10



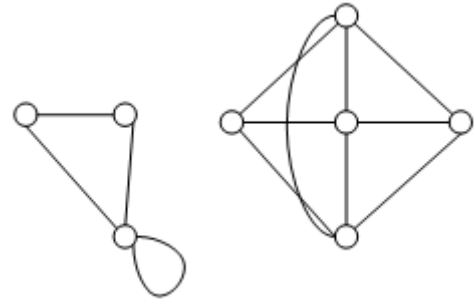
11



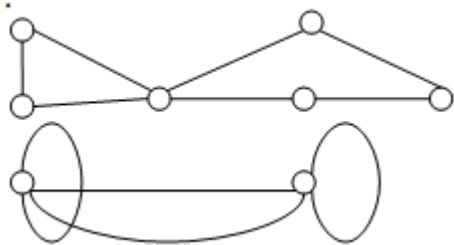
12



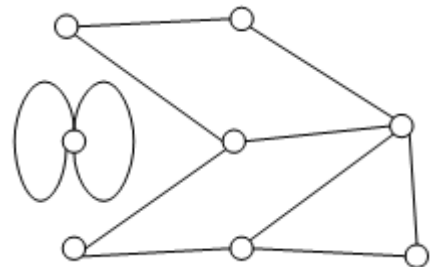
13



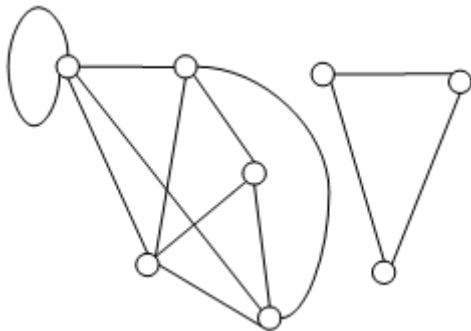
14



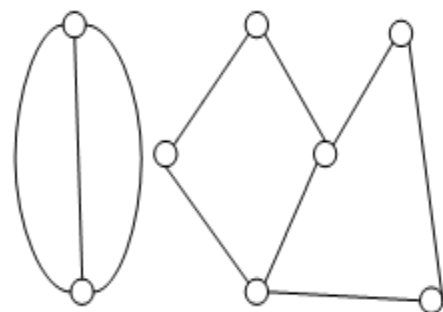
15



16



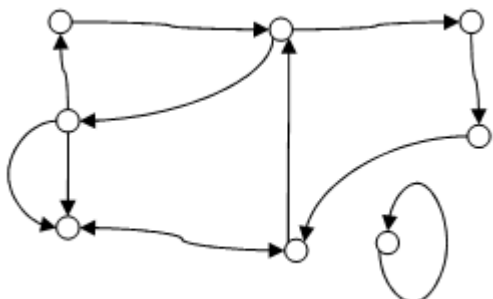
17



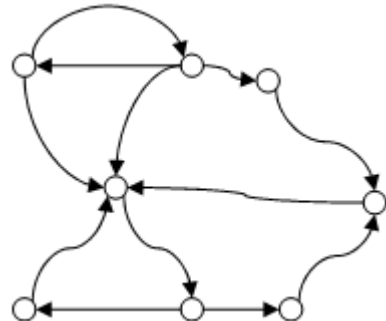
18

Задание 1.2.2

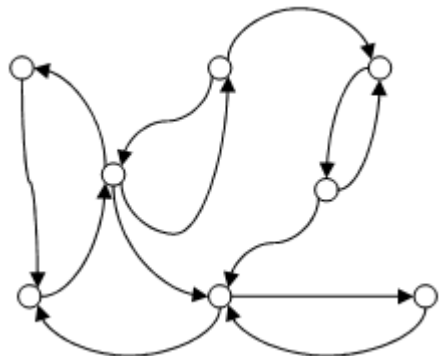
Найти инварианты ориентированных графов (число вершин, число дуг, число компонент связности, цикломатическое число, хроматическое число, плотность графа, вектор степеней вершин, матрицу смежности, матрицу инциденций).



