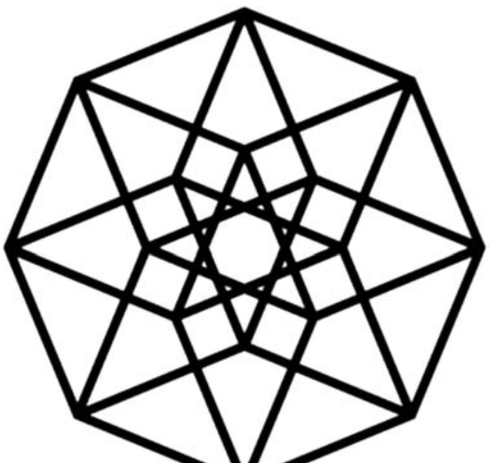
МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Высшая математика»

ДИСКРЕТНАЯ МАТЕМАТИКА

Методические рекомендации к практическим занятиям для студентов направления подготовки 01.03.04 «Прикладная математика» очной формы обучения

Часть 3



Могилев 2022

УДК 519.6
ББК 22.176
Д48

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Высшая математика» «24» февраля 2022 г., протокол № 6

Составители: канд. физ.-мат. наук, доц. И. У. Примак;
канд. физ.-мат. наук, доц. Л. И. Сотская

Рецензент канд. физ.-мат. наук, доц. В. А. Ливинская

Методические рекомендации содержат необходимые для проведения практических занятий теоретическую часть и задачи, а также задания для самостоятельной работы по курсу «Дискретная математика».

Учебно-методическое издание

ДИСКРЕТНАЯ МАТЕМАТИКА

Часть 3

Ответственный за выпуск	В. Г. Замураев
Корректор	И. В. Голубцова
Компьютерная верстка	Е. В. Ковалевская

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 56 экз. Заказ №

Издатель и полиграфическое исполнение:
Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 07.03.2019.
Пр-т Мира, 43, 212022, г. Могилев.

Содержание

1 Практическое занятие № 32. Алфавитное кодирование

На занятии рассматриваются следующие вопросы: основные понятия алфавитного кодирования; таблицы кодов; разделимые схемы; префиксные схемы; неравенство Макмиллана [1–4].

1.1 Теоретическая часть

Алфавитом называется произвольное конечное множество.

Если A – алфавит, то любой элемент $a \in A$ называется ***буквой*** алфавита A.

Словом в алфавите A называется произвольная конечная последовательность букв из A. Множество всех слов в алфавите A обозначается A^*.

Пусть $A = \{a_1, ..., a_n\}$ и $B = \{b_1, ..., b_n\}$ – два алфавита. Функция $F : S \to B^*$, где S – некоторое множество слов в алфавите A ($S \subset A^*$), называется ***кодированием***. Элементы множества S ($\alpha \in S$) называются ***сообщениями***, элементы $\beta = F(\alpha), \beta \in B^*$ – *кодами* (соответствующих сообщений). Обратная функция F^{-1} (если она существует) называется ***декодированием***.

Под алфавитным кодированием понимают схему (таблицу кодов) σ:

$$\sigma = \left\langle a_1 \to \beta_1, ..., a_n \to \beta_n \right\rangle,\ a_i \in A,\ \beta_i \in B^*.$$

Множество кодов букв $V = \{\beta_i\}$ называют множеством ***элементарных кодов*** или просто ***кодом*** алфавита в схеме σ.

Если $\alpha = \alpha_1\alpha_2$, то α_1 называется ***началом***, или ***префиксом***, слова α, а α_2 – ***окончанием***, или ***постфиксом***, слова α. Если при этом $\alpha_1 \neq \Lambda\ (\alpha_2 \neq \Lambda)$, где Λ – пустое слово, то $\alpha_1\ (\alpha_2)$ называется ***собственным началом*** (***собственным окончанием***) слова α. ***Длиной*** слова α называется число букв в нём.

Схема σ называется ***разделимой*** (***однозначной***), если

$$\beta_{i_1}...\beta_{i_k} = \beta_{j_1}...\beta_{j_k} \Rightarrow k = l \text{ и } \forall t \in \{1, ..., k\}\ i_t = j_t.$$

Алфавитное кодирование с разделимой схемой допускает декодирование.

Схема σ называется ***префиксной***, если элементарный код одной буквы не является префиксом элементарного кода другой буквы:

$$(\forall i \neq j\ \ \beta_i, \beta_j \in V) \Rightarrow (\forall \beta \in B^*\ \ \beta_i \neq \beta_j\beta).$$

Теорема. Префиксная схема является разделимой.

Замечание. Свойство быть префиксной является достаточным, но не является необходимым для разделимости схемы.

Теорема. Если схема σ разделима, то длины элементарных кодов $l_i = |\beta_i|$, $i = \overline{1,n}$, удовлетворяют неравенству ***Макмиллана***

$$\sum_{i=1}^{n} \frac{1}{2^{l_i}} \le 1.$$

Теорема. Если числа $l_1, l_2, ..., l_n$ удовлетворяет неравенству Макмиллана, то существует разделимая схема алфавитного кодирования σ.

Замечание. Если числа $l_1, l_2, ..., l_n$ удовлетворяет неравенству Макмиллана, то существует префиксная схема алфавитного кодирования σ с теми же числами.

1.2 Задачи к занятию

1 Пусть $A = \{a_1, ..., a_5\}$ – алфавит языка сообщений, для которого задан код $V = \{10, 12, 012, 101, 2100\}$ в алфавите $B = \{0, 1, 2\}$. Выяснить, является ли слово α кодом некоторого сообщения. В случае положительного ответа, выяснить, является ли α кодом ровно одного сообщения:

а) $\alpha = 10120121012100$; б) $\alpha = 1012101201210012$; в) $\alpha = 0121001210201$; г) $\alpha = 120120121001210$; д) $\alpha = 1010122100$; е) $\alpha = 12101210012$; ж) $\alpha = 101212101012$; з) $\alpha = 1010012100101$.

2 Выяснить, обладает ли код V свойством префикса:

а) $V = \{0, 10, 11, 1110\}$; б) $V = \{01, 11, 10, 001\}$; в) $V = \{1001, 000, 001, 0110, 010, 1000, 0111\}$; г) $V = \{110, 011, 1011, 0100, 11011\}$; д) $V = \{02, 2, 11, 012, 102, 011, 0121\}$; е) $V = \{12, 22, 0011, 101, 0100, 20, 2100\}$.

3 Выяснить, обладает ли схема алфавитного кодирования $\sigma = \langle a \to 111, в \to 10101, г \to 010, л \to 001, о \to 000, с \to 110 \rangle$ свойством префикса. Справедливо ли неравенство Макмиллана для спектра длин заданного кода? С помощью кода, задаваемого схемой:

а) закодируйте слово *слава*; б) закодируйте слово *голоса*; в) декодируйте слово 110001000010; г) декодируйте слово 10101000001000110; д) декодируйте слово 01000000100010101111.

4 Выяснить, является ли код V с кодирующим алфавитом $B = \{0, 1, 2\}$ однозначно декодируемым:

а) $V = \{01, 201, 112, 122, 0112\}$; б) $V = \{0, 01, 0010001001\}$; в) $V = \{001, 021, 102, 201, 001121, 01012101\}$; г) $V = \{20, 01202, 22, 2001, 2012010, 10201121, 1112\}$; д) $V = \{001, 021, 102, 201, 001121, 01012101\}$; е) $V = \{01, 011, 100, 2100, 10110, 00112\}$; ж) $V = \{01, 12, 021, 0102, 10112\}$; з) $V = \{01, 12, 012, 111, 0102, 10112, 01112\}$.

5 Выбрать максимальное по числу элементов подмножество B множества A с условием, что двоичные разложения наименьшей длины чисел из B представляют собой префиксный код:

а) $A = \{1, 5, 6, 7, 12, 13, 17\}$; б) $A = \{1, 3, 6, 8, 10, 13, 19, 33, 37\}$; в) $A = \{2, 6, 7, 9, 12, 15, 18, 35, 36, 37\}$; г) $A = \{2, 3, 7, 8, 11, 12, 13, 14\}$; д) $A = \{1, 2, 5, 8, 9, 10, 13, 15\}$;

е) $A = \{5,7,9,10,12,14,17,23,24\}$; ж) $A = \{4,6,7,10,13,15,20,23\}$; з) $A = \{1,2,5,8,9,12,13,14\}$.

6 С помощью неравенства Макмиллана выяснить, может ли набор чисел L быть набором длин кодовых слов двоичного взаимно однозначного кода:

а) $L = \{1,2,2,3\}$; б) $L = \{2,2,2,4,4,4\}$; в) $L = \{1,2,4,4,4,4\}$; г) $L = \{2,2,3,4,4\}$; д) $L = \{1,2,3,4,4,5\}$.

1.3 Домашнее задание

1 Выяснить, является ли код V с кодирующим алфавитом $B = \{0,1\}$ однозначно декодируемым. Если нет, то привести кодовое слово, допускающее два различных декодирования:

а) $V = \{101,10,100,001,010\}$; б) $V = \{101,10,100,001,010\}$; в) $V = \{101,10,100,001,010\}$; г) $V = \{10,010,0,1001\}$; д) $V = \{10,010,0101,101\}$ е) $V = \{1,10,010,01001\}$; ж) $V = \{1,10,010,01001\}$; з) $V = \{011,01010,01010,010,1010100\}$; и) $V = \{101,010,1,1001\}$; к) $V = \{10,010,01,1001\}$; л) $V = \{100,010,0,1001\}$; м) $V = \{110,011,001,0110,110,0\}$; н) $V = \{01,001,000,1001\}$; о) $V = \{10,01,100,0100,0000\}$; п) $= \{0,101010,01010101\}$.

2 Построить префиксный код с заданным набором длин кодовых слов L:

а) $L = \{2,2,3,3,4,4,4,4\}$; б) $L = \{2,2,2,4,4,4\}$; в) $L = \{2,2,3,4,4\}$; г) $L = \{2,3,3,3,3,4,4\}$; д) $L = \{1,3,3,4,4,4,4\}$; е) $L = \{1,2,3,4,5,6,6\}$; ж) $L = \{2,2,2,4,4,5\}$; з) $L = \{2,3,4,5,5,5,5\}$; и) $L = \{3,3,4,4,4,4\}$; к) $L = \{2,3,3,4\ 4,5,6\}$; л) $L = \{2,3,3,4,4,5,6\}$; м) $L = \{2,3,3,3,4,4,4,4\}$; н) $L = \{1,3,3,4,4,4,5\}$.

3 Выбрать максимальное по числу элементов подмножество B множества A с условием, что двоичные разложения наименьшей длины чисел из B представляют собой однозначно декодируемый код:

а) $A = \{1,5,6,7,12,13,17\}$; б) $A = \{1,3,6,8,10,13,19,33,37\}$; в) $A = \{2,6,7,9,12,15,18,35,36,37\}$; г) $A = \{2,3,7,8,11,12,13,14\}$; д) $A = \{1,2,5,8,9,10,13,15\}$; е) $A = \{5,7,9,10,12,14,17,23,24\}$; ж) $A = \{4,6,7,10,13,15,20,23\}$; з) $A = \{1,2,5,8,9,12,13,14\}$.

2 Практическое занятие № 33. Кодирование с минимальной избыточностью

На занятии рассматриваются следующие вопросы: минимизация длины кода сообщения; цена кодирования; алгоритм Фано; оптимальное кодирование; алгоритм Хаффмена [1–5].

2.1 Теоретическая часть

Пусть $A = \{a_1,...,a_n\}$ и $B = \{b_1,...,b_n\}$ – два алфавита, алфавитное кодирование задано разделимой схемой $\sigma = \langle a_1 \to \beta_1,...,a_n \to \beta_n \rangle$, $a_i \in A$, $\beta_i \in B^*$ и $V = \{\beta_i\}$ –

алфавитный код, заданный схемой σ. Пусть заданы частоты появления символов из алфавита A и, как следствие, их вероятности $p_i = p(a_i)$, удовлетворяющие условиям $p_i > 0$, $i = \overline{1,n}$, $p_1 + p_2 + ... + p_n = 1$. Набор $P = (p_1, ..., p_n)$ называют ***распределением***. Тогда ***средней длиной*** (***ценой***), или ***избыточностью*** кодирования σ при распределении вероятностей P, называют величину $l_\sigma(P) = \sum_{i=1}^{n} p_i l_i$, где $l_i = |\beta_i|$ $(i = \overline{1,n})$.

Разделимая схема $\sigma = \langle a_1 \to \beta_1, ..., a_n \to \beta_n \rangle$, такая что $\forall i\ |\beta_i| = \log_2(n-1) + 1$, называется схемой ***равномерного*** кодирования.

Алфавитное разделимое кодирование σ_o называется кодированием с ***минимальной избыточностью***, или ***оптимальным кодированием***, для распределения вероятностей P, если $l_{\sigma_o}(P) = \inf_\sigma l_\sigma(P)$, где инфинум берется по всем разделимым схемам σ. Код, соответствующий схеме кодирования σ_o, называется ***оптимальным***.

Рассмотрим предложенный Р. Фано приближенный алгоритм построения оптимальных префиксных кодов.

Алгоритм Фано

Это способ построения кода, близкого к оптимальному. Алгоритм состоит из следующих шагов.

1 Список букв алфавита A упорядочивается в порядке невозрастания вероятностей их появления в сообщениях.

2 Список букв делится на две последовательные части с примерно равными суммами вероятностей.

3 Буквам первой части приписывается 0, а буквам второй части – 1.

4 Действия пп. 2 и 3 применяются к каждой части до тех пор, пока все полученные части не станут содержать по одной букве.

5 Каждой букве ставится в соответствие элементарный код, состоящий из 0 и 1, последовательно приписанных частям, содержащим эту букву.

