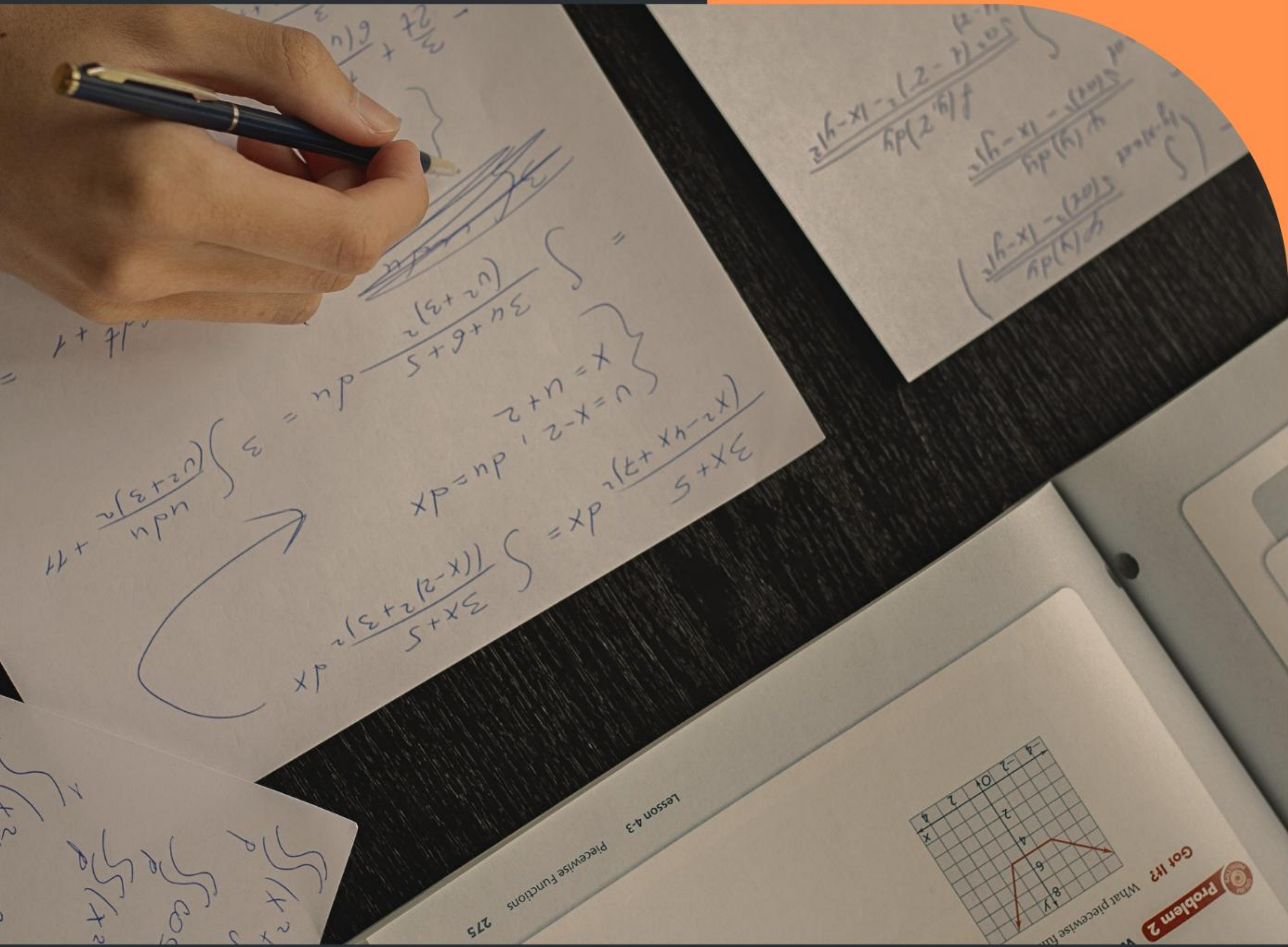
НОО "ПРОФЕССИОНАЛЬНАЯ НАУКА"



ДИСКРЕТНАЯ МАТЕМАТИКА

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ ДЛЯ СТУДЕНТОВ ОЧНО-ЗАОЧНОЙ ФОРМЫ ОБУЧЕНИЯ НАПРАВЛЕНИЯ ПОДГОТОВКИ 09.03.03 «ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ЭКОНОМИКЕ»

Н.С. КОШЕВАЯ

WWW.SCIPRO.RU

СОЧИНСКИЙ ИНСТИТУТ (ФИЛИАЛ)
Федерального государственного автономного образовательного учреждения высшего образования
«РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ
ИМ. П. ЛУМУМБЫ»

Дискретная математика

Учебно-методическое пособие для студентов
очно-заочной формы обучения
направления подготовки 09.03.03
«Информационные технологии в экономике»

Составитель Н.С. Кошевая

Сочи
2025

УДК 519
ББК 22.18

Главный редактор: Краснова Наталья Александровна – кандидат экономических наук, доцент, руководитель НОО «Профессиональная наука»

Технический редактор: Гусева Ю.О.

Автор-составитель:

Кошевая Наталья Сергеевна - ст. преподаватель кафедры «Математика и информационные технологии». Сочинский институт (филиал) ФГАОУ ВО «Российский университет дружбы народов им. П. Лумумбы»

Рецензенты:

Воротников В.И. - д. ф.-м. н., проф.
Постников В. В. - к. м. н.

Дискретная математика [Электронный ресурс]: учебно-методическое издание – Эл. изд. - Электрон. текстовые дан. (1 файл pdf: 75 с.). - Н.С. Кошевая. 2025. – Режим доступа: http://scipro.ru/conf/discrete_mathematics02_25.pdf. Сист. требования: Adobe Reader; экран 10'.

ISBN 978-5-908003-00-1

В методическом пособии изложены основные понятия дискретной математики.
Здесь представлены разделы: комбинаторики, элементы математической логики, булевы функции, теория множеств, элементы теории алгоритмов, графы. Изложение иллюстрируется большим количеством примеров. Освоить материал помогают контрольный тест и вопросы.
Для студентов очно-заочной формы обучения направления подготовки 09.03.03 «Информационные технологии в экономике»
Утверждено к печати на заседании кафедры «Математика и информационные технологии»(протокол № 05-01-03 /15 от 11.12. 2024)

ISBN 978-5-908003-00-1
9 785908 003001 >

Содержание

Тема 1. Комбинаторика

1.1. Основные понятия

Комбинаторика – раздел математики, в котором изучаются различные комбинации (соединения) элементов конечных множеств.

В комбинаторных задачах различают упорядоченные и неупорядоченные конечные множества.

Конечное множество называется упорядоченным, если для двух любых элементов a и b этого множества имеет место одно из следующих отношений порядка: «либо $a \le b$ (a не превосходит b)»; «либо $b \le a$ (b не превосходит a).

Свойства упорядоченного множества:

- рефлективность: любой элемент не превосходит самого себя;
- антисимметричность: если a не превосходит b, a b не превосходит a, то a и b совпадают;
- транзитивность: если a не превосходит b, a b не превосходит c, то a не превосходит c.

Два множества, составленные из одних и тех же элементов, но с разными отношениями порядка, считаются различными упорядоченными множествами. Поэтому при задании упорядоченного множества через его элементы необходимо также задать их порядок (указать правило, по которому любые два элемента множества можно сравнить).

Многие комбинаторные задачи решаются с помощью двух правил: правило сложения и правило умножения.

Правило сложения: если некоторый объект «a» можно выбрать n способами, а объект b можно выбрать m способами, причем первый и второй способы не пересекаются, то любой из объектов a или b можно выбрать n+m способами.

Правило умножения: если из некоторого конечного множества первый объект a можно выбрать n способами, а второй объект b можно выбрать m способами, то оба объекта в указанном порядке можно выбрать $n \cdot m$ способами.

Пример 1.1

Сколько различных трехзначных чисел можно составить из цифр: 0, 2, 3,5, 7, если:

а) цифры в числе не повторяются;

б) цифры в числе повторяются.

Решение.

а) Для составления трехзначного числа на первое место можно выбрать только 4 цифры – 2,5,3,7 (ноль не выбирается), на второе место можно поставить одно из оставшихся цифр – их также 4 (так как одно уже выбрали), на третье место можно поставить одно из оставшихся трех цифр.

Таким образом, согласно правилу умножения количество трехзначных чисел без повторения цифр будет равно $4 \cdot 4 \cdot 3 = 48$.

б) Если цифры в числе могут повторяться, то количество трехзначных чисел будет уже рано $4 \cdot 5 \cdot 5 = 100$.

Пример 1.2

Сколько чисел, содержащие не менее трех различных цифр, можно составить из цифр: 2,3,4,5,6?

Решение.

По правилу умножения трехзначных чисел можно составить $5 \cdot 4 \cdot 3 = 60$ способами, четырехзначных – $5 \cdot 4 \cdot 3 \cdot 2 = 120$ способами, пятизначных –

$5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120$. По правилу сложения, всего можно составить 60+120+120 = 300 чисел, состоящих не менее, чем из трех различных цифр.

В комбинаторике существует две схемы составления комбинаций m элементов из заданного множества: *без повторения*, когда при составлении комбинаций элементы не повторяются и *с повторением*, когда при составлении комбинаций осуществляется повторное использование элементов.

1.2. Комбинаторика без повторения

Рассмотрим множество, состоящее из n элементов, в котором, переставляя элементы данного множества, получим различные комбинации, называемые *перестановкой.*

Для нахождения числа перестановок используют формулу:

$$P(n) = n! \quad (1)$$

$n! = 1 \cdot 2 \cdot 3 \cdot \ldots \cdot n$

$0! = 1 \qquad 1! = 1$

В перестановке важен порядок.

Пример 1.3

Сколько различных пятизначных чисел можно составить из цифр 1,2,3,4,5?

Решение.

Для составления пятизначного числа поочередно отбирают цифры. Первую цифру можно отобрать и поставить на первое место пятью способами, тогда отобрать и поставить цифру на втрое место можно четырьмя способами, на третье место соответственно – тремя способами, на четвертое – двумя и пятое – одним способом.

Используя формулу перестановки, имеем: $P(5) = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 5! = 120$.

Итак, 120 различных пятизначных чисел можно составить из цифр 1,2,3,4,5.

Размещением из n элементов по k элементов называется любое упорядоченное подмножество данного множества, содержащее k элементов.

Два размещения различны, если они отличаются друг от друга либо составом элементов, либо порядком их расположения.

Число размещений из n элементов по k элементов обозначается A_n^k и вычисляется по формуле:

$$A_n^k = \frac{n!}{(n-k)!} \quad (2)$$

Пример 1.4

Сколько различных четырехзначных чисел можно составить из чисел 1,2, 3…,9, если все цифры в четырехзначных числах различны?

Решение.

Для формирования каждого четырехзначного числа выбираем четыре цифры из девяти, поэтому существует: $A_9^4 = \frac{9!}{(9-4)!} = \frac{9!}{5!} = 6 \cdot 7 \cdot 8 \cdot 9 = 3024$ различных чисел.

Сочетанием из n элементов по k элементов называется любое подмножество данного множества, содержащее k элементов.

Любые два сочетания отличаются друг от друга хотя бы одним элементом

Число сочетаний из n элементов по k элементов обозначается C_n^k и вычисляется по формуле:

$$C_n^k = \frac{n!}{k!\cdot(n-k)!} \qquad (3)$$

Свойства сочетаний:

1. $C_n^k = C_n^{n-k}$

2. $C_n^0 + C_n^1 + ... + C_n^n = 2^n$

3. $C_n^k = C_{n-1}^k + C_{n-1}^{k-1}$

4. $C_n^0 = C_n^n = 1$

Пример 1.5

Если множество содержит 10 элементов, то, сколько оно содержит трехэлементных подмножеств?

Решение.

Поскольку множество не упорядочено и выбираются любые три элемента из десяти, то поэтому всего имеется $C_{10}^3 = \frac{10!}{3!\cdot(10-3)!} = \frac{10!}{3!\cdot 7!} = \frac{8\cdot 9\cdot 10}{1\cdot 2\cdot 3} = 4\cdot 3\cdot 10 = 120$ различных подмножеств.

1.3. Комбинаторика с повторением

Пусть в множестве из n элементов есть k различных типов элементов, при этом первый тип элементов повторяется n_1 раз, второй – n_2 раз, …, k-й тип – n_k раз, причем $n_1+n_2+…+n_k = n$. Тогда перестановки элементов данного множества представляют собой *перестановки с повторениями.*

Число *перестановок с повторениями* обозначаются $P_n(n_1, n_2, …, n_k)$ и вычисляется по формуле:

$$P_n(n_1, n_2, ..., n_k) = \frac{n!}{n_1!\cdot n_2!\cdot \ldots \cdot n_k!} \qquad (4)$$

Пример 1.6

Сколько различных буквенных комбинаций можно составить, переставляя буквы в слове «ЛИЛИЯ»?

Решение.

Вообще из пяти букв можно составить $P_5 = 5! = 120$ различных пятибуквенных комбинаций. В слове «ЛИЛИЯ» две одинаковые буквы «Л» и «И», перестановка которых не поменяет комбинацию. Поэтому число перестановок с повторением меньше числа перестановок без повторений во столько раз, сколько можно переставлять повторяющиеся буквы. Отсюда количество различных комбинаций при перестановке букв в слове «ЛИЛИЯ» будет равно: $P_5(2,2) = \frac{P_5}{P_2\cdot P_2} = \frac{5!}{2!\cdot 2!} = \frac{120}{4} = 30$

Если при упорядоченной выборке k элементов из n элементы возвращаются обратно, то полученные выборки представляют собой *размещения с повторениями.*

Число всех размещений с повторениями из n элементов по k обозначается $\left(A_n^k\right)^*$ и вычисляется по формуле:

$$\left(A_n^k\right)^* = n^k \tag{5}$$

Пример 1.7

Учитывая порядок установки программы на компьютер, сколькими способами можно установить, пять программ на восемь компьютеров?

Решение.

Каждую из пяти программ можно загрузить в любой из восьми компьютеров в установленном порядке, причем в один компьютер можно загрузить несколько программ от одной до пяти, таким образом, общее число установки программ равно $\left(A_n^k\right)^* = 8^5 = 32\,768$.

Если при упорядоченной выборке k элементов из n элементы возвращаются обратно без последующего упорядочивания, то полученные выборки представляют собой *сочетания с повторениями.*

Число всех сочетаний с повторениями из n элементов по k обозначается $\left(C_n^k\right)^*$ и вычисляется по формуле:

$$\left(C_n^k\right)^* = C_{n+k-1}^k \tag{6}$$

Пример 1.8

В магазине имеется три вида программ. Сколькими способами можно приобрести семь программ.

Решение.

Поскольку имеет 3 вида программ, а приобрести нужно 7 программ, то число возможных наборов равно: $\left(C_3^7\right)^* = C_{3+7-1}^7 = C_9^7 = \frac{9!}{7!\cdot(9-7)!} = \frac{9!}{7!\cdot 2!} = \frac{8\cdot 9}{1\cdot 2} = 4\cdot 9 = 36$

Итоговую сводку формул представляем в таблице №1

Таблица №1

Формулы комбинаторики

Комбинаторика	Перестановка	Размещение	Сочетание
с повторением	$P(n) = n!$	$A_n^k = \frac{n!}{(n-k)!}$	$C_n^k = \frac{n!}{k!\cdot(n-k)!}$
без повторения	$P_n^*(n_1, n_2, \ldots, n_k) = \frac{n!}{n_1!\cdot n_2!\cdot \ldots \cdot n_k!}$	$\left(A_n^k\right)^* = n^k$	$\left(C_n^k\right)^* = C_{n+k-1}^k$

1.4. Бином Ньютона

Представление выражения *(a+b)^n*, где n – натуральное число, в виде многочлена, осуществляется с помощью формулы, называемой формулой *«бином Ньютона»*:

$$(a+b)^n = \sum_{k=0}^{n} C_n^k \cdot a^k \cdot b^{n-k} \tag{7}$$

где C_n^k - биноминальные коэффициенты.

Свойства биноминальных коэффициентов:

1Сумма коэффициентов разложения $(a+b)^n$ равна 2^n.

2Коэффициенты членов, равноудаленных от концов разложения равны.

3Сумма коэффициентов четных членов разложения равна сумме коэффициентов нечетных членов разложения. Каждая сумма равна 2^{n-1}.

Пример 1.9 Разложить в многочлен $(a+b)^6$.

Решение.

$$(a+b)^6 = a^6 + C_6^5 \cdot a^5 \cdot b + C_6^4 \cdot a^4 \cdot b^2 + C_6^3 \cdot a^3 \cdot b^3 + C_6^2 \cdot a^2 \cdot b^4 + C_6^1 \cdot a \cdot b^5 + b^6 =$$

$$= a^6 + \frac{6!}{5! \cdot 1!} \cdot a^5 \cdot b + \frac{6!}{4! \cdot 2!} \cdot a^4 \cdot b^2 + \frac{6!}{3! \cdot 3!} \cdot a^3 \cdot b^3 + \frac{6!}{2! \cdot 4!} \cdot a^2 \cdot b^4 + \frac{6!}{1! \cdot 5!} \cdot a \cdot b^5 + b^6 =$$

$$= a^6 + 6a^5 b + 15a^4 b^2 + 20a^3 b^3 + 15a^2 b^4 + 6ab^5 + b^6$$

Биноминальные коэффициенты можно вычислить с помощью ***треугольника Паскаля,*** схема составления которого заключается в следующем:

1. в первой строке пишем две единицы;
2. во второй строке три числа: по краям записываем единицы, в середине записываем сумму единиц первой строки;
3. каждая последующая строка начинается и заканчивается единицей, а промежуточные числа этих строк — это сумма соседних чисел из предыдущей строки.

1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1

Треугольник Паскаля

Пример 1.10

Используя треугольник Паскаля разложить в многочлен $(a+b)^5$

Решение.

Биноминальные коэффициенты выберем из пятой строки треугольника Паскаля и подставим в многочлен:

$$(a+b)^5 = a^5 + 5a^4b + 10a^3b^2 + 10a^2b^3 + 5ab^4 + b^5$$

1.5. Метод математической индукции

Доказательство истинности утверждений для всех натуральных чисел является одной из сложнейших проблем, поскольку, даже убедившись в истинности этого утверждения для большого количества натуральных чисел, нельзя говорит об истинности утверждения для всех натуральных чисел.

Однако с помощью метода математической индукции можно разрешить данный вопрос.

Принцип математической индукции.

Предположим, что для утверждения *P(n)* выполнены следующие условия:

1. при n=1, P (1) – истинна; (база индукции)
2. следует предположение, что для любого натурального k (k<n), утверждение P(k) – истинна;
3. из предположения истинности утверждения P(k) следует истинность утверждение истинности (k+1). (индуктивный переход)
4. тогда утверждение P(n) истинно для любого натурального n.

Пример 1.11

Доказать методом математической индукции истинность формулы: $1+2+3+...+n=\frac{n\cdot(n+1)}{2}$

Решение.

1. При n =1 получаем $1=\frac{1\cdot(1+1)}{2}\Rightarrow 1=1$. Формула верна.

1 Предположим, что формула верна для любого k (k<n): $1+2+3+...+k=\frac{k\cdot(k+1)}{2}$.

2 Если утверждение $1+2+3+...+k=\frac{k\cdot(k+1)}{2}$ верно для любого k, тогда оно верно и для (k+1).

$$\frac{k\cdot(k+1)}{2}+(k+1)=\frac{(k+1)\cdot((k+1)+1)}{2}$$

3 $$\frac{k^2+k+2k+2}{2}=\frac{k^2+2k+k+2}{2}$$

$$\frac{k^2+3k+2}{2}=\frac{k^2+3k+2}{2}$$

Из предположения истинности формулы $1+2+3+...+k=\frac{k\cdot(k+1)}{2}$ следует истинность формулы для *k+1*

Следовательно, формула истинна для любого натурального n.

Пример 1.12

Показать, что при любых n выражение 7^{2n} -1 делится 24.

Решение.

1. При n = 1 (7^2 -1): 24= 48:24 =2 – истинна.
2. Предположим, что условие выполняется для любого k (k<n), т.е. 7^{2k} -1.
3. Из предположения истинности условия, что для любого k (k<n) выражение 7^{2k} -1 делится 24, должна следовать истинность, что условие истинна и для (k+1).

Тогда,

($7^{2(k+1)}$ – 1): 24 = (7^{2k} 7^2 -1): 24= [$7^2\cdot$ ((7^{2k} -1) +48)]: 24 = [49$\cdot$ (7^{2k} – 1)]: 24 + [49 $\cdot$48]:24.

Итак, первое слагаемое делится на 24, так как выражение (7^{2k} -1) делится на 24 (доказано в первом пункте). Второе слагаемое также делится на 24, так как содержит множитель 48.

Из предположения истинности условия при любых k выражение 7^{2k} -1 делится 24, следует истинность условия при любом (k+1), следовательно, условие истинна для любого натурального n.

Контрольные вопросы для самопроверки

Тема 1 «Комбинаторика»
1. Основные правила комбинаторики.
2. Размещение без повторения. Определение. Пример.
3. Размещение с повторением. Определение. Пример.
4. Перестановка без повторения. Определение. Пример.
5. Перестановка с повторением. Определение. Пример.
6. Сочетание без повторения. Определение. Пример.
7. Сочетание с повторением. Определение. Пример.
8. Метод математической индукции.

Тема 2. Элементы математической логики

2.1. Логика высказывания

Логика – это наука о рассуждениях, которая позволяет определить, истинна или ложь то или иное математическое утверждение, исходя из совокупности первичных предположений, называемых *аксиомами*.

Логика применяется в информатике для построения компьютерных программ и доказательство из корректности.

Основной единицей изучения математической логики является *высказывание.*

Высказыванием называется повествовательное предложение, про которое можно сказать истинно оно или ложно.

Восклицательное и вопросительное предложение не является высказыванием.

Высказывания в математической логике обозначаются заглавными буквами.

Истинность или ложность кого - либо высказывания определяет его *истинностное значение.*

Обозначается истинностное значение высказывания буквами «Л», если высказывание ложно, и «И», если высказывание истинно.

В русском языке различают простые и сложные предложения. Сложное предложение составляют из простых с помощью союзов - связок. В математической логике также различают простые и сложные высказывания.

Высказывание называется ***простым***, если оно не содержит логических связок.

Высказывание называется ***сложным***, если оно составлено из простых высказываний, связанных между собой логическими связками.

К логическим связкам относятся следующие операции математической логики: конъюнкция, дизъюнкция, отрицание, импликация, сумма по модулю два, эквивалентность.