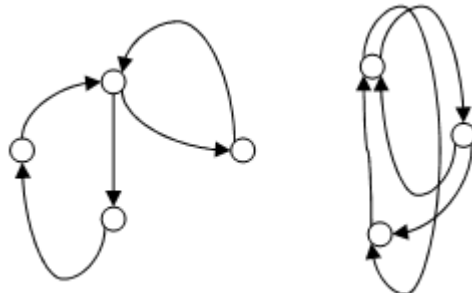
1



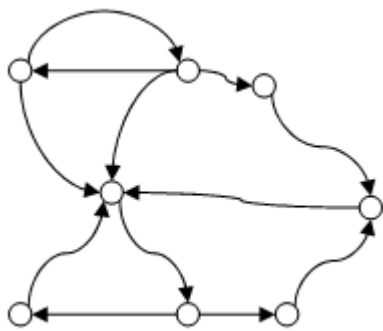
2



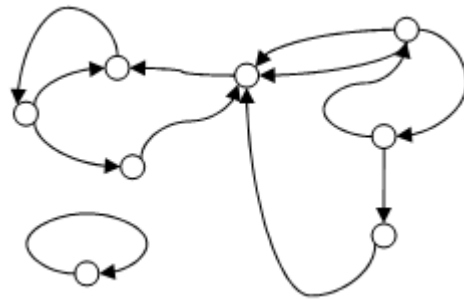
3



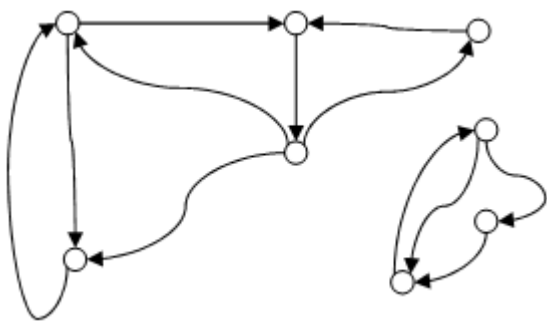
4



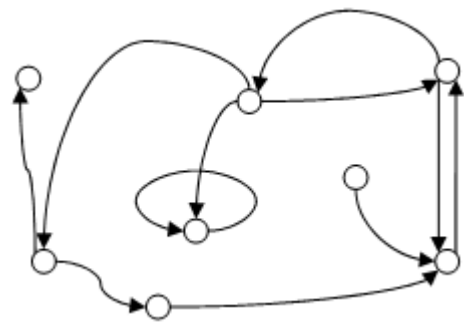
5



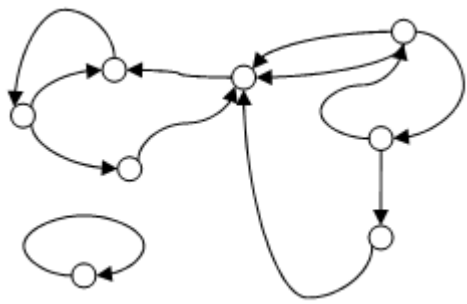
6



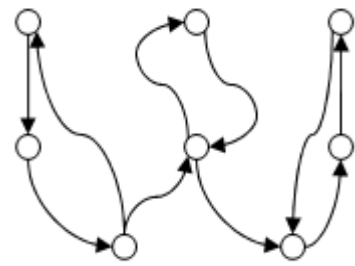
7



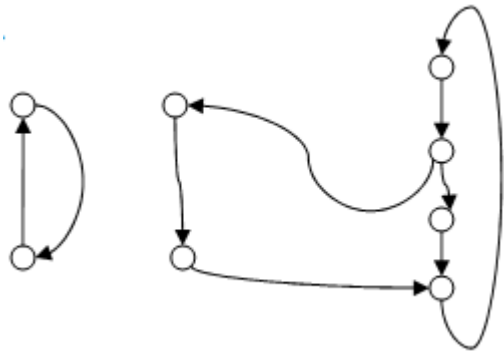
8



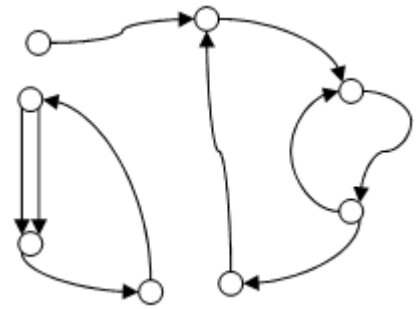
9



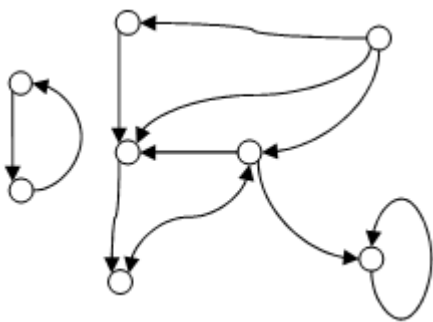
10



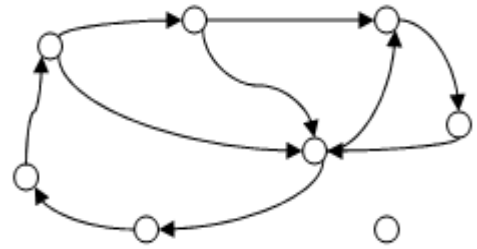
11



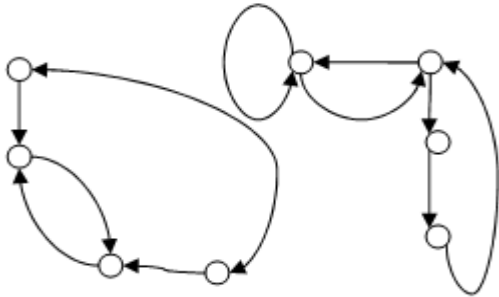
12



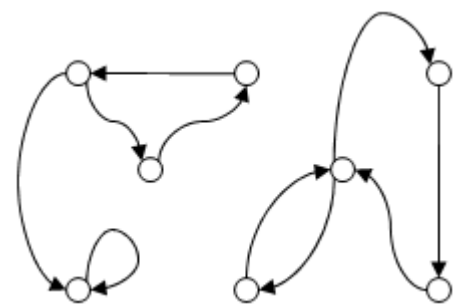
13



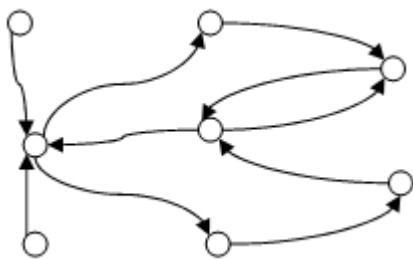
14



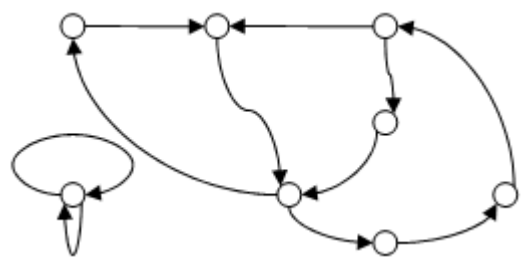
15



16



17

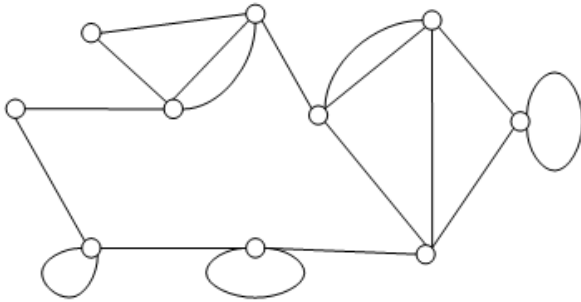


18

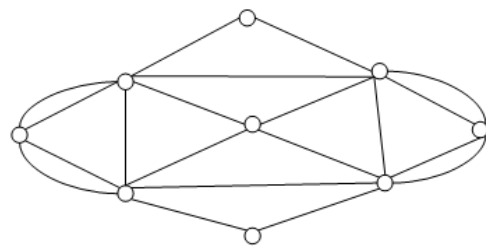
Задание 1.2.3

- 1) Проверить, является ли граф, изображенный на рисунке, эйлеровым или полуйлеровым.
- 2) Найти эйлеров путь или эйлеров цикл.
- 3) Проверить наличие достаточного условия гамильтоновости графа, изображенного на рисунке, по теореме Дирака.
- 4) Найти гамильтонов путь или цикл, если они существуют.
- 5) Сделать вывод о гамильтоновости или полугамильтоновости заданного графа.

