

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Шебзухов Тетяна Александровна

Должность: Директор Пятигорского государственного автономного образовательного учреждения
федерального университета

Дата подписания: 13.06.2024 16:10:31

Уникальный программный ключ:

d74ce93cd40e39275c3ba2f58486412a1c8ef96f

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**Федеральное государственное автономное образовательное учреждение
высшего образования**

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Пятигорский институт (филиал) СКФУ

Колледж Пятигорского института (филиал) СКФУ

УТВЕРЖДАЮ

Директор Пятигорского института
(филиал) СКФУ
Т.А.Шебзухова

**ДИСКРЕТНАЯ МАТЕМАТИКА С ЭЛЕМЕНТАМИ
МАТЕМАТИЧЕСКОЙ ЛОГИКИ
МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ПРАКТИЧЕСКИХ
ЗАНЯТИЙ**

Специальности СПО

09.02.07 Информационные системы и программирование

Квалификация специалист по информационным системам

Пятигорск 2024

Методические указания для практических занятий по дисциплине Дискретная математика с элементами математической логики составлены в соответствии с требованиями ФГОС СПО. Предназначены для студентов, обучающихся по специальности 09.02.07 Информационные системы и программирование.

Пояснительная записка

Данные методические указания предназначены для закрепления теоретических знаний приобретения необходимых практических навыков и умений по программе дисциплины "Дискретная математика" для специальности СПО 09.02.07 Информационные системы и программирование.

Практические занятия составлены в соответствии с требованиями Федерального государственного образовательного стандарта. В результате освоения учебной дисциплины обучающийся должен **уметь:**

- формулировать задачи логического характера и применять средства математической
- логики для их решения;□
-
-

применять законы алгебры логики;□

определять типы графов и давать их характеристики;□

строить простейшие автоматы;□

В результате освоения учебной дисциплины обучающийся должен **знать:**

- основные понятия и приемы дискретной математики;□ □ □

логические операции, формулы логики, законы алгебры логики;□ □ □

основные классы функций, полноту множества функций, теорему Поста;□

- основные понятия теории множеств, теоретико-множественные операции и их связь с логическими операциями;□
- логику предикатов, бинарных отношений и их виды;□
- элементы теории отображений и алгебры подстановок;□
- метод математической индукции;□
- алгоритмическое перечисление основных комбинаторных объектов;□
- основные понятия теории графов, элементы теории автоматов□

Практическая работа № 1

Тема: Построение таблиц истинности логических функций.

Вариант 1

1. Составить таблицу истинности логического выражения

$$(A \& B) \leftrightarrow (\neg B \rightarrow \neg A) \text{ XOR } A$$

$$(\neg A \vee B) \wedge (\neg B \vee \neg A)$$

2. Построить логическую схему $(\neg A \& B) \vee (A \& \neg B) \vee (A \& B)$ функции
3. Упростить логическое выражение

- Определить, являются ли два высказывания эквивалентными $A \& \neg(\neg B \vee C)$ и $A \& B \& \neg C$
- Определить истинность или ложность высказываний $(\neg(X < 5) \vee (X < 3)) \& (\neg(X < 2) \vee (X < 1))$ при $X=1$

Вариант 2

- Составить таблицу истинности логического выражения $(\neg B \& A) \leftrightarrow (A \rightarrow \neg B) \text{ XOR } B$
- Построить логическую схему $(A \& B) \vee ((A \vee B) \wedge \neg A)$
 $\neg((\neg A \vee B) \& A) \wedge (\neg A \vee \neg B)$ функции
- Упростить логическое выражение
- Определить, являются ли два высказывания эквивалентными $\neg(\neg A \& B \vee A \& (B \vee \neg C))$ и $\neg B \& (\neg A \vee C)$
- Определить истинность или ложность высказываний $(\neg(X < 5) \vee (X < 3)) \& (\neg(X < 2) \vee (X < 1))$ при $X=3$

Вариант 3

- Составить таблицу истинности логического выражения $\neg(B \vee A) \leftrightarrow (\neg A \rightarrow B) \text{ XOR } A$
- Построить логическую схему $(\neg A \vee B) \wedge (A \vee \neg B) \wedge (B \vee A)$ функции
- Упростить логическое выражение
- Определить, являются ли два высказывания эквивалентными $\neg C \vee \neg B \vee \neg(A \vee \neg C)$ и $\neg A \& B \vee \neg C \& B$
- Определить истинность или ложность высказываний $X > 1 \& (\neg(X < 5) \vee (X < 3))$ при $X=2$

Вариант 4

- Составить таблицу истинности логического выражения $(\neg A \rightarrow \neg B) \leftrightarrow (B \& A) \text{ XOR } B$
- Построить логическую схему $\neg((\neg A \vee B) \wedge (\neg B \vee A)) \vee (A \vee B)$ функции
- Упростить логическое выражение

4. Определить, являются ли два высказывания эквивалентными $\neg(A \vee \neg B) \vee \neg B \& C$ и $\neg A \& (BC)$
5. Определить истинность или ложность высказываний $\neg((X > 2) \vee (X < 2)) \vee (X > 4)$ при $X=1$

Вариант 5

1. Составить таблицу истинности логического выражения $(A \& B) \leftrightarrow (\neg A \& B) \text{ XOR } B$
2. Построить логическую схему $\neg(A \& B) \vee (\neg(B \vee A))$ функции
3. Упростить логическое $\neg A \vee B \vee \neg(\neg B \vee A) \vee A \& B$ выражение
4. Определить, являются ли два высказывания эквивалентными
5. Определить истинность или ложность высказываний $X > 1 \& (\neg(X < 5) \vee (X < 3))$ при $X=2$

Практическая работа № 2

Тема: Минимизация логических функций по законам алгебры логики.

Цель работы обобщение, систематизация, углубление, закрепление полученных знаний по теме Упрощение формул логики.

Теоретические основы: Вопросы для актуализации опорных знаний: Что такое равносильные выражения? Перечислить Равносильности, выражающие одни логические операции через другие. 3. Устно Пусть $a = \text{“это утро ясное”}$, $b = \text{“это утро теплое”}$. Выразите следующие формулы на обычном языке: Ответ: а) “это утро ясное и тёплое”; ж) “это утро не ясное или не тёплое”; б) “это утро ясное и оно не тёплое”; з) “это утро не ясное и не тёплое”; в) “это утро не ясное и оно не тёплое”; и) “это утро ясное или не тёплое”; г) “это утро не ясное или оно тёплое”; к) “если это утро ясное, то оно не тёплое”; д) “это утро ясное или оно не тёплое”; л) “если это утро не ясное, то оно тёплое”; е) “это утро не ясное или оно не тёплое”; м) “это утро ясное и не тёплое”. 4. Пройти комплекс тестовых заданий презентация Проверочная4.ppt Задания для практического занятия и инструктаж по их выполнению Записать теоретические сведения в тетрадь. Разобрать материал презентации по данной теме Упрощение.ppt. Разобрать примеры. Выполнить письменно практические задания по данной теме. Теоретические сведения Для упрощения логических выражений используют законы алгебры логики. Они формулируются для базовых логических операций

— "НЕ", "И" и "ИЛИ". Используя законы логики, формулы склеивания и поглощения и свойства логических операций, можно сложную логическую функцию заменить более простой, но равносильной ей функцией. Этот процесс называется минимизацией функции. Минимизация необходима для того, чтобы функциональные схемы не были слишком громоздкими и не использовали лишних элементов. Чем меньше в функции, получаемой при минимизации, входных переменных и используемых логических операций, тем проще логическая схема, меньше в ней логических элементов. Минимизация необходима и при составлении сложных логических выражений в программах. Пример 1. Упростить формулу $(A+B) \cdot (A+C)$ Решение. $C+B \cdot A+B \cdot C$ а) Раскроем скобки $(A+B) \cdot (A+C)$, следовательно, б) По закону идемпотентности $A \cdot A$ $C+B \cdot A+B \cdot C$ $C+B \cdot A+B \cdot C$

в) В высказываниях A и $A \cdot C$ вынесем за скобки A и используя свойство $C \cdot C = C$, получим $A \cdot A \cdot C = A$. Аналогично пункту в) вынесем за скобки высказывание A . г $C \cdot C \cdot C$

Таким образом, мы доказали закон дистрибутивности (распределительный). Преобразования “поглощение” и “склеивание” - поглощение Решение. Пример 2.