Определение. Конъюнкцией двух высказываний А и В называется высказывание истинное тогда и только тогда, когда оба высказывания истинны, во всех остальных случаях высказывание ложно.

Конъюнкция в математике определена как умножение и обозначается $\wedge$.

Конъюнкция определяет логическую связку «и».

Таблица истинности конъюнкции имеет вид (таб.2.1):

Таблица 2.1

Таблица истинности для конъюнкции

A	B	$A \wedge B$
л	л	л
л	и	л
и	л	л
и	и	и

Определение. Дизъюнкцией двух высказываний А и В называется высказывание ложное тогда и только тогда, когда оба высказывания ложь, во всех остальных случаях высказывание истинно.

Дизъюнкция в математике определена как сложение и обозначается $\vee$.

Дизъюнкция определяет логическую связку «или».

Таблица истинности дизъюнкции имеет вид (таб. 2.2):

Таблица 2.2

Таблица истинности для дизъюнкции

A	B	$A \vee B$
л	л	л
л	и	и
и	л	и
и	и	и

Определение. Отрицанием (инверсией) высказывания называется высказывание истинное тогда и только тогда, когда данное высказывание ложно, и ложно, когда данное высказывание истинно.

Отрицание определяет логическую связку «не».

Таблица истинности отрицания имеет вид (таб. 2.3):

Таблица 2.3

Таблица истинности отрицания

A	$\overline{A}$
л	и
и	л

Определение. Импликацией двух высказываний А и В называется высказывание ложное тогда и только тогда, когда посылка «А» истинна, а заключение «В» –ложно во всех остальных случаях высказывание истинно.

Импликация определяет логическую связку «если А, то В» и обозначается $\rightarrow$

Таблица истинности импликации имеет вид (таб. 2.4):

Таблица 2.4

Таблица истинности импликация

A	B	$A \rightarrow B$
л	л	и
л	и	и
и	л	л
и	и	и

Определение. Суммой по модулю два (неравнозначность) двух высказываний А и В называется высказывание ложное тогда и только тогда, когда истинность двух высказываний совпадает и истинно, когда не совпадает.

Сумма по модулю два определяет логическую связку «или» и обозначается $\oplus$.

Таблица истинности суммы по модулю два имеет вид (таб. 2.5):

Таблица 2.5

Таблица истинности суммы по модулю два

A	B	$A \oplus B$
л	л	л
л	и	и
и	л	и
и	и	л

Определение. Эквивалентностью двух высказываний A и B называется высказывание ложное тогда и только тогда, когда истинность двух высказываний не совпадает и истинно, когда совпадает.

Эквивалентность определяет логическую связку «A тогда и только тогда, когда B» и обозначается $\leftrightarrow$.

Таблица истинности эквивалентности имеет вид (таб. 2.6):

Таблица 2.6

Таблица истинности эквивалентности

A	B	$A \leftrightarrow B$
л	л	и
л	и	л
и	л	л
и	и	и

Буквы, обозначающие высказывания, логические связки и скобки составляют *алфавит языков логики высказываний*: алгебры логики и исчисления высказываний.

Алфавит исчисления высказывания состоит из следующих символов:

- любая переменная, обозначающая высказывание;
- логические связки: $\wedge, \vee,$ $,$ $\rightarrow, \oplus, \leftrightarrow$;
- скобки, запятые.

Определение. ***Логической формулой*** называется формула, если она удовлетворяет следующим условиям:

- любая переменная, обозначающая высказывание – формула;

- если A и B - формулы, то $(A \wedge B)$, $(A \vee B)$, $(\overline{A})$, $(A \rightarrow B)$, $(A \oplus B)$, $(A \Leftrightarrow B)$ – формулы;

- других формул нет.

Подформулы:

1. Подформулой переменной является сама переменная.
2. Для подформулы $\overline{A}$ подформулой является $\overline{A}$ и все подформулы формулы *A.*
3. Для формулы вида $(A * B)$, где «*» любая логическая связка, подформулами являются

(A $*$ B) и все подформулы формул A и B.

Высказывание, истинное во всех случаях независимо от переменной, называется ***тавтологией***.

Высказывание, построенное так, что оно ложно в каждом случае, независимо от аргумента, называется ***противоречием.***

Пример 2.1

Записать логической формулой следующих высказываний:

а) «Если компьютер при запуске не выдает ошибку при проверке оперативной памяти, то она исправна. Если при запуске он выдает ошибку при проверке оперативной памяти и память установлена правильно, то либо оперативная память с дефектом, либо дефектна материнская плата».

Решение.

Высказывание состоит из следующих простых высказываний:

- A – «компьютер выдает ошибку при проверке оперативной памяти»;
- B – «оперативная память исправна»;
- C – «оперативная память установлена правильно»;
- D – «материнская плата дефектна».

С учетом введенных обозначений простых высказываний и логических связок сложное высказывание может быть представлено в виде следующей логической формулы:

$(\overline{A} \rightarrow B) \wedge ((A \wedge C) \rightarrow (\overline{B} \vee D))$.

б) «Если допоздна работаешь с компьютером и при этом много пьешь кофе, то утром просыпаешься в дурном расположении духа или с головной болью».

Решение.

Высказывание состоит из следующих простых высказываний:

- X – «допоздна работаешь с компьютером»;
- Y – «много пьешь кофе»;
- Z – «просыпаешься в дурном расположении духа»;
- U – «просыпаешься с головной болью».

С учетом введенных обозначений простых высказываний и логических связок сложное высказывание может быть представлено в виде следующей логической формулы:

$(X \wedge Y) \rightarrow (Z \vee U)$.

Пример 2.2

Определите, является ли последовательность символов формулой:

а) $((P \wedge Q)\ R) \rightarrow \overline{B}$;

б) $(P \rightarrow Q) \vee (Q \rightarrow P)$.

Решение.

а) Согласно определению логической формулы выражение содержит переменные *P, Q, R, B* которые являются формулами, $\overline{B}$ и $(P \wedge Q)$ – являются формулами, $((P \wedge Q)R)$ – не является формулой, поскольку между скобкой и переменной *R* отсутствует логическая связка.

Вывод: данное выражение не является логической формулой.

б) Переменные *P, Q* являются формулами, $(P \rightarrow Q)$ и $(Q \rightarrow P)$ являются формулами, следовательно, $(P \rightarrow Q) \vee (Q \rightarrow P)$ – является формулой.

Пример 2.3

Выпишите все возможные подформулы из следующей формулы:

$((P \leftrightarrow (Q \wedge \overline{R})) \vee (P \wedge Q)) \rightarrow (P \vee (\overline{R} \wedge Q))$.

Решение.

Подформула – это такая связная часть данной формулы, которая сама является формулой. Связность означает, что подформулу можно без разрыва «наложить» на данную формулу.

Подформулы удобно перечислять последовательно, по числу логических связок, занятых в ней.

1) Подформулами будут все переменные входящие в данную формулу: *P, Q, R (формулы с нулевым числом логических связок);*

2) Выделим подформулы с одной логической связкой: $\overline{R}$, $P \wedge Q$;

3) Выделим подформулы с двумя логическими связками: $(Q \wedge \overline{R})$, $(\overline{R} \wedge Q)$;

4) Выделим подформулы с тремя логическими связками: $P \leftrightarrow (Q \wedge \overline{R})$, $P \vee (\overline{R} \wedge Q)$;

5) Подформулы с четырьмя логическими связками нет;

6) Выделим подформулу с пятью логическими связками: $(P \leftrightarrow (Q \wedge \overline{R})) \vee (P \wedge Q$;

7) Последняя подформула совпадает с исходной формулой:

$((P \leftrightarrow (Q \wedge \overline{R})) \vee (P \wedge Q)) \rightarrow (P \vee (\overline{R} \wedge Q))$.

Таким образом, у данной формулы 11 подформул.

Пример 2.4

Определите логическое значение высказывания исходя из значений предыдущих высказываний:

а) $\begin{cases} A \rightarrow B = и; \\ A \leftrightarrow B = л \end{cases} \Rightarrow B \rightarrow A - ?$

б) $\begin{cases} A \rightarrow B = л; \\ ((\overline{A} \wedge B) \rightarrow (\overline{A} \vee B)) - ? \end{cases}$

Решение.

а) Импликация А и В может быть истинной в трех случаях, когда: 1) «А» - л и «В» -л; 2) «А»- л и «В» -и; 3) «А»- и «В» -и. Эквивалентность А и В ложна, когда истинность значений дух высказываний не совпадает, следовательно, выбираем вариант «А»- л и «В» -и.

Отсюда получаем, что высказывание $B \rightarrow A$ = л, так как и$\rightarrow$л = л.

б) Импликация А и В ложна только в том случае, если «А» - и, «В» - л, тогда имеем:

$$((\overline{A} \wedge B) \rightarrow (\overline{A} \vee B)) = ((\overline{и} \wedge л) \rightarrow (\overline{и} \vee л) = ((л \wedge л) \rightarrow (л \vee л)) = л \rightarrow л = и$$

2.2. Булевы функции

Рассмотрим простые высказывания в виде переменных x_i, а значения истинности высказываний представим в виде числовых значений «и» – 1, «л» – 0, тогда такие переменные будем назвать *булевыми переменными*, которые принимают свои значения на множестве E= {0,1}.

Тогда сложные высказывания можно рассматривать как функции, определенные на множестве E^n всевозможных значений n булевых переменных и принимающих свои значения из множества E. Такие функции называются ***булевыми функциями.***

Всякая булева функция рассматривается как отображение f: $E^n \rightarrow E$.

Каждый элемент множества E^n называют набором значений n булевых переменных. Существует 2^n различных наборов n булевых переменных.

На каждом наборе булева функция *f* может принимать два значения: 0 и 1. Поэтому существует 2^{2^n} различных булевых функций *n* переменных.

Способы задания булевых функций:

- таблицей истинности;

Любая булева функция может быть задана таблицей, в которой перечисляются всевозможные значения переменных и для каждого набора значений переменных указывается соответствующее ему значение булевой функции. Такая таблица называется ***таблицей истинности булевой функции***

- аналитической формой (префиксной, инфиксной);

-совершенной формой;

-минимизированной формой;

-векторной формой;

- в n- разрядном двоичном коде.

Булевы функции нуля, одной, двух переменных называют элементарными. Функции нуля переменных – это константы 0 и 1.

Особую роль в алгебре логики играют булевы функции одной и двух переменных – *унарные и бинарные логические операции.*

Для унарной операции существует 4 булевы функции: сохраняющие константы 0 и 1, функция, повторяющая переменную и функция отрицания переменной.

Представим булеву функцию одной переменной в виде таблицы истинности (Таблица 2.7)

Таблица 2.7

Булевы функции оной переменной

x	f_1	f_2	f_3	f_4
0	0	1	0	1
1	0	1	1	0

f_1 – константа 0.

f_2 – константа 1.

f_3 – повторения переменной.

f_4 – отрицание переменной.

Для бинарной операции существует 16 булевых функций.

Представим булеву функцию двух переменных в виде таблицы истинности (Таблица 2.8)

Таблица 2.8

Булевы функции двух переменных

x_1	x_2	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}
0	0	0	1	0	0	1	1	0	0	1	1	0	1	0	0	1	1
0	1	0	1	0	1	1	0	0	1	1	0	1	0	0	1	1	0
1	0	0	1	1	0	0	1	0	1	0	1	1	0	1	0	1	0
1	1	0	1	1	1	0	0	1	1	1	1	0	1	0	0	0	0

f_1 – сохраняющая константу 0;
f_2 – сохраняющая константу 1;
f_3 – повторение первой переменной;
f_4 – повторение второй переменной;
f_5 – отрицание первой переменной;
f_6 – отрицание второй переменной;
f_7 – конъюнкция;
f_8 – дизъюнкция;
f_9 – импликация ($x_1 \rightarrow x_2$)
f_{10} – импликация ($x_2 \rightarrow x_1$)
f_{11} – сумма по модулю два;
f_{12} – эквивалентность;
f_{13} – отрицание импликации $x_1 \rightarrow x_2$;
f_{14} – отрицание импликации $x_2 \rightarrow x_1$;
f_{15} – штрих Шеффера (отрицание конъюнкции);
f_{16} – стрелка Пирса (отрицание дизъюнкции).

Эквивалентными ил равносильными называются формулы, представляющие одну и ту же функцию.

Метод установления эквивалентности двух формул:

- по каждой формуле создается таблица истинности;

-полученные таблицы сравниваются по каждому набору значений переменных.

Пример 2.2

Булева функция трех переменных задана формулой в префиксной форме:

$f\ (x_1, x_2, x_3) = f_3(f_1(x_3,\ x_1),\ f_2(x_1,\ f_3(x_1,\ f_4(x_2))))$.

а) Представить f в инфиксной форме, если $f_1 - \vee$, $f_2 - \wedge$, $f_3 - \oplus$, $f_4 - \neg$

б) Вычислить значение функции в наборе (0,1,0).

Решение.

а) Булева функция будем иметь следующую инфиксную форму:

$f(x_1,x_2,x_3) = (x_3 \vee x_1)\ \oplus\ (x_1 \wedge\ (x_1 \oplus (\overline{x_2})))$.

б) Вычислим функцию f на наборе (0,1,0)

$f\ (0,1,0) = (0 \vee 0)\ \oplus\ (0 \wedge\ (0 \oplus (\bar{1})))= 0\ \oplus (0 \wedge (0 \oplus 0)) = 0 \oplus (0 \wedge 0) = 0 \oplus 0 = 0$

Пример 2.2

Составить таблицу истинности булевой функции: $f(x_1,x_2,x_3) = (x_1 \leftrightarrow x_2)\ \rightarrow\ ((x_1 \wedge\ x_3)\ \vee x_2)$.

Решение:

Для построения таблицы истинности функции f вычислим ее значения на каждом из восьми наборов значений:

1. Для определения истинностных значений в колонке 4 смотрим значения 1 и 2 колонки;
2. Для определения истинностных значений в колонке 5 смотрим значения 1 и 3 колонки;
3. Для определения истинностных значений в колонке 6 смотрим значения 5 и 3 колонки;
4. Для определения истинностных значений в колонке 7 смотрим значения от 4 к 6 колонки.

Таблица 2.9

Таблица истинности

1	2	3	4	5	6	7
x_1	x_2	x_3	$x_1 \leftrightarrow x_2$	$x_1 \wedge x_3$	$(x_1 \wedge x_3) \vee x_2$	$(x_1 \leftrightarrow x_2) \rightarrow ((x_1 \wedge x_3) \vee x_2)$.
0	0	0	1	0	0	0
0	0	1	1	0	0	0
0	1	0	0	0	1	1
0	1	1	0	0	1	1
1	0	0	0	0	0	1
1	0	1	0	1	1	1
1	1	0	1	0	1	1
1	1	1	1	1	1	1

Пример 2.3

Доказать эквивалентность формул: $x_1 \downarrow x_2 = \overline{x_1 \vee x_2} = \overline{x}_1 \& \overline{x}_2$

Решение.

Для доказательства эквивалентности формул составим таблицу истинности для каждой из них:

Таблица 2.10

Таблица истинности функций примера 2.3

x_1	x_2	$x_1 \downarrow x_2$	$x_1 \vee x_2$	$\overline{x_1 \vee x_2}$	$\overline{x}_1$	$\overline{x}_2$	$\overline{x}_1 \& \overline{x}_2$
0	0	**1**	0	**1**	1	1	**1**
0	1	**0**	1	**0**	1	0	**0**
1	0	**0**	1	**0**	0	1	**0**
1	1	**0**	1	**0**	0	0	**0**

Формулы $x_1 \downarrow x_2$, $\overline{x_1 \vee x_2}$, $\overline{x}_1 \& \overline{x}_2$ принимают одинаковые значения на одних и тех же наборах значений, следовательно, данные формулы эквиваленты.

2.3. Нормальные формы булевых функций

Пусть $x_1, x_2, \ldots, x_n$ простые высказывания. Состояние простого высказывания x в простой и инверсной форме называются **термами.** Любой вариант связки конъюнкции или дизъюнкции полного набора термов называют **конституентой.**

Элементарной конъюнкцией (дизъюнкцией) называется выражение, состоящее из конечного числа термов, взятых в этом выражении не более одного раза и разделенных операциями конъюнкцией (дизъюнкцией): $x_1 \wedge x_2 \wedge \ldots \wedge x_n$ ($x_1 \vee x_2 \vee \ldots \vee x_n$).

Выражение, представляющее собой, дизъюнкцию элементарных конъюнкций (конъюнкцию элементарных дизъюнкций) называется ***нормальной дизъюнктивной формой*** – ДНФ (***нормальной конъюнктивной формой*** – НКФ).

Пример 2.4

Привести к дизъюнктивной нормальной форме функцию:

$$f(x,y,z)=x\,\&\,y\vee\bar{x}\,\&\,(y\vee x\,\&\,z)\,\&\,\overline{(x\,\&\,(\bar{y}\vee z)\vee y\,\&\,z)}$$

Решение.

$$f(x,y,z)=(x\,\&\,y)\vee\bar{x}\,\&\,(y\vee x\,\&\,z)\,\&\,\overline{(x\,\&\,(\bar{y}\vee z)\vee y\,\&\,z)}=(x\,\&\,y)\vee((\bar{x}\,\&\,y)\vee(x\,\&\,\bar{x}\,\&\,z))\,\&$$

$$\&\,((\overline{x\,\&\,(\bar{y}\vee z)})\,\&\,\overline{(y\,\&\,z)})=(x\,\&\,y)\vee(\bar{x}\,\&\,y)\,\&\,((\bar{x}\vee\overline{(\bar{y}\vee z)})\,\&\,(\bar{y}\vee\bar{z}))=(x\,\&\,y)\vee(\bar{x}\,\&\,y)\,\&$$

$$\&\,((\bar{x}\vee(y\,\&\,\bar{z}))\,\&\,(\bar{y}\vee\bar{z}))=(x\,\&\,y)\vee(\bar{x}\,\&\,y)\,\&\,((\bar{x}\,\&\,\bar{y})\vee(\bar{x}\,\&\,\bar{z})\vee(y\,\&\,\bar{z}\,\&\,\bar{y})\vee(y\,\&\,\bar{z}\,\&\,\bar{z}))=$$

$$=(x\,\&\,y)\vee(\bar{x}\,\&\,y)\,\&\,((\bar{x}\,\&\,\bar{y})\vee(\bar{x}\,\&\,\bar{z})\vee(y\,\&\,\bar{z}))=(x\,\&\,y)\vee(\bar{x}\,\&\,y\,\&\,\bar{x}\,\&\,\bar{y})\vee(\bar{x}\,\&\,y\,\&\,y\,\&\,\bar{z})\vee$$

$$\vee(\bar{x}\,\&\,y\,\&\,y\,\&\,\bar{z})=(x\,\&\,y)\vee(\bar{x}\,\&\,y\,\&\,\bar{z})=y\,\&\,(x\vee(\bar{x}\,\&\,\bar{z}))=y\,\&\,(x\vee\bar{z})=(x\,\&\,y)\vee(y\,\&\,\bar{z})$$

Пример 2.5

Привести функцию $f(x,y,z)=(x\,\&\,\bar{y})\vee(\bar{x}\,\&\,y)\vee(x\,\&\,\bar{z})$ к конъюнктивной нормальной форме.

Решение.

$$f(x,y,z)=(x\,\&\,\bar{y})\vee(\bar{x}\,\&\,y)\vee(x\,\&\,\bar{z})=\overline{\overline{(x\,\&\,\bar{y})\vee(\bar{x}\,\&\,y)\vee(x\,\&\,\bar{z})}}=\overline{(\bar{x}\vee y)\,\&\,(x\vee\bar{y})\,\&\,(\bar{x}\vee z)}=$$

$$=\overline{((\bar{x}\,\&\,x)\vee(\bar{x}\,\&\,\bar{y})\vee(x\,\&\,y)\vee(y\,\&\,\bar{y}))\,\&\,(\bar{x}\vee z)}=\overline{((\bar{x}\,\&\,\bar{y})\vee(x\,\&\,y))\,\&\,(\bar{x}\vee z)}=$$

$$=\overline{(\bar{x}\,\&\,\bar{y}\,\&\,\bar{x})\vee(\bar{x}\,\&\,\bar{y}\,\&\,z)\vee(x\,\&\,y\,\&\,\bar{x})\vee(x\,\&\,y\,\&\,z)}=\overline{(\bar{x}\,\&\,\bar{y})\vee(\bar{x}\,\&\,\bar{y}\,\&\,z)\vee(x\,\&\,y\,\&\,z)}=$$

$$=\overline{(\bar{x}\,\&\,\bar{y})\vee(x\,\&\,y\,\&\,z)}=\overline{(\bar{x}\,\&\,\bar{y})}\,\&\,\overline{(x\,\&\,y\,\&\,z)}=(x\vee y)\,\&\,(\bar{x}\vee\bar{y}\vee\bar{z})$$

Нормальная форма называется ***совершенной,*** если в каждый ее элементарной конъюнкции (дизъюнкции) представлены все переменные входящие в функцию (с отрицанием и без отрицания).

Существуют следующие способы представления булевых функций в совершенной форме.

а) посредством таблиц истинности;

б) аналитический.

а) По данной таблице истинности составление СКНФ и СДНФ осуществляется следующим образом:

- ***Совершенная дизъюнктивная нормальная форма (СДНФ)***

При составлении СДНФ выбираются наборы в таблице истинности, на которых функция принимает значение 1. Все значения переменных в этом наборе равных 0 берут с отрицанием, все значения переменных в этом наборе равных 1 берут без отрицания.

Соединив се переменные, соответствующие этому набору знаком конъюнкции (&), мы получим элементарную конъюнкцию. Тогда дизъюнкции всех элементарных конъюнкций, соответствующих наборам значений переменных, где функция равна единице, и определяет СДНФ.

Пример 2.6

Построить СДНФ для функции, таблица истинности которой имеет вид:

Таблица 2.11

Таблица истинности булевой функции

x_1	x_2	x_3	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

$f(x_1, x_2, x_3) = (\bar{x}_1 \& \bar{x}_2 \& x_3) \vee (\bar{x}_1 \& x_2 \& \bar{x}_3) \vee (x_1 \& \bar{x}_2 \& x_3)$ - СДНФ

- *Совершенная конъюнктивная нормальная форма (СКНФ)*

При составлении СКНФ выбираются наборы в таблице истинности, на которых функция принимает значение 0. Все значения переменных в этом наборе равных 1 берут с отрицанием, все значения переменных в этом наборе равных 0 берут без отрицания.