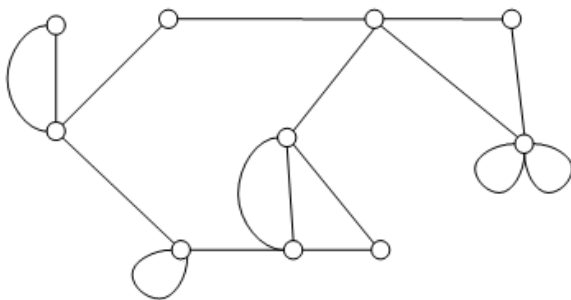
1



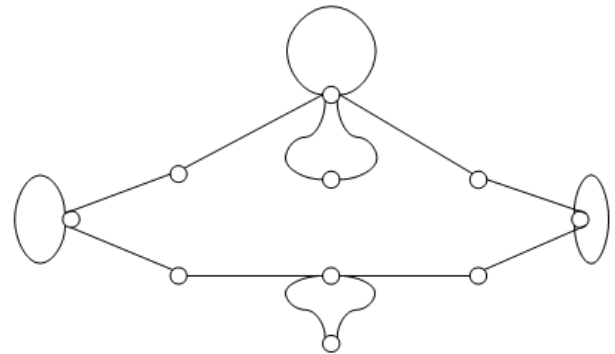
4



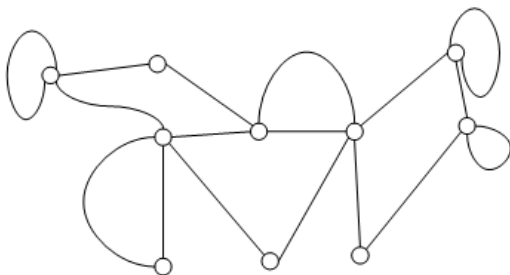
2



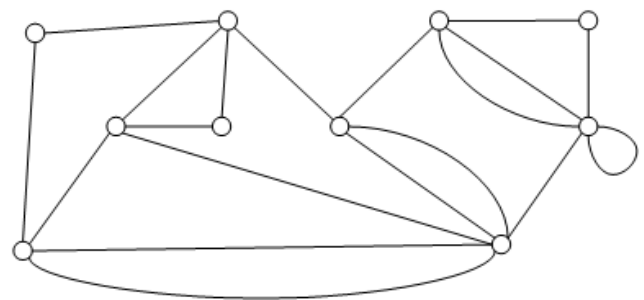
5



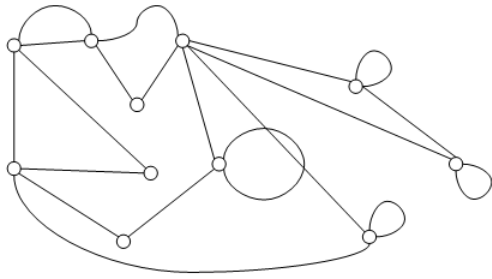
3



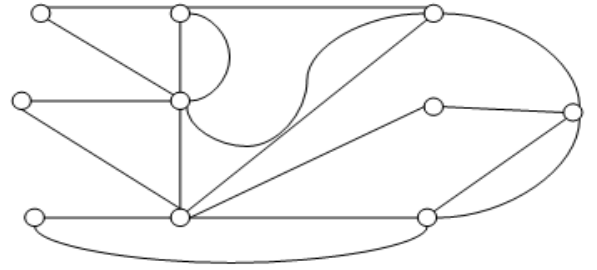
6



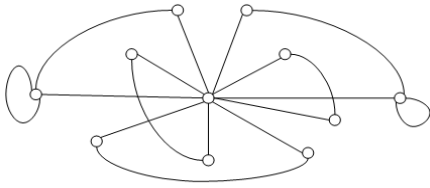
7



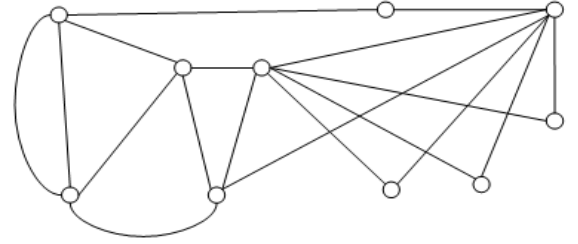
10



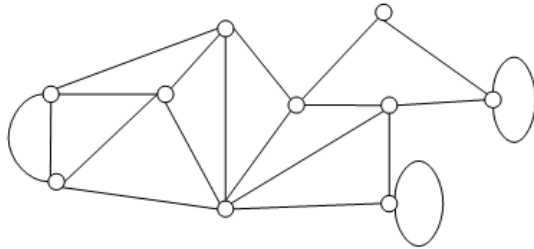
8



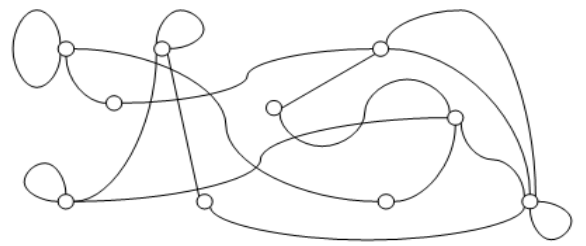