Теорема. Если код с длинами $l_1, l_2, ..., l_n$ является оптимальным для распределения вероятностей $P = \{p_1, p_2, ..., p_n\}$, то оптимальным также будет и код с длинами $l_1, l_2, ..., l_{n-1} - 1$ для распределения вероятностей $P = \{p_1, p_2, ..., p_{n-1} + p_n\}$.

Эта теорема позволяет сводить задачу построения оптимального кода мощности n к задаче построения оптимального кода мощности $n-1$. На основе этой редукции Д. Хаффмен предложил способ построения оптимальных кодов.

Алгоритм Хаффмена

Для построения оптимального кода мощности n достаточно $n-2$ раза повторить следующую процедуру:

1) упорядочить вероятности текущего (исходного) распределения в порядке невозрастания;

2) выделить две наименьшие вероятности, приписав им 0 и 1;

3) заменить две наименьшие вероятности одной, равной их сумме.

После окончания этого процесса остаются две самые большие вероятности, которым также приписываются символы 0 и 1. Это оптимальный код для двухбуквенного алфавита. Далее на основе теоремы редукции из оптимального кода для двух букв строится оптимальный код для трех букв и так далее до построения оптимального кода исходной мощности n.

2.2 Задачи к занятию

1 По заданному распределению вероятностей P построить алфавитное префиксное кодирование по алгоритмам Фано и Хаффмена. Сравнить их избыточность:

а) $P=\{0,55;0,25;0,15;0,05\}$; б) $P=\{0,55;0,25;0,15;0,05\}$; в) $P=\{0,45;0,30;$ $30;0,20;0,05\}$; г) $P=\{0,50;0,25;0,15;0,10\}$; д) $P=\{0,60;0,20;0,15;$ $0,05\}$; е) $P=\{0,40;0,25;0,20;0,15\}$; ж) $P=\{0,40;0,25;0,20;0,15\}$ з) $P=\{0,40;$ $0,25;0,20;0,15\}$; и) $P=\{0,40;0,25;0,20;0,15\}$; к) $P=\{0,35;0,30;0,25;0,10\}$; л) $P=\{0,30;0,25;0,23;0,22\}$; м) $P=\{0,45;0,40;0,10;0,05\}$; н) $P=\{0,50;0,30;$ $0,15;0,05\}$; о) $P=\{0,40;0,25;0,20;0,15\}$; п) $P=\{0,45;0,40;0,10;0,05\}$.

2 Выяснить, является ли код оптимальным для распределения вероятностей $P=\{0,15;0,25;0,05;0,01;0,09;0,25;0,15;0,05\}$:

а) $V=\{001,010,10,11,101,011,0110,00\}$; б) $V=\{0000,001,1001,10010,0111,$ $111,0010,1110\}$; в) $V=\{000,001,010,011,100,101,110,111\}$; г) $V=\{100,00,$ $11110,11111,1110,01,101,110\}$; д) $V=\{000,01,1110,11111,110,10,001,11110\}$; е) $V=\{000,10,0111,01111,011,01,100,11111\}$; ж) $V=\{01,02,10,12,20,21,22,00\}$.

3 Выяснить, существует ли двоичный код с минимальной избыточностью, обладающий заданой последовательностью L длин кодовых слов:

а) $L=\{2,3,3,3\}$; б) $L=\{3,3,3,3\}$; в) $L=\{1,3,3,3,3\}$; г) $L=\{1,2,3,4\}$; д) $L=\{1,2,3,\ 4,4\}$; е) $L=\{1,2,3,4,4,4\}$; ж) $L=\{1,2,3,4,4,4,4\}$; з) $L=\{1,3,3,3,3,$ $3,3\}$; и) $L=\{1,\ 4,4,4,4,4,4,4,4\}$; к) $L=\{3,3,3,3,3,3\}$.

2.3 Домашнее задание

1 По заданному распределению вероятностей P построить алфавитное префиксное кодирование по алгоритмам Фано и Хаффмена. Сравнить их избыточность:

а) $P=\{0,45;0,40;0,10;0,05\}$; б) $P=\{0,45;0,23;0,17;0,15\}$; в) $P=\{0,47;\ 0,36;$ $0,14;\ 0,03\}$; г) $P=\{0,58;\ 0,29;\ 0,12;\ 0,01\}$; д) $P=\{0,58;0,29;\ 0,12;\ 0,01\}$; е) $P=\{0,40;\ 0,28;\ 0,20;\ 0,12\}$; ж) $P=\{0,40;\ 0,28;\ 0,20;\ 0,12\}$; з) $P=\{0,45;\ 0,33;$

0,14; 0,08$\}$; и) $P=\{0,45;\ 0,33;\ 0,14;\ 0,08\}$; к) $P=\{0,40; 0,23; 0,20; 0,17\}$; л) $P=\{0,35;\ 0,26;\ 0,25;\ 0,14\}$; м) $P=\{0,35;\ 0,26;\ 0,25;\ 0,14\}$.

3 Практическое занятие № 34. Помехоустойчивое кодирование

На занятии рассматриваются следующие вопросы: кодирование с исправлением ошибок; возможность исправления всех ошибок; кодовое расстояние; код Хэмминга для исправления одного замещения [1–5].

3.1 Теоретическая часть

Пусть имеется канал связи C, содержащий источник помех: $S \xrightarrow{C} S'$, $S \in A^*$, $S' \in B^*$, где $S-$ множество переданных, а $S'-$ соответствующее множество принятых по каналу сообщений. Кодирование F, обладающее таким свойством, что

$$S \xrightarrow{F} K \xrightarrow{C} K' \xrightarrow{F^{-1}} S, \forall s \in S, F^{-1}(C(F(s))) = s,$$

называется ***помехоустойчивым***, или ***самокорректирующимся***, или кодированием с ***исправлением ошибок***.

Различают ошибки следующих типов: $0 \to 1$, $1 \to 0$ – ошибки замещения разряда; $0 \to \Lambda$, $1 \to \Lambda -$ ошибки выпадения разряда; $\Lambda \to 1$, $\Lambda \to 0$ – ошибки вставки разряда.

Пусть $\alpha = \alpha_1\alpha_2...\alpha_n$ и $\beta = \beta_1\beta_2...\beta_n$ – двоичные слова длины n.

Число $\rho = \sum_{i=1}^{n} |a_i - \beta_i|$ называется ***расстоянием*** Хемминга между α и β. Число ρ равно количеству координат, в которых слова α и β различаются. Если $\rho = 1$, то слова α и β называются ***соседними***.

Множество V называется кодом с растоянием d, если $\min\limits_{\alpha \neq \beta} \rho(\alpha, \beta) = d$. Число $d -$ ***кодовое расстояние*** множества V, обозначаемое как $d(V)$.

Код V ***обнаруживает*** t ошибок, если любое слово γ, получаемое из $\forall \alpha \in V$ в результате не более t ошибок, не принадлежит V.

Теорема. Код V ***обнаруживает*** t ошибок тогда и только тогда, когда кодовое расстояние $d(V) \geq t+1$.

Теорема. Код V ***исправляет*** t ошибок тогда и только тогда, когда кодовое расстояние $d(V) \geq 2t+1$.

Код V обнаруживает $d(V)-1$ ошибку и исправляет $[(d(V)-1)/2]$ ошибок.

Коды Хэмминга

Коды Хемминга являются самоконтролирующимися кодами. Для их построения достаточно приписать к каждому слову добавочные (контрольные) двоичные разряды и выбрать цифры в этих разрядах так, чтобы общее количество единиц в изображении любого числа было, например, четным. Одиночная ошибка (будем говорить об ошибке замещения) в каком-либо разряде передаваемого слова (в том числе, может быть, и в контрольном разряде) изменит четность общего количества единиц. В этом случае счетчики по модулю 2, подсчитывающие количество единиц, которые содержатся среди двоичных цифр числа, могут давать сигнал о наличии ошибок.

Кодирование. Закодируем слово $\alpha = \alpha_1\alpha_2...\alpha_m$ словом $\beta = \beta_1\beta_2...\beta_{m+k}$, где k – количество контрольных разрядов, которое выбирается так, чтобы удовлетворялось неравенство $2^k \geq m+k+1$. Расположение контрольных разрядов в коде Хемминга осуществляется в позициях, номера которых являются степенями двойки (1, 2, 4, 8, 16…). Информационные разряды кода занимают оставшиеся свободные позиции. Контрольные разряды p_i определяются следующим образом: $p_i = \beta_{2^i+1} \oplus \beta_{2^i+2} \oplus ...$, где сумма берется по всем β_j $(2^i < j \leq m+k)$, у которых в разложении индекса j коэффициент при 2^i равен 1.

Пример – Построим код Хэмминга для слова $\alpha = 1011$. Здесь $m = 4$. Согласно неравенству возьмем $k = 3$. Построим кодовое слово $\beta = \beta_1\beta_2\beta_3\beta_4\beta_5\beta_6\beta_7 = p_0 p_1 1 p_2 001$.

$$p_0 = \beta_1 = \beta_3 \oplus \beta_5 \oplus \beta_7 = 0,\ p_1 = \beta_2 = \beta_3 \oplus \beta_6 \oplus \beta_7 = 0,\ p_2 = \beta_4 = \beta_5 \oplus \beta_6 \oplus \beta_7 = 1.$$

Итак, кодовое слово $\beta = 0011001$.

Декодирование. Пусть получено слово, записанное с использованием кода Хэмминга, $\beta = \beta_1\beta_2...\beta_n$ $(n = m+k)$, в котором имеет место одна ошибка замещения. Необходимо определить разряд, в котором присутствует ошибка, и восстановить исходное слово. Определим число информационных разрядов как $m = \left[\log_2\left(\dfrac{2^n}{n+1}\right)\right]$. По слову $\beta = \beta_1\beta_2...\beta_n$ ищем k сумм вида $\upsilon_i = \beta_{2^i} \oplus \beta_{2^i+1} \oplus ...$, $i = 0,...,k-1$, где в i-сумму входят все β_j, $2^i \leq j \leq m+k$, у которых в разложении индекса j коэффициент при 2^i равен 1. Номер разряда ошибки определяется как $\sum\limits_{0 \leq i < k} \upsilon_i 2^i$. Восстановление исходного слова предполагает инвертирование разряда с ошибкой и удаление контрольных разрядов в полученном слове.

Пример – Декодируем $\beta = 1001110$ для $n = 7, m = 4, k = 3$. Запишем

$$\upsilon_0 = \beta_1 \oplus \beta_3 \oplus \beta_5 \oplus \beta_7 = 0,\ \upsilon_1 = \beta_2 \oplus \beta_3 \oplus \beta_6 \oplus \beta_7 = 1,\ \upsilon_2 = \beta_4 \oplus \beta_5 \oplus \beta_6 \oplus \beta_7 = 1.$$

Номер разряда ошибки $\sum\limits_{i=0}^{2} \upsilon_i 2^i = 0 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2 = 6$. Следовательно, $\alpha = \alpha_1\alpha_2\alpha_3\alpha_4 = \beta_3\beta_5\overline{\beta_6}\beta_7 = 01000$.

3.2 Задачи к занятию

1 Сколько информационных символов содержится в коде, исправляющем одиночную ошибку при числе информационных комбинаций $N = 32$?

2 Определить избыточность корректирующего кода при общем числе кодовых комбинаций $N = 256$.

3 Для каждого множества B найти кодовое расстояние, число ошибок, которые код обнаруживает, и число ошибок, которые код исправляет:

а) $B = \{11000, 10101, 01110, 10110\}$; б) $B = \{00110, 11100, 01110, 10010\}$; в) $B = \{1100, 1010, 0110, 1011, 0101\}$; г) $B = \{110, 010, 011, 101, 111, 001, 100\}$; д) $B = \{110, 010, 011, 101, 111, 001, 100\}$; е) $B = \{100, 110, 011, 101, 010, 111\}$; ж) $B = \{111100, 110011, 001111\}$; з) $B = \{101010, 010110, 000011\}$; и) $B = \{01101010, 11000110, 00011001\}$; к) $B = \{10101100, 11100110, 10011101\}$; л) $B = \{0110, 1010, 1100, 0110, 0001, 1001\}$; м) $B = \{1010, 1100, 1110, 0110, 1001, 1101\}$.

4 Построить код Хэмминга для сообщения α:

а) $\alpha = 10011$; б) $\alpha = 11001$; в) $\alpha = 11001$; г) $\alpha = 10110$; д) $\alpha = 01011$; е) $\alpha = 01101$; ж) $\alpha = 11010$; з) $\alpha = 11011$; и) $\alpha = 01111$; к) $\alpha = 11110$; л) $\alpha = 11111$; м) $\alpha = 01110$; н) $\alpha = 10111$; о) $\alpha = 10010$; п) $\alpha = 10101$; р) $\alpha = 11100$; с) $\alpha = 11101$; т) $\alpha = 01100$; у) $\alpha = 01010$.

5 По каналу, допускающему одну ошибку замещения, была получена последовательность β, построенная по методу Хэмминга. Восстановить исходное сообщение α:

а) $\beta = 101110$; б) $\beta = 011110$; в) $\beta = 110011$; г) $\beta = 1001011$; д) $\beta = 0101101$; е) $\beta = 101110110$; ж) $\beta = 101001101$; з) $\beta = 1011101$; и) $\beta = 10111011$; к) $\beta = 10100110$; л) $\beta = 1101110$; м) $\beta = 1010110$; н) $\beta = 010110$; о) $\beta = 100001$; п) $\beta = 0110100$; р) $\beta = 1010010$; с) $\beta = 0100110$.