Соединив се переменные, соответствующие этому набору знаком дизъюнкции ($\vee$), мы получим элементарную дизъюнкцию. Тогда конъюнкцию всех элементарных дизъюнкций, соответствующих наборам значений переменных, где функция равна 0 и определяет СКНФ.

Пример 2.7

Построить СКНФ для функции, заданной в таблице истинности 2.11:

$f(x_1, x_2, x_3) = (x_1 \vee x_2 \vee x_3) \& (x_1 \vee \bar{x}_2 \vee \bar{x}_3) \& (\bar{x}_1 \vee x_2 \vee x_3) \& (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \& (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)$ - СКНФ

б) Аналитический способ представления нормальной булевой функции в совершенной форме осуществляется на основе законов алгебры логики.

Законы алгебры логики

1. *Коммутативный закон*

- конъюнкции: $x_1 \& x_2 = x_2 \& x_1$
- дизъюнкции: $x_1 \vee x_2 = x_2 \vee x_1$

2. *Дистрибутивный закон*

- конъюнкции относительно дизъюнкции: $x_1 \& (x_2 \vee x_3) = (x_1 \& x_2) \vee (x_1 \& x_3)$
- дизъюнкции относительно конъюнкции: $x_1 \vee (x_2 \& x_3) = (x_1 \vee x_2) \& (x_1 \vee x_3)$

3. *Ассоциативный закон*

- конъюнкции: $x_1 \& (x_2 \& x_3) = (x_1 \& x_2) \& x_3$
- дизъюнкции: $x_1 \vee (x_2 \vee x_3) = (x_1 \vee x_2) \vee x_3$

4. *Закон идемпотентности*

- конъюнкции: $x_1 \& x_1 = x_1$
- дизъюнкции: $x_1 \vee x_1 = x_1$

5. *Закон поглощения*

- конъюнкции: $(x_1 \& x_2) \vee x_2 = x_2$
- дизъюнкции: $(x_1 \vee x_2) \& x_2 = x_2$

6. *Закон двойного отрицания*: $\overline{\overline{x}} = x$

7. *Законы де Моргана*

- $\overline{x_1 \vee x_2} = \overline{x_1} \wedge \overline{x_2}$
- $\overline{x_1 \wedge x_2} = \overline{x_1} \vee \overline{x_2}$

8. *Операции с константами*

$x \wedge 1 = x \qquad x \wedge 0 = 0 \qquad \overline{0} = 1$

$x \vee 1 = 1 \qquad x \vee 0 = x \qquad \overline{1} = 0$

9. Закон противоречия

$x \wedge \overline{x} = 0$

10.Закон исключенного третьего

$x \vee \overline{x} = 1$

Пример 2.8

Аналитическим способом получить СДНФ функции *f(x,y,z,u)= (x&y)* $\vee$ *(x&z)* $\vee$ *(z&u),* используя законы алгебры логики.

Решение.

Используя закон исключенного третьего, приведем исходную функцию к виду:

$$f(x,y,z,u) = (x \& y \& (z \vee \overline{z}) \& (u \vee \overline{u})) \vee (x \& z \& (y \vee \overline{y}) \& (u \vee \overline{u})) \vee (z \& u \& (x \vee \overline{x}) \& (y \vee \overline{y}))$$

Раскрывая скобки, и на основе закона идемпотентности приведем данную функцию к СДНФ.

$$f(x,y,z,u) = (x \& y \& (z \vee \overline{z}) \& (u \vee \overline{u})) \vee (x \& z \& (y \vee \overline{y}) \& (u \vee \overline{u})) \vee (z \& u \& (x \vee \overline{x}) \& (y \vee \overline{y})) =$$
$$= (x \& y \& z \& (u \vee \overline{u})) \vee (x \& y \& \overline{z} \& (u \vee \overline{u})) \vee (x \& z \& \overline{y} \& (u \vee \overline{u})) \vee (z \& u \& x \& (y \vee \overline{y})) \vee$$
$$\vee (z \& u \& \overline{x} \& (y \vee \overline{y})) = (x \& y \& z \& u) \vee (x \& y \& z \& \overline{u}) \vee (x \& y \& \overline{z} \& u) \vee (x \& y \& \overline{z} \& \overline{u}) \vee$$
$$\vee (x \& \overline{y} \& z \& u) \vee (x \& \overline{y} \& z \& \overline{u}) \vee (x \& y \& z \& u) \vee (x \& \overline{y} \& z \& u) \vee (\overline{x} \& y \& z \& u) \vee (\overline{x} \& \overline{y} \& z \& u) =$$
$$= (x \& y \& z \& u) \vee (x \& y \& z \& \overline{u}) \vee (x \& y \& \overline{z} \& u) \vee (x \& y \& \overline{z} \& \overline{u}) \vee (x \& \overline{y} \& z \& u) \vee (x \& \overline{y} \& z \& \overline{u}) \vee$$
$$\vee (\overline{x} \& y \& z \& u) \vee (\overline{x} \& \overline{y} \& z \& u)$$

2.4. Минимизация булевых функций

Минимизация нормальных форм осуществляется на основе закона склеивания:

$$(x_1 \& x_2) \vee (x_1 \& \overline{x_2}) = x_1 \& (x_2 \vee \overline{x_2}) = x_1 \& 1 = x_1$$
$$(x_1 \vee x_2) \& (x_1 \vee \overline{x_2}) = x_1 \vee (x_2 \& \overline{x_2}) = x_1 \vee 0 = x_1$$

Одну и ту же конституенту можно склеивать с другими конституентами многократно, так как в логике действует закон идемпотентности: $x_1 \& x_1 = x_1$, $x_1 \vee x_1 = x_1$, поэтому любую конституенту можно размножить.

Пример 2.9

Минимизировать булеву функцию:

$$f(z_1, z_2, z_3) = (\overline{z_1} \& \overline{z_2} \& \overline{z_3}) \vee (\overline{z_1} \& \overline{z_2} \& z_3) \vee (\overline{z_1} \& z_2 \& \overline{z_3}) \vee (z_1 \& \overline{z_2} \& z_3) \vee (z_1 \& z_2 \& z_3)$$

Решение.

$$f(z_1, z_2, z_3) = (\overline{z_1} \& \overline{z_2} \& \overline{z_3}) \vee (\overline{z_1} \& \overline{z_2} \& z_3) \vee (\overline{z_1} \& z_2 \& \overline{z_3}) \vee (z_1 \& \overline{z_2} \& z_3) \vee (z_1 \& z_2 \& z_3) =$$
$$= (\overline{z_1} \& \overline{z_2}) \& \underbrace{(z_3 \vee \overline{z_3})}_{1} \vee (\overline{z_1} \& \overline{z_3}) \& \underbrace{(\overline{z_2} \vee z_2)}_{1} \vee (z_1 \& z_3) \& \underbrace{(\overline{z_2} \vee z_2)}_{1} \vee (\overline{z_2} \& z_3) \& \underbrace{(\overline{z_1} \vee z_1)}_{1} =$$
$$= (\overline{z_1} \& \overline{z_2}) \vee (\overline{z_1} \& \overline{z_3}) \vee (z_1 \& z_3) \vee (\overline{z_2} \& z_3)$$

Пример 2.10

Минимизировать булеву функцию:

$$f(z_1, z_2, z_3) = (\bar{z}_1 \,\&\, \bar{z}_2 \,\&\, \bar{z}_3) \vee (\bar{z}_1 \,\&\, z_2 \,\&\, \bar{z}_3) \vee (z_1 \,\&\, z_2 \,\&\, \bar{z}_3) \vee (z_1 \,\&\, z_2 \,\&\, z_3)$$

Решение.

$$f(z_1, z_2, z_3) = (\bar{z}_1 \,\&\, \bar{z}_2 \,\&\, \bar{z}_3) \vee (\bar{z}_1 \,\&\, z_2 \,\&\, \bar{z}_3) \vee (z_1 \,\&\, z_2 \,\&\, \bar{z}_3) \vee (z_1 \,\&\, z_2 \,\&\, z_3) =$$

$$= (\bar{z}_1 \,\&\, \bar{z}_3) \,\&\, \underbrace{(\bar{z}_2 \vee z_2)}_{1} \vee (z_2 \,\&\, \bar{z}_3) \,\&\, \underbrace{(\bar{z}_1 \vee z_1)}_{1} \vee (z_1 \,\&\, z_2) \,\&\, \underbrace{(\bar{z}_3 \vee z_3)}_{1} =$$

$$= (\bar{z}_1 \,\&\, \bar{z}_3) \vee (z_2 \,\&\, \bar{z}_3) \vee (z_1 \,\&\, z_2)$$

Применение этих законов позволяет найти более компактные аналитические выражения для заданной функции.

После того как найдены *минимальные нормальные формы,* их рекомендуется проверить на всех наборах аргументов x. Именно полный набор из n термов образует *конституенту*. В процессе минимизации некоторые термы из конституент пропадут. Оставшуюся часть дизъюнкта или конъюнкта называют ***импликантой.***

Импликанты появляются в результате склейки смежных конституент, различающихся одним термом. Однако для функции, зависящих от многих переменных неконтролируемый процесс склейки неизбежно приводит к лишним импликантам. Требуемое число импликант может оказаться гораздо меньше возможного числа смежных склеек. В таких случаях истинную минимизированную нормальную форму получают с помощью *специальных методов минимизации.*

Рассматриваемые далее методы минимизации будут касаться только СДНФ, из которой на основании принципа двойственности легко могут быть распространены и на СКНФ.

- **Минимизация с использованием карт Карно.**

Карта Карно – это таблица, каждый элемент которой является элементарной конъюнкцией. Строиться она таким образом, чтобы две соседние в строке или в столбце элементарные конъюнкции отличались только одним термом.

Заполнение карт Карно:

1. выбираем первый минтерм функции и находим его координаты в таблице;

- ставим единицу, когда работаем с ДНФ;
- ставим ноль, когда работаем с КНФ.

2. осуществляем склейку элементов при помощи формул:

$$(x_1 \,\&\, x_2) \vee (x_1 \,\&\, \bar{x}_2) = x_1 \,\&\, (x_2 \vee \bar{x}_2) = x_1 \,\&\, 1 = x_1$$

$$(x_1 \vee x_2) \,\&\, (x_1 \vee \bar{x}_2) = x_1 \vee (x_2 \,\&\, \bar{x}_2) = x_1 \vee 0 = x_1$$

$$x \wedge 1 = x \qquad x \wedge 0 = 0 \qquad \bar{0} = 1$$

$$x \vee 1 = 1 \qquad x \vee 0 = x \qquad \bar{1} = 0$$

$$x \wedge \bar{x} = 0 \qquad x \vee \bar{x} = 1$$

Правила склейки:

- только парное количество элементов;
- элементы соединяются по ребру;
- крайние противоположные друг другу элементы (сворачивание карты в трубочку)

3. записывается минимизированная формула.

Пример 2.11

Минимизировать функцию

$f(x_1, x_2, x_3, x_4) = (\bar{x}_1 \& \bar{x}_2 \& \bar{x}_3 \& \bar{x}_4) \vee (\bar{x}_1 \& \bar{x}_2 \& x_3 \& \bar{x}_4) \vee (\bar{x}_1 \& x_2 \& x_3 \& \bar{x}_4) \vee (x_1 \& x_2 \& \bar{x}_3 \& \bar{x}_4) \vee$
$\vee (x_1 \& x_2 \& \bar{x}_3 \& x_4)$

посредством карт Карно.

Решение.

Заполним карту Карно (рис.2.1):

Карта Карно

Элементарные конъюнкции	$\bar{x}_3 \& \bar{x}_4$	$\bar{x}_3 \& x_4$	$x_3 \& x_4$	$x_3 \& \bar{x}_4$
$\bar{x}_1 \& \bar{x}_2$		1		1
$\bar{x}_1 \& x_2$				1
$x_1 \& x_2$				
$x_1 \& \bar{x}_2$	1	1		

Рис 2.1

1. Минтер $\bar{x}_1 \& \bar{x}_2 \& \bar{x}_3 \& x_4$ просто перепишем, так как его нечем склеить;
2. Выделим первую пару для склейки:

– по вертикали мы выделим одну общую координату $x_3 \& \bar{x}_4$ для обоих минтермов $\bar{x}_1 \& \bar{x}_2$ $\bar{x}_1 \& x_2$;

- сокращаем $\bar{x}_2$ и x_2 получаем $\bar{x}_1 \& x_3 \& \bar{x}_4$.

3. Выделим вторую пару для склейки:

- по горизонтали мы выделим общую координату $x_1 \& \bar{x}_2$ для обоих минтермов $\bar{x}_3 \& \bar{x}_4$ и $\bar{x}_3 \& x_4$;

- сокращаем $\bar{x}_4$ и x_4 получаем $x_1 \& \bar{x}_2 \& \bar{x}_3$

Минимизированная форма булевой функции имеет вид:

$f(x_1, x_2, x_3, x_4) = (\bar{x}_1 \& \bar{x}_2 \& \bar{x}_3 \& x_4) \vee (\bar{x}_1 \& x_3 \& \bar{x}_4) \vee (x_1 \& \bar{x}_2 \& \bar{x}_3)$

Пример. 2.12

Минимизировать функцию

$f(z_1, z_2, z_3) = (\bar{z}_1 \& \bar{z}_2 \& \bar{z}_3) \vee (\bar{z}_1 \& z_2 \& \bar{z}_3) \vee (z_1 \& z_2 \& \bar{z}_3) \vee (z_1 \& z_2 \& z_3)$ посредством карт Карно.

Решение.

1.Заполним карту Карно (рис.2.2):

2 Выделим первую пару для склейки:

- по горизонтали выделим общую координату $z_1 \& z_2$ для двух термов x_3 и $\bar{x}_3$;

- сокращаем на x_3 и $\bar{x}_3$ и получаем $z_1 \& z_2$.

3. Выделим вторую пару для склейки:

- по вертикали выделим общий терм $\bar{x}_3$ для обоих минтермов $\bar{z}_1 \& \bar{z}_2$ и $\bar{z}_1 \& z_2$;
- сокращаем на x_2 и $\bar{x}_2$ и получаем $\bar{z}_1 \& \bar{z}_3$.

4. Выделим третью пару для склейки:

- по вертикали выделим общий терм $\bar{x}_3$ для обоих минтермов $\bar{z}_1 \& z_2$ и $z_1 \& z_2$;
- сокращаем на x_1 и $\bar{x}_1$ и получаем $z_2 \& \bar{z}_3$.

Карта Карно

Элементарные конъюнкции	x_3	$\bar{x}_3$
$\bar{z}_1 \& \bar{z}_2$		1
$\bar{z}_1 \& z_2$		1
$z_1 \& z_2$	1	1
$z_1 \& \bar{z}_2$		

Рис. 2.2

Минимизированная форма булевой функции имеет вид:

$f(z_1, z_2, z_3) = (z_1 \& z_2) \vee (\bar{z}_1 \& \bar{z}_3) \vee (z_2 \& \bar{z}_3)$.

- **Минимизация методом Куайна.**

На первом этапе минимизации исходную СДНФ можно упростить, используя закон склеивания, путем попарного перебора конституент, а затем импликант.

На втором этапе используют таблицу Куайна, где по вертикали перечислены все полученные на первом этапе импликанты, а по горизонтали – исходные конституенты.

Ставим единицу на пересечении строк и столбцов, где импликанта «накрывает» коституенту.

На третьем этапе в каждой графе нужно исключить избыточные единицы.

Выбор единиц производится из соображений минимальности числа термов.

Пример 2.13

Минимизировать булеву функцию *f* (2,5,6,7,10,12,13,14) =1 методом Куайна.

Решение.

Переведем числа 2,5,6,7,10,12,13,14 в двоичную систему счисления, в которой:

2 - 0010; 5 – 0101; 6 – 0110; 7 – 0111; 10 – 1010; 12 – 1100; 13 – 1101; 14 – 1110.

Выразим двоичную функцию *f* (0010, 0101, 0110, 0111, 1010, 1100, 1101, 1110) =1 в СДНФ.

Получим: *f* (0010, 0101, 0110, 0111, 1010, 1100, 1101, 1110) =

$$= \begin{matrix} (\bar{z} \& \bar{u} \& y \& \bar{x}) \vee (\bar{z} \& u \& \bar{y} \& x) \vee (\bar{z} \& u \& y \& \bar{x}) \vee (\bar{z} \& u \& y \& x) \vee (z \& \bar{u} \& y \& \bar{x}) \vee \\ \vee (z \& u \& \bar{y} \& \bar{x}) \vee (z \& u \& \bar{y} \& x) \vee (z \& u \& y \& \bar{x}) \end{matrix}$$

Упростим полученное СДНФ, используя законы склеивания, перебирая контитуенты, а затем импликанты.

В результате получаем:

$$f(x,y,u,z) = (y \,\&\, \overline{x}) \vee (\overline{z} \,\&\, u \,\&\, x) \vee (\overline{z} \,\&\, u \,\&\, y) \vee (u \,\&\, \overline{y} \,\&\, x) \vee (z \,\&\, u \,\&\, \overline{y}) \vee (z \,\&\, u \,\&\, \overline{x})$$

Построим таблицу Куайна (таб. 2.12):

Таблица 2.12

Таблица Куайна

z, u, y, x	0010	0101	0110	0111	1010	1100	1101	1110
– –10	**1**		**1**		**1**			**1**
01 – 1		**1**		**1**				
011 –			1	1				
– 101		1					1	
110 –						**1**	**1**	
11 – 0						1		1

Ставим единицы там, где импликанта «накрывает» конституенту.

После заполнения таблицы Куайна получили по две единице в каждом столбце, между тем, достаточно иметь только одну единицу в столбце. Поэтому по возможности исключаем избыточные единицы. Выбор единиц производится из соображения минимальности термов.

Выбираем строку «– –10» по максимальному числу единиц и выделяем их полужирным шифром. Поскольку в столбце «0110» единица уже имеется, то вторую единицу можно уже исключить («011–» – исключается). Аналогично исключается строка «11 – 0».

В строке «01–1» выделяем полужирным шрифтом единицы, которые исключают строки «–101» в столбце 0101 и «011–» в столбце 0111. Два оставшихся столбца «накрываются» единицами строки «110–».

В итоге минимизированная функция состоит из трех импликант:

$$f(x,y,u,z) = (\overline{x} \,\&\, y) \vee (x \,\&\, u \,\&\, \overline{z}) \vee (\overline{y} \,\&\, u \,\&\, z)$$

2.5. *Классы булевых функций*

Общее число булевых функций равно 2^{2^n}. Из множества этих функций выделяют некоторые классы функций, которые играют важную роль в теории булевых функций. Такими классами являются: класс функций, сохраняющих константу 0, класс функций, сохраняющих константу 1, класс линейных функций, класс монотонных функций, класс самодвойственных функций.

- Функций $f(x_1, x_2, \ldots, x_n)$ принадлежит классу сохраняющих константу 0 (K_0), если $f(0,0,\ldots 0)=0$.

Пример 2.14

Определить, принадлежит ли классу K_0 функция $f(x,y) = x \oplus y$.

Решение. $f(0,0) = 0 \oplus 0 = 0$ – функция принадлежит классу K_0.

Пример 2.15

Определить, принадлежит ли классу K_0 функция $f(x,y) = x \Leftrightarrow y$.

Решение. $f(0,0) = 0 \Leftrightarrow 0 = 1$ – функция не принадлежит классу K_0.

- Функций $f(x_1, x_2, \ldots, x_n)$ принадлежит классу сохраняющих константу 1 (K_1), если $f(1,1,\ldots 1)=1$

Пример 2.16

Определить, принадлежит ли классу K_1 функция $f(x,y)=x\oplus y$.

Решение. $f(1,1)=1\oplus 1=0$ – функция не принадлежит классу K_1.

Пример 2.17

Определить, принадлежит ли классу K_1 функция $f(x,y)=x\Leftrightarrow y$.

Решение. $f(1,1)=1\Leftrightarrow 1=1$ – функция принадлежит классу K_1.

- Функций $f(x_1,x_2,\ldots,x_n)$ называется линейной функцией, если она может быть представлена в виде: $c_0\oplus c_1x_1\oplus c_2x_2\oplus\ldots\oplus c_nx_n$.

Множество всех линейных функций обозначается L.

Выражение $c_0\oplus c_1x_1\oplus c_2x_2\oplus\ldots\oplus c_nx_n$ называется полиномом Жегалкина.

Теорема Жегалкина. Любую булеву функцию можно представить единственным полиномом Жегалкина.

Полином Жегалкина строится с помощью треугольника Паскаля. Верхняя строка треугольника — это строка значений исходной функции. В каждой строке, начиная со второй, любой элемент треугольника равен сумме по модулю 2 двух соседних элементов предыдущей строки.

Выбираем из левой стороны треугольника Паскаля единицы, в которых определены соответствующие наборы переменных исходной функции. Набору (000) соответствует слагаемое 1. Соединив переменные знаком конъюнкции, мы получим слагаемое в полиноме Жегалкина.

Пример 2.18

Построить полином Жегалкина для функции заданной следующей таблицей истинности (таб. 2.13):

Таблица 2.13

Таблица истинности булевой функции

x_1	x_2	x_3	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Решение.

Заполним таблицу и построим треугольник Паскаля (таб.2.14):

Таблица 2.14

x_1	x_2	x_3	f	Треугольник Паскаля	Слагаемые
0	0	0	0	01100100	1
0	0	1	1	**1010110**	**x_3**
0	1	0	1	**111101**	**x_2**
0	1	1	0	00011	x_2&x_3
1	0	0	0	0010	x_1
1	0	1	1	011	x_1&x_3
1	1	0	0	**10**	**x_1&x_2**
1	1	1	0	**1**	**x_1&x_2&x_3**

Выбираем слагаемые для полинома Жегалкина из тех строк, где слева в треугольнике Паскаля стоят единицы. Это слагаемые x_3, x_2, (x_1&x_2) **и** (x_1&x_2&x_3).

Из данных слагаемых составим полином Жегалкина, который будет иметь вид:

$$x_3 \oplus x_2 \oplus (x_1 \& x_2) \oplus (x_1 \& x_2 \& x_3)$$

Функция $f^*(x_1, x_2, ..., x_n) = \overline{f}(\overline{x_1}, \overline{x_2}, ..., \overline{x_n})$ *называется двойственной* функцией для функции *f* (x_1, x_2, …, x_n).

При построении двойственной функции на место каждой переменной ставится ее отрицание и берется отрицание от всей функции.

Пример 2.19

Определить двойственную функцию для функции $f(x, y) = x \& y$

Решение.