Упростить выражение A - Решение. Пример 3. Упростить выражение склеивание Всякую формулу можно преобразовать так, что в ней не будет отрицаний сложных высказываний - все отрицания будут применяться только к простым высказываниям. Пример 4. Преобразовать формулу так, чтобы не было отрицаний сложных высказываний. Решение. 1. Воспользуемся формулой де Моргана, получим: 2. Для выражения применим еще раз формулу де Моргана, получим: Любую формулу можно тождественно преобразовать так, что в ней не будут использованы: - знаки логического

5
слож
ения;

- знаки логического умножения, а будут использованы: - знаки отрицания и логического умножения - знаки отрицания и логического сложения. Пример 5. Преобразовать формулу так, чтобы в ней не использовались знаки логического сложения. Решение. Воспользуемся законом двойного отрицания, а затем формулой де Моргана. Пример 6. Преобразовать формулу так, чтобы в ней не использовались знаки логического умножения. Решение. Используя формулы де Моргана и закон двойного отрицания получим: Замена эквиваленции и импликации на конъюнкцию, дизъюнкцию и отрицание. До сих пор мы занимались равносильными преобразованиями формул, не “. Сейчас покажем, “ и

“Содержащих знаков импликации и эквиваленции “, можно заменить равносильной ей или что всякую формулу, содержащую формулой, не содержащей этих знаков. Имеют место следующие равносильности: (1) (2) Докажем равносильность (1) с помощью таблицы истинности: $X \ Y \ X \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1$

Эквиваленция выражается через конъюнкцию и импликацию Из (3) и (1) получаем $X \ Y$ (4) Эта равносильность выражает эквиваленцию через конъюнкцию, дизъюнкцию и отрицание. Из равносильностей

(3) и (2) получаем равносильность, (5) - выражающая эквиваленцию через конъюнкцию и отрицание. Вывод: В алгебре логики всякую логическую функцию можно выразить через другие логические функции, но их должно быть, по меньшей мере, 2 операции, при этом одной из них обязательно должно быть отрицание.

$$(A \rightarrow B) \& (A \rightarrow C)$$

Пример 1. Упростить

$$\overline{AB} + \overline{B}, \quad \overline{BC} + \overline{C}, \quad \overline{AC} + \overline{BC}$$

Пример 2. Упростить выражения

так, чтобы в полученных формулах не содержалось отрицания сложных высказываний.

Контрольные вопросы:

1. Что значит формализовать высказывание?

2. Сколько будет строк в таблице истинности, если в высказывание 2 переменных? 3 переменных?
3. Дать определение импликации. Что такое эквивалентность?
4. Определите тип высказывания и вид логической операции с соответствующей логической связкой:
 - а) Всякий прямоугольник имеет прямые углы и параллельные противоположные стороны;
 - б) Треугольники с равными сторонами не являются равнобедренными;
 - в) На следующем уроке будет либо история, либо химия;
 - г) Завтра я пойду в школу и библиотеку; д) Либо он заболел, либо забыл о нашей договорённости;
 - е) Утром мы обычно ходим на лыжах или катаемся на коньках.

Практическая работа № 3

Тема: Получение СКНФ и СДНФ по таблице истинности и их минимизация.

Цель работы: Научиться минимизировать логические функции разными методами

Теоретическая часть: Рассмотрим булевы функции, в виде суперпозиции элементарных функций. И ИЛИ, НЕ. Используя законы алгебры логики можно заменить громоздкие булевы функции им равносильными, но более простыми. Такой процесс называется **минимизацией** булевых функций. Его проводят для упрощения сложных логических выражений в программах, а также для того, чтобы построенные на их основе функциональные схемы не содержали лишних элементов.

Минимизировать булевы функции надо приводя их к так называемой нормальной форме. Существуют 2 разновидности нормальных форм: 1 – дизъюнктивная нормальная форма (ДНФ), и 2 – конъюнктивная нормальная функция (КНФ).

Элементарной конъюнкцией называется конъюнкция, составленная из переменных или их отрицаний, в которой каждая переменная встречается не более одного раза.

Дизъюнктивной нормальной формой (ДНФ) называется формула, имеющая вид дизъюнкции элементарных конъюнкций (в вырожденном случае это может быть одна конъюнкция).

Элементарной дизъюнкцией называется дизъюнкция, составленная из переменных или их отрицаний, в которой каждая переменная встречается не более одного раза.

Конъюнктивной нормальной формой (КНФ) называется формула, имеющая вид конъюнкции элементарных дизъюнкций (в вырожденном случае это может быть одна дизъюнкция).

Для каждой формулы логики высказываний функции F имеется равносильная ей дизъюнктивная нормальная форма (ДНФ) и конъюнктивная нормальная форма (КНФ).

Алгоритм приведения формул логики высказываний к ДНФ (КНФ)

1. Избавиться от всех логических операций, содержащихся в формуле, заменив их основными: конъюнкцией, дизъюнкцией и отрицанием. Это можно сделать, используя равносильные формулы.

2. Заменить знак отрицания, относящийся к выражениям вида знаками отрицания, относящимся к отдельным переменным высказывания на основании законов Де Моргана.

3. Избавиться от знаков двойного отрицания.

4. Применить если нужно к операциям конъюнкции и дизъюнкции свойства дистрибутивности и формулы поглощения.

Совершенной дизъюнктивной нормальной формой (СДНФ) называется такая дизъюнктивная нормальная форма, у которой в каждую конъюнкцию входят все переменные данного списка (либо сами, либо их отрицания), причем в одном и том же порядке.

Свойства СДНФ:

1. ДНФ не содержит двух одинаковых конъюнкций (Из формулы удалены одинаковые элементарные конъюнкции, кроме одной, согласно правилу идемпотентности).

2. Ни одна конъюнкция не содержит одновременно двух одинаковых переменных. 3. Ни одна конъюнкция не содержит одновременно некоторую переменную и ее отрицание (Из формулы удалены элементарные конъюнкции, включающие одновременно переменную и ее отрицание, согласно закону инверсии).

4. Каждая конъюнкция СДНФ содержит либо переменную, либо ее отрицание для всех переменных, входящих в формулу (т.е. каждая элементарная конъюнкция включает все логические переменные, входящие в эту формулу).

Если в логической функции не выполняется последнее требование, то в неполную элементарную конъюнкцию необходимо ввести дополнительный множитель, включающий дизъюнкцию отсутствующей переменной и ее отрицание. Это всегда можно сделать, согласно закону инверсии.

7

Переход от ДНФ к СДНФ: если в какой-то простой конъюнкции недостает переменной, например, z , вставляем в нее выражение $z \vee \bar{z} = 1$, после чего раскрываем скобки (при этом повторяющиеся дизъюнктивные слагаемые не пишем).

Совершенной конъюнктивной нормальной формой (СКНФ) называется такая КНФ, у которой в каждую простую дизъюнкцию входят все переменные данного списка (либо сами, либо их отрицания), причем в одинаковом порядке.

Свойства СКНФ:

1. КНФ не содержит двух одинаковых дизъюнкций.

2. Ни одна из дизъюнкций не содержит одновременно двух одинаковых переменных.

3. Ни одна из дизъюнкций не содержит одновременно некоторую переменную и ее отрицание.

4. Каждая дизъюнкция СКНФ содержит либо переменную, либо ее отрицание для всех переменных, входящих в формулу.

Если логическая функция имеет вид КНФ, то привести ее к виду совершенной КНФ (сокращенно СКНФ) можно, дополнив каждую элементарную дизъюнкцию логическим нулем, который в следующем шаге заменяется на конъюнкцию недостающей переменной и ее отрицание. Полученная нормальная конъюнктивная форма является совершенной.

Переход от КНФ к СКНФ: если в простой дизъюнкции не хватает какой-то переменной (например, z , то добавляем в нее выражение $z \cdot \bar{z} = 0$ (это не меняет самой дизъюнкции), после чего раскрываем скобки с использованием распределительного закона)

Минимизация логических функций по картам Карно.