3.3 Домашнее задание

1 Для каждого множества B найти кодовое расстояние, число ошибок, которые код обнаруживает, и число ошибок, которые код исправляет:

а) $B = \{101, 001, 010, 110, 000, 011\}$; б) $B = \{00111, 01010, 10001, 01001\}$; в) $B = \{11001, 00011, 10001, 01101\}$; г) $B = \{01010, 11101, 00110, 10011\}$; д) $B = \{01100, 00010, 11001, 10101\}$; е) $B = \{100100, 010010, 111010, 001101\}$.

2 Построить макет кода Хэмминга, определить значения корректирующих разрядов для кодовой комбинации 00101 кода Хаффмеана.

3 Дана последовательность 10011010. Закодировать кодом Хэмминга.

4 Закодировать сообщение habr кодом Хэмминга, имея следующее бинарное представление: h – 01000100, a – 00111101, b – 00111110, r – 01001000 (примечание – исходное сообщение надо разбить на два блока по 16 бит).

5 Пользуясь кодом Хэмминга, найти ошибку в сообщении 1111 1011 0010 1100 1101 1100 110.

4 Практическое занятие № 35. Сжатие данных

На занятии рассматриваются следующие вопросы: сжатие текстов; предварительное построение словаря; алгоритм Лемпела – Зива [1].

4.1 Теоретическая часть

Методы кодирования, которые позволяют построить (без потери информации) коды сообщений, имеющие меньшую длину по сравнению с исходным сообщением, называются методами ***сжатия*** (или ***упаковки***) данных. Качество сжатия определяется ***коэффициентом сжатия***, который обычно измеряется в процентах и показывает, на сколько процентов кодированное сообщение короче исходного. Эти методы предполагают следующий алгоритм кодирования:

1) исходное сообщение по некоторому алгоритму разбивается на последовательности символов, называемых ***словами*** (слово может иметь одно или несколько вхождений в исходный текст сообщения);

2) полученное множество слов считается буквами нового алфавита. Для этого алфавита строится разделимая схема алфавитного кодирования (равномерного или оптимального). Полученная схема называется ***словарем***, т. к. она сопоставляет слову код;

3) при декодировании исходное сообщение восстанавливается путем замены кодов слов на слова из словоря.

Алгоритм Лемпела – Зива

В этом алгоритме используется следующая идея, которая известна также как ***адаптивное сжатие***. За один проход по тексту одновременно динамически строится словарь и кодируется текст. Вначале словарь содержит пустое слово, имеющее код 0. Далее в тексте последовательно выделяются слова. Выделяемое слово – это максимально длинное слово из имеющихся в словаре плюс еще один символ. В сжатое представление записываются найденный код слова и расширяющая буква, а словарь пополняется расширенной комбинацией.

При использовании алгоритма Лемпела – Зива словарь не хранится – за счет того, что при декодировании используется этот же алгоритм построения словаря, словарь динамически восстанавливается.

4.2 Задачи к занятию

Выполнить кодирование фразы с помощью метода Лемпеля – Зива LZ77:

а) кукуркукурекурекун; б) упупапекапекаупуп; в) лорлоралоранранлоран; г) пропронепронепрнепрона ; д) какатанекатанекатата; е) менменаменаменатеп;

ж) долделдолдилделдил; з) sarsalsarsanlasanl; и) kloklonkolonklonkl; к) tertrektekertektrek; л) bigbonebigborebigbo; м) commercommecommerce; н) comconcomconacom; о) webwerbweberweberweb; п) porpoterpoterporter; р) mantopmentopomantomen; с) roporopoterropoterter; т) webwerbweberweberweb; у) sionsinossionsinos; ф) mantopmentopomantomen.

4.3 Домашнее задание

1 Выполнить кодирование фразы с помощью метода Лемпеля – Зива LZ77. Рассчитать количество информации, передаваемой при равномерном кодировании кодом ASCII (8 бит на символ) и словарном кодировании LZ77. Сравнить полученные значения:

а) Richard's wretched ratchet wrench; б) 2 She slits the sheet she sits on; в) How much wood could a woodchuck chuck; г) She sees seas slapping shores; д) Six stick shifts stuck shut; е) The batter with the butter is the batter that is better; ж) Clowns grow glowing crowns; з) Blake's black bike's back brake bracket block broke; и) Peter Piper picked a peck of pickled peppers; к) Rubber baby buggy bumpers.

5 Практическое занятие № 36. Шифрование

На занятии рассматриваются следующие вопросы: шифрование с помощью случайных чисел; шифрование с открытым ключом [1].

5.1 Теоретическая часть

Шифрование – это кодирование данных с целью защиты от несанкционированного доступа. Область знаний о шифрах, методах их создания и раскрытия называется ***криптографией***.

Процесс кодирования называется ***шифрованием***, процесс декодирования – ***расшифрованием***. Кодируемое сообщение называется ***шифровкой***, метод кодирования – ***шифром***. Основное требование к шифру состоит в том, чтобы расшифрование было возможно только при наличии санкции, т. е. некоторой дополнительной информации (или устройства), которая называется ***ключом*** шифра. Процесс декодирования шифровки без ключа называется ***дешифрованием***.

Рассмотрим шифрование с помощью случайных чисел.

Определение. Под операцией $a \bmod b$ понимают определение остатка, возникающего при целочисленном делении a на b.

Пусть имеется ***генератор псевдослучайных чисел***, работающий по определенному алгоритму, в частности: $T_{i+1} = (a \cdot T_i + b) \bmod c$, где T_i – предыдущее псевдослучайное число; T_{i+1} – последующее псевдослучайное число; коэффициенты a, b, c постоянны и известны.

Процесс шифрования определяется следующим образом. Шифруемое сообщение представляется в виде последовательности слов $S_0, S_1, \ldots$, каждое длиной n, которые складываются по модулю два со словами последовательности $T_0,\ T_1, \ldots,$ т. е. $C_i = S_i \oplus T_i$.

Последовательность $T_0,\ T_1, \ldots$ называется ***гаммой шифра***. Процесс расшифровывания заключается в том, чтобы еще раз сложить шифрованную последовательность с той же самой гаммой шифра: $S_i = C_i \oplus T_i$.

Ключом такого шифра является значение T_0.

Шифры, в которых для шифрования и расшифрования используется один и тот же ключ, называются ***симметричными***.

Рассмотрим шифры с ***открытым ключом***. Эти шифры не являются симметричными – для шифрования и расшифрования используются разные ключи. При этом ключ, используемый для шифрования, является открытым (не секретным) и может быть сообщен всем желающим отправить шифрованное сообщение, а ключ, используемый для расшифрования, является закрытым и хранится в секрете получателем шифрованных сообщений. Даже знание всего зашифрованного сообщения и открытого ключа, с помощью которого оно было зашифровано, не позволяет дешифровать сообщение без знания закрытого ключа.

Определение. Число a ***сравнимо по модулю*** n с числом b (обозначение: $a \equiv b(\bmod n)$, если a и b при делении на n дают один и тот же остаток:

$$a \equiv b(\bmod n) \Leftrightarrow a \bmod n = b \bmod n.$$

Определение. Числа называются взаимно простыми, если их наибольший общий делитель равен единице.

Шифрование с открытым ключом производится следующим образом.

1 Получателем сообщений производится генерация открытого ключа (пара чисел n и e) и закрытого ключа (число d).

Для этого:

– выбираются два простых числа: p и q;

– определяется первая часть открытого ключа: $n = pq$;

– определяется вторая часть открытого ключа – выбирается небольшое нечетное число e, взаимно простое с числом $(p-1)(q-1)$;

– определяется закрытый ключ: $d \equiv e^{-1} \bmod (p-1)(q-1)$.

После этого открытый ключ (числа n и e) сообщается всем отправителям сообщений.

2 Отправитель шифрует сообщение (разбивая его, если нужно, на слова S_i длиной менее $\log_2 n$ разрядов): $C_i \equiv (S_i)^e \bmod n$, и отправляет получателю.

3 Получатель расшифровывает сообщение с помощью закрытого ключа d: $P_i \equiv (C_i)^d \bmod n$.

Пример – Генерация ключей: $p = 3, q = 11$; $n = pq = 3 \cdot 11 = 33$; $e = 7$; $d \equiv 7^{-1} \bmod 20 = 3$.

Пусть $S_1 = 3$, $S_2 = 1$, $S_3 = 2$ ($S_1, S_2, S_3 < n = 33$). Тогда код определяется так: $C_1 = 3^7 \bmod 33 = 9$; $C_2 = 1^7 \bmod 33 = 1, C_3 = 2^7 \bmod 33 = 29$.

При расшифровке имеем следующее: $P_1 = 9^3 \bmod 33 = 3$; $P_2 = 1^3 \bmod 33 = 1$; $P_3 = 29^3 \bmod 33 = 2$.

5.2 Задачи к занятию

1 Решить сравнения:

а) $33x \equiv 6 \bmod 69$; б) $1287x \equiv 447 \bmod 516$; в) $3x \equiv 7 \bmod 11$; г) $11x \equiv 21 \bmod 15$; д) $12x \equiv 5 \bmod 21$; е) $(58x) \bmod 47 = 87$.

2 Зашифровать слово ШИФРОВКА с помощью симметричного ключа, если $T_0 = 86, a = 9$, $b = 13, c = 256$.

3 Зашифровать слово АЛФАВИТ на следующей гамме: ИДФНТВХ.

4 Зашифровать и расшифровать слово КРИПТОЛОГИЯ с помощью открытого ключа, используя следующие параметры: $p = 5, q = 17, e = 13$.

5 Известен открытый ключ: $p = 11, q = 7, e = 13$. Найти закрытый ключ d.

6 Зашифровать открытый текст TO BE OR NOT TO BE THAT IS THE QUESTION, используя секретный ключ RELATIONS.

7 Известен ключ 1001001110111101110. Закодировать последовательность 1001110011100111001 для передачи по каналу связи.

8 Шифр Цезаря. Этот шифр реализует следующее преобразование текста: каждая буква исходного текста заменяется следующей после нее буквой в алфавите, который считается написанным по кругу. Используя этот шифр, зашифровать текст «Красна птица перьями, а человек – знанием».

9 Шифр Виженера. Используя в качестве ключевого слова слово ЛЕТО, закодировать слова ИНФОРМАТИКА, ШКОЛА.

10 Шифр Атбаш. Некоторые фрагменты библейских текстов зашифрованы с помощью шифра, который назывался Атбаш. Правило шифрования состояло в замене i – й буквы алфавита буквой с номером $n - i + 1$, где n – число букв в алфавите (значит, первая буква заменяется последней, вторая – предпоследней и т. д.). Закодировать слова СТУПА, ГОРОД, ПАРК.

5.3 Домашнее задание

1 Решить сравнения:

а) $24x \equiv 6 \bmod 81$; б) $27x \equiv 12 \bmod 15$; в) $91x \equiv 26 \bmod 169$; г) $28x \equiv 56 \bmod 49$; д) $91x \equiv 26 \bmod 169$.

2 Известен ключ 1111001101110010110. Раскодировать последовательность 1001110011100111001, принятую по каналу связи.

3 Известна переданная последовательность 1111100000, известна принятая последовательность 0101010101. Найти ключ, при помощи которого было закодировано сообщение.

4 Используя шифр Цезаря, декодировать следующие фразы: ИОБОЙЁ – РПМПГЙОБ ФНЯ.

5 Слова МСЦЮЭОЁЩ, ЮЧОЫЧЁ получены с помощью шифра Виженера с ключевым словом ЛЕТО. Восстановить исходные слова.

6 Шифр «Перестановки». Кодирование осуществляется перестановкой букв в слове по одному и тому же общему правилу. Восстановить слова и определить правило перестановки: РОПЕФТЬЛ, АКСРЕН.

6 Практическое занятие № 37. Определения графов. Элементы графов

На занятии рассматриваются следующие вопросы: основные определения графов: смежность, диаграммы, орграфы, псевдографы, мультиграфы и гиперграфы, изоморфизм графов, подграфы, валентность, маршруты, цепи, циклы, связность, расстояние между вершинами, ярусы и диаметр графа, эксцентриситет и центр [1, 6–12].

6.1 Теоретическая часть

Графом $G = (V,\, E)$ называется совокупность двух множеств – непустого множества, V (множества ***вершин***) и множества E неупорядоченных пар различных элементов множества V ($E-$ множество ***ребер***).

Пусть v_1, v_2 – вершины, $e = (v_1, v_2)-$ соединяющее их ребро. Тогда вершина v_1 и ребро e ***инцедентны***, вершина v_2 и ребро e также ***инцедентны***. Два ребра, инцедентные одной вершине, называются ***смежными***; две вершины, инцедентные одному ребру, также называются ***смежными***.

Множество вершин, смежных с вершиной v, называется множеством смежности вершины v и обозначается $\Gamma(v)$.

Граф изображают ***диаграммой***, в которой вершины изображаются точками (или кружками), ребра – линиями.

Если элементами множества E являются ***упорядоченные пары***, то граф называется ***ориентированным*** (или ***орграфом***). В этом случае элементы множества V называются ***узлами***, а элементы множества E – ***дугами***.

Если элементом множества E может быть пара ***одинаковых*** (не различных) элементов V, то такой элемент множества E называется петлей, а граф называется ***графом с петлями*** (или ***псевдографом***).

Если E является не множеством, а ***набором***, содержащим несколько одинаковых элементов, то эти элементы называются ***кратными ребрами***, а граф называется ***мультиграфом***. Если элементами множества E являются не обязательно двухэлементные, а любые подмножества множества V, то такие элементы множества E называются ***гипердугами***, а граф называется ***гиперграфом***.

В дальнейшем под выражением «граф $G=(V,\ E)$» будем понимать неориентированный граф без петель и кратных ребер.