$\overline{f}(\overline{x}, \overline{y}) = \overline{\overline{x} \& \overline{y}} = \overline{\overline{x}} \vee \overline{\overline{y}} = x \vee y$

Функция $x \vee y$ двойственная для функции $x \& y$: $(x \& y)^* = x \vee y$

Двойственную функцию можно получить и по таблице истинности исходной функции *f*.

Для этого строим функцию $\overline{f}$ и записываем ее значения в обратном порядке. Получим таблицу истинности двойственной функции.

Пример 2.20

Определить двойственную функцию для функции *f* (0010)

Решение.

Определим $\overline{f}$: $\overline{f}$ (1101). Поставим набор значений в обратном порядке, получим двойственную функцию: f^* (1011).

Функция, совпадающая со своей двойственной функцией, называется ***самодвойственной функцией.***

Класс самодвойственных функций обозначается S.

Критерий самодвойственности функции. Для самодвойственности функции необходимо и достаточно, чтобы на любых двух противоположных наборах значений переменных функция принимала разные значения.

Пример 2.21

Определить является ли функция *f* (10101110) самодвойственной функцией.

Решение.

Определим $\overline{f}$: $\overline{f}$(01010001). Поставим набор значений в обратном порядке, получим двойственную функцию: f^* (10001010). $f \neq f^*$. Функция не является самодвойственной.

Пример 2.22

Определить является ли функция *f* (01110001) самодвойственной функцией.

Решение.

Определим $\overline{f}$: $\overline{f}$(10001110). Поставим набор значений в обратном порядке, получим двойственную функцию: f^* (01110001). $f = f^*$. Функция является самодвойственной.

Если для двух наборов длины n из нулей и единиц x = (x_1, x_2, …, x_n) и y = (y_1, y_2, …, y_n) выполнены условия $x_1 \leq y_1, x_2 \leq y_2, ..., x_n \leq y_n, \text{то } x \leq y$. При невыполнении этих условий наборы x и y *несравнимы*. Наборы разной длины также являются несравнимы.

Пример 2.23

Определить, сравнимы ли наборы (1,0) и (1,0,0).

Решение.

Так как наборы (1,0) и (1,0,0) разной длины, то они являются несравнимыми.

Пример 2.24

Определить, сравнимы ли наборы (0,1,1,0) и (0,1,0,0).

Решение.

Так как 0=0, 1=1, 1>0, 0=0, то наборы (0,1,1,0) и (0,1,0,0) – сравнимы.

Пример 2.25

Определить, сравнимы ли наборы (0,1,0,1) и (0,1,1,0).

Решение.

Так как 0=0, 1=1, 0<1, 1>0 – наборы несравнимы.

- Функция *f* (x_1, x_2, …, x_n) называется ***монотонной,*** если для всех наборов
x = (x_1, x_2, …, x_n) и
y = (y_1, y_2, …, y_n) из условия $x \leq y$ следует условие $f(x) \leq f(y)$.

Пример 2.26

Определить, является ли функция, представленная таблицей 2.15 монотонной функцией.

Таблица 2.15

Таблица истинности булевой функции

x_1	x_2	x_3	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Решение.

Так как (0,1,0) $\leq$ (0,1,1), а *f* (0,1,0,)> *f* (0,1,1), то функция *f* не является монотонной функцией.

2.6. Теорема Поста

Рассмотрим систему булевых функций:

$$\begin{cases} f(x_1, x_2, \dots, x_n) \\ f_1(x_1, x_2, \dots, x_m) \\ \qquad \dots \\ f_n(x_1, x_2, \dots, x_m) \end{cases} \qquad (1^*)$$

Функция $f(x_1, x_2, \dots, x_n) = f(f_1(x_1, x_2, \dots, x_m), \dots, f_n(x_1, x_2, \dots, x_m))$ является *суперпозицией* функций системы (1^*).

Множество булевых функций называется замкнутым, если в результате суперпозиции функций рассматриваемого множества, могут быть получены только функции, принадлежащие данному множеству.

Теорема. Каждый из классов булевых функций, а именно, класс функций, сохраняющих константу 0, класс функций, сохраняющих константу 1, класс линейных функций, класс монотонных функций, класс самодвойственных функций и множество всех булевых функций является замкнутым.

Система функций (1^*) называется полной на множестве всех булевых функций, если в результате суперпозиции функций этой системы можно получить любую булеву функцию.

Теорема Поста. Для того, чтобы система булевых функций была полной, необходимо и достаточно, чтобы она содержала хотя бы одну функцию: не сохраняющую константу 0, не сохраняющую константу 1, не самодвойственную, не линейную, не монотонную.

Пример 2.27

Проверить штрих Шеффера на полноту.

Решение.

Представим функцию *f* (x, y) = x|y в виде таблицы (таб. 2.16)

Таблица 2.16

Таблица истинности штрих Шеффера

x	y	*f*
0	0	1
0	1	1
1	0	1
1	1	0

1. *f* (0,0) = 1 – не принадлежит классу K_0;

2. *f* (1,1) = 0 – не принадлежит классу K_1;

3. (0,0) $\leq$(1,1), а f (0,0)> *f* (1,1) – не монотонная;

4. $f(1,1,1,0)$, а $f^*(0,0,0,1)$, $f \neq f^*$ – не самодвойственная;

5. Построим полином Жегалкина

x	y	Треугольник Паскаля	Слагаемые
0	0	1110	1
0	1	001	y
1	0	01	x
1	1	1	xy

$1 \oplus xy$ – не линейная функция.

Ответ. Штрих Шеффера образует полную систему функций.

Пример 2.28

Проверить на полноту систему $\{\vee, \oplus\}$.

Решение.

1. $x \vee y = (0 \vee 0) = 0$ и $x \oplus y = (0 \oplus 0) = 0$. Обе функции принадлежат классу K_0, следовательно, система не является полной.

Пример 2.29

Проверить на полноту систему $\{\vee, \neg\}$.

1. $f(\neg x) = (\neg 0) = 1$ – не принадлежит классу K_0;

2. $f(\neg x) = (\neg 1) = 0$ – не принадлежит классу K_1;

3. *0<1, а* $f(\neg 0) > f(\neg 1)$ – не принадлежит классу монотонных функций;

4. $f(0,0,0,1)$, а $f^*(1,1,1,0)$, $f \neq f^*$ - не самодвойственная функция;

5. $y \oplus x \oplus xy$ – не принадлежит классу линейных функций.

Система $\{\vee, \neg\}$ является полной.

Контрольные вопросы для самопроверки

Тема 2 «Основы математическая логика»

1. Простое высказывание. Определение. Пример.
2. Сложное высказывание. Определение. Пример.
3. Логические функции: конъюнкция. Определение. Таблица истинности.
4. Логические функции: дизъюнкция. Определение. Таблица истинности.
5. Логические функции: инверсия. Определение. Таблица истинности.
6. Логические функции: импликация. Определение. Таблица истинности.
7. Логические функции: эквивалентность. Определение. Таблица истинности.
8. Логические функции: сумма по модулю два. Определение. Таблица истинности.
9. Суперпозиция логических функций; логические формулы; равносильные логические формулы.
10. Дизъюнктивная нормальная форма логической функции.
11. Конъюнктивная нормальная форма логической функции.
12. Булевы операции и их основные свойства. Привести доказательство одного из свойств.
13. Функционально полная система логических функций: определение, пример.

Тема 3. Функционально-бинарные отношения

3.1. Отображения

Бинарное отношение называется функциональным, если из aRb и aRc следует b=c.

Таким образом, бинарное отношение называется функциональным, если оно не содержит двух различных упорядоченных пар с одинаковой первой компонентой.

Так как всякое бинарное множество рассматривается как подмножество прямого произведения $R \subset A \times B$, то в этом случае говорят, что задано отношение R от A до B. Стоит отметить, что множество A может не совпадать с R_-, а множество B не совпадает с R_+ бинарного отношения R.

Функциональное отношение $R \subset A \times B$ называется *отображением*, если $R_- = A$.

Если $R \subset A \times B$ – отображение, то записывают R: $A \to B$.

В отображении R: $A \to B$, где имеются упорядоченные пары aRb, $a \in A$, $b \in B$, элемент a называется прообразом элемента b при отображении R, а элемент b – образом элемента a при отображении R.

Отображение $R \subset A \times B$ называется функцией, если множества A и B числовые.

Если рассматриваются *n* – местные функции (функции от нескольких переменных), то функция *f* от *n* переменных есть отображения множества A^n в множество B: $A^n \to B$.

Если отображение *f* есть функция отA до B, и $b \in B$ есть образ прообраза $a \in A$, то это записывается в виде: $f(a)=b$.

Принципиальным отличием отображения и функции состоит в том, что отображение определяется для любого множества A и B, а функция только на числовых множествах.

Классификация отображений и функций

Различают сюръективное, инъективное и биективное отображение (функцию).

- отображение (функция) называется сюръективным, если $R_+ = b$;

- отображение (функция) называется инъективным, если в нем разным прообразам соответствуют разные образы, т.е. из условия $a \neq b$ следует условие $F(a) \neq F(b)$;

- отображение (функция) называется биективным (взаимно однозначное), если оно сюръективное и инъективное.

3.2. Подстановка

Рассмотрим множество $X_n = \{1, 2, \ldots, n\}$, где $n \in N$ и каждый элемент представлен один раз.

Взаимно однозначное отображение ϕ множества X на себя называется подстановкой:

$\phi : X_n \to X_n$.

Отношение $X_n \to X_n$ бинарное, которое записывается в виде матрицы:

$\phi = \begin{pmatrix} x_1 & x_2 & x_3 & \ldots & x_n \\ \phi(x_1) & \phi(x_2) & \phi(x_3) & \ldots & \phi(x_n) \end{pmatrix}$, где в первой строке представлены прообразы, а во второй – образы.

Если прообразы записаны в порядке возрастания, то такую подстановку называют *канонической.*

Подстановка вида ϕ= (1, 2, ..., *n) =e* называется *тождественной,* так как для любого прообраза справедливо равенство $e(x_i) = x_i$.

Произведение подстановок

Произведение подстановок осуществляется в соответствии с общим правилом композиции: *(vm)(i)= v(m(i))*.

Рассмотрим подстановки:

$$\eta = \begin{pmatrix} x_1 & x_2 & x_3 & \dots & x_n \\ \eta(x_1) & \eta(x_2) & \eta(x_3) & \dots & \eta(x_n) \end{pmatrix} \quad \phi = \begin{pmatrix} x_1 & x_2 & x_3 & \dots & x_n \\ \phi(x_1) & \phi(x_2) & \phi(x_3) & \dots & \phi(x_n) \end{pmatrix}$$

Произведение данных подстановок запишем в виде:

$$\eta \cdot \phi = \begin{pmatrix} x_1 & x_2 & x_3 & \dots & x_n \\ \eta(x_1) & \eta(x_2) & \eta(x_3) & \dots & \eta(x_n) \end{pmatrix} \cdot \begin{pmatrix} x_1 & x_2 & x_3 & \dots & x_n \\ \phi(x_1) & \phi(x_2) & \phi(x_3) & \dots & \phi(x_n) \end{pmatrix}$$

С помощью перестановки столбцов нужно добиться, чтобы первая строка в первой подстановке совпадала со второй строкой второй подстановки, тогда результатом произведения будет подстановка, в которой первая строка совпадает с первой строкой второго множителя, а вторая строка со второй строкой первого множителя.

Пример 3. 2.1

Найти произведение подстановок $\phi \cdot \eta \quad u \quad \eta \cdot \phi$, если

$$\phi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 3 & 1 \end{pmatrix} \qquad \eta = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 4 & 2 \end{pmatrix}$$

Решение.

$$\phi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 3 & 1 \end{pmatrix} \qquad \eta = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 4 & 2 \end{pmatrix}$$

$$\phi \cdot \eta = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 3 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 4 & 2 \end{pmatrix} = \begin{pmatrix} 3 & 1 & 4 & 2 \\ 3 & 2 & 1 & 4 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 4 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 2 & 1 & 4 \end{pmatrix}$$

$$\eta \cdot \phi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 4 & 2 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 3 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 4 & 3 & 1 \\ 1 & 2 & 4 & 3 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 3 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 4 & 3 \end{pmatrix}$$

Натуральной степенью подстановки ϕ называется подстановка $\phi^n = \underbrace{\phi \cdot \phi \cdot \ldots \cdot \phi}_{n}$ (произведение n подстановок каждая из которых равна ϕ).

Обратная подстановка.

Для каждой подстановки ϕ можно указать такую обратную подстановку ϕ^{-1}, при которой выполняется равенство: $\phi\, \phi^{-1} = \phi^{-1} \phi = e$.

Для нахождения обратной подстановки необходимо в исходной подстановке поменять местами строки, а затем меняя местами столбцы привести подстановку к каноническому виду.

Пример 3.2.2

Для подстановки $\eta = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 4 & 2 \end{pmatrix}$ найти обратную подстановку.

Решение.

$$\eta = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 4 & 2 \end{pmatrix}$$

$$\eta^{-1} = \begin{pmatrix} 3 & 1 & 4 & 2 \\ 1 & 2 & 3 & 4 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 1 & 3 \end{pmatrix}$$

$$\eta \cdot \eta^{-1} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 4 & 2 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 1 & 3 \end{pmatrix} = \begin{pmatrix} 2 & 4 & 1 & 3 \\ 1 & 2 & 3 & 4 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 1 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix}$$

Свойства произведения:

1. Умножение выполняется только для подстановок одной степени.
2. Для умножения подстановок не выполняется коммуникативный закон.
3. Для умножения подстановок выполняется сочетательный закон.
4. Подстановка не изменяется, если ее умножить на тождественную подстановку.
5. Умножение данной подстановки и обратной равно тождественной подстановке.
6. Обратная подстановка единственна.
7. Подстановка обратная единичной также является тождественной.
8. При умножении степеней подстановки, достаточно возвести данную подстановку в степень, которая равна сумме данных степеней.

Циклы подстановки

Подстановка вида $i_1 \to i_2, i_2 \to i_3, ..., i_{m-1} \to i_m, i_m \to i_1$ называется циклом длины m.

Цикл длины 2 называется транспозицией. Циклы, не содержащие общие элементы, называются независимыми.

Теорема. Каждую подстановку можно разложить в произведение независимых циклов. Это разложение единственно с точностью до порядка циклов.

Пример 3.2.3

Разложить подстановку $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 3 & 4 & 5 & 2 & 1 & 6 & 6 & 8 \end{pmatrix}$ в произведение независимых циклов.

Решение.

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 3 & 4 & 5 & 2 & 1 & 7 & 6 & 8 \end{pmatrix} = (1,3,5) \cdot (2,4) \cdot (6,7) \cdot (8)$$

Знак подстановки

Инверсией подстановки называется рокировка (перестановка) двух соседних значений в нижней строке канонической подстановки. При этом подстановка меняется.

Пусть n число инверсий, приводящих подстановку к тождественной, тогда функция $\varepsilon : \eta \to (-1)^n$ называется четностью подстановки η. Если $\varepsilon(\eta) = 1$, то подстановка называется четной, если $\varepsilon(\eta) = -1$, то подстановка называется нечетной.

Пример 3. 2

Найти число инверсий, четность подстановки $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 6 & 4 & 3 & 1 & 2 & 5 \end{pmatrix}$.

Решение.

$$\begin{pmatrix}1&2&3&4&5&6\\6&4&3&1&2&5\end{pmatrix}\to\begin{pmatrix}1&2&3&4&5&6\\6&4&1&3&2&5\end{pmatrix}\to\begin{pmatrix}1&2&3&4&5&6\\6&1&4&3&2&5\end{pmatrix}\to\begin{pmatrix}1&2&3&4&5&6\\1&6&4&3&2&5\end{pmatrix}\to$$

$$\to\begin{pmatrix}1&2&3&4&5&6\\1&6&4&2&3&5\end{pmatrix}\to\begin{pmatrix}1&2&3&4&5&6\\1&6&2&4&3&5\end{pmatrix}\to\begin{pmatrix}1&2&3&4&5&6\\1&2&6&4&3&5\end{pmatrix}\to\begin{pmatrix}1&2&3&4&5&6\\1&2&6&3&4&5\end{pmatrix}\to$$

$$\to\begin{pmatrix}1&2&3&4&5&6\\1&2&3&6&4&5\end{pmatrix}\to\begin{pmatrix}1&2&3&4&5&6\\1&2&3&4&6&5\end{pmatrix}\to\begin{pmatrix}1&2&3&4&5&6\\1&2&3&4&5&6\end{pmatrix}$$

Таким образом, число инверсий равно 10, отсюда $(-1)^{10} = 1$, следовательно, подстановка четная.

Транспозицией подстановки называется рокировка (перестановка) двух любых значений в нижней строке канонической подстановки. Инверсия частный случай транспозиции. Поэтому четное (нечетное) число транспозиций не меняет (меняет) знак подстановки.

Пример 3. 2.4

Привести подстановку $\begin{pmatrix}1&2&3&4&5&6\\6&4&3&1&2&5\end{pmatrix}$ с помощью транспозиций к тождественной.

Решение.

$$\begin{pmatrix}1&2&3&4&5&6\\6&4&3&1&2&5\end{pmatrix}\to\begin{pmatrix}1&2&3&4&5&6\\1&4&3&6&2&5\end{pmatrix}\to\begin{pmatrix}1&2&3&4&5&6\\1&2&3&6&4&5\end{pmatrix}\to\begin{pmatrix}1&2&3&4&5&6\\1&2&3&4&6&5\end{pmatrix}\to$$

$$\to\begin{pmatrix}1&2&3&4&5&6\\1&2&3&4&5&6\end{pmatrix}$$

Таким образом, число транспозиций равно 4, отсюда $(-1)^4 = 1$, следовательно, подстановка четная.

Контрольные вопросы для самопроверки

Тема 3 «Функционально-бинарные отображения»

1. Понятия множества и элемента множества, примеры. Подмножество, собственное подмножество: определения, примеры. Универсальное множество: определение, пример.
2. Равные множества: определение, пример. Мощность конечного множества, пустое множество: определение, примеры.
3. Способы задания множества.
4. Объединение и пересечение множеств: определения и примеры.
5. Разность множеств, дополнение множества: определения и примеры.
6. Основные тождества теории множеств. Привести пример доказательства для одного из тождеств.
7. Бинарное отношение на множестве: определение и пример.
8. Способы задания бинарных отношений.
9. Свойства бинарного отношения: рефлективность, симметричность, транзитивность.
10. Отношение эквивалентности, разбиение множества на классы эквивалентности.
11. Антирефлексивность и антисимметричность; отношения строгого и нестрогого порядков.

Тема 4. Алгоритмы

4.1. Понятие алгоритма

Алгоритм – набор инструкций, описывающих порядок действий исполнителя для достижения некоторого результата.

Основные требования, предъявляемые к алгоритмам:

1. **Возможность работы с множеством данных.** Каждый алгоритм должен применяться к некоторому множеству данных, которые называют *входными данными*. В ходе его работы появляются промежуточные результаты, которые называют *промежуточными данными*. По завершении работы каждый алгоритм должен выдавать некоторые результаты, которые называют выходными данными. Таким образом, данные, с которыми должен работать алгоритм делятся на входные, промежуточные и выходные. В качестве данных выступают: числа, матрицы, векторы и т.д.
2. **Наличие памяти и возможность обращаться к ней.** Под памятью алгоритма понимается возможность в течение определенного времени хранения информации о необходимых для работы данных.
3. **Свойство дискретности.** Алгоритм должен представлять собой набор четко определенных действий, число которых конечно.
4. **Свойство детерминированности.** Переход от одного действия (шага) к другому осуществляется по строго определенным правилам (возможность осуществлять эти действия случайным образом исключена).
5. **Свойство результативности.** Через конечное число шагов должно быть получено некоторое множество выходных данных.
6. **Возможность реализации.** Под возможностью реализации алгоритма понимание наличие средств, с помощью которых может быть организован процесс работы этого алгоритма.

4.2. Алгоритмы упорядочивания множества

Рассмотрим неупорядоченное множество A = {a_1, a_2, …, a_n}, элементы которого записаны в памяти компьютера как элементы массива: A = (a[1], a[2], …, a[n]), где a[i] – содержимое ячейки с номером i.

В некоторых приложениях требуется упорядочить запись элементов в ячейках массива так, чтобы выполнялось соотношение a [i] $\le$ a [j], если i < j, где i и j – номера ячеек массива.

Символ $\le$ определяет отношения порядка. Процедуру упорядочивания элементов массива называют *сортировкой.*

Существует ряд методов упорядочивания множеств.

- **Пузырьковая сортировка.**

Рассматривать данный метод удобно на основе вектора – столбца, элементы которого являются элементы массива.

Метод состоит в том, что на каждой итерации метода наиболее «легкий» элемент (элемент с меньшим значением) оказывается выше всех более «тяжелых» элементов массива (элементы с большим значением). Это и дает название методу.

Каждая k- я итерация начинается рассматриваться с последнего n-го элемента массива. При рассмотрении i –го элемента массива, происходит его сравнение с (i-1) –м элементом. Если a[i-1] < a[i], то элементы массива меняются местами. В противном случае расположение элементов не меняется. Далее происходит переход к рассмотрению (i-1) – го элемента массива. Итерация заканчивается при сравнении a[k+1] –го элемента с элементом a[k]. В результате сравнения самых «легкий» элемент оказывается «наверху».

Сортировка заканчивается, когда процесс ее выполнения не будет ни одной замены местами элементов массива.

Пример 4. 2.1

Пузырьковым методом отсортировать массив целых чисел A {8,4,6,3,7,2,1,5}.

Решение.

$$\begin{pmatrix}8\\4\\6\\3\\7\\2\\1\\5\end{pmatrix}\Rightarrow\begin{pmatrix}1\\8\\4\\6\\3\\7\\2\\5\end{pmatrix}\Rightarrow\begin{pmatrix}1\\2\\8\\4\\6\\3\\7\\5\end{pmatrix}\Rightarrow\begin{pmatrix}1\\2\\3\\8\\4\\6\\5\\7\end{pmatrix}\Rightarrow\begin{pmatrix}1\\2\\3\\4\\8\\5\\6\\7\end{pmatrix}\Rightarrow\begin{pmatrix}1\\2\\3\\4\\5\\8\\6\\7\end{pmatrix}\Rightarrow\begin{pmatrix}1\\2\\3\\4\\5\\6\\8\\7\end{pmatrix}\Rightarrow\begin{pmatrix}1\\2\\3\\4\\5\\6\\7\\8\end{pmatrix}$$

Число операций, которые нужно выполнить (в худшем случае) для получения отсортированного массива определяется: $\sum_{k=1}^{n=1} 2(n-k) = 2n^2 - 2n - 2\frac{n(n-1)}{2} = n^2 - n$ операциями, поскольку, на первой итерации требуется выполнить n-1 сравнений, и, возможно, столько же перемен местами сравниваемых элементов. На второй итерации уже потребуется выполнить n-2 сравнений и столько же перемен местами элементов. На k-ой итерации потребуется выполнить n-k сравнений и столько же перемен местами элементов.