Карты Карно - это графическое представление операций попарного неполного склеивания и элементарного поглощения.

Карты Карно рассматриваются как перестроенная соответствующим образом таблица истинности функции.

Карты Карно - определенная плоская развертка n-мерного булева куба.

Строится таблица истинности функции определенным образом. Каждая клетка таблицы соответствует вполне определенной вершине булева куба. Нулевые значения не записываются.

Карта Карно для функции 4-х переменных:

$x_1 x_2$ $x_3 x_4$	00	01	11	10
00	1	1	1	
01			1	
11		1		
10	1	1		

Благодаря использованию кода Грея в ней верхняя строка является соседней с нижней, а правый столбец соседний с левым, т.е. карта Карно рассматривается как поверхность фигуры под названием тор ("бублик"). p-клетки - клетки карты Карно, соответствующие единичному значению функции.

8

Соседние наборы - наборы, которые различаются только одним аргументом (одной орбитой).

Правила склейки:

□ склейку клеток карты Карно можно осуществлять по единицам (если необходимо получить ДНФ) или по нулям (если требуется КНФ); □ □

□

n склеивать

можно только прямоугольные области с числом единиц (нулей) 2, где

n

— целое число. Для карт Карно с числом переменных более четырёх могут получаться более сложные области, о чём будет сказано в следующих разделах; □ □

□ область, которая подвергается склейке должна содержать только единицы □ □ □ (нули); крайние клетки каждой горизонтали и каждой вертикали также граничат между □ собой и могут объединяться в прямоугольники. Следствием этого правила является смежность всех четырёх угловых ячеек карты Карно для $N=4$. Если во всех четырёх угловых ячейках стоят единицы (нули) они могут быть объединены в квадрат; все единицы □ (нули) должны попасть в какую-либо область; □

□

□ с точки зрения минимальности ДНФ (КНФ) число областей должно быть как можно меньше (каждая область представляет собой терм), а число клеток в области должно □ быть как можно больше (чем больше клеток в области, тем меньше переменных содержит n ячеек содержит $N-n$ переменных); терм. Терм размером $2^□$

□ одна ячейка карты Карно может входить сразу в несколько областей. Это следует из очевидного свойства булевых функций: повторение уже существующего слагаемого □ □

(сомножителя) не влияет на функцию: $A \vee A = A; A \cdot A = A;$ □

□

□ в отличие от СДНФ (СКНФ), ДНФ (КНФ) не единственны. Возможно несколько эквивалентных друг другу ДНФ (КНФ), которые соответствуют разным способам покрытия карты Карно прямоугольными областями. □ □

После заполнения Карты Карно значениями из таблицы истинности берём первую область и смотрим, какие переменные не меняются в пределах этой области, выписываем конъюнкцию этих переменных; если неменяющаяся переменная нулевая, □ □ проставляем над ней инверсию. Берём следующую область, выполняем то же самое, что и для □ первой, и т. д. для всех областей. Конъюнкции областей объединяем дизъюнкцией. □

Для КНФ всё то же самое, только рассматриваем клетки с нулями, неменяющиеся переменные в пределах одной области объединяем в дизъюнкции (инверсии проставляем над единичными переменными), а дизъюнкции областей объединяем в конъюнкцию. На этом минимизация считается законченной. □ □

□ **Пример 1** □

У мальчика Коли есть мама, папа, дедушка и бабушка. Коля пойдёт гулять на улицу, □ если □ и только если ему разрешат хотя бы двое родственников.

Для краткости обозначим родственников Коли через буквы:

мама — x_1

папа — x_2 дедушка —
 x_3 бабушка — x_4

Условимся обозначать согласие родственников единицей, несогласие - нулём.

Возможность пойти погулять обозначим буквой f , Коля идёт гулять — $f = 1$, Коля гулять не идёт — $f = 0$.

Составим таблицу истинности:

№	X1	X2	X3	X4	f
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	1

Перерисуем таблицу истинности в 2-х мерный вид, переставим в ней строки и столбцы в соответствии с кодом Грея и заполним её значениями из таблицы истинности:

$x_3 \ x_4$					
		00	01	11	10
$x_1 \ x_2$	00	0	0	1	0
	01	0	1	1	1
	11	1	1	1	1
	10	0	1	1	1

Минимизируем в соответствии с **правилами склейки**: □

все области содержат 2^n клеток; □

□ так как Карта Карно на четыре переменные, оси располагаются на границах Карты □
и их не видно;

так как Карта Карно на четыре переменные, все области симметрично осей —

□

□ смежные между собой; □ области S3, S4, S5, S6

□ максимально большие; □ все области пересекаются

□ (необязательное условие). □

X3 X4				
X1 X2	00	01	11	10
00	0	0	1	0
01	0	1	1	1
11	1	1	1	1
10	0	1	1	1

Diagram showing prime implicants S1, S2, S3, S4, S5, S6 on the Karnaugh map. S1 is a vertical rectangle covering (11,1), (10,1), (11,0), (10,0). S2 is a horizontal rectangle covering (11,1), (11,0), (10,1), (10,0). S3 is a vertical rectangle covering (01,1), (01,0), (11,1), (11,0). S4 is a horizontal rectangle covering (01,1), (01,0), (11,1), (11,0). S5 is a vertical rectangle covering (11,1), (10,1), (11,0), (10,0). S6 is a horizontal rectangle covering (11,1), (10,1), (11,0), (10,0).

В данном случае рациональный вариант только один.

$$f(X1, X2, X3, X4) = S1 \vee S2 \vee S3 \vee S4 \vee S5 \vee S6 = \\ = X3X4 \vee X1X2 \vee X2X4 \vee X1X4 \vee X1X3 \vee X2X3$$

Составим мин. КНФ:

X3 X4				
X1 X2	00	01	11	10
00	0	0	1	0
01	0	1	1	1
11	1	1	1	1
10	0	1	1	1

Diagram showing prime implicants S1, S2, S3, S4 on the Karnaugh map. S1 is a vertical rectangle covering (01,1), (01,0), (11,1), (11,0). S2 is a horizontal rectangle covering (01,1), (01,0), (11,1), (11,0). S3 is a vertical rectangle covering (11,1), (10,1), (11,0), (10,0). S4 is a horizontal rectangle covering (11,1), (10,1), (11,0), (10,0).

$$f(X1, X2, X3, X4) = (S1) (S2) (S3) (S4) = \\ = (X1 \vee X2 \vee X3)(X1 \vee X3 \vee X4)(X2 \vee X3 \vee X4)(X1 \vee X2 \vee X4)$$

Задания для практического занятия:

□

□

1. Для

функции из таблицы 1,

соответствующей номеру

своего варианта, выполнить

следующее: □ составить

таблицу истинности; записать

СДНФ и СКНФ функции;

доказать эквивалентность

СДНФ и СКНФ. Таблица 1 -

Варианты задания

№ варианта	Функция
1	$f(x, y, z) = \bar{x} \& y \vee (\bar{x} \vee z)$
2	$f(x, y, z) = z \rightarrow (\bar{x} \vee \bar{y})$
3	$f(x, y, z) = x \& y \rightarrow (\bar{x} \vee \bar{y})$
4	$f(x, y, z) = (x \& y \& z) \sim (\bar{x} \vee y)$
5	$f(x, y, z) = (\bar{x} \vee \bar{y}) \sim y$

- ☐ Составить таблицу истинности.
- ☐ Записать СДНФ и СКНФ функции.
- ☐ Упростить выражение для СДНФ и СКНФ, используя карту Карно.
- ☐ Составить схему устройства, реализующего заданную СДНФ и СКНФ после

2. Для функции из таблицы 2, соответствующей номеру своего варианта, выполнить следующее:

упрощения.