Два графа $G_1=(V_1,\ E_1)$ и $G_2=(V_2,\ E_2)$ называют ***изоморфными*** (обозначается $G_1 \sim G_2$), если существует биекция $h: V_1 \to V_2$, сохраняющая смежность:

$$e_1=(u,v)\in E_1 \Rightarrow e_2=\big(h(u),h(v)\big)\in E_2,\ e_2=(u,v)\in E_2 \Rightarrow e_1=\big(h^{-1}(u),h^{-1}(v)\big)\in E_1.$$

Граф $G'=(V',\ E')$ называется ***подграфом*** $G=(V,\ E)$ (обозначается $G'\subset G$), $V'\subset V$ и/или $E'\subset E$. В случае $V'=V$, G' называется ***остовным подграфом*** G. Если $V'\subset V\ \&\ E'\subset E\ \&(V'\neq V \vee E'\neq E)$, то граф G' называется ***собственным*** подграфом графа G.

Подграф $G'=(V',\ E')$ называется ***правильным*** подграфом графа $G=(V,\ E)$, если G' содержит всевозможные ребра G.

Количество ребер, инцидентных вершине v, называется ***степенью*** (или ***валентностью***) вершины v и обозначается $d(v)$: $d(v)=|\Gamma(v)|$.

Если степени всех вершин равны k, то граф называется ***регулярным*** степени k.

Если степень вершины равна 0 ($d(v)=0$), то вершина называется ***изолированной***. Если степень вершины равна 1 ($d(v)=1$), то вершина называется ***концевой***, или ***висячей***.

Для орграфа число дуг, из вершины υ, называется полустепенью исхода (обозначается $d^-(v)$), а входящих – полустепенью захода (обозначается $d^+(v)$).

***Теорема* (*Эйлера*).** Сумма степеней вершин графа равна удвоеному количеству ребер: $\sum\limits_{v\in V} d^-(v)+\sum\limits_{v\in V} d^+(v)=2q$ (q – число ребер).

Маршрутом в графе называется чередующася последовательность вершин и ребер $v_0,e_1,v_1,e_2,v_2,...,e_k,v_k$, в которой любые два соседних элемента инцидентны. Если $v_0=v_k$, то маршрут называют ***замкнутым***, иначе – ***открытым***.

Если все ребра различны, то маршрут называется ***цепью***. Если все вершины (а значит, и ребра) различны, то маршрут называется ***простой цепью***. В цепи $v_0,e_1,v_1,e_2,v_2,...,e_k,v_k$ вершины v_0 и v_k называются ***концами*** цепи. Цепь, соединяющая вершины u и v, обозначается $\langle u,v\rangle$.

Замкнутая цепь называется ***циклом***. Граф без циклов называется ***ациклическим***. Для орграфов цепь называется ***путем***, а цикл – ***контуром***.

Длиной маршрута называется количество ребер в нем (с повторениями).

Растоянием между вершинами u и v (обозначается $d(u,v)$) называется длина кратчайшей цепи $\langle u,v\rangle$.

Диаметр графа, он определяется как наибольшее расстояние между вершинами.

Эксцентриситет вершины – это расстояние от нее до самой удаленной вершины. Диаметр графа, таким образом, – это максимальный из эксцентриситетов его вершин. Минимальный из эксцентриситетов вершин называется ***радиусом*** графа.

Вершина графа, имеющая минимальный эксцентриситет, называется ***центральной***. Иначе говоря, центральная вершина – это такая, эксцентриситет которой равен радиусу графа. ***Центр*** графа – множество всех его центральных вершин.

Множество вершин, находящихся на одинаковом расстоянии n от вершины v, называется ***ярусом***.

Маршрут называют ***эйлеровым***, если он замкнут и проходит через каждое ребро графа только по одному разу.

Маршрут называют ***гамильтоновым***, если он замкнут и проходит через каждую вершину графа только по одному разу.

Говорят, что вершины в графе связаны, если существует соединяющая их (простая) цепь. Граф, в котором все вершины связаны, называется ***связанным***.

Отношение связанности вершин является эквивалентностью. Классы эквивалентности по отношению связанности называются ***компонетами связаности*** графа. Число компонент связанности графа G обозначается $k(G)$.

Граф является связанным тогда и только тогда, когда $k(G)=1$. Если $k(G)>1$, то $G-$ ***несвязный*** граф.

6.2 Задачи к занятию

1 В графе n вершин и m ребер. Сколько у него:

а) остовных подграфов; б) порожденных подграфов?

2 Определить число графов с n вершинами, в которых допускаются ребра следующих типов:

а) неориентированные и петли; б) ориентированные и петли; в) ориентированные, но не петли.

3 Определить число ориентированных графов с n вершинами, в которых каждая пара различных вершин соединена:

а) не более чем одним ребром; б) точно одним ребром.

4 Выяснить, существуют ли графы с набором степеней:

а) (0,2,2,3,3); б) (2,2,2,3,3); в) (2,2,3,3,3); г) (0,1,2,3,4).

5 При каких n существуют графы с n вершинами, каждая из которых имеет степень 3? степень 4?

6 Определить число графов с n вершинами, в которых:

а) данные k вершин являются изолированными; б) нет изолированных вершин (примените метод включения и исключения).

7 Определить среди следующих графов изоморфные (рисунок 1).

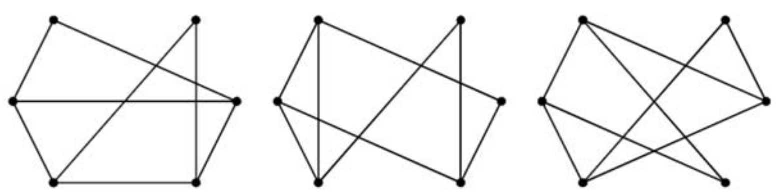


Рисунок 1 – Графы к задаче 7

8 Перечислить все попарно неизоморфные графы:

а) с четырьмя вершинами; б) с шестью вершинами и тремя ребрами; в) с шестью вершинами и 13 ребрами.

9 Перечислить все попарно неизоморфные ориентированные графы без петель с тремя вершинами и тремя ребрами.

10 Указать номера вопросов, на которые было отвечено «да». Возможен ли однородный граф, в котором:

а) пять вершин и степень каждой вершины равна 3; б) шесть вершин и степень каждой из них равна 4; в) четыре вершины и шесть ребер; г) пять нечетных вершин и шесть ребер; д) семь вершин и степень каждой вершины равна 5; е) шесть вершин и девять ребер; ж) восемь вершин и степень каждой из них равна 3?

11 Степень вершины полного графа равна 7. Из графа удалили несколько ребер так, что степень каждой вершины получившегося частичного графа стала равной 5. Сколько ребер удалили? Сколько ребер осталось?

12 Найти степень вершины полного графа, имеющего 91 ребро.

13 В однородном графе степень вершины равна 5. Число ребер равно 35. Найти число вершин.

14 В графе G пять вершин v_i $(i=\overline{1,5})$, причем $d(v_1)=d(v_2)$, а степени остальных вершин различны и не совпадают с $d(v_1)$. Найти $d(v)$.

15 Доказать, что граф с 45 вершинами и 950 ребрами связен.

6.3 Домашнее задание

1 Выяснить, существуют ли графы с набором степеней:

а) (0; 2; 2; 3; 2); б) (1; 1; 2; 2; 2); в) (1; 2; 3; 4; 5); г) (1; 1; 1; 3; 3).

2 Графы, изображенные на рисунке 2, разбить на классы попарно неизоморфных графов.

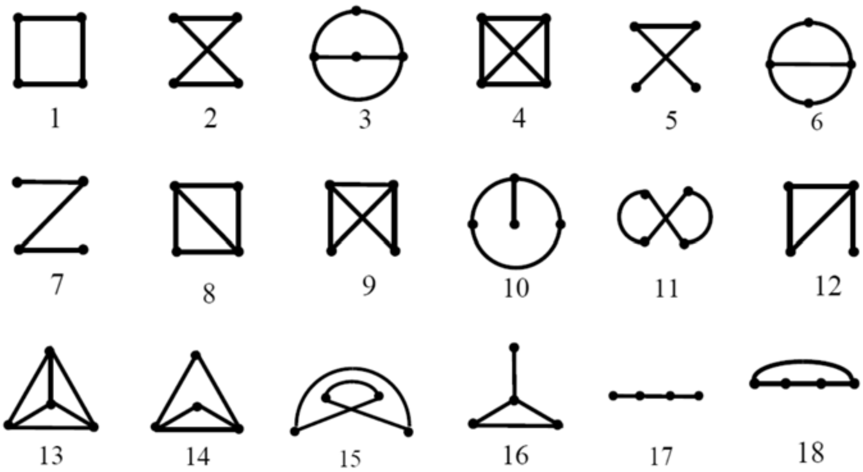


Рисунок 2 – Графы к задаче 2

3 Может ли существовать граф с 15 вершинами, степень каждой из которых равна 5?

4 Найти диаметр графа G , заданного множеством вершин $V\left(G\right)=\{1,2,3,$ $4,5,6,7,8\}$ и множеством рёбер $E\left(G\right)=\{\left(1,\ 2\right),\left(1,\ 4\right),\left(2,\ 4\right),\left(2,\ 5\right),\left(3,\ 6\right),\left(4,\ 7\right),$ $\left(4,8\right),\left(5,\ 6\right),\left(5,\ 8\right)\}$.

5 В графе G семь вершин v_i $(i=\overline{1,7})$ и $d(v_1)=6$, $d(v_2)=5$, $d(v_3)=d(v_4)=3$, $d(v_5)=d(v_6)=2$, $d(v_7)=1$. Какие вершины смежны v_3?

6 Построить граф, центр которого:

а) состоит ровно из одной вершины; б) состоит из двух вершин; в) состоит из трех вершин и не совпадает с множеством всех вершин; г) совпадает с множеством всех вершин.

7 Доказать, что граф с 53 вершинами, каждая из которых имеет степень не менее 26, связен.

7 Практические занятия № 38 и 39. Операции над графами. Представление графов

На занятии расматриваются следующие вопросы: виды графов; двудольные графы; направленные орграфы и сети; операции над графами; матрицы смежности и инциденций; списки смежности; массив дуг; обходы графов; алгоритм Фалкерсона (графический метод упорядочения элементов графа); матричный способ упорядочения вершин графа [1, 6–11].

7.1 Теоретическая часть

Граф, состоящий из одной вершины, называется ***тривиальным***.

Граф, в котором каждая пара вершин смежна, называется ***полным***.

Двудольным граф – это граф $G=\left(V,\ E\right)$, такой что множество V разбито на два непересекающихся множества V_1 и V_2, причем всякое ребро из E инциденто вершине из V_1 и вершине из V_2. Множества V_1 и V_2 называются ***долями*** двудольнего графа.

Теорема. Граф является двудольным тогда и только тогда, когда все его простые циклы имеют четную длину.

Если в графе ориентировать все ребра, то получится орграф, который называется направленым. Направленный орграф, полученный из полного графа, называется ***турниром***.

Если в орграфе для вершины $d^+(v)=0$, то такая вершина называется ***источником***, в случае $d^-(v)=0$, то вершина – ***стоком***. Направленный орграф с одним источником и одним стоком называется ***сетью***.

Дополнением графа $G=\left(V,\ E\right)$ называется граф вида $\overline{G}(\overline{V},\overline{E})$, где $\overline{V}=V$, а $\overline{E}=\{(u,v):u,v\in V,(u,v)\notin E\}$. Дополнение графа содержит все вершины исходного графа, а вершины в нем смежны тогда и только тогда, когда они не смежны в исходном графе.

Объединением (***наложением***) графов $G_1 = (V_1,\ E_1)$ и $G_2 = (V_2,\ E_2)$ называют граф вида $G = G_1 \cup G_1 = (V,\ E)$, в котором $V = V_1 \cup V_2$; $E = E_1 \cup E_2$.

Сложением (***соединением***) графов $G_1 = (V_1,\ E_1)$ и $G_2 = (V_2,\ E_2)$ называется граф вида $G = G_1(V_1,\ E_1) + G_2(V_2,\ E_2) = (V,\ E)$, в котором $V = V_1 \cup V_2$ и $E = E_1 \cup E_2 \cup \{(u,v) : u \in V_1, v \in V_2\}$. Соединение двух графов содержит все вершины и ребра исходных графов, к тому же каждая вершина первого графа смежна с каждой вершиной второго.

Пересечением двух графов $G_1 = (V_1,\ E_1)$ и $G_2 = (V_2,\ E_2)$ называют граф вида $G = G_1 \cap G_1 = (V,\ E)$, где $V = V_1 \cap V_2$; $E = E_1 \cap E_2$.

Произведением двух графов $G_1 = (V_1,\ E_1)$ и $G_2 = (V_2,\ E_2)$ называют граф вида $G = G_1 \times G_2 = (V,\ E)$, в котором $V = V_1 \times V_2$, а множество ребер определяется следующим образом: вершины (u_1, u_2) и (v_1, v_2) смежны тогда и только тогда, когда либо $u_1 = v_1$ и $(u_2, v_2) \in E_2$, либо $u_2 = v_2$ и $(u_1, v_1) \in E_1$.

Удаление ребра $e \in E_1$ из графа $G_1 = (V_1,\ E_1)$ дает граф $G = G_1 - e = (V,\ E)$, в котором $V = V_1$; $E = E_1 \setminus \{e\}$.

Добавление ребра $e \notin E_1$ к графу $G_1 = (V_1,\ E_1)$ дает граф $G = G_1 + e = (V,\ E)$, в котором $V = V_1$; $E = E_1 \cup \{e\}$.

Удаление вершины $v \in V_1$ из графа $G_1 = (V_1,\ E_1)$ дает граф $G = G_1 - v = (V,\ E)$, в котором $V = V_1 \setminus \{v\}$; $E = E_1 \setminus \{e = (v_1, v_2) : v_1 = v \vee v_2 = v\}$.

Добавление вершины $v \notin V_1$ к графу $G_1 = (V_1,\ E_1)$ дает граф $G = G_1 + v = (V,\ E)$, в котором $V = V_1 \cup \{v\}$; $E = E_1$.