- **Сортировка выбором**.

В процессе сортировки находится наименьший элемент, содержащийся в рассматриваемой части массива, который помещается взамен его первого элемента. Затем процедура повторяется – рассматривается неотсортированная часть массива, которая уменьшена по сравнению с предыдущей итерацией на один элемент.

Таким образом, на k-й итерации, как и в пузырьковой сортировке, рассматривают массив a[k]…a[n], находят наименьший элемент, меняют его с a[k]. Сортировка заканчивается после выполнения (n-1) –й итерации.

Пример 4.2.2

Сортировкой выбором упорядочить массив A = {8,4,6,3,7,2,1,5}.

Решение.

(8,4,6,3,7,2,1,5) – исходный массив;
(1,4,6,3,7,2,8,5) – 1-я итерация;
(1,2,6,3,7,4,8,5) – 2-я итерация;
(1,2,3,6,7,4,8,5) – 3 –я итерация;
(1,2,3,4,7,6,8,5) – 4-я итерация;
(1,2,3,4,5,6,8,7) – 5-я итерация;
(1,2,3,4,5,6,8,7) – 6-я итерация;
(1,2,3,4,5,6,7,8) – 7-я итерация;

Пример 4.2.3

Сортировкой выбором упорядочить массив B = {3,2,7,9,4,5,8,6,1}

Решение.

(3,2,7,9,4,5,8,6,1) – исходный массив;
(1,2,7,9,4,5,8,6,3) – 1-я итерация;
(1,2,7,9,4,5,8,6,3) – 2-я итерация;
(1,2,3,9,4,5,8,6,7) – 3-я итерация;
(1,2,3,4,9,5,8,6,7) – 4-я итерация;
(1,2,3,4,5,9,8,6,7) – 5-я итерация;
(1,2,3,4,5,6,8,9,7) – 6-я итерация;
(1,2,3,4,5,6,7,9,8) – 7-я итерация;
(1,2,3,4,5,6,7,8,9) – 8-я итерация.

На первой итерации требуется выполнить n-1 сравнений и две операции перемен двух элементов местами, на второй итерации требуется выполнить n-2 сравнений и две операции перемещения элементов, а на k-й итерации выполняется n-k сравнений и снова две операции перемен элементов местами. Тогда число операций, необходимых для сортировки массива, в худшем случае, определяется $\sum_{k=1}^{n=1}(n-k+2)=n^2-n-\frac{n(n-1)}{2}+2(n-1)=\frac{n^2+3n}{2}-2$ операциями.

- Сортировка вставками.

Так как на первом шаге метода полагается, что первый элемент массива уже отсортирован, тогда на k-й итерации метода имеем отсортированную часть a [1], …, a[k] массива и неотсортированную часть массива a [k+1], …, a[n]. К первой части присоединяем новый элемент a [k+1] и осуществляется поиск места его расположения. При сравнении его с a[k] элементом при условии a[k]> a [k+1], элементы меняются местами. Итерация заканчивается, когда в результате сравнения не происходит перемены мест этих элементов, тогда остается выполнить n-1 итераций.

Пример 4.2.4

Сортировкой вставкой упорядочить массив A = {8,4,6,3,7,2,1,5}.

Решение.

(8,4,6,3,7,2,1,5) – исходный массив;
(4,8,6,3,7,2,1,5) – 1-я итерация;
(4,6,8,3,7,2,1,5) – 2-я итерация;
(3,4,6,8,7,2,1,5) – 3-я итерация;
(3,4,6,7,8,2,1,5) – 4-я итерация;
(2,3,4,6,7,8,1,5) – 5 -я итерация;
(1,2,3,4,6,7,8,5) – 6 -я итерация;
(1,2,3,4,5,6,7,8) – 7-я итерация.

На первой итерации требуется выполнить одно сравнение и одну операцию перемен местами сравниваемых элементов. На второй итерации соответственно осуществляется два сравнения и столько же изменений местами элементов. На k-й итерации потребуется выполнить k сравнений и столько же перемен местами элементов, следовательно, в худшем случае потребуется выполнить $\sum_{k=1}^{n=1}2k=2\frac{n(n-1)}{2}=n^2-n$ операций.

- Квадратичная сортировка.

Квадратичная сортировка имеет меньшее число сравнений по сравнению с быстрой сортировкой, однако требует дополнительного объема памяти.

Исходный массив разделяют на m групп A_1, A_2, …A_m по $\sqrt{n}$ элементов в каждой.

На первой итерации из каждой группы выбирается наименьший элемент, который пересылается во вспомогательный массив U, вмещающий m элементов. В массиве U находится наименьший элемент и пересылается в результирующий массив. Массив U пополняется наименьшим элементом из той группы, которой принадлежал элемент, отправленный в результирующий массив. На этом первая итерация заканчивается. На второй итерации опять находится наименьший элемент в массиве U и оправляется в результирующий массив, после чего массив U пополняется наименьшим элементом по аналогии с первой итерацией. В каждой группе выбранный элемент заменяется большим фиктивным элементом. Работа заканчивается тогда, когда все группы будут заполнены фиктивными элементами.

Пример 4.2.5

Квадратичной сортировкой упорядочить массив B = {3,2,7,9,4,5,8,6,1}.

Решение.

Так как в массиве 9 элементов разобьем его на три группы: $\sqrt{9} = 3$

A_1 = {3,2,7}, A_2 = {9,4,5}, A_3 = {8,6,1}

Первая итерация:

Формируем вспомогательный массив, заполнив его наименьшими элементами из каждой группы: U = {2,4,1}

A_1 = {3,7}, A_2 = {9,5}, A_3 = {8,6}

Оправляем наименьший элемент из вспомогательного массива в результирующий массив: R = {1}

Вторая итерация:

Пополняем вспомогательный массив, наименьшим элементом из группы A_3:
U = {2,4,6}

A_1 = {3,7}, A_2 = {9,5}, A_3 = {8}

Оправляем наименьший элемент из вспомогательного массива в результирующий массив: R = {1,2}

Третья итерация:

Пополняем вспомогательный массив, наименьшим элементом из группы A_1:
U = {3,4,6}

A_1 = {7}, A_2 = {9,5}, A_3 = {8}

Оправляем наименьший элемент из вспомогательного массива в результирующий массив: R = {1, 2, 3}

Четвертая итерация:

Пополняем вспомогательный массив, наименьшим элементом из группы A_1:
U = {7,4,6}

A_1 = {*f*}, A_2 = {9,5}, A_3 = {8}

Оправляем наименьший элемент из вспомогательного массива в результирующий массив: R = {1, 2, 3, 4}

Пятая итерация:

Пополняем вспомогательный массив, наименьшим элементом из группы A_2:
U = {7, 5, 6}

A_1 = {*f*}, A_2 = {9}, A_3 = {8}

Оправляем наименьший элемент в результирующий массив: R = {1, 2, 3, 4, 5}

Шестая итерация:

Пополняем вспомогательный массив, наименьшим элементом из группы A_2:

U = {7, 9, 6}

$$A_1 = \{f\},\ A_2 = \{f\},\ A_3 = \{8\}$$

Оправляем наименьший элемент в результирующий массив: R = {1, 2, 3, 4, 5, 6}

Седьмая итерация:

Пополняем вспомогательный массив, наименьшим элементом из группы A_2:

U = {7, 9, 8}

$$A_1 = \{f\},\ A_2 = \{f\},\ A_3 = \{f\}$$

Оправляем наименьший элемент в результирующий массив: R = {1, 2, 3, 4, 5, 6, 7}

U = {*f*, 8, 9}

Восьмая итерация:

Оправляем наименьший элемент в результирующий массив: R = {1, 2, 3, 4, 5, 6, 7, 8}

U = {*f*, *f*, 9}

Девятая итерация:

Оправляем наименьший элемент в результирующий массив:

R = {1, 2, 3, 4, 5, 6, 7, 8, 9} U = {*f*, *f*, *f*}.

Всего выполняется n итераций, поэтому для упорядочивания массива квадратичной сортировкой необходимо выполнить $(\sqrt{n} + m) \cdot n$ сравнений.

- **Быстрая сортировка.**

Вначале сортировки выбирается последний элемент и называется *опорным.* Затем вводятся два понятия «*наименьший индекс*», который называют «стена» и «*текущий элемент*», первый левый элемент от «стены».

Сравниваем все элементы с опорным элементом: все элементы больше опорного остаются на своих местах (справа от стены), а элементы меньше опорного меняются местами с текущим элементом и перемещаются левее от стены. В результате сравнения массив разделится на две части: элементы меньше опорного и элементы больше опорного. Затем опорный элемент меняем местами с текущим элементом, т.е. располагаем его между меньшими и большими элементами и продолжаем сортировку с новым опорным элементом.

Пример 4.2.6

Упорядочить массив B = {3,2,7,9,4,5,8,6,1} методом быстрой сортировки.

Решение.

B = {3,2,7,9,4,5,8,6,1}. Опорный элемент 1, текущий элемент 3.

1. Все элементы больше опорного 1, следовательно, остаются на своих местах (справа от 3).

Меняем местами опорный элемент 1 и текущий 3 – (1|2,7,9,4,5,8,6,3) и располагаем единицу левее «стены».

2. Опорный элемент 3, текущий элемент 2.

Сравниваем все элементы с опорным элементом 3. Текущий элемент 2 меньше опорного элемента 3, следовательно, перемещаем его левее «стены». (1,2 |7,9,4,5,8,6,3).

3. Опорный элемент 3, текущий элемент 7.

Все элементы левее «стены» больше 3, следовательно, остаются левее стены, а опорный элемент 3 меняется местами с текущим элементом 7 и перемещается левее «стены»:

(1,2,3|9,4,5,8,6,7).

4. Опорный элемент 7, текущий элемент 9.

Четыре меньше 7, поэтому меняем местами 4 и 9 (1,2,3 |4,9,5,8,6,7) и перемещаем «стену»

(1,2,3,4| 9,5,8,6,7) правее элемента 4.

5. Элемент 5 меньше 7, меняем местами текущий элемент 9 и 5 перемещаем «стену» (1,2,3,4, 5|9,8,6,7) правее элемента 5.
6. Элемент 8 больше опорного элемента 7, следовательно, остается правее «стены» (1,2,3,4, 5|9,8,6,7).
7. Элемент 6 меньше опорного элемента 7, следовательно, меняем местами текущий элемент 9 и 6 перемещаем «стену» (1,2,3,4, 5, 6 |8, 9,7) правее элемента
8. Опорный элемент 7, текущий элемент 8.

Все элементы больше опорного числа 7, следовательно, меняем местами 7 и 8 и переносим элемент 7 левее «стены» (1,2,3,4, 5, 6, 7 | 9,8).

9. Опорный элемент 8, текущий элемент 9.

Элемент 9 больше опорного элемента, следовательно, остается правее «стены».

10. В заключении располагаем опорный элемент 8 между меньшими и большими элементами массива, поменяв местами опорный и текущий элементы (1,2,3,4,5,6,7,8,9).

- **Сортировка Хоара.** Сортировка Хоара одна из разновидностей быстрой сортировки.

Вначале сортировки выбирается опорный элемент (первый в массиве) и сравнивается с остальными элементами. Если элемент больше опорного, то элементы местами не меняются, если меньше, то меняются местами.

После всех сравнений опорный элемент разбивает массив на две части (подмассивы) меньше опорного и больше опорного и больше в сравнении не участвует.

На каждом подмассиве исходная процедура продолжается до тех пор, пока длина каждого подмассива будет не более единицы.

Пример 4.2.7

Упорядочить массив B = {3,2,7,9,4,5,8,6,1} методом быстрой сортировки.

Решение.

3,2,7,9,4,5,8,6,1 - опорный элемент 3	
1,2,7, 9, 4, 5,8,6, 3	1 2 3 7, 4, 5, 8, 6, 9
1, 2, 7,9, 4, 5,8,6,3	1 2 3 7, 4, 5, 8, 6, 9
1, 2, 3, 9, 4, 5,8,6,7	1 2 3 7, 4, 5, 8, 6 9
1, 2, 3, 9, 4, 5,8,6,7	1 2 3 6, 4, 5, 8, 7 9
1, 2, 3, 9, 4, 5,8,6,7	1 2 3 6, 4, 5, 8, 7 9
1, 2, 3, 9, 4, 5,8,6,7	1 2 3 6, 4, 5, 8, 7 9
1, 2, 3, 9, 4, 5,8,6,7	1 2 3 6, 4, 5, 7, 8 9
1, 2, 3, 9, 4, 5,8,6,7	1 2 3 6, 4, 5 7 8 9
1 2 3 9, 4, 5,8,6,7	1 2 3 5, 4, 6 7 8 9
1 2 3 7, 4, 5, 8, 6, 9	1 2 3 5, 4, 6 7 8 9
1 2 3 7, 4, 5, 8, 6, 9	1 2 3 5, 4 6 7 8 9
1 2 3 7, 4, 5, 8, 6, 9	1 2 3 4, 5 6 7 8 9
	1 2 3 4 5 6 7 8 9

Контрольные вопросы для самопроверки

Тема 4 «Теория алгоритмов»

1. Алгоритм. Определения. Основные требования.
2. Рекурсивные функции.
3. Алгоритм сортировки: сортировка выбором.
4. Алгоритм сортировки: пузырьковая сортировка.
5. Алгоритм сортировки: быстрая сортировка.
6. Алгоритм сортировки: сортировка слиянием.
7. Алгоритм сортировки: сортировка вставками.

Тема 5. Теория графов

5.1. Основные понятия

Графом называют совокупность множеств вершин $V= \{v_1, v_2, \ldots, v_n\}$ и ребер (дуг) $E = \{e_1, e_2, \ldots, e_k\}$, обозначаемый $G= (V, E)$. Каждое ребро из множества E определяется парой вершин v_i и v_j, которые оно соединяет. Такие вершины называются концевым для ребра e_k. Ребро графа задается неупорядоченной парой вершин (v_i, v_j), а дуга графа упорядоченной парой вершин. Дугу в графе называют ориентированным ребром.

Различают следующие графы: ориентированный и неориентированный.

Граф называется неориентированным, если он не имеет дуг.

Граф называется ориентированным, если он состоит из одних дуг (ориентированных ребер).

Если граф содержит как ориентированные, таки неориентированные ребра, то такой граф называется смешанным.

Поскольку множество вершин является конечным множество, то граф является конечным.

Конечные графы изображаются в виде диаграммы, на которой вершины обозначаются **точками,** а ребра, соединяющие две точки в виде **линии** между этими точками.

Две вершины называются **смежными,** если они являются концами одного ребра.

Две концевые точки одного ребра являются инцидентными к данному ребру.

Два ребра называются **смежными,** если они имеют общую вершину.

Ребро называется **инцидентным** вершине, если эта вершина является концевой для данного ребра.

Любое ребро в графе представляется как бинарное множество вершин. Такие графы называются **простыми.**

Ребро, которое соединяет одну и туже вершину называется **петлей.**

Если в графе есть две вершины, соединены несколькими ребрами, то такой граф называется **мультиграфом.**

Вершину графа называют **изолированной,** если она не инцидентна ни одному ребру.

Ориентированное ребро имеет *начальную* и *конечную* вершины.

Пример 5.1.1

По заданному графу G (рисунок 5.1) указать:

- вершины,
- ребра,
- изолированные вершины,
- висячие вершины,
- кратные ребра,
- петли,

Перечислить:

- смежные вершины,
- инцидентные ребра вершинам.

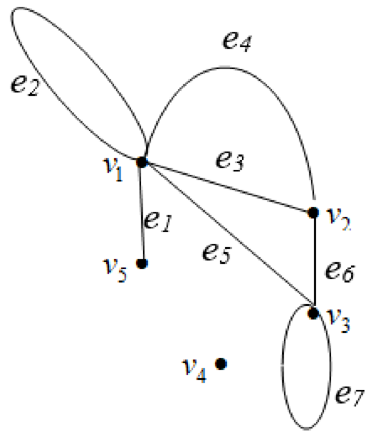


Рис. 5.1 Граф G

Решение.

- вершины: v_1, v_2, v_3, v_4, v_5;
- ребра: e_1, e_2, e_3, e_4, e_5, e_6, e_7;
- изолированная вершина: v_4;
- висячая вершина: v_5;
- кратные ребра: e_3, e_4;
- петли: e_2, e_7;
- смежные вершины: v_1 и v_5, v_1 и v_2, . , v_1 и v_3, , v_2 и v_3;
- ребро e_1 инцидентно вершинам v_1 и v_5, ребра e_3 и e_4 инцидентны вершинам v_1 и v_2,
- ребро e_5 инцидентно вершинам v_1 и v_3, ребро e_7 инцидентно вершинам v_2 и v_3.

Виды графов:(Рис. 5.2)

- полный граф: граф, у которого каждая пара его вершин соединена ребром;
- нуль-граф: граф, у которого нет ни одного ребра;
- пустой граф: граф, у которого нет ни ребер, ни вершин.

Степенью вершины называется число ребер инцидентных данной вершине, обозначается deg (v).

Свойства:

- Сумма степеней вершин всегда четно;
- В любом графе количество вершин нечетной степени четно.

В ориентированном графе определяются: степень выхода (начало дуги) $\deg_{-}$ (v). и степень входа (конец дуги) $\deg_{+}$ (v).

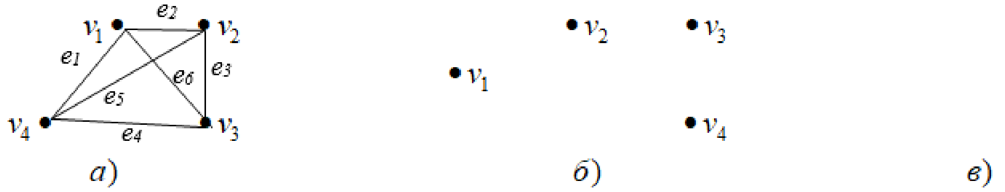


Рис 5.2

На рисунке 5.2 изображены: а) полный граф; б) нуль – граф; в) пустой граф.

Пример 5.1.2

По рисунку 5.3 определить степень вершин:

а) в неориентированном графе;

б) в ориентированном графе.

Рис. 5.3

Решение.

а) $\deg(v_1) = 3$; $\deg(v_2) = 2$; $\deg(v_3) = 1$; $\deg(v_4) = 4$.

б) $\deg_-(v_1) = 2$; $\deg_+(v_2) = 2$; $\deg_-(v_3) = 1$; $\deg_+(v_3) = 2$; $\deg_-(v_4) = 1$.

5.2. Части графов

Пусть имеется граф G = {V, E}.

Граф G' = {V', E'} называется частью графа G = {V, E}, если $V' \subset V, E' \subset E$.

Существуют различные части графа G. К таким частям относятся *подграф* и *суграф*.

Граф G'={V', E'}называется ***подграфом*** графа G={V, E}, если $V' \subset V, E' \subset E$ и множество ребер E' графа G' образовано теми и только теми ребрами графа G, обе концевые вершины каждого из которых принадлежат множеству V'.

Пример 5.2.1

Для данного графа G на рисунке 5.4 изобразить подграф.

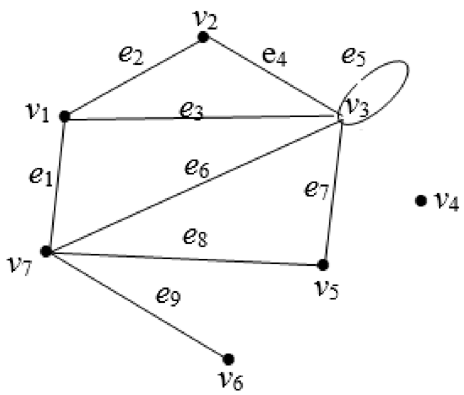


Рис. 5.4 Граф G

Решение.

На рисунке 5.5 представлен подграф графа G.

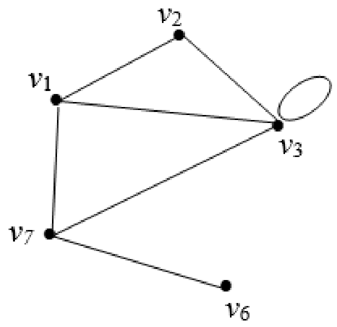


Рис. 5.5 Подграф

Граф G'={V',E'}называется ***суграфом*** графа G={V,E}, если $V' = V,\ E' \subset E$, множество вершин V' и V графов G' и G совпадают, а подмножество ребер E' графа G' образовано некоторыми ребрами графа G.

Пример 5.2.2

Изобразить суграф для графа G на рисунке 5.4

Решение.

На рисунке 5.6 представлен суграф графа G (рис 5.4), в котором множество вершин совпадает с множеством вершин исходного графа, но отсутствуют ребра e_1, e_5.

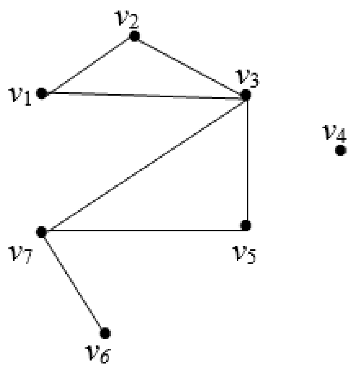


Рис. 5.6 Суграф

Путь (маршрут) – это совокупность ребер, которые соединены вершинами так, что вдоль них можно двигаться по графу.

Путем длиной k из вершины v_0 в v_k называется последовательность $v_0\ e_1\ v_1\ e_2\ v_2 \ldots v_{k-2} e_{k-1} v_{k-1} e_k v_k$. Таким образом, путь длиной k имеет k ребер.

Простым путем из v_0 в v_k называется путь, в котором вершины не повторяются.

Если существует путь из v_0 в v_k, то существует простой путь v_0 в v_k.

Граф G является связным тогда и только тогда, когда между двумя его любыми вершинами существует простой путь.

Циклом графа G называется путь ненулевой длины, соединяющий начальную вершину с самой собой при этом, ни одна вершина не повторяется.

Цикл называется n–циклом, если он содержит «n» различных ребер и «n» различных вершин.

В маршруте одно и то же ребро может встретиться несколько раз. Если в маршруте ребро встречается один раз, то этот маршрут **называется цепью.**

Пример 5.2.3

По данному графу G (рис. 5.7) найти: маршрут, цепь, простую цепь, цикл, простой цикл.