№	Состояние входа				Функция f по вариантам				
	X1	X2	X3	X4	1	2	3	4	5
0	0	0	0	0	1	0	0	0	1
1	0	0	0	1	1	0	1	0	0
2	0	0	1	0	0	0	0	0	0
3	0	0	1	1	0	0	1	0	1
4	0	1	0	0	0	1	0	0	0
5	0	1	0	1	0	1	0	1	0
6	0	1	1	0	0	0	1	0	0
7	0	1	1	1	0	0	1	1	1
8	1	0	0	0	1	0	0	0	1
9	1	0	0	1	1	0	1	0	1
10	1	0	1	0	0	0	0	0	0
11	1	0	1	1	0	0	1	0	1
12	1	1	0	0	0	1	0	1	0
13	1	1	0	1	1	1	0	1	0
14	1	1	1	0	1	0	0	1	0
15	1	1	1	1	0	0	0	1	0

Контрольные вопросы:

1. Что называют простой конъюнкцией и простой дизъюнкцией?
2. Что такое ДНФ и КНФ?
3. В чем отличие СДНФ/СКНФ от ДНФ/КНФ?
4. Для чего нужно минимизировать логические функции?
5. Какие алгебраические преобразования наиболее часто используются при минимизации функций?
6. В чем суть метода карт Карно?

Практическая работа № 4**Тема: Определение тождественности логических функций.**

Цель работы: изучить законы определения тождественно истинных и тождественно ложных формул

Теоретическая часть: Все формулы алгебры логики делятся на тождественно истинные, тождественно ложные и выполнимые. Определения тождественно истинных и тождественно ложных формул даны выше.

Формулу F называют выполнимой, если она принимает значение “истина” (1) хотя бы на одном наборе значений входящих в неё переменных и не является тождественно истинной.

В связи с этим возникает задача получения ответа на вопрос: к какому классу относится данная формула?

Эта задача называется проблемой разрешимости.

Очевидно, что проблема разрешимости алгебры логики разрешима.

Действительно, для каждой формулы алгебры логики может быть составлена таблица истинности, которая и даст ответ на поставленный вопрос. Однако практическое использование таблиц истинности при большом количестве переменных x_i затруднительно.

Существует другой способ, позволяющий, не используя таблицы истинности, определить, к какому классу относится формула F . Этот способ основан на приведении формулы к нормальной форме (ДНФ или КНФ) и использовании алгоритма, который позволяет определить, является ли рассматриваемая формула тождественно истинной или не является. Одновременно с этим решается вопрос о том, будет ли формула A выполнимой. Предположим, что мы имеем критерий тождественной истинности для формул алгебры

Рассмотрим механизм его применения. Применим критерий тождественной

истинности к формуле A . Если окажется, что A формула – тождественно истинная, то задача решена. Если же окажется, что формула $\neg A$ тождественно истинна, то применим

критерий тождественной истинности к формуле $\overline{A}$. Если окажется, что формула $\overline{A}$ – тождественно истинная, то ясно, что формула A – тождественно ложная, и задача решена. Если же формула $\neg \overline{A}$ тождественно истинная, то остаётся единственный возможный

результат: A формула выполнима.

Установим теперь критерий тождественной истинности произвольной формулы алгебры логики. С этой целью предварительно сформулируем и докажем критерий тождественной истинности элементарной дизъюнкции.

Теорема 1.

Для того, чтобы элементарная дизъюнкция была тождественно истинной, необходимо и достаточно, чтобы в ней содержалась переменная и её отрицание.

Доказательство необходимости.

Пусть элементарная дизъюнкция тождественно истинна, но в неё одновременно не входит некоторая переменная и её отрицание. Придадим каждой переменной, входящей в элементарную дизъюнкцию без знака отрицания, значение “ложь”, а каждой переменной, входящей в элементарную дизъюнкцию под знаком отрицания – значение “истина”. Тогда,

очевидно, вся элементарная дизъюнкция примет значение “ложь”, что противоречит условию.

Доказательство достаточности.

Пусть теперь элементарная дизъюнкция содержит переменную и её отрицание. Так как $x_i \vee \bar{x}_i \equiv 1$, то и вся элементарная дизъюнкция будет тождественно истинной. Критерий тождественной истинности элементарной дизъюнкции позволяет сформулировать и доказать критерий тождественной истинности произвольной формулы алгебры логики.

Теорема 2.

Для того, чтобы формула алгебры логики A была тождественно истинна, необходимо и достаточно, чтобы любая элементарная дизъюнкция, входящая в КНФ A , содержала переменную и её отрицание. Доказательство необходимости.

Но A - тождественно истинна. Тогда и КНФ $A \equiv A_1 \wedge A_2 \wedge \dots \wedge A_n$ где A_i элементарные дизъюнкции. Пусть A_i - тождественно истинна.

().

Так как КНФ $A \equiv A_i$ $A_i \equiv 1$, то каждая A_i тождественно истинна.

элементарная дизъюнкция содержит переменную и её отрицание. Доказательство достаточности.

Пусть любая элементарная A_i входящая в КНФ A дизъюнкция, содержит переменную и её отрицание. Тогда по теореме 1 $A_i \equiv 1$, $i = \overline{1, n}$.

Например, выясним, является ли формула $A \equiv y \vee \bar{y} \wedge x \vee \bar{x} \wedge \bar{y}$ тождественно истинной.

При этом и КНФ $A \equiv (y \vee \bar{y}) \wedge (x \vee \bar{x}) \wedge (y \vee \bar{y})$ тождественно истинной.

элементарная дизъюнкция $y \vee \bar{y}$ и $y \vee x \vee \bar{x}$, $A \equiv 1$.

Так как то ясно, что

каждая

входящая в КНФ A ,

содержит переменную и её отрицание. Следовательно,

Аналогично имеют место теоремы, дающие критерии тождественной ложности элементарной конъюнкции и любой формулы алгебры логики.

Теорема 3.

Для того, чтобы элементарная конъюнкция была тождественно ложной, необходимо и достаточно, чтобы в ней содержалась переменная и её отрицание.

Теорема 4.

Для того, чтобы формула алгебры логики A была тождественно ложной, необходимо и достаточно, чтобы любая конъюнкция, входящая в ДНФ A , содержала переменную и её отрицание.

Практическая часть:

Пример. Проверить тождественность логических функций:

$$F(x_1, x_2, x_3) = (x_1 x_2 x_3) \vee (x_1 x_2 \bar{x}_3) \vee (x_1 \bar{x}_2 x_3) \vee (\bar{x}_1 x_2 x_3) \vee (\bar{x}_1 \bar{x}_2 \bar{x}_3)$$

$$P(x_1, x_2, x_3) = (x_1 x_2 \bar{x}_3) \vee (\bar{x}_1 x_2 x_3)$$

Пример. Проверить тождественность логических функций:

$F = (x_1 x_2 \vee x_2 x_3 \vee x_1 x_2 \vee x_1 x_3 \vee x_2 x_3) \vee f(x_1, x_2, x_3)$, которая принимает единичные значения на наборах 2, 3.

Пример. Проверить тождественность логических функций:

$$F(x, y, z) = (x \vee y \vee z) \wedge f(x_1, x_2, x_3)$$

Функция f имеет следующую минимальную форму $f(x, y, z) = (x \vee y \vee z) \wedge (x \vee y \vee z)$

Пример. Путём тождественных преобразований получить минимальную форму записи логической функции F1 и проверить, является ли она тождественной функции F2. Заданы функции F1 и F2:

$$F1(x_1, x_2, x_3) = (x_1 x_2 \vee x_1 x_3 \vee x_2 x_3) \vee (x_1 x_2 \vee x_1 x_3 \vee x_2 x_3)$$

$$F2(x_2, x_3) = (x_2 \vee x_3)$$

Пример. Путём тождественных преобразований логической функции F1 получить её минимальную форму записи и проверить, является ли она тождественной функции F2, определённой на данных наборах. Заданы функции F1 и F2:

$$F1(x_1, x_2, x_3) = (x_1 x_2 \vee x_1 x_3 \vee x_2 x_3) \vee (x_1 x_2 \vee x_1 x_3 \vee x_2 x_3); F2=1 \text{ на наборах } 4, 5, 6.$$

Пример. Заданы функции F1 и F2:

F1=0 на наборе 0, 2, 3, 4, 5, 6, 7,

$$F_2 = \overline{x_1}x_2x_3 + x_1\overline{x_2}x_3 + x_1x_2\overline{x_3} + x_1x_2x_3$$

Пример. Логические функции F_1 и F_2 заданы таблично (на наборах):

$F_1=0$ на наборах 0, 4, 6, $F_2=0$

на наборах 0, 3, 6, 7.