11



9



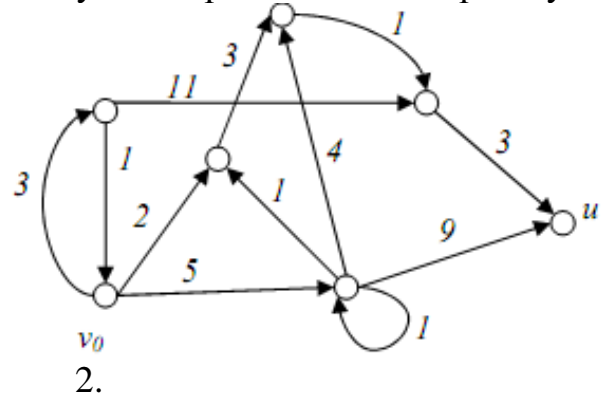
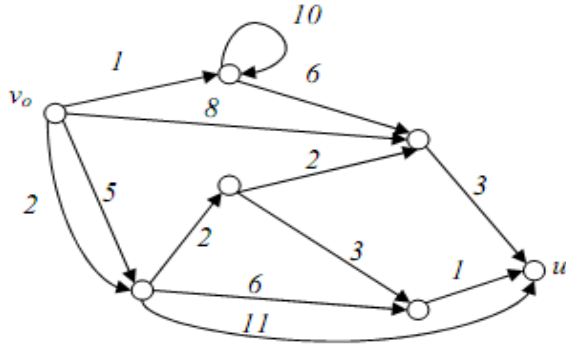
12



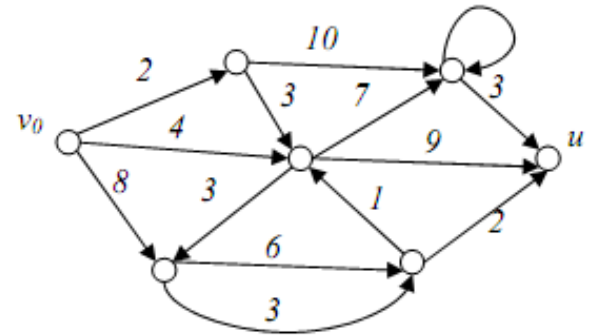
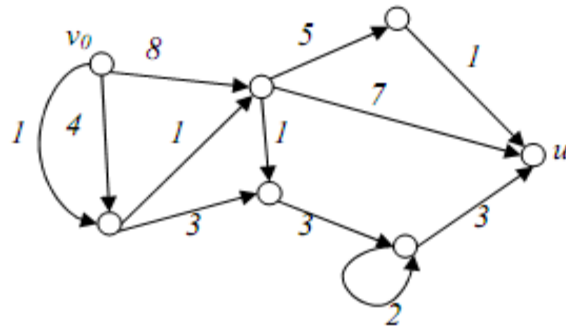
ЗАДАНИЕ 1.2.4

Найти кратчайший путь и его длину из вершины v_0 в вершину u .

1

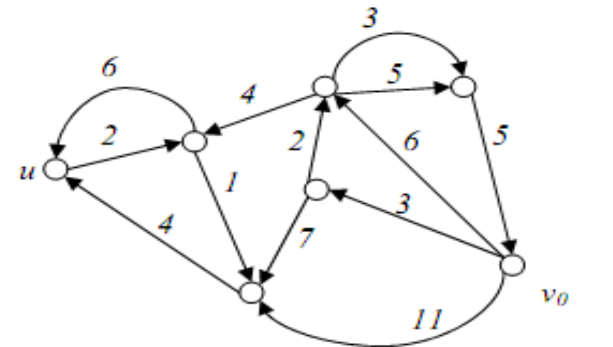
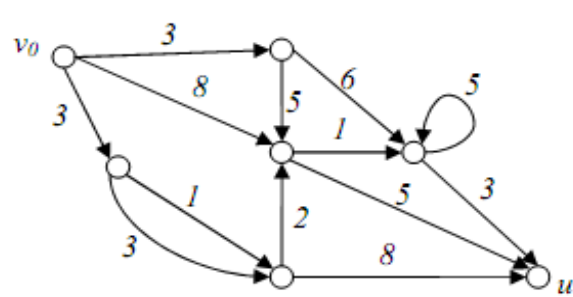


3



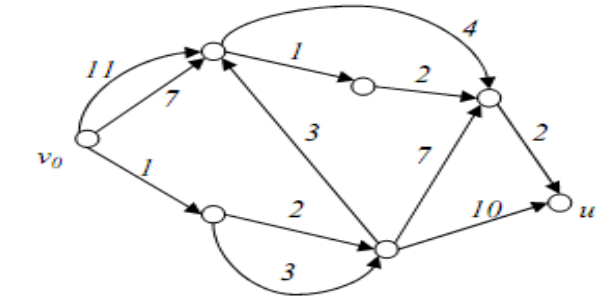
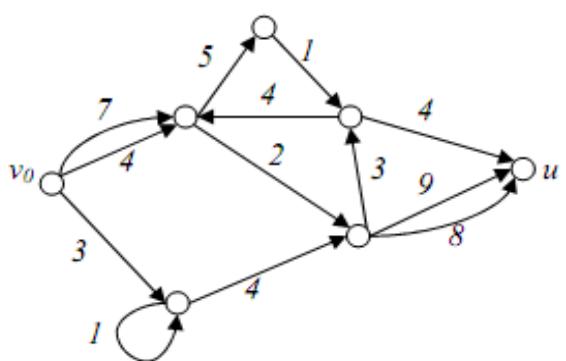
4