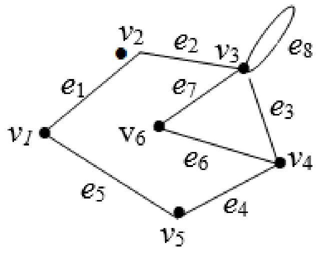


Рис. 5.7

Решение.

- Маршрут длины 6, соединяющий вершины v_1 и v_2 определяется ребрами:

$e_1, e_2, e_3, e_6, e_7, e_2$.

- Цепь длины 5 определяется ребрами: e_1, e_2, e_3, e_6, e_7. Это цепь не является простой, так как при обходе мы дважды прошли через вершину v_3.

- Простая цепь, это цепь без повторяющихся вершин: e_1, e_2, e_3.

- Цикл представляется ребрами e_3, e_6, e_7, e_8.

- Простой цикл представляется ребрами e_3, e_6, e_7.

Цикл, который включает все ребра и все вершины графа G называется **эйлеровым циклом.**

Путь, который включает каждое ребро графа G только один раз, называется эйлеровым путем.

Пример 5.2.4

Изобразить граф с эйлеровым путем.

Решение.

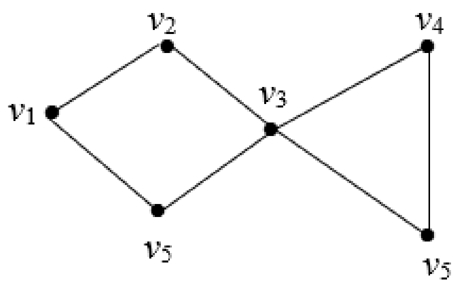


Рис. 5.8

На рисунке 5.8 изображен граф с эйлеровым путем, так как путь включает каждое ребро графа только один раз.

Эйлеровый путь, который не является эйлеровым циклом, называется *собственным эйлеровым путем.*

Граф G называется **двудольным,** если множество V можно представить как объединение непересекающихся множеств V_1 и V_2, так что каждое ребро графа имеет концевые вершины, принадлежащие подмножествам V_1 и V_2.

Полный двудольный граф — это граф, у которого каждые две вершины соединены ребром, но при этом вершины принадлежат разным подмножествам.

Пример 5.2.5

Изобразить двудольный граф.

Решение.

Рисунок 5.9 – двудольный граф, так как множество V можно представить как объединение непересекающихся множеств V_1 и V_2, так что каждое ребро графа имеет концевые вершины, принадлежащие подмножествам V_1 и V_2:

$V_1 = \{v_1, v_2, v_3\}$ и $V_2 = \{v_4, v_5\}$.

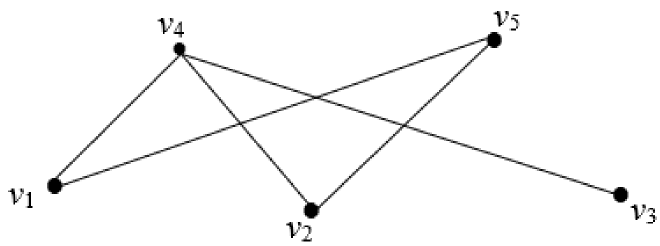


Рис. 5.9

5.3. Способы задания графов

Словесное описание графа.

Пример 5.3.1

Изобразить ориентированный граф G с кратными дугами e_1 с началом в вершине v_1 и концом в вершине v_2 и e_2 с началом в вершине v_2 и концом в вершине v_1. Также e_4 с началом в вершине v_3 и концом в вершине v_4 и e_5 с началом в вершине v_4 и концом в вершине v_3. Петлей при вершине v_5 и дугами e_3 с началом в вершине v_2 и концом в вершине v_3, e_6 с началом в вершине v_5 и концом в вершине v_4.

Решение.

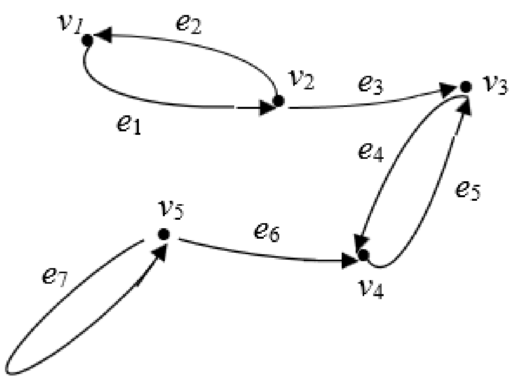


Рис. 5.10 Ориентированный граф G

Граф G (рис. 5.10) является графом по описанию в примере 5.3.1.

Список ребер:

- либо в виде бинарного списка вершин инцидентных ребрам графа;

- либо в виде двух массивов g ($g_1, g_2, \dots, g_n$), h ($h_1, h_2, \dots, h_n$), где *i* - е ребро графа выходит из вершины g_i и входит в вершину h_i.

Пример 5.3.2

Изобразить неориентированный граф G представленный следующим списком ребер {(1,2), (1,3), (2,3), (2,4), (3,1)}.

Решение.

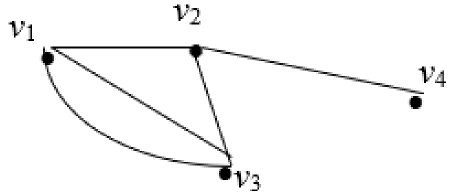


Рис. 5.11

Пример 5.3.3

Изобразить неориентированный граф, который представлен следующими массивами: g = {1,1,1,2,2,3,3,3,3,5,5,6,7}   h = {2,3,7,1,3,3,1,5,7,3,7,7,6}.

Решение.

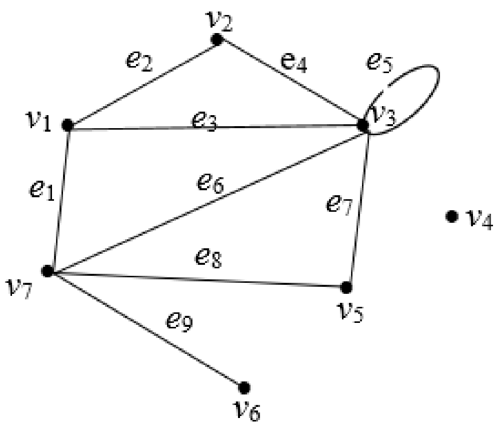


Рис. 5.12

Искомый граф изображен на рисунке 5.12

Пример. 5.3.3

Изобразите ориентированный граф, который представлен следующими массивами: *f* = {2,2,3,4, d = {1,4,2,3}.

Решение.

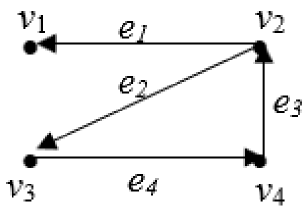


Рис. 5. 13

Граф, представленный массивами *f* = {2,2,3,4}, d = {1,4,2,3} изображен на рисунке 5.13.

Структура смежности.

В ориентированном графе вершина *w* называется последователем другой вершины *v*, если существует дуга, направленная из v и w. Вершина v называется предшественником w.

Граф может быть задан *структурой смежности*, т.е. списком всех последователей каждой вершины, другими словами, для каждой вершины v задается Adj (v) – список всех последователей v.

Граф, заданный структурой смежности, обозначается G (V, Г), где Г отображение V на V.

Пример 5.3.4

Поданной структуре смежности постройте неориентированный граф:

Структура смежности

v	*Adj(v)*
1	2, 3, 7
2	1, 3
3	1, 2, 3, 5, 7
4	
5	3, 7
6	7
7	1, 3, 5, 6.

Решение.

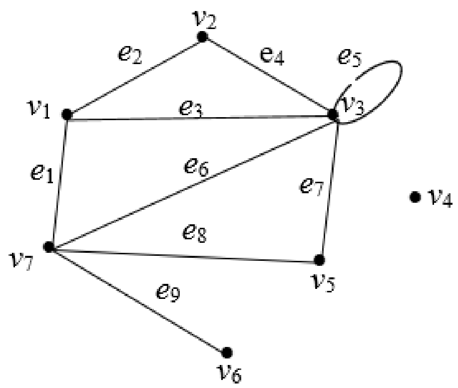


Рис. 5. 14 Неориентированный граф

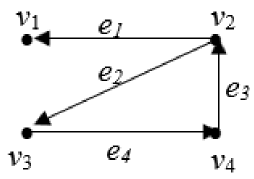
Пример 5.3.5

По данной структуре смежности постройте ориентированный граф:

Структура смежности

v	*Adj(v)*
1	
2	1, 4
3	2
4	3

Решение.



Граф OG Рис. 5.15

На рисунке 5.15 искомый граф.

Матрица смежности.

Это матрица n-го порядка, в которой количество строк и столбцов соответствуют количеству вершин. Если вершины смежные, то в матрице $v_{ij} = 1$, если не смежные, то $v_{ij} = 0$.

Матрица смежности неориентированного графа симметрична относительно главной диагонали. Матрица смежности ориентированного графа не симметрична.

Пример 5.3.6

Для данного неориентированного графа G (рис. 5.16) построить матрицу смежности.

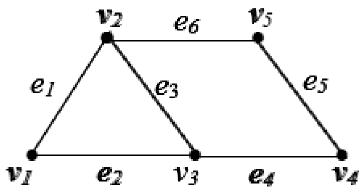


Рис. 5.16

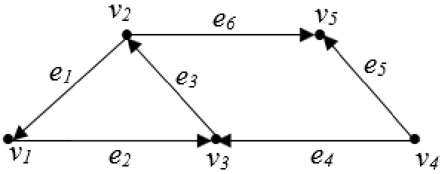
Решение.

Так как вершин 5, то размер матрицы смежности будет 5×5:

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

Пример 5.3.7

Для данного ориентированного графа OG (рис. 5.17) построить матрицу смежности.



Граф OG Рис. 5.17

Решение.

Так как вершин 5, то размер матрицы смежности будет 5×5:

Матрица смежности ориентированного графа имеет вид:

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

Матрица инцидентности.

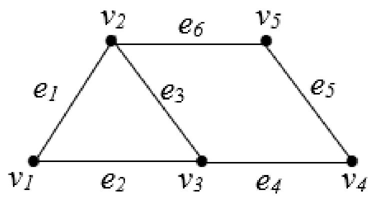
Матрица инцидентности – это матрица размера m×n, в которой строкам поставлены в соответствие вершины графа, а столбцам – ребра.

Если граф G неориентированный, то матрица инцидентности такого графа является булевой матрицей, в которой q_{ij} = 1, если вершина v_i инцидентна ребру e_j и q_{ij} = 0 в противном случае.

Если граф G ориентированный, то q_{ij} =1, если в графе имеется дуга e_j (v_0 v_1) в которой v_0 – начальная вершина, q_{ij} = - 1, если в графе имеется дуга e_j (v_1 v_0) в которой v_0 – конечная вершина и q_{ij} = 0 во всех других случаях.

Пример 5.3.8

Для данного неориентированного графа G (рис. 5.18) построить матрицу инцидентности.



Граф G Рис. 5.18

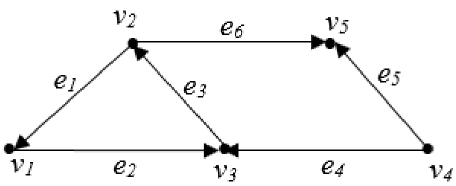
Решение.

Так как вершин 5, а ребер 6 то размер матрицы инцидентности будет 5×6:

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

Пример 5.3.9

Для данного ориентированного графа OG (рис. 5.19) построить матрицу инцидентности.



Граф OG Рис. 5.19

Решение.

Так как вершин 5, а ребер 6 то размер матрицы инцидентности будет 5×6:

Матрица инцидентности ориентированного графа имеет вид:

$$A(OG)=\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

5.4. Операции на графах

1. *Объединением* двух графов G_1 (V_1E_1) и G_2 (V_2E_2) называется граф G (V E), который представлен объединением двух графов, через объединение их вершин и ребер. Граф-объединение содержит все вершины и ребра, принадлежащие хотя бы одному из графов.

Пример 5.4.1

Графы G_1 и G_2 заданы матрицами смежности A и B. Представить геометрически данные графы и граф их объединения, если:

$$A=\begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix} \qquad B=\begin{pmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Решение.

Геометрическое представление графов G_1 и G_2 представлено на рисунке 5.20.

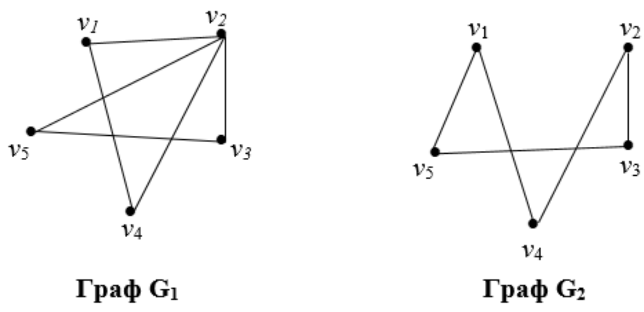


Рис 5.20

Граф объединение $G_1 \cup G_2 = \{(V_{G_1} \cup V_{G_2});(E_{G_1} \cup E_{G_2})\}$ будет иметь вид (рис. 5.21):

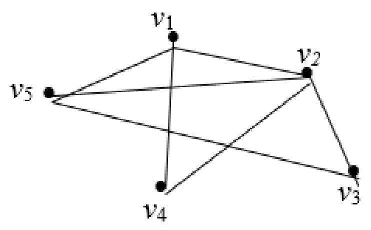
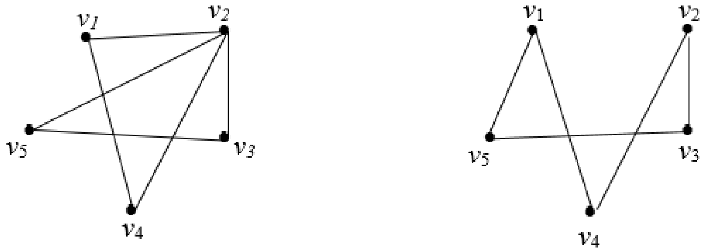


Рис. 5.21 $G_1 \cup G_2$

2. *Пересечением* двух графов G_1 (V_1E_1) и G_2 (V_2E_2) называется граф G (V E), который представлен пересечением двух графов, через пересечение их вершин и ребер. Граф-пересечение содержит те, и только те вершины и ребра, которые принадлежащие одновременно двум графам.

Пример 5.4.2

По данным графам G_1 и G_2 (рис. 5.22) найти граф G – пересечение G_1 и G_2, если:



Граф G_1 **Граф G_2**

Рис. 5.22

Решение.

$G = G_1 \cap G_2 = \{(V_{G_1} \cap V_{G_2});(E_{G_1} \cap E_{G_2})\}$.

Граф G – пересечения G_1 и G_2 представлен на рисунке 5.23.

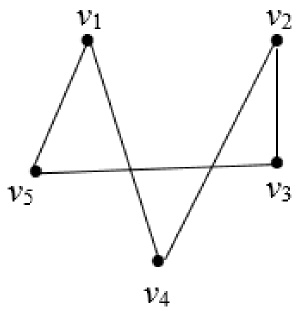


Рис. 5.23 Граф-пересечение

3. *Дополнением* графа G (V E) называется граф $\overline{G}=\{V,\overline{E}\}$, в котором содержится тоже множество вершин, что и в исходном графе, ребра соединяют только те вершины, которые в исходном графе не были соединены.

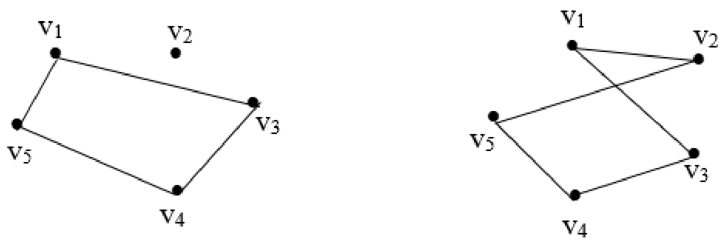
Пример 5.4.3

Изобразить дополнение к графам G_1 и G_2, представленных на рисунке 5.24.

Решение.

Для представления дополнения к графу G_1 изображаем все вершины исходного графа и соединяем только те вершины, которые не были соединены в исходном графе: (v_1, v_3), (v_1, v_5), (v_3, v_4), (v_4, v_5). Рисунок 5.24 а).

Для представления дополнения к графу G_2 изображаем все вершины исходного графа и соединяем только те вершины, которые не были соединены в исходном графе: (v_1, v_2), (v_1, v_3), (v_2, v_5), (v_3, v_4), (v_4, v_5). Рисунок 5.24 б).



а) б)

Граф $\overline{G}_1$ **Граф $\overline{G}_2$**

Рис. 5.24

4.Композицией двух графов G_1 (V_1E_1) и G_2 (V_2E_2) называется граф G (V E), в котором имеется ребро (v_i v_j) если и только если имеются ребра (v_i c) принадлежащее E_1 и (c v_j) принадлежащее E_2.

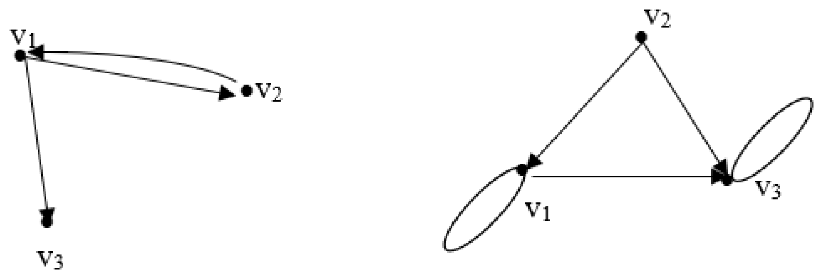
Композицией графов G_1 (V_1E_1) и G_2 (V_2E_2) – это композиция бинарных отношений, определяемых множествами ребер этих графов.

Если исходный граф неориентированный, то каждое ребро {xy} рассматривается как две упорядоченные пары {x, y} и {y, x}. Результативная композиция может быть получена композиций двух бинарных отношений, определяемых ребрами заданных графов. Полученный граф может быть смешанным.

Если исходные графы ориентированные, то множества дуг таких графов есть обычные бинарные отношения. Результат – множество дуг, представляющих композицию заданных бинарных отношений.

Пример 5.4.4

Определить композицию графов G_1 и G_2, которые представлены в виде (Рис. 5.25):



Граф G_1 Граф G_2

Рис. 5.25

Решение.

Представим графы G_1 и G_2 в виде бинарных отношений их вершин:

G_1 = {(1,2), (1,3), (2,1)};

G_2 = {(2,1), (2,3), (3,3), (1,3), (1,1)}.

Комбинация графов $G_1 \circ G_2$ будет представлена ребрами:

$G_1 \circ G_2$ = {(1,1), (1,3), (2,3), (1,3), (2,1)}. Ребро (1,3) проводим один раз.

Графически композиция графа представлена на рисунке 5.26.

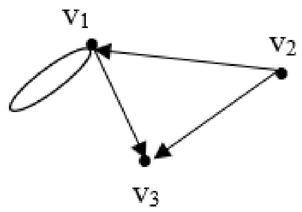


Рис. 5.26 $G_1 \circ G_2$

Комбинация графов $G_2 \circ G_1$ будет представлена ребрами: $G_2 \circ G_1 = \{(2,2), (2,3), (1,2), (1,3)\}$

Графически композиция графа представлена на рисунке 5.27.

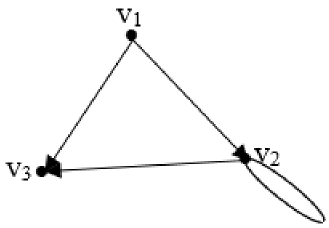


Рис. 5.27 $G_2 \circ G_1$

4. Операция *удаления вершины из графа* приводит к построению такого графа, в котором содержаться те и только те ребра графа G, которые не инцидентны вершине v.

Пример 5. 4.5

В графе G, представленного на рисунке 5.28, удалить вершину v_3.

Решение.

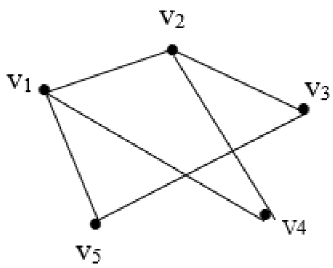


Рис. 5.28 Граф G

После удаления вершины v_3 получаем граф G', который представляем на рисунке 5.29

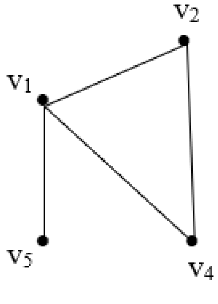


Рис. 5.29 Граф G'

5. Операция *стягивания ребра* состоит в удалении ребра между вершинами v_i и v_j и отождествлении этих вершин в одну вершину v. Множество ребер инцидентных вершинам v_i и v_j в новом графе полагают инцидентными вершине v. Остальные ребра и вершины в результирующем графе остаются такими же, как в исходном графе.

Пример 5.4.6

В графе G (Рис. 5.30) осуществить операцию стягивания ребра, посредством удаления ребра инцидентного вершинам v_1 и v_2.

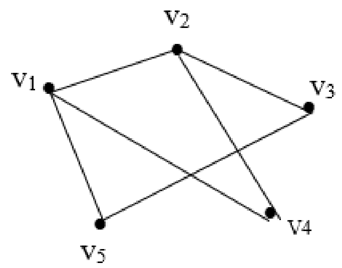


Рис. 5.30 Граф G

Решение.

После удаления ребра инцидентного вершинам v_1 и v_2, данные вершины отождествляются в единую вершину v, а ребра инцидентные данным вершинам (v_1, v_4) и (v_2, v_4) совпадут. В итоге получим граф G'', представлен на рисунке 5.31.

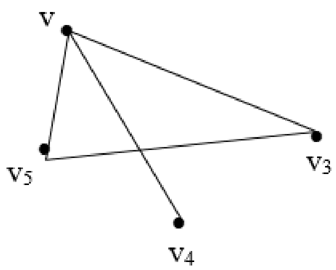


Рис. 5.31 Граф G''

5.5. Связность графа. Деревья.

Граф G называется ***связным,*** если между двумя его любимы вершинами существует маршрут. Связной граф образует компоненту связности. Один граф может иметь несколько компонент связности.

Неформально, компонент связности в графе представляется изображением графа без отрыва пера от листа, при этом некоторые ребра можно проходить несколько раз.