Пусть дан граф $G = \left(V,\ E\right)$.

Матрица смежности – квадратная матрица порядка n, где $n -$ количество вершин графа, такая что $M(G) = (m_{ij})$ и

$$m_{ij} = \begin{cases} 1, \text{ если } (i, j) \in E, \\ 0, \text{ если } (i, j) \notin E. \end{cases}$$

Матрица инцидентности – прямоугольная матрица, в которой число строк равно количеству вершин, а число столбцов – количеству ребер, причем $H(G) = (h_{ij})$ и

$$h_{ij} = \begin{cases} 1, \text{ если вершина } i \text{ и ребро } j \text{ инцидентны}, \\ 0, \text{ если вершина } i \text{ и ребро } j \text{ не инцидентны}. \end{cases}$$

Для орграфа матрица инцидентности имеет особенности:

$$h_{ij} = \begin{cases} 1, \text{ если ребро } j \text{ заходит в вершину } i, \\ 0, \text{ если вершина } i \text{ и ребро } j \text{ не инцидентны}, \\ -1, \text{ если ребро } j \text{ исходит из вершины } i. \end{cases}$$

Списком ребер (***масивом дуг***) будем называть таблицу, состоящую из двух столбцов, в левом из которых перечисляются все ребра $e_i \in E$, а в правом – инцидентные им вершины; для неорентированного графа порядок вершин в строке произвольный, для орграфа первым стоит номер начала ребра.

В ***списке смежности*** каждому номеру вершины сопоставляется список номеров смежных с ней вершин (или ссылок на эти вершины). Каждому ребру соответствует один элемент этого списка.

Под ***упорядочиванием вершин*** связного орграфа без контуров понимают такое разбиение его вершин на группы, при котором:

1) вершины первой группы не имеют предшествующих (таких вершин может быть несколько и они еще называются минорантами), а вершины последней – последующих (таких вершин также может быть несколько, их называют мажорантами);

2) вершины любой другой группы не имеют предшествующих в следующей группе;

3) вершины одной и той же группы дугами не соединяются.

Аналогичным образом вводится понятие упорядочивания дуг. В результате упорядочивания элементов получают граф, изоморфный данному. Упорядочивание элементов выполняют графическим или матричным способом.

Графический способ упорядочивания вершин **(*алгоритм Фалкерсона*)**.

Шаг 1. Находят вершины графа, в которые не входит ни одна дуга. Они образуют первую группу. Нумеруют вершины группы в натуральном порядке: 1, 2 … . При этом присвоение номеров вершинам внутри группы может быть сделано не единственным образом, что не имеет значения.

Шаг 2. Мысленно вычеркивают все пронумерованные вершины и дуги, из них выходящие. В получившемся графе найдется, по крайней мере, одна вершина, в которую не входит ни одна дуга. Этой вершине, входящей во вторую группу, присваивают очередной номер и т. д.

Этот шаг повторяют до тех пор, пока все вершины не будут упорядочены (пронумерованы). Аналогичным образом упорядочивают дуги орграфа.

Матричный алгоритм упорядочивания вершин (выделение уровней) орграфа без контуров основывается на анализе матрицы смежности орграфа [4–6].

7.2 Задачи к занятиям

1 Полный граф имеет 105 ребер. Найти число его вершин.

2 Каково наибольшее число ребер в двудольном графе с вершинами n?

3 В полном двудольном графе содержится 119 ребер. Найти величины $|V_1|$ и $|V_2|$, если известно, что $|V_1| > 1$, $|V_2| > 15$.

4 В связном двудольном графе $|V_1| = 7$, $|V_2| = 10$. Сколько ребер содержит граф, если при удалении любого ребра граф становится несвязным?

5 Для неорентированных графов $G_1 = (V_1, E_1)$ и $G_2 = (V_2, E_2)$, где $V_1 = \{v_1, ..., v_4\}$, $V_2 = \{v_2, ..., v_6\}$, $E_1 = \{(v_1, v_2), (v_1, v_3), (v_2, v_3), (v_2, v_4), (v_3, v_4)\}$, $E_2 = \{(v_2, v_3), (v_2, v_4), (v_2, v_6), (v_3, v_4), (v_3, v_6), (v_4, v_5), (v_5, v_6)\}$, найти и изобразить графически:

а) $\overline{G}_1$ и $\overline{G}_2$; б) $G_1 \cup G_2$; в) $G_1 \cap G_2$; г) $G_1 + G_2$; д) $G_1 \times G_2$; е) $G_2 - v_4$; ж) $G_2 - \{v_4, v_5\}$.

6 Дан орграф $G = (V, E)$, где $V = \{v_1, ..., v_6\}$, $E = \{(v_1, v_2), (v_1, v_3), (v_1, v_5), (v_2, v_4), (v_2, v_5), (v_3, v_5), (v_3, v_6), (v_4, v_3), (v_4, v_6), (v_6, v_5)\}$. Составить:

а) матрицу смежности вершин; б) матрицу инциденций; в) изобразить граф графически; г) матрицу смежности дуг.

7 Для графов G_1 и G_2, представленных матрицами смежности

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix} \text{ и } \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix} \text{ соответственно, найти:}$$

а) $\overline{G}_1$ и $\overline{G}_2$; б) $G_1 \cup G_2$; в) $G_1 \cap G_2$.

8 Изобразить графы, соответствующие представленным матрицам смежности вершин. Составить матрицы инцидентности этих графов. Указать степени вершин графа. Найти радиус, диаметр, центр графа.

$$\text{а) } \begin{pmatrix} 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}; \text{ б) } \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}; \text{ в) } \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 2 & 2 \\ 0 & 0 & 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 2 & 0 & 1 \end{pmatrix}.$$

9 Неорентированный граф, представленный матрицей инцидентности

$$\begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}:$$

а) изобразить; б) указать степени вершин графа; в) найти длину пути из вершины v_2 в v_5, указать все цепи, соединяющие вершины v_2 в вершину v_5; г) построить простой цикл, содержащий вершину v_4.

10 Изоморфны ли графы, заданные следующими списками смежности:

а) 1) 2, 3, 4; 2) 1, 4, 5; 3) 1, 5; 4) 1, 2; 5) 2, 3; б) 1) 2, 3; 2) 1, 3, 4, 5; 3) 1, 2; 4) 2, 5; 5) 2, 4.

11 Дан орграф $G=(V,\ E)$, где $V=\{v_1,\ldots,v_6\}$ – множество вершин, E – множество дуг. Упорядочить вершины орграфа и построить изоморфный граф:

а) $E=\{(v_1,v_2),(v_1,v_3),(v_1,v_5),(v_2,v_4),(v_2,v_5),(v_3,v_5),(v_3,v_6),(v_4,v_3),(v_4,v_6),(v_6,v_5)\}$; б) $E=\{(v_1,v_2),(v_1,v_3),(v_1,v_6),(v_2,v_3),(v_2,v_5),(v_3,v_4),(v_3,v_6),(v_5,v_3),(v_5,v_4),(v_6,v_4)\}$; в) $E=\{(v_1,v_6),(v_1,v_4),(v_1,v_5),(v_2,v_6),(v_2,v_5),(v_2,v_1),(v_2,v_3),(v_3,v_1),(v_5,v_4),(v_5,v_6)\}$.

12 Для орграфов задачи 10 указать степени вершин графа, найти радиус, диаметр и центр.

7.3 Домашнее задание

1 Дополнение полного двудольного графа содержит 31 ребро. Найти $|V_1|$ и $|V_2|$.

2 Указать номера вопросов, на которые надо ответить «да»:

а) может ли двудольный граф содержать петли; б) верно ли, что нуль–граф, содержащий семь вершин, является двудольным; в) является ли двудольным граф, содержащий одну вершину; г) входит ли пустой граф в множество двудольных графов; д) может ли быть двудольным простой граф, содержащий 35 ребер; е) во всяком ли полном двудольном графе существует гамильтонов цикл; ж) существует ли двудольный граф, содержащий замкнутую эйлерову цепь; з) существуют ли связные двудольные графы, в которых все вершины множества V_1 являются четными, а все вершины множества V_2 – нечетными ?

3 Найти $G_1 \cup G_2$, $G_1 \cap G_2$, $G_1 \times G_2$ для следующих графов (рисунок 3).

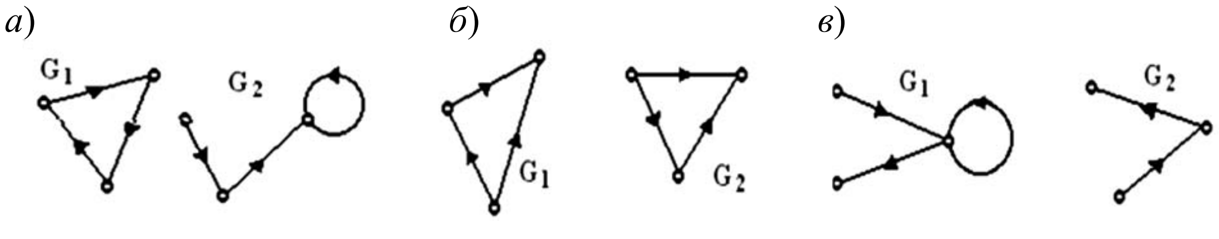


Рисунок 3 – Графы к задаче 3

4 Изобразить графы, соответствующие представленным матрицам инцидентности:

$$
\text{а)}\begin{pmatrix}1&0&1&1\\0&0&1&0\\1&0&0&1\\0&0&0&0\\0&1&0&0\end{pmatrix};\
\text{б)}\begin{pmatrix}1&0&0&1&1&1&0\\1&1&0&0&0&0&0\\0&1&1&1&0&0&1\\0&0&0&0&0&0&0\\0&0&0&0&1&1&0\end{pmatrix};\
\text{в)}\begin{pmatrix}-1&0&1&1\\0&0&-1&0\\1&0&0&-1\\0&0&0&0\\0&1&0&0\end{pmatrix};\
\text{г)}\begin{pmatrix}1&0&0&1&1&-1&0\\-1&-1&0&0&0&0&0\\0&1&1&-1&0&0&1\\0&0&0&0&0&0&0\\0&0&0&0&-1&1&0\end{pmatrix}.
$$

5 Показать изоморфизм двух графов на рисунке 4.

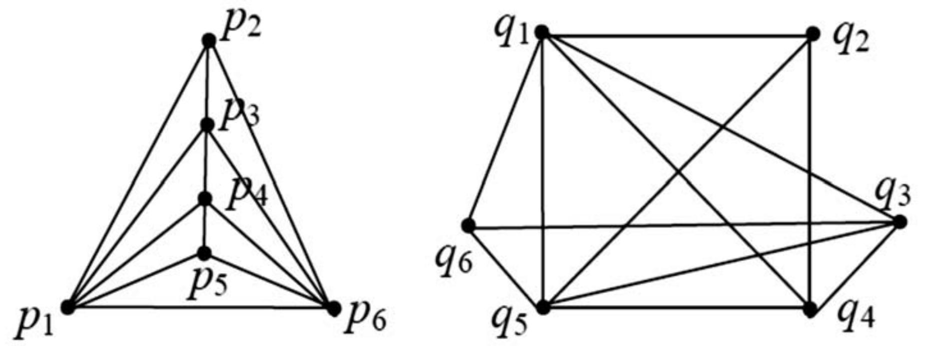


Рисунок 4 – Графы к задаче 5

6 Показать изоморфизм двух графов в задаче 5 с помощью матриц инцидентности.

7 Определить среди графов, заданных следующими списками смежности, изоморфные:

а) 1) 2, 3, 5; 2) 1, 3; 3) 1, 2, 4, 5, 6; 4) 3, 6; 5) 1, 3; 6) 3, 4; б) 1) 2, 3, 4, 5, 6; 2) 1, 6; 3) 1, 6; 4) 1, 5; 5) 1, 4; 6) 1, 2, 3; в) 1) 2, 5; 2) 1, 3, 4, 5, 6; 3) 2, 4; 4) 2, 3; 5) 1, 2, 6; 6) 2, 5.

8 Дан орграф $G=(V,\ E)$, где $V=\{v_1,...,v_6\}$ – множество вершин, E – множество дуг. Упорядочить вершины орграфа и построить изоморфный граф:

а) $E=\{(v_2,v_4),(v,v_1),(v_4,v_1),(v_5,v_1),(v_5,v_2),(v_5,v_4),(v_5,v_6),(v_6,v_1),(v_6,v_2),$ $(v_6,v_3)\}$; б) $E=\{(v_1,v_3),(v_1,v_4),(v_1,v_5),(v_2,v_1),(v_2,v_5),(v_2,v_6),(v_4,v_3),(v_5,v_4),$ $(v_6,v_1),(v_6,v_5)\}$; в) $E=\{(v_1,v_4),(v_2,v_4),(v_2,v_5),(v_3,v_1),(v_3,v_2),(v_3,v_6),(v_5,v_4),$ $(v_6,v_1),(v_6,v_5),(v_6,v_2)\}$.

9 Для орграфов задачи 8 указать степени вершин графа, найти радиус, диаметр и центр.

8 Практическое занятие № 40. Потоки в сетях

На занятии расматриваются следующие вопросы: определение потока; теорема Форда – Фалкерсона; алгоритм нахождения максимального потока [1, 6–12].

8.1 Теоретическая часть

Сетью называют ориентированный связный граф $G(V,E)$ с множеством вершин V и множеством дуг E, где каждой дуге $(v_i,v_j)\in E$ придается числовая характеристика r_{ij}, называемая ***пропускной способностью***. В G выделяют две фиксированные вершины – s и t; s называют ***истоком***, t – ***стоком***, а остальные вершины – ***промежуточными***. Пропускная способность дуги $r_{ij} \geq 0$, и если $(v_i, v_j) \notin E$, то $r_{ij} = 0$.