5

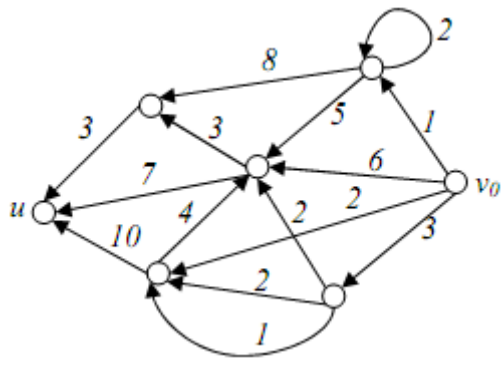


6

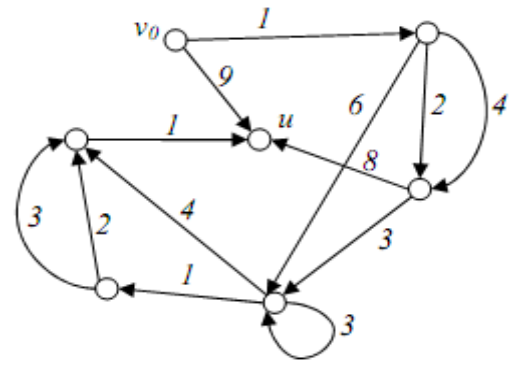
7



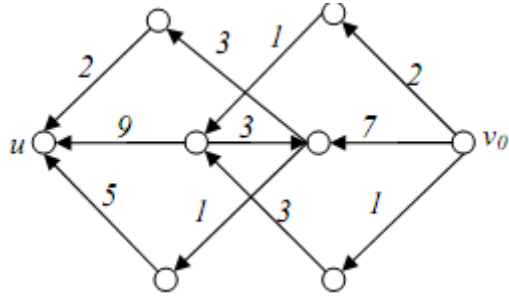
8



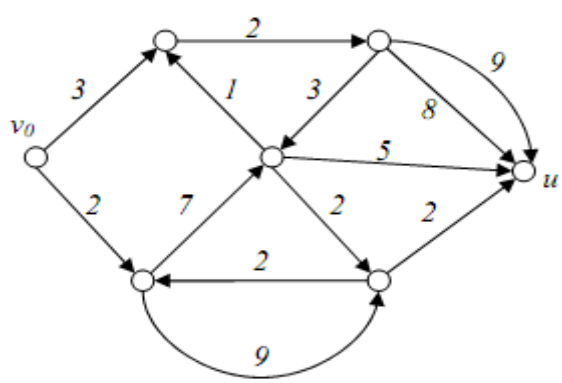
9



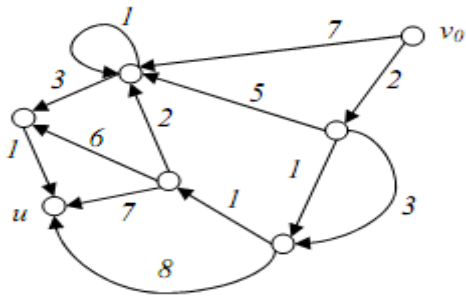
10



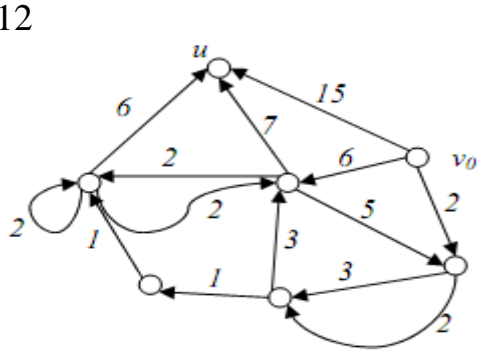
11



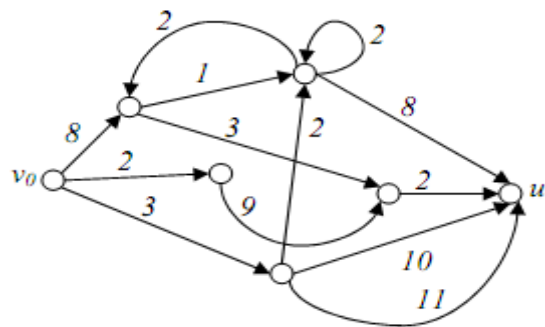
12



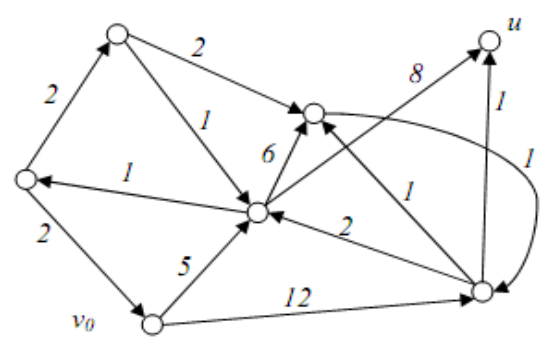
13



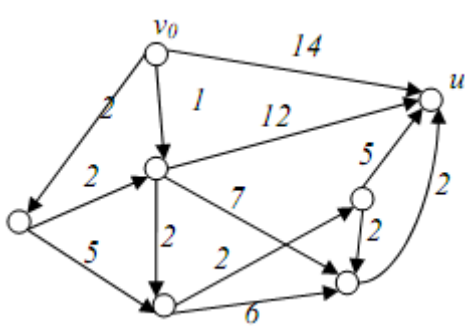
14



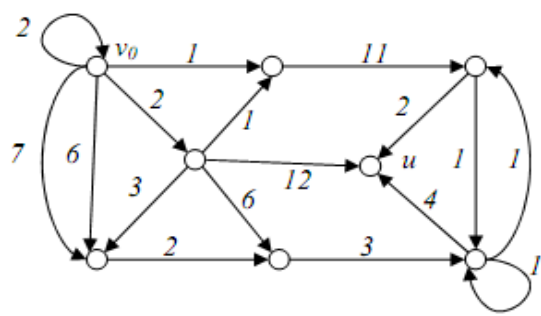
15



16

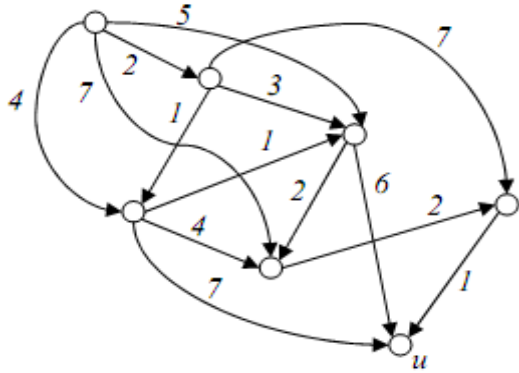


17

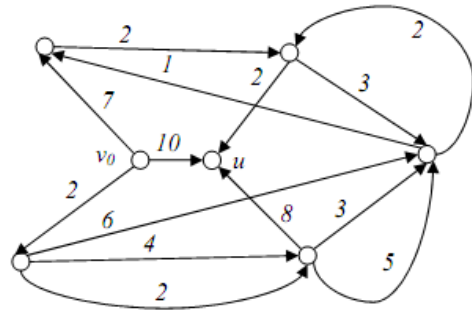


18

19



20



ЗАДАНИЕ 1.2.5

Используя графы из предыдущего задания, предварительно заменив дуги на рёбра, найти минимальное остовное дерево:

для **четных номеров вариантов** - с помощью жадного алгоритма;

для **нечетных номеров вариантов** - с помощью алгоритма ближайшего соседа.

2. КОНЕЧНЫЕ АВТОМАТЫ

2.1. АВТОМАТЫ-ПРЕОБРАЗОВАТЕЛИ

Абстрактным конечным автоматом называется пятерка $A = (X, Y, S, f, g)$, где:

- X - конечное множество, называемое входным алфавитом;
- Y - конечное множество, называемое выходным алфавитом;
- S - конечное множество, называемое алфавитом состояний;
- $f : X \times S \rightarrow S$ функция переходов;
- $g : X \times S \rightarrow Y$ функция выходов.

Произвольные конечные последовательности символов

- алфавита X называются входными словами,
- алфавита Y – выходными словами,
- алфавита S – словами состояний.

Число символов в слове называется его длиной. Допускается существование пустых слов Λ . При функционировании автомата по входному слову x и начальному состоянию s_0 находятся выходное слово y и слово состояний s . Если в автомате выделено начальное состояние s_0 , то он называется инициальным. Инициальный автомат задается шестеркой $A = (X, Y, S, f, g, s_0)$.

Способы задания конечных автоматов.