Необходимо получить их минимальную форму записи.

Пример. Задана функция F_2 аналитически. $F_2 = \overline{x_1}x_2x_3 + x_1\overline{x_2}x_3 + x_1x_2\overline{x_3} + x_1x_2x_3$

Необходимо получить её таблицу истинности.

$$1) \overline{x_1}x_2x_3 + x_1\overline{x_2}x_3 + x_1x_2\overline{x_3} + x_1x_2x_3 = \overline{x_1}x_2x_3 + x_1\overline{x_2}x_3 + x_1x_2(\overline{x_3} + x_3) = \overline{x_1}x_2x_3 + x_1\overline{x_2}x_3 + x_1x_2$$

$$= \overline{x_1}x_2x_3 + x_1\overline{x_2}x_3 + x_1x_2$$

$$2) (\overline{x_1}x_2x_3 + x_1\overline{x_2}x_3) + (x_1x_2\overline{x_3} + x_1x_2x_3) = \overline{x_1}x_2x_3 + x_1\overline{x_2}x_3 + x_1x_2(\overline{x_3} + x_3) = \overline{x_1}x_2x_3 + x_1\overline{x_2}x_3 + x_1x_2$$

$$= \overline{x_1}x_2x_3 + x_1\overline{x_2}x_3 + x_1x_2$$

Практическая работа № 5

Тема: Выполнение бинарных алгебраических операций над множествами

Цель работы: изучение основных форм представления бинарного отношения, анализ свойств бинарных отношений, выполнение операций на д бинарными отношениями.

Теоретическое обоснование

Кортеж — это упорядоченная последовательность элементов с возможными повторениями.

Элементы кортежа перечисляются в угловых скобках. **Пример 1.** $\langle a, b, c \rangle$

Порядок записи имеет значение: $\langle a, b \rangle$ не равно $\langle b, a \rangle$

Повторы имеют значение: $\langle a, b \rangle$ не равно $\langle a, b, a \rangle$ Длина кортежа — это число элементов в нем.

Прямое (декартово) произведение множеств А и В — это множество всех кортежей длины 2, в которых первый элемент принадлежит множеству А, второй элемент—множеству В.

Обозначение:

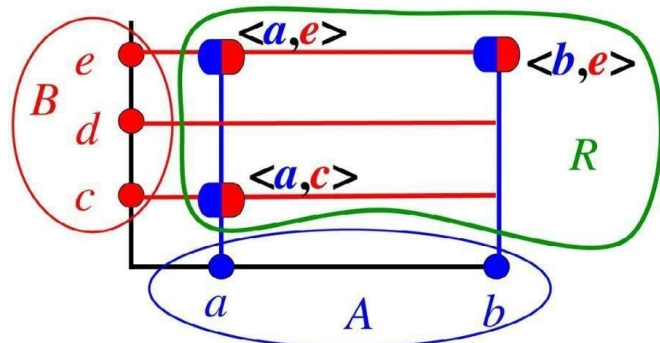
$A \times B$ — прямое произведение множеств А и В.

Бинарное отношение между двумя множествами — это подмножество их прямого произведения.

БИНАРНОЕ ОТНОШЕНИЕ МЕЖДУ МНОЖЕСТВАМИ

$$\begin{aligned} A &= \{a, b\} \\ B &= \{c, d, e\} \\ R &\subseteq A \times B \\ R &= \{\langle a, c \rangle, \langle a, e \rangle, \langle b, e \rangle\} \end{aligned}$$

А и В.



15 Пример 2.

Бинарное отношение между множествами

$A = \{a, b, c, d\}$ — множество **студентов**

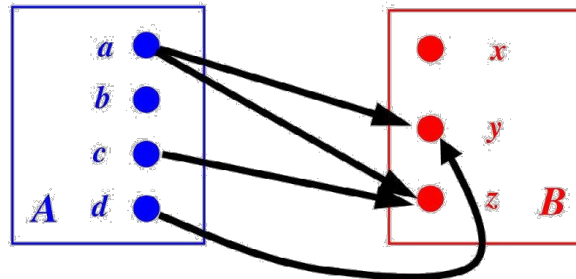
$B = \{x, y, z\}$ — множество **спортивных секций**

$R \subseteq A \times B$ — бинарное отношение между множествами A и B

$R = \{\langle a, y \rangle, \langle a, z \rangle, \langle c, z \rangle, \langle d, y \rangle\}$ — бинарное отношение

«студент занимается в секции»

Например: $\langle a, y \rangle$ — студент a занимается в секции y



Всего есть $2^{|A| \cdot |B|}$ бинарных отношений между конечными множествами A и B .

Если $R \subseteq A \times A$ то R — это бинарное отношение на множестве A .

Пример 3.

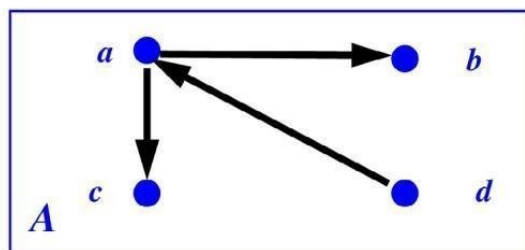
Бинарное отношение на множестве

$A = \{a, b, c, d\}$ — множество **боксеров**

$R \subseteq A \times A$ — бинарное отношение на множестве A

$R = \{\langle a, b \rangle, \langle a, c \rangle, \langle d, a \rangle\}$ — бинарное отношение «выиграл»

Например: $\langle a, b \rangle$ — боксер a выиграл у боксера b



Способы задания

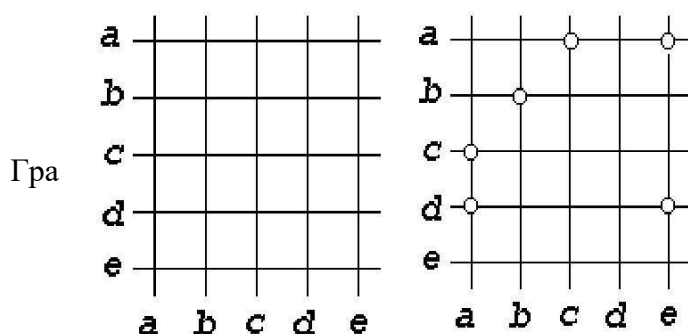
Перечисление всех пар из базового множества A и базового множества B

$A = \{a_1, a_2\}$ $B = \{b_1, b_2, b_3\}$, $R = \{(a_1, b_1), (a_1, b_3), (a_2, b_1)\}$

Отношения могут задаваться формулами. Формулы $y = x^2 + 5x - 6$ или $x + y < 5$ задают бинарные отношения на множестве действительных чисел.

Графический метод задания

$R = \{(a, d), (a, c), (b, b), (c, a), (e, d), (e, a)\}$



Графическое представление

Граф - фигура состоящая из точек

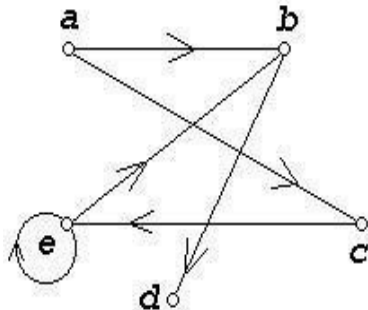
(вершин) соединенных

линиями (дугами). Вершины графа

соответствуют элементам множества A ,

то есть x_i , а наличие дуги, соединяющей вершины x_i и x_j , означает, что $(x_i, x_j) \in R$. Чтобы подчеркнуть упорядоченность пары на дуге ставится стрелка.

$$A = \{(a, b), (a, c), (b, d), (c, e), (e, b), (e, e)\}$$



Матричная форма задания (матрица смежности бинарного отношения)

Пусть на некотором конечном множестве X задано отношение A . Упорядочим какимлибо образом элементы множества $X = \{x_1, x_2, \dots, x_n\}$ и определим **матрицу отношения A**

=

$[a_{ij}]$ следующим образом:

$$a_{ij} = \begin{cases} 1, & \text{если } (x_i, x_j) \text{ принадлежит } \alpha, \\ 0, & \text{если } (x_i, x_j) \text{ не принадлежит } \alpha. \end{cases} \quad A = \begin{matrix} & \begin{matrix} a & b & c & d & e \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \\ e \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix} \end{matrix}$$