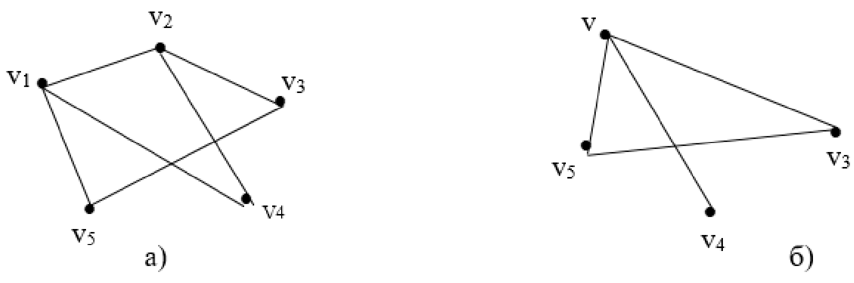


Рис. 5.32

На рисунке 5.32 рисунок а) связной граф; б) не связной граф.
Связной граф G не имеющий циклов называется ***деревом***.

Пример 5.5.1
Определить какой из графов а) или б) на рисунке 5.33 является деревом.

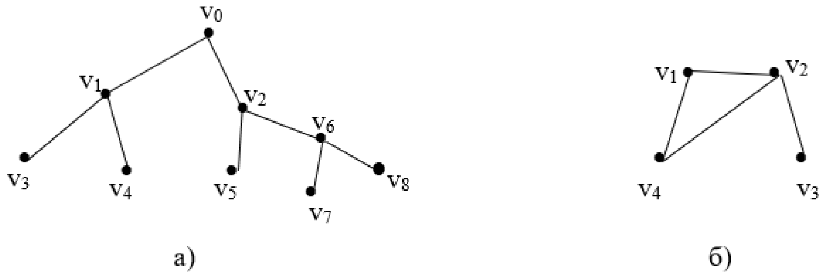


Рис. 5.33

Решение.
На рисунке 5.33 деревом является граф а). Граф б) не является графом, поскольку содержит цикл: v_1, v_2, v_4, v_1.

Если граф G имеет несколько компонентов связности и не имеет циклов, то такой граф называется ***лесом.***

Ориентированное дерево – это ориентированный граф без циклов, удовлетворяющий следующим условиям:

- имеется в точности одна вершина, называемая корнем, в которое не входит ни одно ребро;
- в каждую вершину (кроме корня) входит ровно одно ребро;
- из корня в каждую вершину идет только один путь.

Пусть граф G – является ориентированным деревом, тогда если имеется путь из вершины v_0 к вершине v_1, то вершина v_0 называется ***предком***, а вершина v_1 называется ***потомком.***

Если $(v_0\ v_1) \in E$, то v_0 – подлинный предок (отец) вершины v_1, v_1 – подлинный потомок (сын) вершины v_0.

Если вершина не имеет подлинных потомком, то она называется ***листом.***

Вершина v_i и ее потомок вместе образуют ***поддерево*** графа G, в котором вершина v_i будет являться корнем поддерева.

Глубина вершины v в дереве – это длина пути из корня в v.

Высота вершины v – это длина самого длинного пути из v в какой-нибудь лист.

Высотой дерева называется высота его корня.

Уровень вершины v в дереве равен разности высоты дерева и глубины вершины v.

Пример 5.5.2
По графу G, изображенного на рисунке 5.34 определить:

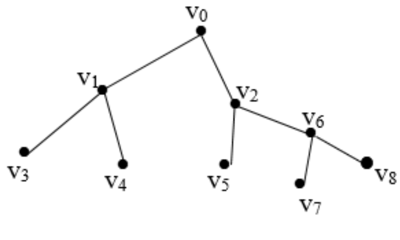


Рис 5.34

- корень дерева и его листы;
- потомков и предков вершины v_2;
- поддерево;

-глубину вершины v_6;

- высоту вершины v_2;

- высоту дерева.

Решение.

- Корнем дерева в графе на рисунке 5.34 является вершина v_0; листы: v_3, v_4, v_5, v_7, v_8;

- предком вершины v_2 является вершина v_0, потомками вершины v_2 являются вершины:

v_5, v_6, v_7, v_8;

- вершины v_1, v_3, v_4 образуют поддерево, в котором вершина v_1 является корнем;

- глубина вершины v_6 равна 2;

- высота вершины v_2 равна 2;

- высота дерева равна 3.

Упорядоченным деревом называется дерево, в котором множество сыновей каждой вершины упорядоченно в некотором порядке (Рис. 5.35).

При изображении упорядоченного дерева полагают, что множество сыновей каждой вершины упорядочено слева направо.

Бинарным деревом называется такое упорядоченное дерево, в котором:

- каждый сын произвольной вершины рассматривается либо как правый, либо как левый сын;

- каждая вершина имеет не более одного левого и не более одно правого сына.

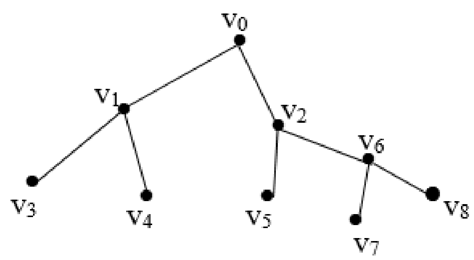


Рис. 5.35 Упорядоченное дерево

5.6. Поиск в глубину

Для определения связности графа используется один из алгоритмов поиска в графе – это **поиск в глубину**. Если граф связный, то в процессе поиска в глубину отмечается каждая вершина графа. В процессе строится некоторое дерево.

Процесс поиска в глубину начинается с произвольной точки v_0. Она становится текущей. Далее для любой текущей точки v_i находится такая смежная вершина v_j, которая является помеченной. Ребро v_i v_j присоединяется к строящемуся дереву T.

Если для текущей вершины смежная непомеченная вершина не находиться, то происходит возврат к вершине, которая была текущей до нее.

Конец работы алгоритма определяется условием, требующим выполнить возврат из начальной вершины v_0.

Если построенное в результате поиска в глубину дерево T содержит все вершины графа, то этот граф связной.

Пример 5.6.1

Определить с помощью алгоритма поиска в глубину является ли граф G на рисунке 5.36 связным.

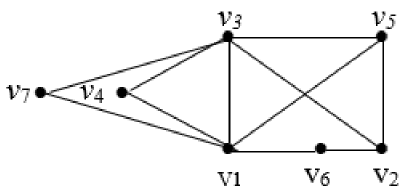


Рис. 5.36

Решение.

Поиск в глубину представлен на рисунке 5.37. Утолщенные ребра принадлежат построенному дереву поиску.

Начиная с вершины v_1, строим дерево – поиск. На основе поиска делаем вывод, что граф, изображенный на рисунке 5.37 связный, поскольку в ходе поиска были отмечены все вершины графа.

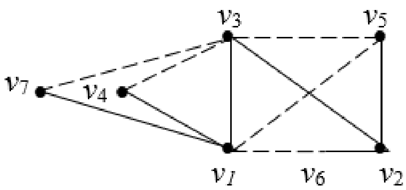


Рис. 5.37

5.7. Числа графов

Помимо степеней вершин в графе определяются следующие числа:
- *цикломатическое число* – это число, определяемое формулой:

$$m - n + p = \lambda(G) \qquad (1)$$

где m – число ребер, n – число вершин, число компонент связности графа, p – число компонент связности графа.

Другими словами, цикломатическое число определяет число ребер, которые необходимо удалить из графа G, чтобы он не имел циклов.

- *хроматическое число графа* – это такое число k, которое показывает наименьшее число различно *раскрашенных* вершин графа. (Раскрашивание концевых вершин разными цветами).

Определенного соотношения, которое помогло бы вычесть хроматическое число, не имеется. Наиболее простая оценка этого числа имеет вид: $k(G) \le \deg_{\max}(v) + 1$

- число *внутренней устойчивости* графа – это число вершин графа в его наибольшем независимом множестве.

Наибольшим независимым множеством вершин называется подмножество вершин графа, содержащее наибольшее количество не смежных вершин.

- *кликовое число* графа – это число вершин в наибольшей по включению в клике.

Клика – это любой полный подграф графа G. Кликой является всякая вершина и любое ребро исходного графа.

- *число паросочетания* – это число ребер в наибольшем паросочетании в графа G.

Два ребра называются независимыми, если они не инцидентны одной и той же вершине. Совокупность попарно независимых ребер графа G называют паросочетанием.

- *число вершинного покрытия* – число элементов в наименьшем вершинном покрытии.

Подмножество вершин графа G называют вершинным покрытием этого графа, если каждое его ребро имеет хотя бы одну концевую вершину в данном подмножестве. Вершинное покрытие называют наименьшим, если оно имеет наименьшее число вершин по сравнению с другими вершинными покрытиями графа G.

- *число реберного покрытия* – число элементов в наименьшем реберном покрытии.

Подмножество ребер графа G называют реберным покрытием этого графа, если всякая вершина этого графа является концевой хотя бы для одного ребра рассматриваемого подмножества ребер. Реберное покрытие называют наименьшим, если оно имеет наименьшее число элементов по сравнению с другими реберными покрытиями графа G.

Пример 5.7.1

Определить все числа графа G представленного на рисунке 5.38

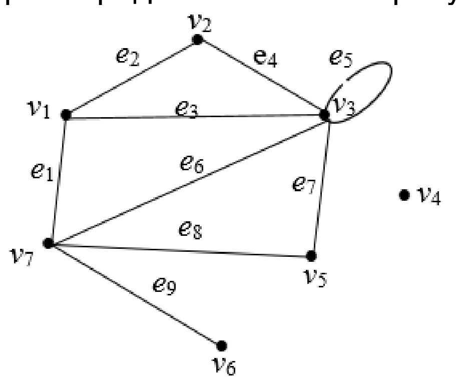


Рис. 5.38 Граф G

Решение.

Цикломатическое число. Граф G на рисунке 5.38 имеет 9 – ребер, 7 – вершин, 2 – компонента связности. Отсюда цикломатическое число равно 9 – 7+2 = 4.

Хроматическое число. Хроматическое число графа G на рисунке 5.34 равно 3.

Число внутренней устойчивости. Так как вершины v_1, v_5, v_6 не смежные и образуют наибольшее независимое множество, то число внутренней устойчивости графа G равно 3: $\alpha_0(G)=3$. $G_2 = \{(v_1, v_7), (v_1, v_3), (v_3, v_7)\}$; $G_3 = \{(v_3, v_5), (v_3, v_7), (v_5, v_7)\}$.

Кликовое число графа G равно 3, так как наибольшее число вершин в полных подграфах данного графа равно 3.

Число паросочетаний.

Перечислим множества независимых ребер: $\{e_1, e_4\}$, $\{e_1, e_5\}$, $\{e_1, e_7\}$, $\{e_2, e_6\}$, $\{e_2, e_8\}$,

$\{e_3, e_8\}$, $\{e_4, e_8\}$, $\{e_4, e_9\}$, $\{e_2, e_5, e_9\}$, $\{e_2, e_7, e_9\}$, $\{e_2, e_5, e_8\}$. Наибольшее паросочетание равно 3.

Число вершинного покрытия. Вершины v_1, v_3, v_7 образуют множество концевых вершин, которые являются хотя бы одним концом каждого ребра, принадлежащего графу G. Поскольку это множество наименьшее, следовательно, число вершинного покрытия равно 3.

Число реберного покрытия.

Наименьшее число ребер, которые содержат все вершины графа G равно 4, так как определяются ребрами e_1, e_2, e_7, e_9. Следовательно число реберного покрытия равно 4.

5.8. Эйлеров и гамильтонов графы

Цикл графа G, в котором содержатся все его вершины и каждое ребро встречается в нем только один раз, называется ***эйлеровым.***

(рисуем, не отрывая пера и не проходя дважды по одному и тому же ребру)

Цикл графа G, проходящий через каждую его вершину по одному разу, называется ***гамильтоновым.***

Полный граф является эйлеровым и гамильтоновым.

Критерий существования в графе эйлерова цикла: Связный и неориентированный граф содержит эйлеров цикл тогда и только тогда, когда каждая его вершина имеет счетную степень.

Для гамильтонова критерия нет. Есть критерии частных случаев.

Если для любой пары вершин (v_i v_j) графа сумма локальных степеней больше или равна числу вершин графа, то в таком графе существует гамильтонов цикл.

Пример 5.8.1

Представить эйлеров граф

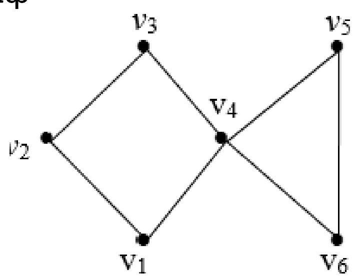


Рис. 5.39

Решение.

На рисунке 5.39 изображен эйлеров граф, так как он имеет цикл, в котором содержатся все его вершины и каждое ребро встречается в нем только один раз.

Пример 5.8.2

Представить граф с гамильтоновым циклом.

Решение.

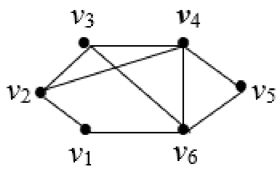


Рис. 5.40

На рисунке 5.40 изображен гамильтонов граф, поскольку, в каждом цикле данного графа через каждую вершину проходишь только один раз.

5.9. Отношения между графами

Отображение *f* графа G (V,E) на граф G^* (V^* E^*) называется ***гомоморфизмом (похожий, подобный)*** из G в G^*: *f*: G $\rightarrow$ G' если оно удовлетворяет следующим условиям:

- *если* $e \in E$, *то* $f(e) \in E^*$ $(f(E) \subseteq f(E^*))$;
- *если* $v \in V$, *то* $f(v) \in V^*$ $(f(V) \subseteq f(V^*))$;
- *если вершины v и w инцидентны ребру e графа G, то вершины f(v) и f(w) инцидентны ребру f(e) графа G'.*

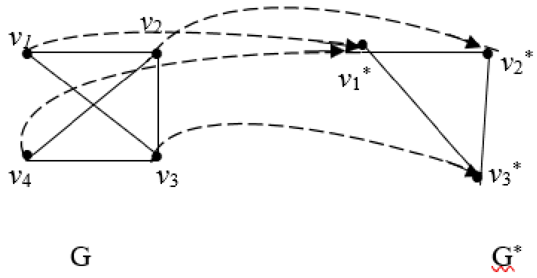


Рис. 5.41 Гомоморфизм графов

Свойства отношений между графами:

- Если отображение *f* – гомоморфизм из *G* в *G**, то *f(G) (f(V), f(E))* – подграф графа *G.*
- Если граф *G* – связный и *f* – гомоморфизм, то граф *f(G)* – связный.
- Если граф *G* – полный и *f* – гомоморфизм, то граф *f(G)* – полный.

Графы *G(V,E)* и G^* (V^* E^*) называются ***изоморфными (схожие по форме и свойствам структуры),*** в случае возможности установления между их вершинами и ребрами взаимно однозначное соответствие: $V \leftrightarrow V^*$, $E \leftrightarrow E^*$, такое, что, если v_i и v_j из *V* соответствуют v_i^* и v_j^* из V^*, то ребро $(v_i, v_j) \in E$ соответствует ребру $(v_i^*, v_j^*) \in E^*$.

Если графы изоморфны, то один из графов можно получить из другого некоторой подстановкой вершин, переводящий один граф в другой.

Изоморфные преобразования графов, представленных в виде матриц смежности и инцидентности, состоят в перестановке строк или столбцов.

Пример 5.9.1

Изобразить для графа G на рисунке 5.42 изоморфные графы.

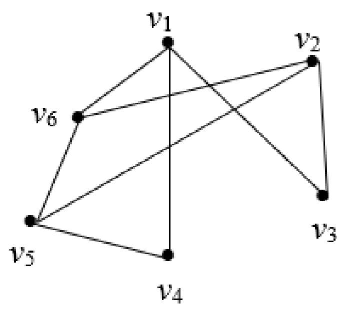


Рис. 5.42

Решение.

а) Изоморфный граф для графа G на рисунке 5.42, получим, если определим иное соединение вершин (рис.5.43):

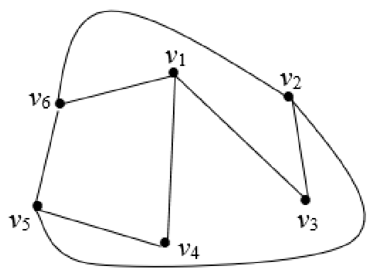


Рис 5.43

б) Изоморфный граф для графа G на рисунке 5.42, получим, если определим иное расположение вершин (рис. 5.44):

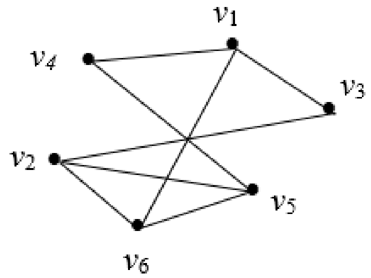


Рис. 5.44

На рисунках 5.42, 5.43, 5.44 изображены изоморфные графы.

Если граф $G(V,E)$ содержит ребро $e\ (v_i,v_j)$ и граф $G'(V'E')$ получен из графа $G(V,E)$ добавлением новой вершины v в множестве V и заменой ребра $(v_i,\ v_j)$ на ребра $(v_i,\ v)$ и $(v,\ v_j)$, то граф $G'(V'\ E')$ называется ***расширением графа*** $G(V,E)$.

Пример 5.9.2

Представить расширение графа G изображенного на рисунке 5.45.

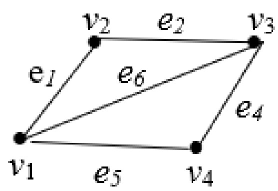


Рис. 5.45

Решение.

Расширение графа G получим после добавления вершины v_5 на ребре e_6. Рисунок 5.46.

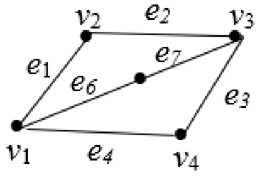


Рис. 5.46

Если графы $G_1,\ G_2,\ \ldots,\ G_n$ таковы, что G_{i+1} является расширением графа G_i, то граф G_n называется ***производным*** (произведенный из другой формы) для G_1.

Пример 5.9.3

Представить производный граф от графа G' изображенного на рисунке 5.47.

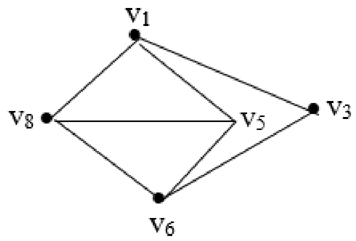


Рис. 5.47 Граф G'

Решение.

Производный граф G представляем на рисунке 5.48.

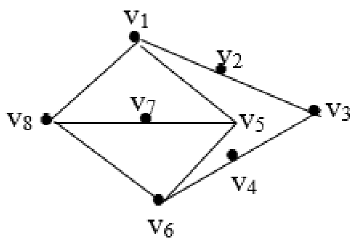


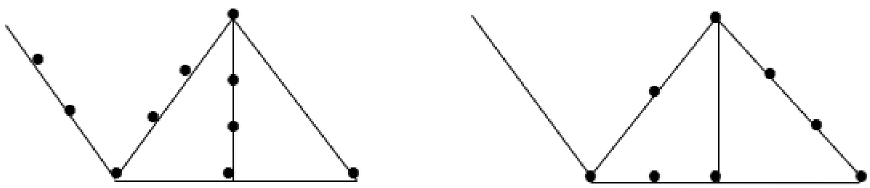
Рис. 5.48Граф G

Графы *G и G'* называются ***гомеоморфные (связанный),*** если существует граф *G'*, такой, что оба графа *G и G'* являются производными от графа *G'*.

Пример 5. 38

Изобразить гомеоморфные графы.

Решение.



Граф G **Граф H**

Рис. 5. 49

На рисунке 5.49 представлены гомеоморфные графы G и H.

Свойства:

- Если графы *G и G'* гомеоморфные, то у них одинаковое количество вершин с нечетной степенью.
- Если графы *G и G'* гомеоморфные, то граф *G* имеет эйлеров цикл, тогда и только тогда, когда граф *G'* имеет эйлеров цикл.

5.10. Планарные графы

Планарным называется граф, который может быть изображен на плоскости так, что его ребра не пересекаются.

Если граф планарный, то его можно разделить по ребрам на несколько частей, считая внешнюю часть. Такие части называются *гранями.* Граница каждой грани является циклом.

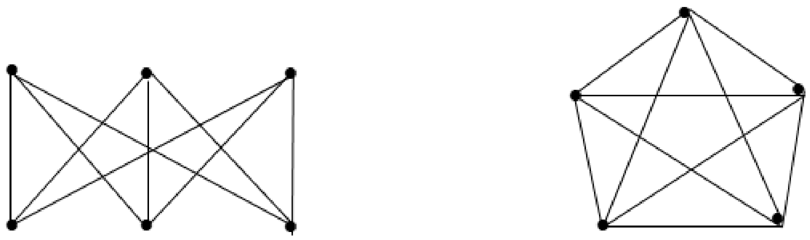
Грань планарного графа – это максимальный участок плоскости, такой, что любые две точки этого участка могут быть соединены кривой, не пересекающей ребро графа.

Определение планарности графа осуществляется с помощью теоремы Эйлера: если G – связный планарный граф, содержащий v- вершин, e – ребер и f граней, то $v-e+f=2$.

Свойства планарных графов:

1. Полный двудольный граф не является планарным.
2. Полный граф с пятью вершинами не является планарным
3. Каждый планарный граф содержит вершину степени 5 или менее.
4. Если два связных графа гомеоморфный, то они либо оба планарные, либо оба не планарные.
5. Произвольный граф, гомеоморфный полному двудольному или полному графу с пятью вершинами не является планарным.

Теорема Куратовского. *Граф является планарным тогда и только тогда, когда он не содержит подграф гомеоморфный ($K_{3,3}$) или (K_5)*



Граф $K_{3,3}$ **Граф K_5**

Рис. 5.50

Пример. 5.10.1

Показать, что граф, изображенный на рисунке 5. 51 планарный.

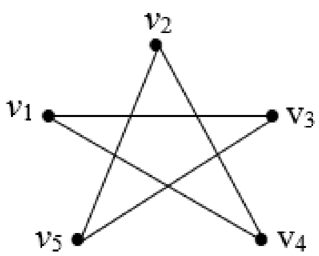


Рис. 5.51

Решение.

1. Передвигая вершину v_4, получаем более простой граф. Рисунок 5.52.

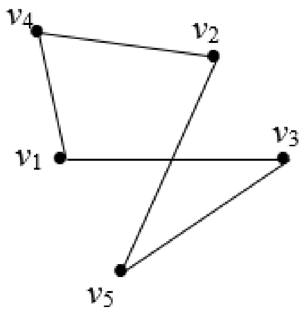


Рис. 5.52

2. Передвигая вершину v_5, получаем планарный граф, граф без самопересечений. Рисунок 5.53.