Конечные автоматы можно задавать с помощью таблиц, определяющих функции f и g , или графически с помощью диаграммы Мура, представляющей собой взвешенный ориентированный мультиграф. При этом:

- вершины графа соответствуют состояниям, дуги - переходам из состояния в состояние;
- веса дуг – пары $\{x_i/y_j\}$, где x_i - входной, а y_j - соответствующий выходной символ при данном переходе.

Если автомат является инициальным, то начальное состояние обычно помечают символом «*».

Обычно выделяют два основных класса задач, решаемых с помощью автоматов:

- 1) распознавание принадлежности входных слов данному множеству (автоматы-распознаватели);
- 2) преобразование входных слов в выходные в соответствии с

основным предназначением автомата (автоматы-преобразователи).

Точно провести границу между этими типами автоматов не всегда оказывается просто.

Приведем пример функционирования автомата-преобразователя. Пусть функция переходов f и функция выходов g заданы таблично:

Состояния \ Входные символы	S_0	S_1^*	S_2
А	$(S_2, 1)$	$(S_1, 0)$	$(S_2, 0)$
Ш	$(S_1, 0)$	$(S_0, 0)$	$(S_0, 1)$
Л	$(S_1, 1)$	$(S_2, 1)$	$(S_0, 0)$

Составим по данной таблице полное описание автомата с помощью кортежа $A_s = (X, S, Y, f, g, s)$.

Входной алфавит: $X = \{А, Ш, Л\}$, алфавит состояний $S = \{s_0, s_1, s_2\}$, выходной алфавит $Y = \{0, 1\}$, начальное состояние s_1 .

функция переходов $f: X \times S \rightarrow S$:

f	s_0	s_1	s_2
А	s_2	s_1	s_2
Ш	s_1	s_0	s_0
Л	s_1	s_2	s_0

функция выходов $g: X \times S \rightarrow Y$:

g	s_0	s_1	s_2
А	1	0	0
Ш	0	0	1
Л	1	1	0

По заданному описанию построим диаграмму Мура (рис.2.1).

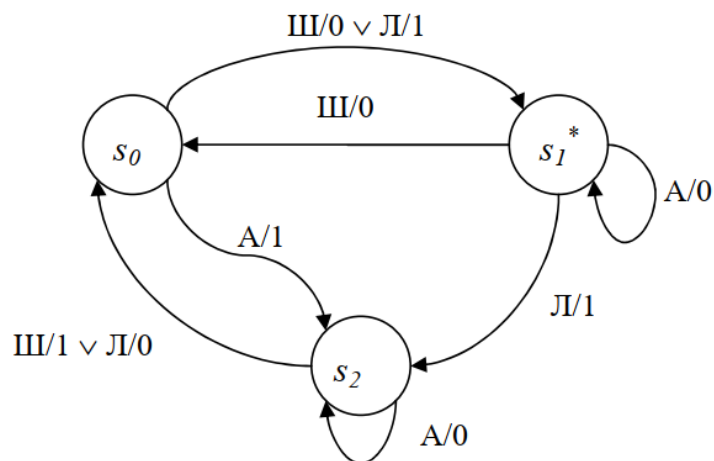


Рис. 2.1. Диаграмма Мура автомата-преобразователя

Для заданного начального состояния автомата s_1 и заданного входного слова $x = ШАЛАШ$ найдем выходное слово y и конечное состояние, в котором будет находиться автомат:

x		Ш	А	Л	А	Ш
y		0	1	0	1	1
s	s_1	s_0	s_2	s_0	s_2	s_0

Конечное состояние - s_0 , выходное слово $y = 01011$.