Матрица записывается в двойных вертикальных черточках

Операции над бинарными отношениями: $\square$ пересечение двух бинарных отношений R_1 и R_2 - это отношение $R_1 R_2 =$ $\square$

$\{(x, y) \mid (x, y) \in R_1 \text{ и } (x, y) \in R_2\}$. $\square$ объединение двух бинарных отношений R_1 и R_2 - это отношение $\square R_1 R_2 =$ $\square$

$\{(x, y) \mid (x, y) \in R_1 \text{ или } (x, y) \in R_2\}$. $\square$ разностью отношений R_1 и R_2 называется такое отношение, что: $R_1 \setminus R_2 = \{$

$(x, y) \mid (x, y) \in R_1 \text{ и } (x, y) \notin R_2\}$ $\square$

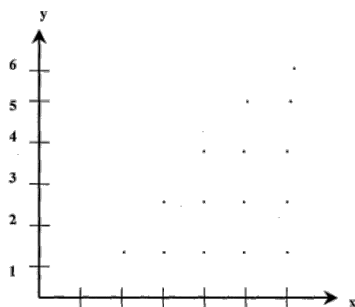
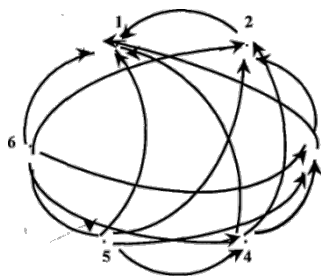
$\square$ дополнение к отношению $R = \{(x, y) \mid (x, y) \in A \setminus R\}$. $\square$

□ обратное отношение $R^{-1} = \{ (x, y) \mid (y, x) \in R \}$. □

□

Пример 4. Дано множество $A = \{ 1, 2, 3, 4, 5, 6 \}$ N. На нем задано бинарное отношение ρ «больше», т. е. $(x, y) \in \rho \Leftrightarrow x > y$. Построить граф и график этого отношения. Какими свойствами обладает это отношение?.

17



Граф указанного отношения

график этого отношения

Рефлексивность: Если бы это отношение было бы рефлексивным, то $x > x$ для $x \in A$. Например, было бы верно $2 > 2$ (ложь). Значит отношение « $>$ » на A не является рефлексивным.

Симметричность: Если бы это отношение было бы симметричным на множестве A , то $x > y \Rightarrow y > x$. Например, $3 > 2 \Rightarrow 2 > 3$ (ложь),. Значит, отношение « $>$ » на A не является симметричным.

Транзитивность: Если бы это отношение было бы транзитивным на множестве A , то $x > y, y > z \Rightarrow x > z$. Это утверждение истинно для любых натуральных чисел, т. е. и чисел из A . Значит, отношение « $>$ » на A является транзитивным.

Асимметричность: Ни для каких чисел $x, y \in A$ не может быть одновременно истинным $x > y$ и $y > x$, т. е. отношение « $>$ » на A асимметрично. Отношение « $>$ » на множестве A является ρ отношением строгого порядка т. к. оно асимметрично и транзитивно.

$$\rho_x \rho_y$$

Связность: Для любых двух элементов $x, y \in A$ верно: $x > y$ или $y > x$ т. е. отношение « $>$ » на

$$\rho_x \rho_y \rho_x$$

множестве A является связным. Т. к. отношение “ $>$ ” на множестве A связное и является отношением строгого порядка, то оно есть отношение строгого линейного порядка.

Практическая работа № 6

Тема: Выполнение операций над бинарными множествами

Цель работы: изучение основных форм представления бинарного отношения, анализ свойств бинарных отношений, выполнение операций над бинарными отношениями.

Задания для практического занятия:

Дано множество натуральных чисел $X = \{1, 2, 3, 4, 5, 6, 7\}$ и определённые на нём бинарные отношения R_1 и R_2 .

№ варианта	Отношение R_1	Отношение R_2
Вариант 1	1)Быть больше 2)Быть эквивалентным по mod 3	1)Быть меньше 2)Быть делимым
Вариант 2	1)Быть больше 2)Быть меньше	1)Быть эквивалентным по mod 2 2)Быть эквивалентным по mod 3 18
Вариант 3	1)Быть больше 2)Быть меньше	1)Быть эквивалентным по mod 3 2)Быть эквивалентным по mod 2
Вариант 4	1)Быть эквивалентным по mod 2 2)Делиться на 3	1)Делиться на 2 2)Быть делимым
Вариант 5	1)Быть эквивалентным по mod 3 2)Быть меньше	1)Делиться на 3 2)Делиться на 2

1.Представить каждое бинарное отношение во всех формах:

☐ графическая форма;☐ граф;☐ матрица смежности.☐

☐ 2.Охарактеризовать каждое бинарное отношение элементарными свойствами:

☐ рефлексивность;☐ ☐

симметричность;☐ ☐

транзитивность.☐

3. Выполнить следующие бинарные операции над отношениями R_1 и R_2 :

☐ объединение; ☐ пересечение;

- разность $R_1 \setminus R_2$ и $R_2 \setminus R_1$; симметрическая разность;
- произведение (композиция) $R_1 \circ R_2$ и $R_2 \circ R_1$.

Контрольные вопросы:

1. Что такое бинарное отношение на множестве?
2. Как можно записать бинарное отношение?
3. Какое отношение называют рефлексивным?
4. Какое отношение является антирефлексивным?
5. Какое отношение называют симметричным?
6. Какое отношение является антисимметричным?
7. Какое отношение называют транзитивным?
8. Что такое эквивалентность на множестве?

Практическая работа №7

Тема: Построение диаграмм Эйлера—Венна.

Цель работы: познакомить студентов с такими понятиями, как множество, элементы множеств, булеан, универсальное множество, мощность множества, операции над множествами, представления множеств с помощью диаграмм, формируя у них терминологический запас.

Теоретическое обоснование

Любое понятие дискретной математики можно определить с помощью понятия множества. Под множеством понимают объединение в одно общее объектов, хорошо различаемых нашей интуицией или нашей мыслью. Объекты, которые образуют множество, будем называть элементами множества и обозначать малыми буквами латинского алфавита. 19 Множество и его элементы обозначаются следующим образом:

$A = \{a_1, a_2, a_3\}$ – множество, состоящее из трех элементов;

$A = \{a_1, a_2, \dots\}$ – множество, состоящее из бесконечного числа элементов. Множество может состоять из элементов, которые сами являются множествами. Нужно различать элемент a и множество, состоящее из единственного элемента a .

Пример. Множество $A = \{1, 2\}$ состоит из двух элементов 1, 2; но множество $\{A\}$ состоит из одного элемента A .

Если элемент a принадлежит множеству A , это записывается следующим образом: $a \in A$. Если элемент a не принадлежит множеству A , то записывают так: $a \notin A$. Если какое-либо множество A включено в другое множество B , то используется запись $A \subset B$. Множество, содержащее конечное число элементов, называется конечным, если множество не содержит

ни одного элемента, то оно называется пустым и обозначается $\emptyset$. Принято считать, что пустое множество является подмножеством любого множества: $\emptyset \subset A$, где A – любое множество. Таким образом, всякое множество содержит в качестве своих подмножеств пустое множество и само себя.

Пример. 1. Множество корней уравнения $\sin x = 2$ является пустым.

2. Пусть A_1 – множество простых чисел, A_2 – множество целых чисел, $a = 4$. Тогда $a \in A_2$, а $a \notin A_1$.

Множество считается заданным, если каким-либо образом указано некоторое свойство, которым обладают все его элементы и не обладают никакие другие объекты.

Множество может быть задано различными способами: перечислением элементов (конечные множества) или указанием их свойств (при этом в обоих случаях при задании множеств используют фигурные скобки).

Примеры задания множеств. 1. Множество M цифр десятичного алфавита можно задать в виде: $M = \{0, 1, \dots, 9\}$ или $M = \{x \mid x - \text{целое}, 0 \leq x \leq 9\}$, где справа от вертикальной черты указывают свойство элементов этого множества. Множество M чётных чисел можно записать в виде: $M = \{x \mid x - \text{чётное число}\}$.