Множество чисел $X = \{x_{ij}\}$ называют ***потоком по сети***, а сами x_{ij}, определенные на дугах $(v_i, v_j) \in E$, – ***потоками в дугах***, если выполняются следующие условия:

1) $x_{ij} = -x_{ji}$;

2) $x_{ij} \leq r_{ij}$ (поток в дуге (v_i, v_j) не может превышать пропускной способности этой дуги);

3) $\sum_{j=1}^{n} x_{ij} = 0 \left(v_i \neq s, t\right)$ (условие сохранения потока в промежуточных вершинах).

Если $x_{ij} < r_{ij}$, то дуга (v_i, v_j) называется ***ненасыщенной***, если же $x_{ij} = r_{ij}$ – ***насыщенной***.

Суммарный поток из истока s равен суммарному потоку в сток t, т. е.

$$f = \sum_{j=1}^{n} x_{kj} = \sum_{i=1}^{n} x_{il} \left(v_k = s, v_l = t\right),$$

где v_j – конечные вершины дуг, исходящих из s;

v_i – начальные вершины дуг, входящих в t. Функцию f называют ***мощностью потока*** на сети.

Разрез сети $S / \overline{S}$ – множество дуг, для которых выполняются следующие требования:

1) $s \in S$; $t \in \overline{S}$;

2) $S \cap \overline{S} = \varnothing$;

3) $S \cup \overline{S} = V$.

Пропускная способность разреза $R\left(S / \overline{S}\right)$ равна сумме пропускных способностей r_{ij} всех дуг, входящих в разрез.

Поток через разрез $X\left(S / \overline{S}\right)$ равен сумме потоков x_{ij} по всем дугам, входящим в разрез.

***Теорема Форда – Фалкерсона*:** максимальный поток по заданной сети равен минимальной пропускной способности разреза, отделяющего s от t.

Алгоритм расчета максимального потока

1 Выписываем полный путь из истока в сток, определяя при этом пропускную способность. Пропускная способность равна наименьшему из значений пропускных способностей дуг, составляющих этот путь. Над каждой дугой пути записываем

поток, равный пропускной способности данного пути. Если поток и пропускная способность совпадают, то ***дуга насыщенная***, выделяем её.

2 Строим другие полные пути и считаем их пропускную способность до тех пор, пока сток достигаем из истока по ненасыщенным дугам. Если какая-либо дуга пути входит в другой путь, то пропускная способность данной дуги равна первоначальной пропускной способности минус поток по другому пути. К потоку данной дуги прибавляем поток по новому пути. Выделяем все насыщенные дуги данного пути.

3 Определяем множество S как исток s и вершины, достигаемые из истока по ненасыщенным дугам. Тогда множество $\overline{S}$ образует остальные вершины сети. Выделяем разрез на сети (т. е. множество дуг сети (v_i, v_j) таких, что $v_i \in S$, а $v_j \in \overline{S}$).

4 Считаем мощность потока сети f и пропускную способность разреза $R\left(S/\overline{S}\right)$. В соответствии с теоремой Форда – Фалкерсона эти величины равны друг другу.

Пример – Рассчитать максимальный поток в сети (рисунок 5) от истока $v_1 = s$ до стока $v_9 = t$.

Решение

Запишем полные пути от истока к стоку, их пропускную способность и потоки:

$L_1 = (v_1, v_2, v_5, v_9)$, $R(L_1) = \min\{r_{12}, r_{25}, r_{59}\} = \min\{4,5,7\} = 4$, $x_{12} = x_{25} = x_{59} = 4$;

$L_2 = (v_1, v_4, v_8, v_9)$, $R(L_2) = \min\{r_{14}, r_{48}, r_{89}\} = \min\{8,3,8\} = 3$, $x_{14} = x_{48} = x_{89} = 3$;

$L_3 = (v_1, v_3, v_7, v_9)$, $R(L_3) = \min\{r_{13}, r_{37}, r_{79}\} = \min\{2,9,4\} = 2$, $x_{13} = x_{37} = x_{79} = 2$;

$L_4 = (v_1, v_4, v_6, v_9)$, дуга (v_1, v_4) входит в путь L_2, поэтому $R(L_4) = \min\{r_{14} - x_{14}, r_{46}, r_{69}\} = \min\{8-3, 5, 7\} = 5$, $x_{14} = 3 + 5 = 8$, $x_{46} = x_{69} = 5$.

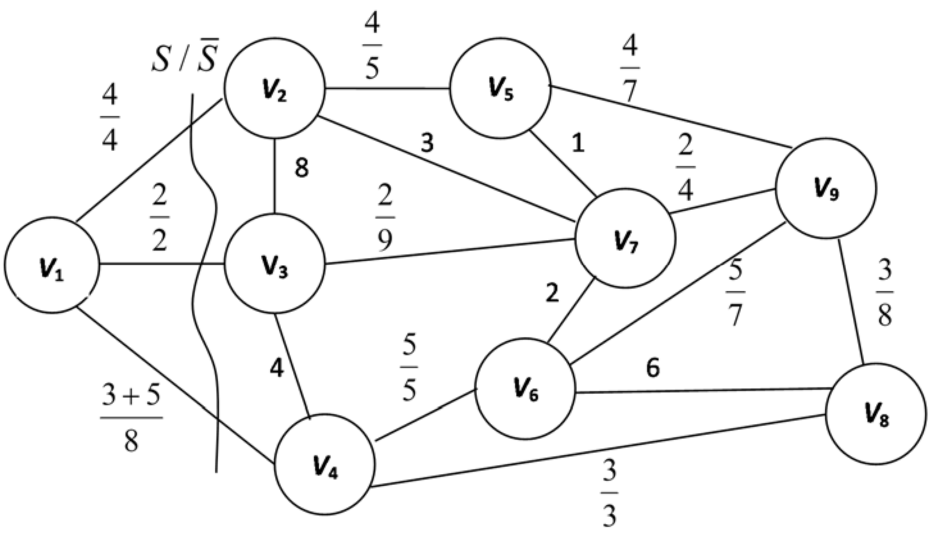


Рисунок 5 – Сеть

Здесь $S=\{v_1\}$, $\overline{S}=\{v_2,v_3,v_4,v_5,v_6,v_7,v_8,v_9\}$. Разрез определяем как $S/\overline{S}=\{(v_1,v_2),(v_1,v_3),(v_1,v_4)\}$. Мощность потока $f=x_{12}+x_{13}+x_{14}=$ $=4+2+8=14$. Пропускная способность разреза $R\left(S/\overline{S}\right)=r_{12}+r_{13}+r_{14}=$ $=4+2+8=14$.

8.2 Задачи к занятию

Для сети, пропускные способности r_{ij} которой указаны в таблице 1, найти максимальный поток от истока $v_1=s$ до стока $v_7=t$.

Таблица 1 – Пропускная способность дуг сети

Номер варианта	$r_{ij}^{\ i,j}$													
1	$18^{1,2}$	$16^{1,3}$	$9^{1,6}$	$8^{2,3}$	$11^{2,4}$	$7^{2,5}$	$13^{2,7}$	$13^{3,5}$	$19^{3,7}$	$10^{4,3}$	$15^{4,6}$	$17^{5,4}$	$28^{5,6}$	$14^{6,7}$
2	$12^{1,2}$	$9^{1,3}$	$11^{1,6}$	$12^{2,3}$	$7^{2,4}$	$12^{3,4}$	$10^{3,5}$	$18^{3,7}$	$12^{4,5}$	$15^{4,6}$	$6^{4,7}$	$8^{5,2}$	$15^{5,7}$	$4^{6,7}$
3	$9^{1,2}$	$11^{1,4}$	$17^{1,6}$	$6^{2,3}$	$8^{2,5}$	$12^{2,7}$	$7^{3,7}$	$5^{4,2}$	$5^{4,6}$	$4^{4,7}$	$7^{5,7}$	$9^{6,7}$		
4	$10^{1,2}$	$5^{1,3}$	$8^{1,6}$	$3^{2,4}$	$3^{2,5}$	$4^{2,7}$	$4^{3,4}$	$5^{3,5}$	$10^{3,6}$	$4^{4,5}$	$9^{4,7}$	$5^{5,6}$	$6^{5,7}$	$7^{6,7}$
5	$5^{1,3}$	$15^{1,5}$	$9^{1,6}$	$6^{2,5}$	$7^{2,7}$	$3^{3,2}$	$4^{3,4}$	$7^{3,6}$	$8^{4,6}$	$3^{4,7}$	$9^{5,6}$	$18^{5,7}$	$5^{6,7}$	
6	$10^{1,2}$	$9^{1,5}$	$26^{1,7}$	$7^{3,2}$	$11^{3,2}$	$8^{4,3}$	$12^{4,6}$	$11^{4,7}$	$13^{5,4}$	$10^{5,6}$	$14^{6,3}$	$8^{6,7}$		
7	$10^{1,2}$	$8^{1,4}$	$8^{2,3}$	$12^{2,4}$	$10^{2,5}$	$6^{2,7}$	$5^{3,6}$	$11^{3,7}$	$4^{4,5}$	$12^{4,6}$	$5^{5,3}$	$9^{5,7}$	$6^{6,5}$	$7^{6,7}$
8	$10^{1,3}$	$18^{1,4}$	$8^{1,5}$	$6^{2,3}$	$11^{2,5}$	$15^{2,6}$	$19^{2,7}$	$12^{3,5}$	$13^{3,7}$	$5^{4,7}$	$18^{5,4}$	$7^{5,6}$	$9^{6,7}$	
9	$7^{1,2}$	$9^{1,3}$	$8^{1,6}$	$11^{2,3}$	$14^{2,4}$	$10^{2,5}$	$6^{2,7}$	$9^{3,5}$	$11^{3,6}$	$19^{3,7}$	$12^{4,6}$	$8^{5,4}$	$14^{5,6}$	$10^{6,7}$
10	$5^{1,2}$	$8^{1,3}$	$9^{2,5}$	$17^{2,6}$	$6^{3,2}$	$10^{3,5}$	$24^{3,7}$	$15^{4,5}$	$6^{4,7}$	$8^{5,6}$	$12^{5,7}$	$7^{5,7}$		
11	$15^{1,3}$	$12^{1,4}$	$11^{1,6}$	$17^{2,4}$	$12^{2,5}$	$14^{2,7}$	$17^{3,5}$	$15^{3,6}$	$21^{3,7}$	$16^{4,5}$	$25^{4,6}$	$13^{5,6}$	$13^{6,2}$	$10^{6,7}$
12	$18^{1,2}$	$15^{1,4}$	$11^{2,3}$	$14^{2,5}$	$16^{2,7}$	$14^{3,6}$	$14^{4,3}$	$19^{4,5}$	$7^{4,7}$	$26^{5,6}$	$19^{6,7}$			
13	$5^{1,2}$	$8^{1,3}$	$11^{1,5}$	$7^{2,5}$	$9^{3,6}$	$7^{4,3}$	$4^{4,7}$	$6^{5,3}$	$10^{5,4}$	$6^{5,7}$	$4^{6,4}$	$5^{6,7}$		
14	$15^{1,3}$	$12^{1,4}$	$11^{1,6}$	$17^{2,4}$	$12^{2,5}$	$14^{2,7}$	$17^{3,5}$	$15^{3,6}$	$21^{3,7}$	$16^{4,5}$	$25^{4,6}$	$13^{5,6}$	$13^{6,2}$	$10^{6,7}$

8.3 Домашнее задание

На заданной сети указаны пропускные способности ребер. Предполагается, что пропускные способности в обоих направлениях одинаковы (рисунок 6). Требуется:

а) сформировать на сети поток максимальной мощности, направленный из истока *I* в сток *S*;

б) выписать ребра, образующие на сети разрез минимальной пропускной способности.

а) б)

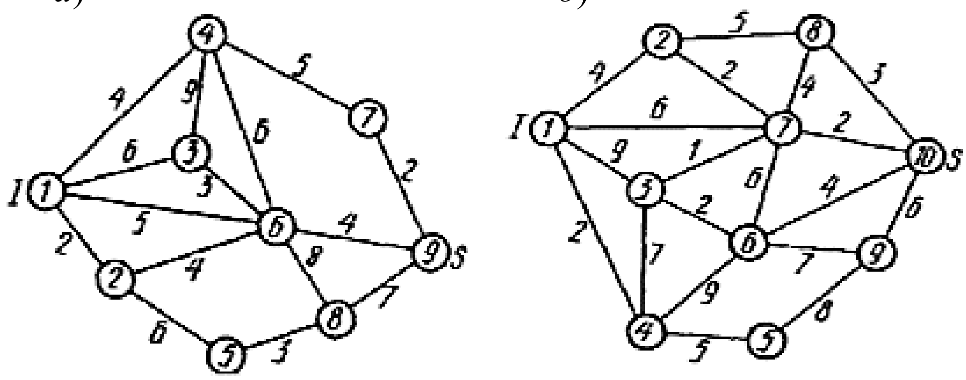


Рисунок 6 – Графы к домашнему заданию

9 Практическое занятие № 41. Связность в орграфах

На занятии расматриваются следующие вопросы: сильная, односторонняя и слабая связность; компоненты сильной связности; выделение компонент сильной связности [1, 6–12].

9.1 Теоретическая часть

В неориентированном графе две вершины либо связаны (если существует соединяющая их цепь), либо не связаны. В ориентированном графе отношение связанности вершин несимметрично, и потому рассматривается сильная, односторонняя и слабая связность.

Пусть $G(V,E)$ – орграф.

Две вершины v_1 и v_2 ***сильно связаны*** в орграфе G, если существует ориентированная цепь (путь) из v_1 в v_2 и из v_2 в v_1.

Две вершины v_1 и v_2 ***односторонне связаны*** в орграфе G, если существует путь либо из v_1 и v_2, либо из v_2 в v_1.