2.2. АВТОМАТЫ-РАСПОЗНОВАТЕЛИ

Автомат-распознаватель позволяет распознать принадлежность входного слова данному множеству. Его описание аналогично описанию автомата-преобразователя.

Приведем пример построения автомата-распознавателя, осуществляющего подсчет количества троек последовательно следующих символов «К». Для этого необходимо найти функции выходов g и переходов f автомата-распознавателя и построить для него диаграмму Мура.

Пусть входной алфавит представляет собой множество $X = \{\#, K\}$, где $\#$ - любой символ, кроме «К»; выходной алфавит – множество $Y = \{0, 1, 2, \dots, N\}$, где N – максимальная длина входного слова.

Для решения задачи достаточно использовать алфавит состояний, включающий 3 состояния:

- s_0 – начальное состояние или состояние, в котором находится автомат после ввода $\#$;
- s_1 – состояние, в котором находится автомат после ввода «К», если предыдущий символ был $\#$ или пустой;
- s_2 – состояние, в котором находится автомат после ввода не менее двух «К».

Таким образом, алфавит состояний представляет собой множество $S = \{s_0, s_1, s_2\}$.

Из описания состояний вытекает, что функция переходов f имеет следующий вид:

f	s_0	s_1	s_2
$\#$	s_0	s_0	s_0
K	s_1	s_2	s_2

Для описания функции выходов g необходимо описать алгоритм подсчета числа троек n последовательно следующих символов «К». Пер-

вый выходной символ $n=0$. Выражение $n:=n+1$ означает, что текущий выходной символ увеличивается на 1. В дальнейшем используется новое значение n .

Таким образом, функция выходов g имеет вид:

g	s_0	s_1	s_2
$\#$	n	n	n
K	n	n	$n:=n+1$

Полученное описание автомата-распознавателя позволяет построить его диаграмму Мура (рис.2.2).

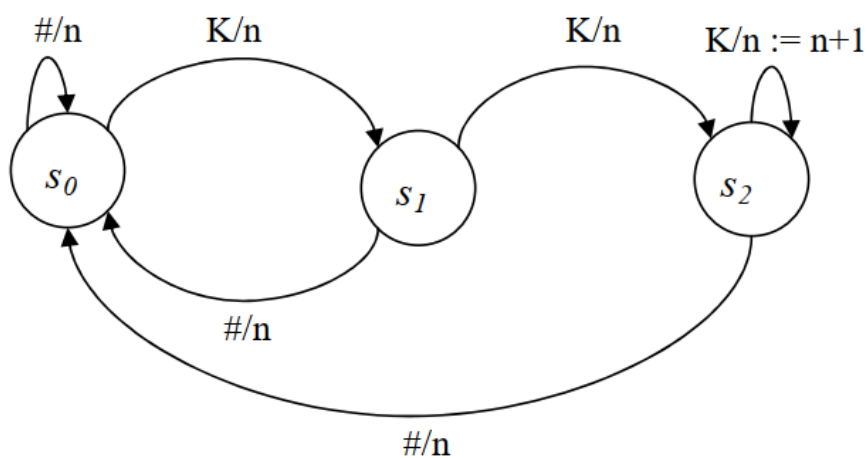


Рис. 2.2. Диаграмма Мура автомата-распознавателя

ЗАДАНИЕ 2.1

Для заданной таблицы конечного абстрактного инициального автомата выполнить следующие действия:

- 1) описать автомат;
- 2) построить диаграмму Мура;
- 3) для заданного начального состояния автомата, отмеченного символом «*» и заданного входного слова x , найти выходное слово y и конечное состояние, в котором будет находиться автомат.

№ варианта	Переходная и выходная функции			
	Состояния	S_0	S_1	S_2
	Входные символы			
1	В	$(S_1, 1)$	$(S_1, 0)$	$(S_2, 0)$
	О	$(S_1, 0)$	$(S_2, 0)$	$(S_0, 1)$
	Л	$(S_0, 1)$	$(S_2, 1)$	$(S_0, 0)$
	Входное слово	ОЛОВО		
2	Т	$(S_2, 1)$	$(S_0, 0)$	$(S_1, 0)$
	О	$(S_0, 0)$	$(S_2, 0)$	$(S_0, 1)$
	Р	$(S_1, 1)$	$(S_2, 1)$	$(S_0, 0)$
	Входное слово	РОТОР		
3	А	$(S_2, 1)$	$(S_1, 0)$	$(S_2, 0)$
	Л	$(S_1, 0)$	$(S_0, 0)$	$(S_0, 1)$
	О	$(S_1, 1)$	$(S_2, 0)$	$(S_1, 0)$
	Входное слово	ЛОЛЛА		
4	М	$(S_1, 1)$	$(S_0, 0)$	$(S_2, 0)$
	А	$(S_2, 0)$	$(S_2, 1)$	$(S_0, 0)$
	Й	$(S_1, 1)$	$(S_2, 1)$	$(S_0, 1)$
	Входное слово	МАМАЙ		
5	П	$(S_0, 1)$	$(S_0, 0)$	$(S_2, 0)$
	Е	$(S_1, 0)$	$(S_0, 0)$	$(S_1, 1)$
	Л	$(S_1, 1)$	$(S_2, 1)$	$(S_2, 0)$