2. Если R – множество точек числовой прямой, то R_n – множество точек n -мерного арифметического пространства; в частности, R_2 – множество точек плоскости, R_3 – множество точек пространства трех измерений.

Для каждого множества M существует множество, элементами которого являются подмножества множества M и только они. Такое множество будем называть семейством множества M или булеаном этого множества и обозначать $B(M)$, а множество M будем называть универсальным (универсумом или пространством) и обозначать I или U . Множество M (универсальное) не должно быть уже объединения рассматриваемых множеств, т. е. оно должно быть равно или содержать объединение рассматриваемых множеств.

Пример. Пусть множество $A = \{1, 2\}$ состоит из двух элементов 1, 2. Тогда множество $B(A)$ включает в себя пустое множество $\emptyset$, два одноэлементных множества $\{1\}$ и $\{2\}$ и само множество $A = \{1, 2\}$, т. е.

$B(A) = \{\emptyset, \{1\}, \{2\}, \{1, 2\}\}$. Мы видим, что множество $B(A)$ состоит из четырех элементов ($4 = 2^2$).

Приведем стандартные обозначения для некоторых наиболее употребительных числовых множеств:

N – множество натуральных чисел (иногда его начинают с 1, иногда с 0; обычно это оговаривается);

P – простые числа;

Z – множество целых чисел (положительные, отрицательные и

0); R – множество действительных чисел. Очевидное соотношение: $N \subset Z \subset R$.

Рассмотрим методы получения новых множеств из уже существующих на примере пространства или множества U , определив в нём 4 операции над множествами A и B : объединение, пересечение, разность, дополнение.

1.2. Операции над множествами

Объединением A и B называется множество $A \cup B$, все элементы которого являются элементами хотя бы одного из множеств A или B :

$$A \cup B = \{x \in x \in A \text{ и / или } x \in B\}.$$

Из определения следует, что $A \subset A \cup B$ и $B \subset A \cup B$. Аналогично определяется объединение нескольких множеств.

Пример. 1. Пусть $A = \{4, 5, 6\}$, $B = \{2, 4, 6\}$. Тогда $A \cup B = \{2, 4, 5, 6\}$.

2. Пусть A – множество чисел, которые делятся на 2, а B – множество чисел, которые делятся на 3: $A = \{2, 4, 6, \dots\}$, $B = \{3, 6, 9, \dots\}$. Тогда $A \cup B$ – множество чисел, которые делятся на 2 или на 3: $A \cup B = \{2, 3, 4, 6, 8, 9, 10, \dots\}$.

Пересечением множеств A и B называется множество $A \cap B$, все элементы которого являются элементами обоих множеств A и B : $A \cap B = \{x \in x \in A \text{ и } x \in B\}$. Из определения следует, что $A \cap B \subset A$, $A \cap B \subset B$ и $A \cap B \subset A \cup B$. Аналогично определяется пересечение нескольких множеств.

Пример. 1. Пусть $A = \{4, 5, 6\}$, $B = \{2, 4, 6\}$. Тогда $A \cap B = \{4, 6\}$.

2. Пусть A – множество чисел, которые делятся на 2, а B – множество чисел, которые делятся на 3: $A = \{2, 4, 6, \dots\}$, $B = \{3, 6, 9, \dots\}$. Тогда $A \cap B$ – множество чисел, которые делятся и на 2, и на 3: $A \cap B = \{6, 12, 18, \dots\}$.

Может оказаться, что множества не имеют ни одного общего элемента. Тогда говорят, что множества не пересекаются или что их пересечение – пустое множество. Пример.

Пусть $A = \{1, 2\}$, $B = \{2, 3\}$, $C = \{3, 4\}$. Тогда $A \cap B \cap C = \emptyset$.

Разностью (относительным дополнением) множества B до множества A называется множество $A \setminus B$, все элементы которого являются элементами множества A , но не являются элементами множества B :

$$A \setminus B = \{x \in A \text{ и } x \notin B\}.$$

Пример. 1. $A = \{4, 5, 6\}$, $B = \{2, 4, 6\}$. $A \setminus B = \{5\}$, $B \setminus A = \{2\}$.

2. $A = \{2, 4, 6, \dots\}$, $B = \{3, 6, 9, \dots\}$. Тогда $A \setminus B$ – множество чисел, которые делятся на 2, но не делятся на 3, а $B \setminus A$ – множество чисел, которые делятся на 3, но не делятся на 2: $A \setminus B = \{2, 4, 8, 10, 14, \dots\}$. $B \setminus A = \{3, 9, 15, 21, 27, \dots\}$.

Симметрической разностью множеств A и B называется множество $A + B$: $A + B = (A \setminus B) \cup (B \setminus A)$.

Пример. 1. $A = \{4, 5, 6\}$, $B = \{2, 4, 6\}$. $A \setminus B = \{5\}$, $B \setminus A = \{2\}$, $A + B = \{2, 5\}$. 2. $A = \{2, 4, 6, \dots\}$, $B = \{3, 6, 9, \dots\}$, $A \setminus B = \{2, 4, 8, 10, 14, \dots\}$. $B \setminus A = \{3, 9, 15, 21, 27, \dots\}$, $A + B = \{2, 3, 4, 8, 9, \dots\}$. Дополнением M множества M является множество $\bar{M} =$

$$\{m \mid m \notin M\}.$$

Пример. Заданы множества $A = \{1, 2, 5, 6\}$ и $B = \{2, 3, 4, 6\}$ на универсальном множестве $U = \{1, 2, 3, 5, 6, 7\}$. Выполнить операции $\bar{A}$, $\bar{B}$.

Решение. В результате выполнения заданных операций получим следующие множества: $\bar{A} = \{3, 7\}$; $\bar{B} = \{1, 5, 7\}$.

Для конечных множеств существует понятие: мощность множества A – число его элементов. Обозначают мощность множества $|A|$.

Пример. $A = \{1, 2, 5, 6\}$, тогда мощность множества $|A| = n(A) = 4$; $|\bar{A}| = 0$; $|\{\bar{A}\}| = 1$.

Также справедливы следующие формулы: для любых множеств A и B $|A \cup B| = |A| + |B| - |A \cap B|$, то есть учитываются общие для обоих множеств элементы.

Пример. $A = \{1, 2, 3\}$ $|A| = 3$; $B = \{1, 2, 3, 4, 5\}$ $|B| = 5$, тогда $A \cup B = \{1, 2, 3, 4, 5\}$ $|A \cup B| = 5$; $A \cap B = \{1, 2, 3\}$ $|A \cap B| = 3$, то есть получим равенство: $|A \cup B| = |A| + |B| - |A \cap B|$ $5 = 3 + 5 - 3$.

Для конечного множества M мощность его булеана $|B(M)|$ равна $2^{|M|}$.

1.3. Применение диаграмм Эйлера-Венна при решении практических задач

Для наглядного представления множеств и отношений между ними используются диаграммы Эйлера-Венна. Универсальное множество изображают в виде прямоугольника, а множества, входящие в универсальное множество, в виде кругов внутри прямоугольника; элементу множества соответствует точка внутри круга (рис. 1).

С помощью диаграмм Эйлера-Венна удобно иллюстрировать операции над множествами. Результирующее множество каждой операции выделено штриховкой.

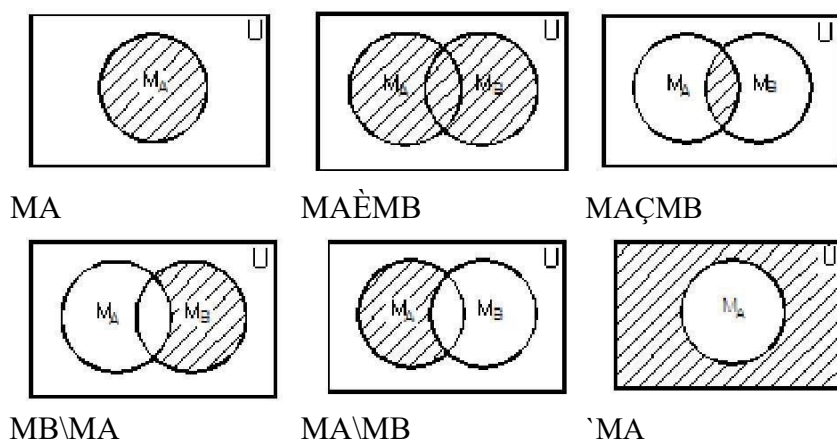


Рис. 1. Представление операций над множествами с помощью диаграмм Эйлера-Венна

Пример. Рассмотрим операцию дополнения множества, являющегося пересечением множеств MA и MB. Необходимо доказать, что её результат совпадает с объединением дополнений этих множеств:

Решение. $M = \overline{M_A \cap M_B} = \overline{M_A} \cup \overline{M_B}$.