Две вершины v_1 и v_2 ***слабо связаны*** в орграфе G, если они связаны в графе G', полученном из G отменой ориентации ребер.

Если все вершины в орграфе сильно (односторонне, слабо) связаны, то орграф называется ***сильно*** (***односторонне***, ***слабо***) ***связанным***.

Из сильной связности следует односторонняя связность, а из односторонней связности следует слабая. Обратное неверно.

Компонентами сильной связности (КСС) орграфа G называются максимальные сильно связные подграфы. Каждая вершина v принадлежит только одной КСС. Если вершина не связана с другими, то считают, что она сама образует КСС.

Рассмотрим множество C_v, определенное следующим образом: $u \in C_v \Leftrightarrow (u = v) \vee$ (существует путь из v в u) & (существует путь из u в v). Тогда можно сказать, что вершина v принадлежит тому КСС, который является подграфом, порожденным C_v.

Числом сильной связности орграфа $G(V,E)$ называется число его различных компонент сильной связности.

Конденсацией G^* (фактор – графом) графа G называется орграф, который получается стягиванием в одну вершину каждой КСС орграфа G.

Для выделения компонент сильной связности используют аппарат матриц достижимости и сильной связности.

Матрицей достижимости орграфа G называется квадратная матрица $R=(r_{ij})$ порядка $|V|$, причем

$$r_{ij}=\begin{cases}1, \text{ если } v_j \text{ достижима из } v_i,\\ 0 - \text{ в противном случае.}\end{cases}$$

Матрицей сильной связности орграфа G называется квадратная матрица $S=(s_{ij})$ порядка $|V|$, причем

$$s_{ij}=\begin{cases}1, \text{ если } v_i \text{ достижима из } v_j \text{ и } v_j \text{ достижима из } v_i,\\ 0 - \text{ в противном случае.}\end{cases}$$

Матрицы достижимости и сильной связности связаны соотношением $S=R*R^T=(r_{ij}\cdot r_{ji})$ (почленное умножение).

Пример – Выделить компоненты сильной связности орграфа G, заданного матрицей смежности

$$\begin{pmatrix}0&0&0&0&0&0&1&1\\1&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&1&0&1&0&0\\0&0&0&0&1&0&0&0\\0&1&0&0&0&1&0&1\\0&1&1&0&0&1&0&0\end{pmatrix}.$$

Решение

Вначале строится матрица достижимости орграфа G, отмечая «куда» можно попасти из каждой вершины

$$R=\begin{pmatrix}1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\0&0&1&0&0&0&0&0\\0&0&0&1&1&1&0&0\\0&0&0&1&1&1&0&0\\0&0&0&1&1&1&0&0\\1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\end{pmatrix}.$$

Тогда

$$S=R*R^T=\begin{pmatrix}1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\0&0&1&0&0&0&0&0\\0&0&0&1&1&1&0&0\\0&0&0&1&1&1&0&0\\0&0&0&1&1&1&0&0\\1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\end{pmatrix}*\begin{pmatrix}1&1&0&0&0&0&1&1\\1&1&0&0&0&0&1&1\\1&1&1&0&0&0&1&1\\1&1&0&1&1&1&1&1\\1&1&0&1&1&1&1&1\\1&1&0&1&1&1&1&1\\1&1&0&0&0&0&1&1\\1&1&0&0&0&0&1&1\end{pmatrix}=\begin{pmatrix}1&1&0&0&0&0&1&1\\1&1&0&0&0&0&1&1\\0&0&1&0&0&0&0&0\\0&0&0&1&1&1&0&0\\0&0&0&1&1&1&0&0\\0&0&0&1&1&1&0&0\\1&1&0&0&0&0&1&1\\1&1&0&0&0&0&1&1\end{pmatrix}.$$

В первой строке матрицы S единицы соответствуют вершинам v_1, v_2, v_7, v_8, следовательно, они образуют компоненту сильной связности $\{v_1, v_2, v_7, v_8\}$. Для выделения остальных компонент в матрице S вычеркиваем соответствующие вершинам v_1, v_2, v_7, v_8 строки и столбцы матрицы и получаем новую матрицу

$$S_1=\begin{pmatrix}1&0&0&0\\0&1&1&1\\0&1&1&1\\0&1&1&1\end{pmatrix},$$

в первой строке которой только одна единица, относящаяся к вершине v_3, что соответствует еще одной компоненте сильной связности $\{v_3\}$. Далее вычеркиваем первые строку и столбец матрицы S_1, получаем новую матрицу

$$S_2=\begin{pmatrix}1&1&1\\1&1&1\\1&1&1\end{pmatrix}.$$

Матрица S_2 имеет одни единицы, относящиеся вершинам v_4, v_5, v_6, что соответствует компоненте сильной связности $\{v_4, v_5, v_6\}$.

Итак, орграф G имеет три компоненты сильной связности: $\{v_1, v_2, v_7, v_8\}$, $\{v_3\}$, $\{v_4, v_5, v_6\}$.

9.2 Задачи к занятию

1 Привести пример неорентированного графа с пятью вершинами, имеющего:

а) одну компоненту связности; б) две компонент связности; в) пять компонент связности.

2 Привести пример орграфа, имеющего четыре вершины и являющегося:

а) сильно связным; б) односторонне связным; в) слабо связным.

3 Для графа G, заданного на рисунке 7, найти:

а) матрицу смежности; б) матрицу инцидентности; в) степени вершин; г) число компонент связности.

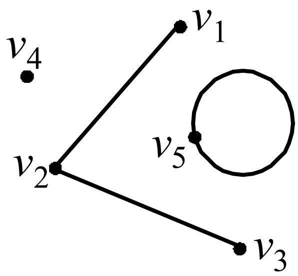


Рисунок 7 – Граф к заданию 3

4 В заданном матрицей смежности орграфе найдите компоненты связности и сильной связности:

$$\text{а)}\begin{pmatrix} 1&1&0&0&0&1\\ 1&1&1&0&0&0\\ 0&0&1&1&1&0\\ 0&0&1&1&0&0\\ 0&0&1&1&1&1\\ 0&0&0&0&0&1 \end{pmatrix};\ \text{б)}\begin{pmatrix} 0&1&0&0&0&0\\ 0&0&0&1&0&0\\ 1&0&0&0&0&0\\ 0&0&0&0&0&1\\ 0&0&1&0&0&0\\ 0&0&0&0&1&0 \end{pmatrix};\ \text{в)}\begin{pmatrix} 0&1&0&0&0&0\\ 1&0&1&0&0&0\\ 0&0&0&1&0&0\\ 0&1&1&1&0&0\\ 1&0&1&0&1&0\\ 0&0&0&0&0&0 \end{pmatrix};$$

$$\text{г)}\begin{pmatrix} 0&0&1&0&0&1\\ 0&0&1&1&0&0\\ 1&0&0&0&0&0\\ 0&0&0&0&0&1\\ 0&0&1&0&0&0\\ 1&0&0&0&1&0 \end{pmatrix};\ \text{д)}\begin{pmatrix} 1&1&0&0&0&1&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&1&1&0&0&0&0\\ 1&0&0&1&1&0&0&0\\ 0&1&0&0&0&1&1&0\\ 0&1&1&0&0&0&0&1\\ 0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&0&0 \end{pmatrix};\ \text{е)}\begin{pmatrix} 0&1&0&0&1&0&0&0\\ 0&1&0&0&0&1&0&0\\ 0&0&1&1&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&1&0\\ 0&1&1&0&0&0&0&1\\ 0&0&0&0&0&0&0&1\\ 0&0&0&0&1&0&0&1 \end{pmatrix}.$$

5 Найти компоненты сильной связности в орграфах, преставленных на рисунке 8. Найти конденсацию.

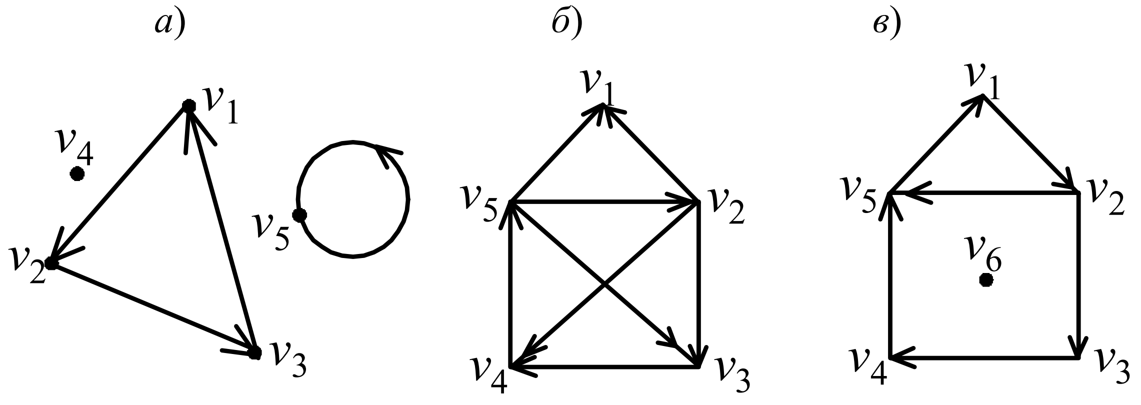


Рисунок 8 – Графы к заданию 5

9.3 Домашнее задание

1 Записать матрицу достижимости и найти компоненты связности и сильной связности для орграфов, заданных матрицей смежности:

$$
\text{а)}\ \begin{pmatrix} 0&1&1&0&1\\ 0&0&1&1&1\\ 1&0&0&1&0\\ 1&0&0&0&0\\ 0&0&0&0&0 \end{pmatrix};\ \text{б)}\ \begin{pmatrix} 0&0&0&1&1&0&0\\ 0&0&0&1&1&0&0\\ 0&0&0&0&0&1&0\\ 0&0&0&0&1&0&0\\ 0&1&0&0&0&0&0\\ 0&0&0&0&0&0&1\\ 0&0&1&0&0&0&0 \end{pmatrix};\ \text{в)}\ \begin{pmatrix} 0&1&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 1&0&0&0&1&1&0&0\\ 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&1&0 \end{pmatrix}
$$

2 Найти компоненты связности и сильной связности для орграфов, представленных на рисунке 9.

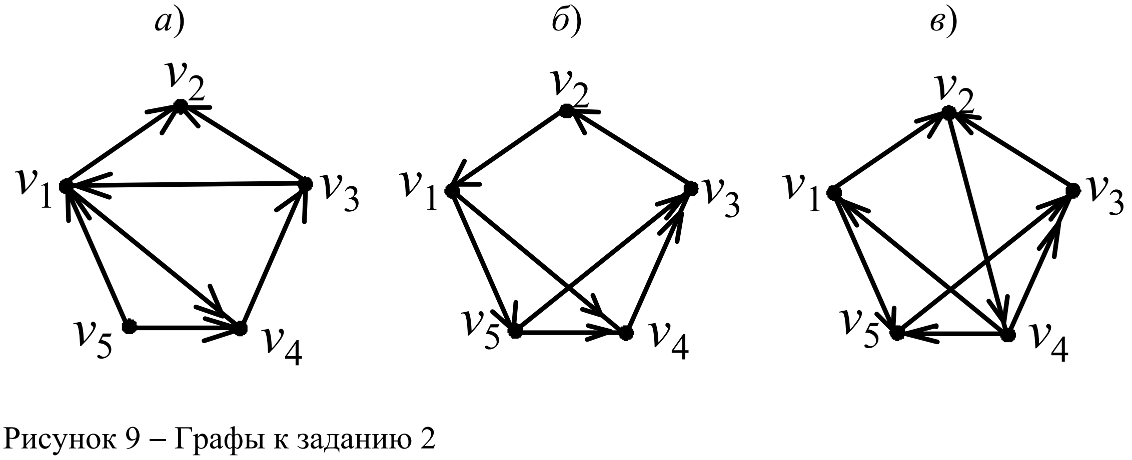


Рисунок 9 – Графы к заданию 2

10 Практическое занятие № 42. Кратчайшие пути

На занятии расматриваются алгоритмы поиска наикратчайшего пути в орграфе [1, 6–11].

10.1 Теоретическая часть

Рассмотрим ориентированный граф $G(V,E)$, в котором $V=\{v_1,v_2,...,v_n\}$ – конечное множество вершин, $E=\{e_1,e_2,...,e_m\}$ – конечное множество дуг. Здесь каждой дуге $e_k=(v_i,v_j),(k=\overline{1,m})$ ставится в соответствие положительное число $c_{ij}\ (i,j=\overline{1,n})$, которое интерпретируется как её длина.

Путь в ориентируемом графе – последовательность сцепленных ориентируемых дуг, в которых начало каждой следующей дуги совпадает с концом предыдущей. Длина любого пути в графе равна сумме длин дуг, входящих в этот путь. Необходимо найти путь минимальной длины (кратчайший путь) из вершины v_s в вершину v_t.

Алгоритм Форда

Рассмотрим алгоритм Форда для определения кратчайшего пути.

В процессе работы алгоритма Форда вершинам графа $v_i\in V$ приписываются числа $d(v_i)$, которые служат оценкой длины кратчайшего пути от вершины v_s к вершине v_i. При этом в самом начале работы алгоритма для всех вершин полагается $d(v_i)=\infty$, кроме $d(v_s)=0$. Затем осуществляется поиск дуги (v_i,v_j) такой, что $d(v_j)-d(v_i)>c_{ij}$, и замена $d(v_j)$ на $d(v_i)+c_{ij}<d(v_j)$. Такое переопределение $d(v_i)$ начинается с дуг, инцидентных v_s, и продолжается до тех пор, пока возможно найти дугу, уменьшающую хотя бы одно значение. Алгоритм Форда состоит из двух этапов. На первом этапе находятся длины кратчайших путей от вершины v_s до всех остальных вершин графа. На втором этапе строится сам путь от вершины v_s к вершине v_t. Продемонстрируем работу алгоритма на конкретном примере.