№ варианта	Переходная и выходная функции			
	Состояния	S_0	S_1	S_2
	Входные символы			
	Входное слово	ПЕПЕЛ		
6	В	$(S_2, 0)$	$(S_1, 1)$	$(S_2, 1)$
	А	$(S_1, 0)$	$(S_2, 0)$	$(S_0, 1)$
	Р	$(S_0, 1)$	$(S_2, 1)$	$(S_1, 0)$
	Входное слово	ВАРВАР		
7	Р	$(S_0, 1)$	$(S_0, 0)$	$(S_2, 0)$
	А	$(S_1, 1)$	$(S_2, 0)$	$(S_1, 0)$
	З	$(S_2, 1)$	$(S_0, 1)$	$(S_0, 1)$
	Входное слово	ЗРАЗА		
8	Х	$(S_0, 1)$	$(S_2, 0)$	$(S_2, 1)$
	О	$(S_1, 0)$	$(S_0, 0)$	$(S_0, 1)$
	Т	$(S_2, 1)$	$(S_0, 1)$	$(S_1, 0)$
	Входное слово	ХОХОТ		
9	И	$(S_0, 1)$	$(S_2, 0)$	$(S_1, 0)$
	К	$(S_2, 0)$	$(S_0, 0)$	$(S_1, 1)$
	Р	$(S_1, 1)$	$(S_2, 1)$	$(S_0, 0)$
	Входное слово	КРИКИ		
10	Р	$(S_2, 1)$	$(S_1, 1)$	$(S_1, 0)$
	Ф	$(S_2, 0)$	$(S_0, 0)$	$(S_1, 1)$
	С	$(S_1, 1)$	$(S_2, 0)$	$(S_0, 0)$
	Входное слово	РСФСР		
11	П	$(S_0, 1)$	$(S_1, 0)$	$(S_1, 0)$
	О	$(S_1, 0)$	$(S_0, 1)$	$(S_1, 1)$
	Т	$(S_2, 1)$	$(S_2, 0)$	$(S_0, 0)$

№ варианта	Переходная и выходная функции			
	Состояния	S_0	S_1	S_2
	Входные символы			
Входное слово	ТОПОТ			
12	С	$(S_2, 0)$	$(S_2, 0)$	$(S_2, 0)$
	О	$(S_1, 0)$	$(S_0, 1)$	$(S_1, 1)$
	Р	$(S_2, 1)$	$(S_2, 1)$	$(S_0, 0)$
	Входное слово	СОРОС		
13	Р	$(S_2, 0)$	$(S_2, 0)$	$(S_2, 0)$
	О	$(S_1, 0)$	$(S_1, 0)$	$(S_0, 0)$
	К	$(S_0, 1)$	$(S_2, 1)$	$(S_1, 0)$
	Входное слово	ОКОРОК		
14	О	$(S_2, 0)$	$(S_1, 1)$	$(S_1, 0)$
	В	$(S_1, 1)$	$(S_0, 0)$	$(S_1, 1)$
	Д	$(S_1, 0)$	$(S_2, 1)$	$(S_0, 0)$
	Входное слово	ДОВОД		
15	С	$(S_2, 0)$	$(S_1, 0)$	$(S_1, 0)$
	А	$(S_1, 0)$	$(S_1, 1)$	$(S_1, 1)$
	М	$(S_2, 1)$	$(S_2, 1)$	$(S_0, 1)$
	Входное слово	САМСА		
16	Г	$(S_2, 1)$	$(S_2, 0)$	$(S_1, 1)$
	А	$(S_0, 0)$	$(S_1, 0)$	$(S_0, 1)$
	Н	$(S_1, 1)$	$(S_2, 1)$	$(S_0, 0)$
	Входное слово	НАГАН		
17	Х	$(S_1, 1)$	$(S_1, 1)$	$(S_1, 0)$
	О	$(S_2, 0)$	$(S_2, 0)$	$(S_0, 1)$
	Д	$(S_2, 1)$	$(S_2, 1)$	$(S_2, 0)$

№ варианта	Переходная и выходная функции			
	Состояния	S_0	S_1	S_2
	Входные символы			
	Входное слово	ДОХОД		
18	Д	$(S_1, 1)$	$(S_1, 0)$	$(S_2, 0)$
	А	$(S_2, 1)$	$(S_0, 0)$	$(S_0, 1)$
	Р	$(S_0, 1)$	$(S_2, 0)$	$(S_0, 0)$
	Входное слово	РАДАР		

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- Алексеев В.В. Элементы теории множеств и теории графов: сборник задач и упражнений по курсу «Дискретная математика». – Саров: СГФТИ, 2001. – 30 с.
- Бернштейн Т.В., Храмова Т.В. Практикум по дискретной математике. – Новосибирск: СибГУТИ, 2011. – 131 с.
- Хрипунова Н.Е., Шурыгина Н.Р., Науменко Л.Ф. Дискретная математика: учебное пособие. – СПб.: СПбВВУРЭ (Военный институт), 2006. – 165 с.

Оглавление

ВВЕДЕНИЕ	3
1. ОСНОВЫ ТЕОРИИ ГРАФОВ	3
1.1. Начальные сведения	3
1.1.1. Основные определения и понятия	4
1.1.2. Инварианты графов	6
1.1.3. Эйлеровы и Гамильтоновы пути и циклы	12
1.1.4. Кратчайший путь в графе	15
1.1.5. Построение минимального остова	17
1.2. Задачи для самостоятельного решения	22
Задание 1.2.1	22
Задание 1.2.2	24
Задание 1.2.3	26
Задание 1.2.4	28
Задание 1.2.5	30
2. Конечные автоматы	31
2.1. Автоматы-преобразователи	31
2.2. Автоматы-распознаватели	33
Задание 2.1	35
Библиографический список	38