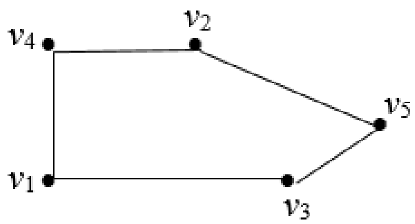


Рис. 5.53

5.11. Определение кратчайшего пути

Для определения кратчайшего пути между вершинами в графе каждому ребру присвоим **вес** (расстояние между вершинами графа), который удовлетворяет следующим условиям:

- вес есть число положительное;
- если ребра нет, то используется знак бесконечности;
- $a+\infty=\infty$

Существует несколько алгоритмов, позволяющих найти кратчайший путь:

1. **Алгоритм Дейкстры**.

Каждой вершине $v_1, v_2, \ldots, v_n$ поставим в соответствие упорядоченную пару $(\infty,0)$. Первая координата вершины означает присвоенное расстояние от вершины v_1 до вершины v_i, а вторая – предыдущую вершину пути.

Шаг 1. Начать с вершины v_1 $(\infty,0)$, заменить ее на v_1 $(0,0)$, сделать ее постоянной, остальные вершины – временные.

Шаг 2. Когда вершина v_k (m, v_u) станет постоянной, для каждой вершины v_j , смежной с v_k, прибавить величину m к расстоянию от вершины v_k до вершины v_j. Если это значение меньше, чем текущее расстояние, присвоенное вершине v_j, заменить текущее расстояние этой суммой и заменить вторую координату на v_k.

Шаг 3. Найти минимум из расстояний, приписанных временным вершинам. Первую вершину с таким расстоянием делаем постоянной.

Шаг 4. Если v_n не постоянная вершина, то возвращаемся к шагу 2.

Шаг 5. Если v_n постоянная вершина, то расстояние, присвоенное вершине, является кратчайшим путем от v_1 до v_n.

Шаг 6. Для нахождения пути начать с вершины v_n, найти предшествующую ей вершину пути (вторая координата). Для каждой вершины пути v_j найти предшествующую ей вершину пути, пока не будет достигнута вершина v_1. Перестановка вершин в обратном порядке и даст кратчайший путь.

Пример 5.11.1

Определить во взвешенном графе G, изображенного на рисунке 5.54, кратчайшее расстояние от вершины A до вершины F.

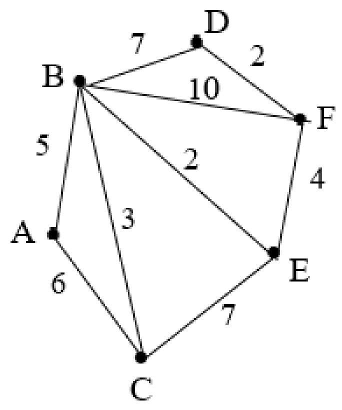


Рис. 5.54

Решение.

1.Зададим первоначальные координаты каждой вершины (рис. 5.55).

Первая компонента координаты покажет длину кратчайшего пути до вершины в момент ее достижения, а вторая компонента указывает на предыдущую вершину кратчайшего пути.

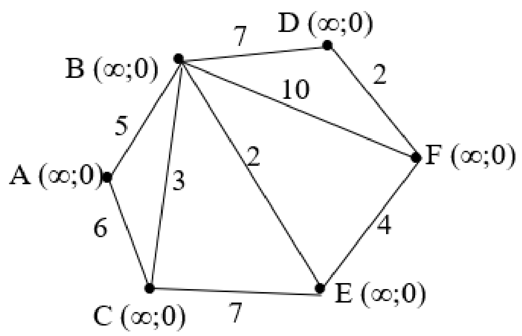


Рис. 5.55

2. Так как движение начинается с точки A, то присвоим ей новые координаты (0;0), определяем ее постоянной вершиной и выделяем жирным шрифтом.

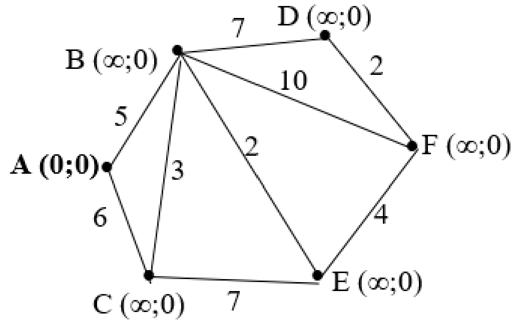


Рис. 5.56

3. С точкой А смежные вершины В и С. Поэтому данным вершинам меняем координаты: В (5; А), С (6; А) (рис. 5.57).

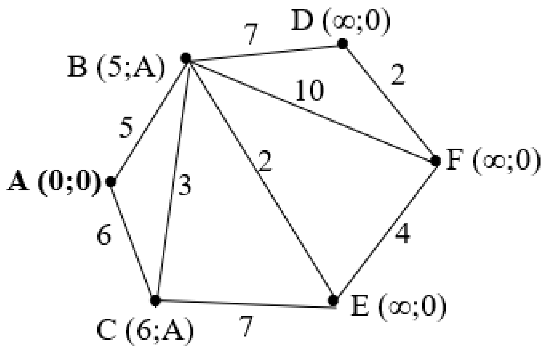


Рис. 5.57

4. Выбираем наименьшее из временных присвоенных значений, т.е. расстояние до вершины В, которое равно 5 и делаем ее постоянной вершиной, отмечая жирным шрифтом (рис. 5.58).

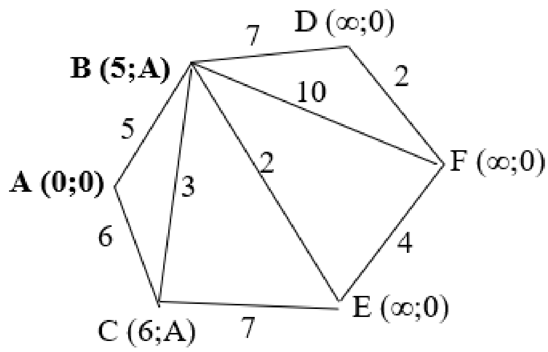


Рис. 5.58

5. Определяем новые расстояния ко всем вершинам смежных с вершиной В: С, D, F, E. Изменения внесем только тем вершинам, до которых новые расстояния меньше старых. В каждом случае прибавляем расстояние от вершины А до вершины В к расстоянию от вершины В до смежных вершин к В. К D: 0+5+7=12; к F: 0+5+10=15; к Е: 0+5+2=7; к С: 0+5+3=8.

Поскольку в вершине С новое расстояние превышает старое, то этой вершине координаты не меняем (рис. 5.59).

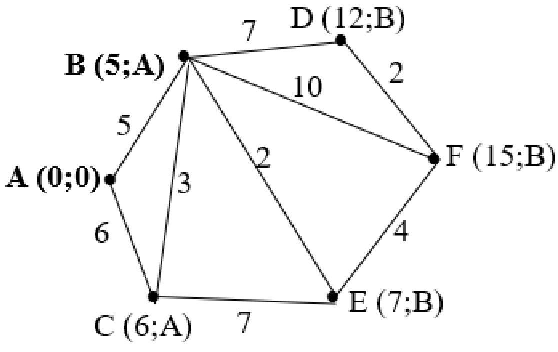


Рис. 5.59

6. Выбираем меньшее из всех новых расстояний {6; 7; 12; 15}. Это величина 6, которая принадлежит вершине С. Делаем данную вершину постоянной, выделяя ее жирным шрифтом (рис. 5.60).

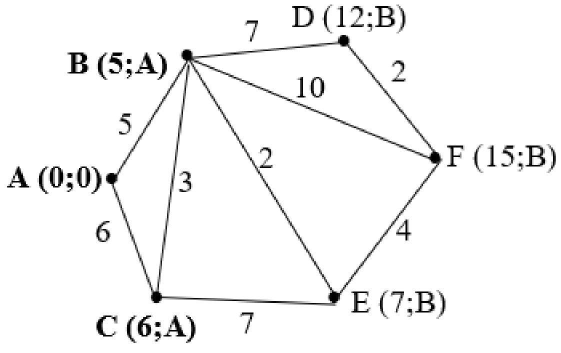


Рис. 5.60

7. Определяем новые расстояния ко всем вершинам смежных с вершиной С. Данный шаг не приводит к изменениям.

8. Выбираем постоянной вершину Е (7; В) (рис. 5.61).

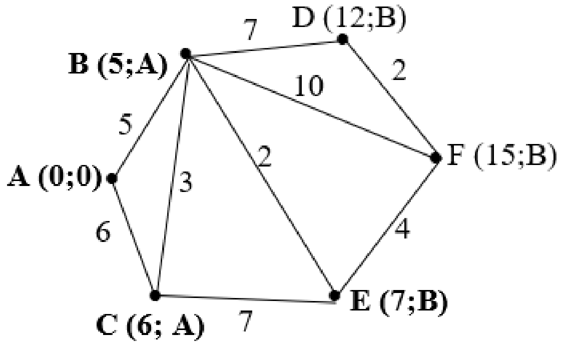


Рис. 5.61

9. Рассматривая новую постоянную вершину Е, меняем координаты F с (15; В) на (11; Е) и делаем ее постоянной (рис. 5.62).

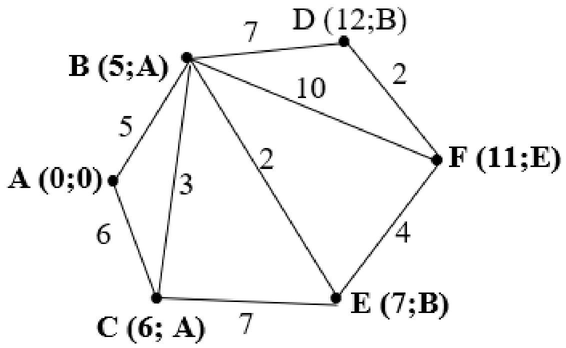


Рис. 5.62

Вершина F стала постоянной, поэтому процесс завершен. Кратчайшее расстояние – 11.

Обозначение пути определяются через предшествующие точки: $F \leftarrow E \leftarrow B \leftarrow A$.

Таким образом, кратчайшим путем является путь ABEF.

Если бы совокупность вершин, смежных с постоянной вершиной, была бы исчерпана до того, как была бы достигнута вершина F, то *задача не имела бы решения.*

Контрольные вопросы для самопроверки

Тема 5 «Теория графов»

1. Граф, вершина графа, ребро графа: определения, примеры.
2. Орграф, вершина орграфа, дуга графа: определения, примеры.
3. Смежные вершины, матрица смежности для графа и орграфа: определения, примеры.
4. Инцидентные вершина и ребро, матрица инцидентности для графа и орграфа: определения, примеры.
5. Диаграмма графа, изоморфизм графов.
6. Плоские графы. Формула Эйлера.
7. Мультиграф и псевдограф: определения и примеры.
8. Степень вершины, вычисление степени вершины по элементам матрицы смежности. Изолированная и висячая вершины: определения, примеры.
9. Полустепени входа и исхода вершины орграфа, вычисление полустепеней по элементам матрицы смежности.
10. Матрицы смежности и инцидентности изоморфных графов.
11. Маршрут, цепь, простая цепь, цикл: определения и примеры.
12. Ориентированный маршрут, цепь, путь, контур: определения и примеры.
13. Подграф, остовый подграф: определения и примеры.
14. Связный граф, компонента графа: определения и примеры. Точка сочленения и мост: определения и примеры.
15. Числа реберной и вершинной связности: определения и примеры.
16. Объединение, произведение, композиция графов: определения и примеры.
17. Дерево: дать равносильные определения понятию, привести пример.
18. Ориентация графа, определить число всех возможных ориентаций для графа с заданным числом ребер.
19. Остовое дерево: определение и пример. Алгоритм построения остового дерева.
20. Взвешенный граф, минимальное остовое дерево: определения и примеры.
21. Обоснование алгоритма построения минимального остового дерева.

Используемая литература

1. Дискретная математика: учебное пособие для вузов / Д. С. Ананичев [и др.]; под научной редакцией А. Н. Сесекина. – Москва: Издательство Юрайт, 2025. – 85 с.

2. Баврин, И. И. Дискретная математика. Учебник и задачник: для вузов / И. И. Баврин. – Москва: Издательство Юрайт, 2025. – 193 с.

3. Гашков, С. Б. Дискретная математика: учебник для вузов / С. Б. Гашков. – 3-е изд., испр. и доп. – Санкт-Петербург: Лань, 2025. – 520 с.

4. Гисин, В. Б. Дискретная математика: учебник и практикум для вузов / В. Б. Гисин. – 2-е изд., перераб. и доп. – Москва: Издательство Юрайт, 2025. – 468 с.

5.Курейчик, В. М. Учебное пособие по курсу «Дискретная математика». Раздел «Теория графов»: [16+] / В. М. Курейчик, В. В. Курейчик, Е. Р. Мунтян; Южный федеральный университет. – Ростов-на-Дону; Таганрог: Южный федеральный университет, 2022. – 166 с.

6.Марченков, С. С. Избранные главы дискретной математики: учебное пособие : [16+] / С. С. Марченков. – Москва: Физматлит, 2023. – 190 с.

7. Палий, И. А. Дискретная математика и математическая логика: учебник для вузов / И. А. Палий. — 3-е изд., испр. и доп. – Москва: Издательство Юрайт, 2025. – 370 с.

8. С. В. Судоплатов, Е. В. Овчинникова. – 5-е изд., испр. и доп. – Москва: Издательство Юрайт, 2025. – 279 с.

9. Таранников, Ю. В. Дискретная математика. Задачник: учебное пособие для вузов / Ю. В. Таранников. – Москва: Издательство Юрайт, 2025. – 385 с.

10. Дискретная математика : учебно-методическое пособие / авт.-сост. В. А. Феофанова, В. И. Воротников ; М-во образования и науки РФ ; ФГАОУ ВПО «УрФУ им. Первого Президента России Б.Н. Ельцина», Нижнетагил. технол. ин-т (фил.). — Нижний Тагил : [НТИ (филиал) УрФУ], 2013. — 256 с.

11. Шевелев, Ю. П. Дискретная математика: учебное пособие для вузов / Ю. П. Шевелев. – 5-е изд., стер. – Санкт-Петербург: Лань, 2024. – 592 с.

Приложение

Приложение 1

Контрольный тест

1. Выбрать множество С, если А = {1;2;3}; В = {2;3;4;}; С = {1;2;3;4}

а) В\А б) А\В в) А $\cap$ В г) А $\cup$ В

2. Выбрать равенство двойственное данному равенству: А $\cup$ А $\cap$ В = А

а) А (А UВ) = АВ б) АUАВ = А

в) А (АUВ) = А г) АВUАВ = А

3. Найти: АUВ если |А|= 10, |В| = 7, | АВ| = 3

а) 14 б) 22 в) 19 г) 18.

4. Даны множества: А = {1;2} В = {2;3}. Найти: В х А

а) {(2;1); (2;2); (3;1); (3;2)} б) {(1;2); (1;1); (2;1); (2;2)}

в) {(1;2); (1;3); (2;2); (2;3)} г) {(2;3); (2;2); (3;2); (3;3)}

5. Выбрать формулу для вычисления P_n

а) n^m б) $\frac{n!}{m!(n-m)!}$ в) n! г) $\frac{n!}{(n-m)!}$

6. Вычислить: P_6 (3; 2;1)

а) 6 б) 30 в) 7 г) 60

7. Вычислить: $\overline{C}_7^6$

а) 924 б) 7 в) 792 г) 15

8. Найти сумму бинарных коэффициентов разложения $(а + в)^6$

а) 256 б) 512 в) 64 г) 128

9 Сколько анаграмм можно составить из слова “мама”

а) 6 б) 360 в) 60 г) 12

10 Выбрать операцию алгебры логики, задаваемую таблицей истинности:

x	y	c
1	1	1
1	0	0
0	1	1
0	0	1

а) c = $x \vee y$ б) c = $x \Leftrightarrow y$ в) c = $x \wedge y$ г) c = $x \Rightarrow y$

11. Выбрать правило исключения альтернативной дизъюнкции $a \oplus b$

а) $ab \vee \overline{a}\overline{b}$ б) $a\overline{b} \vee \overline{a}b$ в) $\overline{a} \wedge \overline{b}$ г) $\overline{a} \vee b$

12. Выбрать логическую операцию, которая выражена через многочлен Жегалкина: $x \oplus 1$

а) $x \vee y$ б) $x \Leftrightarrow y$ в) $\overline{x}$ г) $x \Rightarrow y$

13. Представить в виде многочлена Жегалкина $\overline{xy}$

а) $xy \oplus x \oplus 1$б) $x \oplus y$ в) $xy \oplus 1$ г) $xy \oplus x$

14. Логическая функция задана таблицей истинности. Найти для нее КНФ

x	*y*	*f (x,y)*
1	1	1
1	0	0
0	1	0
0	0	1

а) $(\overline{x} \vee \overline{y})(\overline{x} \vee y)(x \vee \overline{y})$ б) $(x \vee \overline{y})(x \vee y)$

в) $(x \vee y)(\overline{x} \vee y)$ г) $(\overline{x} \vee y)(x \vee \overline{y})$

15. Логическая функция задана таблицей истинности. Найти для нее ДНФ

x	*y*	*f (x, y)*
1	1	1
1	0	0
0	1	0
0	0	1

а) $(x \,\&\, y) \vee (\bar{x} \,\&\, \bar{y})$ б) $(x \vee y) \,\&\, (\bar{x} \vee \bar{y})$

в) $(\bar{x} \,\&\, y) \vee (x \,\&\, \bar{y})$ г) $(\bar{x} \vee y) \,\&\, (x \vee \bar{y})$

16. К какому из классов Поста принадлежит функция $x \oplus y$

а) P_0 б) P_1 в) S г) ни к какому

17. В неориентированном графе последовательность ребер, в которой два соседних ребра имеют общую вершину, называется:

а) простой цепью б) цепью

в) циклический маршрут г) маршрутом

18. Циклический маршрут, который является цепью, называется

а) эйлеров граф б) цикл

в) эйлерова цепь г) эйлеров цикл

19. Связный неориентированный граф, не содержащий циклов, петель и кратных ребер:

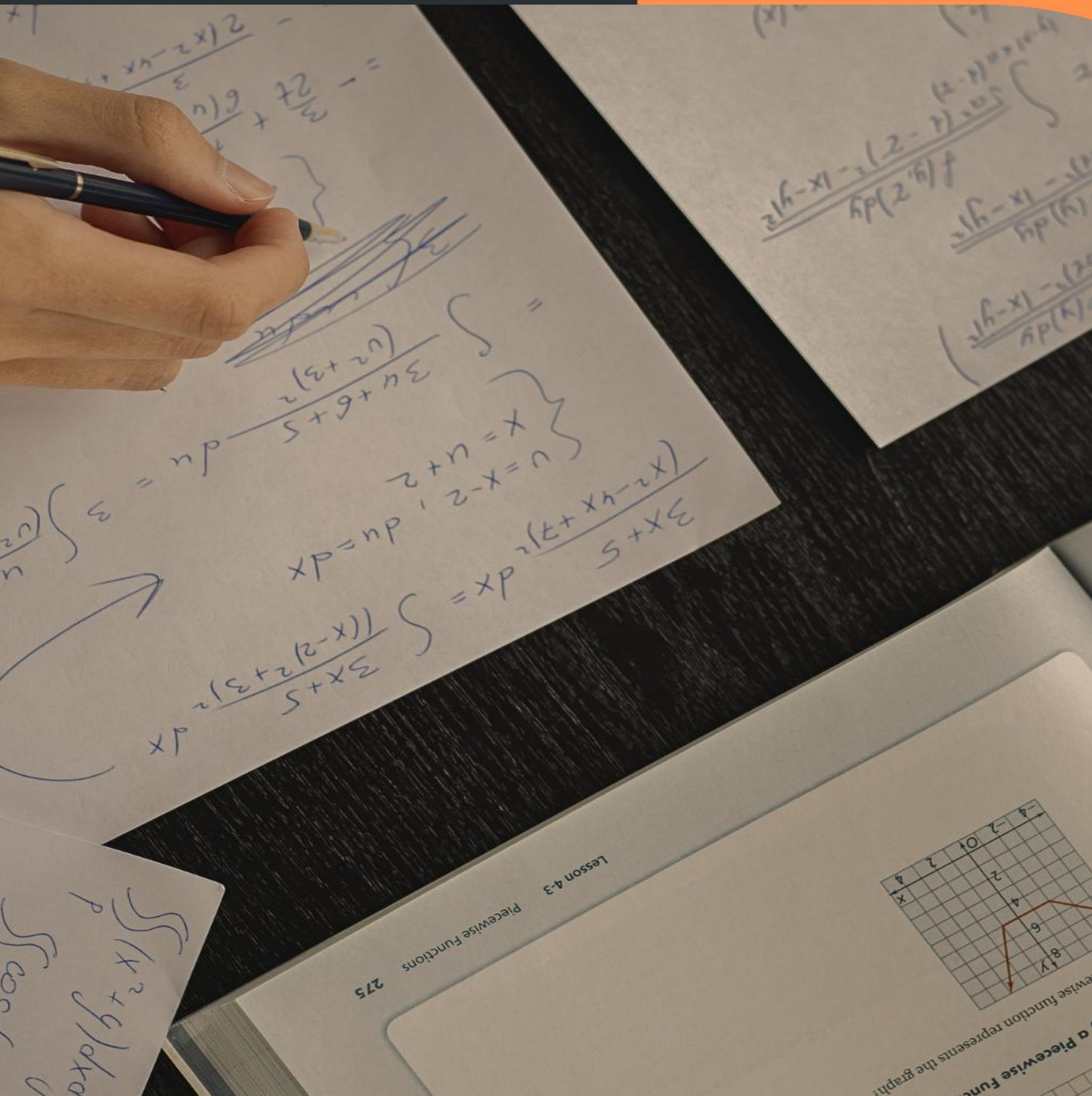
а) плоский граф б) дерево

в) лес г) полный граф

20. Если связи между вершинами графа характеризуются определенной ориентацией, то граф называется:

а) циклическим б) взвешенным

в) конечным г) орграфом



ISBN 978-5-908003-00-1

Усл. печ. л 2,4
Объем издания 15,8 MB
Оформление электронного издания:
НОО Профессиональная наука, mail@scipro.ru
Дата размещения: 05.03.2025 г.
URL: http://scipro.ru/conf/discrete_mathematics02_25.pdf