Пример – Найти в ориентированном графе G (рисунок 10) кратчайший путь из вершины v_1 в вершину v_8 по алгоритму Форда.

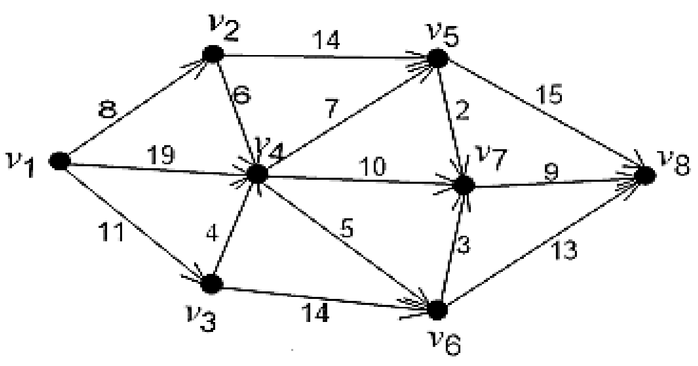


Рисунок 10 – Граф

Решение

В соответствии с рисунком 10 длины дуг представляются следующей матрицей:

$$(c_{ij})=\begin{pmatrix} - & 8 & 11 & 19 & \infty & \infty & \infty & \infty \\ \infty & - & \infty & 6 & 14 & \infty & \infty & \infty \\ \infty & \infty & - & 4 & \infty & 14 & \infty & \infty \\ \infty & \infty & \infty & - & 7 & 5 & 10 & \infty \\ \infty & \infty & \infty & \infty & - & \infty & 2 & 15 \\ \infty & \infty & \infty & \infty & \infty & - & 3 & 13 \\ \infty & \infty & \infty & \infty & \infty & \infty & - & 9 \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & - \end{pmatrix}.$$

Выполним поиск наикратчайшего пути.

Этап 1.

Шаг 1. Присвоение начальных значений $d(v_i)$ $(i=\overline{1,8})$.

Полагаем, $d(v_1)=0$ (конечное значение), $d(v_i)=\infty\,(i=\overline{2,8})$ (временное значение).

Шаг 2. Переопределение значений $d(v_i)$ $(i=\overline{2,8})$.

Из вершины v_1 исходят дуги (v_1,v_2), (v_1,v_3) и (v_1,v_4), для которых выполняется условие $d(v_j)-d(v_1)>c_{1j}$ (j = 2, 3, 4). Переопределим $d(v_j)$ как $d(v_2)=d(v_1)+c_{12}=0+8=8$, $d(v_3)=d(v_1)+c_{13}=0+11=11$, $d(v_4)=d(v_1)+c_{14}=0+19=19$.

В вершине v_4 можно уменьшить значение $d(v_4)$ за счёт дуги (v_2,v_4), т. к. $d(v_4)-d(v_2)=19-8=11>c_{24}=6$. Определяем $d(v_4)=d(v_2)+c_{24}=8+6=14$. Теперь отметим, что дуг, которые могут уменьшить значения $d(v_2)$, $d(v_3)$ и $d(v_4)$, больше нет. Это означает, что найденные $d(v_2)$, $d(v_3)$ и $d(v_4)$ являются конечными значениями, характеризующими длины кратчайших путей до соответствующих вершин. Отметим данный факт на графе (рисунок 11) при указанных вершинах двойной меткой $(d(v_j),\tilde{v}_i)$, где $\tilde{v}_i$ – вершина, дуга $(\tilde{v}_i,v_j)$ из которой определила конечное значение $d(v_j)$.

Переопределим значения $d(v_j)$ $(j=5,6,7)$ с помощью дуг (v_4,v_5), $(v_4,\,v_6)$ и (v_4,v_7). Имеем $d(v_5)=d(v_4)+c_{45}=14+7=21$, $d(v_6)=d(v_4)+c_{46}=14+5=19$, $d(v_7)=d(v_4)+c_{47}=14+10=24$. При этом значение $d(v_7)$ можно уменьшить за счёт дуги (v_6,v_7), т. е. $d(v_7)=d(v_6)+c_{67}=19+3=22$. Теперь дуг, способных уменьшить значения $d(v_5)$, $d(v_6)$ и $d(v_7)$, нет. Это означает, что найденные значения являются конечными, и на рисунке 11 вершины v_5, v_6 и v_7 отметим соответствующими двойными метками.

И, наконец, изменим значение $d(v_8)$. Максимальное уменьшение его достигается за счёт дуги (v_7, v_8). При этом $d(v_8) = d(v_7) + c_{78} = 22 + 9 = 31$. Данное окончательное значение отметим на рисунке 11 двойной отметкой при вершине v_8. Таким образом, найдена длина наикратчайшего пути от вершины v_1 к вершине v_8: $d(v_8) = 31$.

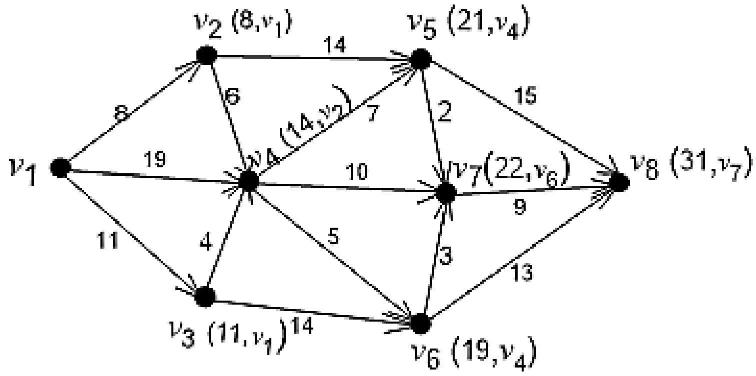


Рисунок 11 – Граф

Этап 2.

Определим наикратчайший путь. Для этого воспользуемся двойными отметками, которые нанесены на рисунок 11. Если анализировать данные отметки, начиная с вершины v_8 и заканчивая вершиной v_1, то получим следующий путь: $v_1, v_2, v_4, v_6, v_7, v_8$.

10.2 Задачи к занятию

Для графа, длины дуг которого заданы таблицей 2, найти наикратчайший путь и его длину.

Таблица 2 – Длины дуг графа *G*

Номер варианта	c_{12}	c_{13}	c_{14}	c_{15}	c_{24}	c_{25}	c_{34}	c_{36}	c_{46}	c_{47}	c_{57}	c_{58}	c_{67}	c_{68}	c_{78}
1	9	5	12	19	6	14	6	14	5	9	2	15	3	13	12
2	5	7	4	15	15	5	6	12	7	2	8	9	5	10	5
3	7	6	9	12	12	7	10	11	9	5	6	7	7	8	7
4	4	10	5	17	11	6	12	7	9	8	5	10	6	7	8
5	6	13	8	21	10	7	7	9	11	8	7	11	8	7	5
6	8	7	5	15	11	5	13	7	9	5	7	9	4	8	6
7	3	10	4	9	7	8	11	4	8	3	9	12	6	11	10
8	2	5	5	10	6	7	6	9	3	8	5	11	4	10	11
9	8	11	5	16	5	6	9	7	8	6	5	7	7	7	6
10	7	8	6	12	7	5	5	7	10	8	7	4	7	5	4
11	4	5	3	23	11	6	4	7	4	5	8	10	6	4	3
12	5	7	9	20	9	8	9	7	6	7	5	11	6	7	5
13	9	6	4	18	6	4	7	6	7	8	4	12	9	6	4
14	7	7	6	11	10	6	8	7	5	5	4	9	7	5	6
15	8	6	11	18	7	12	5	10	2	3	7	10	4	8	9
16	2	4	5	17	5	6	7	8	3	5	9	8	8	6	7

10.3 Домашнее задание

1 По заданной матрице длин дуг (c_{ij}) графа G найти величину минимального пути из вершины v_1 в вершину v_6, указать сам путь и построить граф:

$$\text{а)}\begin{pmatrix} - & 7 & 8 & 13 & \infty & \infty \\ \infty & - & 9 & 7 & 12 & \infty \\ \infty & \infty & - & 6 & 7 & 8 \\ \infty & \infty & \infty & - & 9 & 17 \\ \infty & \infty & \infty & \infty & - & 10 \\ \infty & \infty & \infty & \infty & \infty & - \end{pmatrix};\ \text{б)}\begin{pmatrix} - & 4 & 5 & 10 & 11 & \infty \\ \infty & - & 11 & 3 & 5 & \infty \\ \infty & \infty & - & 6 & 7 & 18 \\ \infty & \infty & \infty & - & 6 & \infty \\ \infty & \infty & \infty & \infty & - & 8 \\ \infty & \infty & \infty & \infty & \infty & - \end{pmatrix};\ \text{в)}\begin{pmatrix} - & \infty & 3 & \infty & \infty & \infty \\ \infty & - & \infty & 4 & 6 & 8 \\ \infty & 8 & - & 5 & 6 & 12 \\ \infty & \infty & \infty & - & 5 & 7 \\ \infty & \infty & \infty & \infty & - & 3 \\ \infty & \infty & \infty & \infty & \infty & - \end{pmatrix}.$$

2 Найти в ориентированном графе G (рисунок 12) кратчайший путь из вершины v_1 в вершину v_8.

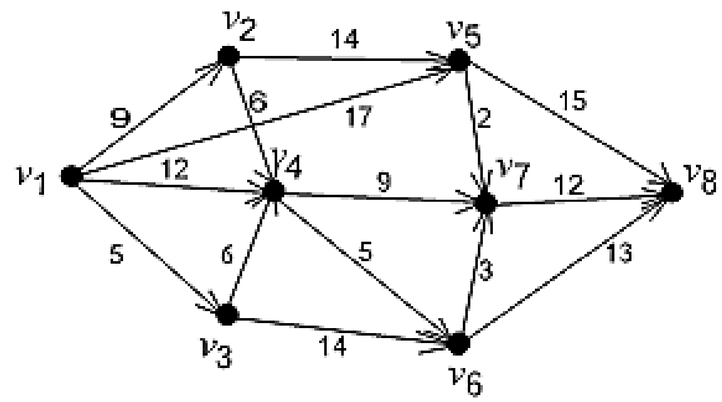


Рисунок 12 – Граф к задаче 2

3 По заданной матрице длин дуг (c_{ij}) графа G найти величину минимального пути из вершины v_1 в вершину v_6, указать сам путь и построить граф:

$$\text{а)}\begin{pmatrix} - & 8 & \infty & 5 & 10 & \infty \\ \infty & - & 7 & \infty & \infty & \infty \\ \infty & \infty & - & \infty & \infty & 8 \\ \infty & 4 & 6 & - & 5 & 11 \\ \infty & 5 & 5 & \infty & - & 3 \\ \infty & \infty & \infty & \infty & \infty & - \end{pmatrix};\qquad \text{б)}\begin{pmatrix} - & 3 & \infty & \infty & \infty & 7 & \infty \\ \infty & - & \infty & 8 & 5 & 10 & \infty \\ \infty & 4 & - & 3 & 6 & \infty & \infty \\ \infty & \infty & \infty & - & \infty & \infty & 5 \\ \infty & \infty & \infty & \infty & - & 6 & 13 \\ \infty & \infty & 7 & 6 & \infty & - & 4 \\ \infty & \infty & \infty & \infty & \infty & \infty & - \end{pmatrix}.$$

Список литературы

1 **Новиков, Ф. А.** Дискретная математика для программистов / Ф. А. Новиков. – Санкт-Петербург: Питер, 2000. – 304 с.

2 **Гаврилов, Г. П.** Задачи и упражнения по дискретной математике: учебное пособие / Г. П. Гаврилов, А. А. Сапоженко. – 3-е изд., перераб. – Москва: ФИЗМАТЛИТ, 2006. – 416 с.

3 **Иорданский, М. А.** Кодирование комбинаторных объектов: учебное пособие / М. А. Иорданский, О. В. Смышляева. – Нижний Новгород: Мининский университет, 2017. – 73 с.

4 **Алексеев, В. Е.** Сборник задач по дискретной математике: учебно-методическое пособие / В. Е. Алексеев, Л. Г. Кисилева, Т. Г. Смирнова. – Нижний Новгород: Нижегородский госуниверситет, 2012. – 80 с.

5 **Зверева, Е. Н.** Сборник примеров и задач по основам теории информации и кодирования сообщений / Е. Н. Зверева, Е. Г. Лебедько. – Санкт-Петербург: НИУ ИТМО, 2014. – 76 с.

6 **Шапорев, С. Д.** Дискретная математика. Курс лекций и практических занятий / С. Д. Шапорев. – Санкт-Петербург: БХВ-Петербург, 2006. – 400 с.

7 **Шевелев, Ю. П.** Дискретная математика: учебное пособие / Ю. П. Шевелев. – Санкт-Петербург: Лань, 2016. – 592 с.

8 **Феофанова, В. А.** Дискретная математика: учебно-методическое пособие / В. А. Феофанова, В. И. Воротников. – Нижний Тагил: НТИ (филиал) УрФУ, 2013. – 256 с.

9 **Авдеюк, О. А.** Лекции и практикум по основам дискретной математики и математической логике / О. А. Авдеюк, Л. В. Дружинина, И. В. Приходькова. – Волгоград: ВолгГТУ, 2019. – 316 с.

10 **Бернштейн, Т. В.** Практикум по дискретной математике / Т. В. Бернштейн, Т. В. Храмова. – Новосибирск: СибГУТИ, 2011. – 131 с.

11 **Павленкова, Е. В.** Сборник заданий по дискретной математике: электронное учебно-методическое пособие / Е. В. Павленкова, Д. Т. Чекмарев. – Нижний Новгород: Нижегородский госуниверситет, 2012. – 68 с.