В этом можно убедиться с помощью диаграмм Эйлера-Венна (рис. 2).

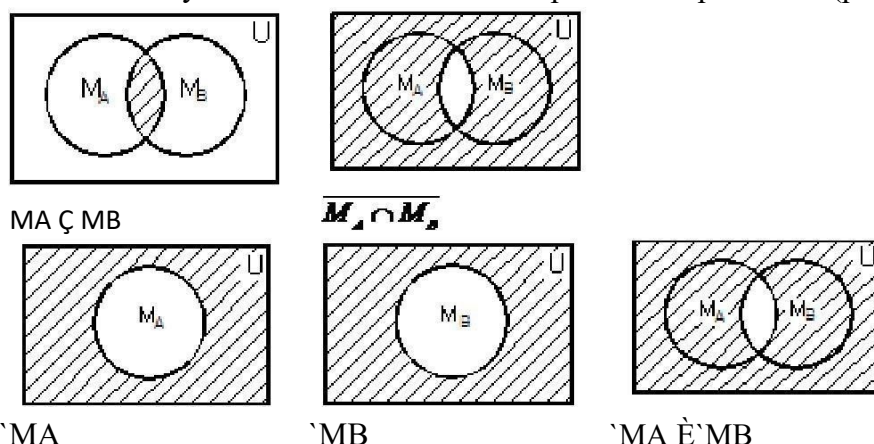


Рис. 2

Для любых подмножеств A, B и C универсального множества U выполняются следующие тождества (основные тождества алгебры множеств):

1. $A \cap B = B \cap A$ (коммутативность ∩). $A \cup B = B \cup A$ (коммутативность ∪).
2. $A \cap (B \cap C) = (A \cap B) \cap C$ (ассоциативность ∩). $A \cup (B \cup C) = (A \cup B) \cup C$ (ассоциативность ∪).
3. $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ (дистрибутивность ∩ относительно ∪). $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ (дистрибутивность ∪ относительно ∩).
4. $A \cap \emptyset = \emptyset$ (свойство нуля). $A \cup \emptyset = A$ (свойство нуля).
5. $A \cap \overline{A} = \emptyset$ (свойство дополнения). $A \cup \overline{A} = U$ (свойство дополнения).
6. $A \cap A = A$ (свойство идемпотентности). $A \cup A = A$ (свойство идемпотентности).
7. $A \cap U = A$ (свойство единицы). $A \cup \emptyset = A$ (свойство единицы).

8. $\overline{A \cup B} = \bar{A} \cap \bar{B}$ (закон де Моргана). 8с. $\overline{A \cap B} = \bar{A} \cup \bar{B}$ (закон де Моргана).
9. $A \cap (A \cup B) = A$ (закон поглощения). 9с. $A \cup (A \cap B) = A$ (закон поглощения).
10. $\overline{\bar{A}} = A$ (инволюция).
11. $A \setminus B = A \cap \bar{B}$

Примеры. 1. Рассмотрим предположение о том, что произвольные множества A и B попарно эквивалентны: 1) $A \subset B$; 2) $A \cap B = A$; 3) $A \cap B = B$.

Решение. Докажем, что из первого предположения следует второе. Действительно, так как $A \subset B \cap A$, то достаточно показать, что в этом случае $A \subset A \cap B$. Но если $x \in A$, то $x \in B$, т. к. $A \subset B$ и, следовательно, $x \in A \cap B$.

Докажем, что из второго предположения следует третье. Так как $A \cap B = A$, то $A \cap B = (A \cap B) \cap B$. По закону поглощения (см. тождество 9) $B \cap (A \cap B) = B$. Отсюда, используя закон коммутативности, получаем $A \cap B = B$.

Докажем, что из третьего предположения следует первое. Так как $A \cap B = A \cap B$, а по условию третьего предположения $A \cap B = B$, то $A \subset B$.

Практическая работа № 8

Тема: Выполнение алгебраических операций над множествами

Цель работы: познакомить студентов с такими понятиями, как множество, элементы множеств, булеан, универсальное множество, мощность множества, операции над множествами, представления множеств с помощью диаграмм, формируя у них терминологический запас.

Задания

Для аудиторных занятий

1. Записать множество M целых чисел x , которые делятся на три и находятся в интервале $3 \leq x \leq 15$. Записать двумя способами.

2. Записать множество A целых чисел x , которые делятся на 2 и на 3 и находятся в интервале $20 \leq x \leq 25$. Записать двумя способами.

3. Принадлежит ли x множеству M , если:

а) $M = \{2, 6, 8, \dots, 50\}$; $x = 35$;

б) $M = \{2, 3, 4, 6, 8, 9, 10, \dots, 100\}$; $x = 23$; в) $M = \{-2, 2, -4, 4, \dots, 120\}$; $x = -30$.

Записать приведенные множества с указанием свойств их элементов.

4. Доказать неравенство: $\{\{1, 2\}, \{2, 3\}\} \neq \{1, 2, 3\}$.

5. Какие из следующих выражений являются истинными и какие ложными:

- а) $\mathbb{A} \hat{=} \{\{\mathbb{A}\}\}$; б) $1 \hat{=} \{\{1, 2\}\}$; в) $\{1, 2\} \hat{=} \{\{1, 2\}\}$; г) $\{1, 2\} \hat{=} \{\{1, 2\}, \{1, 3\}, 1, 2\}$. 6.

Привести примеры таких множеств A, B, C , чтобы были истинными следующие высказывания:

- а) $A \hat{=} B \cup B \hat{=} C \cup A \hat{=} C$;
 б) $A \hat{=} B \cup B \hat{=} C \cup A \hat{=} C$.

7. Какие из следующих множеств конечны и какие бесконечны:

- а) $\{x \in \mathbb{R} \mid x^2 - 5x + 4 = 0\}$;
 б) $\{x \in \mathbb{N} \mid x^2 - 5x + 4 > 0\}$;
 в) $\{x \in \mathbb{N} \mid x^2 / 24\}$.

8. Равны ли множества:

- а) $\{x \in \mathbb{R} \mid x^2 - 2x - 2 = 0\}$ и $\{x \in \mathbb{Q} \mid x^2 - 2x - 2 = 0\}$;
 б) $\{x \in \mathbb{Z} \mid 4/x \in \mathbb{Z} \cup 15/x \in \mathbb{Z}\}$ и $\{x \in \mathbb{Z} \mid 4/x \in \mathbb{Z} \cup 15/x \in \mathbb{Z} \cup 20/x \in \mathbb{Z} \cup 30/x \in \mathbb{Z}\}$.

9. Перечислить элементы следующих множеств:

- а) $\{x \in \mathbb{R} \mid x^2 - 3x + 2 = 0\}$;
 б) $\{x \in \mathbb{R} \mid x^2 + 1 = 0\}$;
 в) множество всех корней уравнения $x^2 + 6x + 9 = 0$.

10. Перечислить все элементы каждого из следующих множеств:

- а) $\{x \in \mathbb{R} \mid x \in \{1\}\}$; в) $\{x \in \mathbb{R} \mid x \in \{1, 2, 3\}\}$;
 б) $\{x \in \mathbb{R} \mid x \in \{1, 2\}\}$; г) $\{x \in \mathbb{R} \mid x \in \mathbb{A}\}$. Для

самостоятельной работы

1. Перечислить элементы следующих множеств:

- а) множество четных чисел от 0 до 20 включительно;
 б) множество всех двузначных чисел, делящихся на 5 и не делящихся на 10;
 в) множество всех последовательностей, содержащих все числа 1, 2, 3, 4, 5 и только эти числа, в которых четные и нечетные числа чередуются. 2.

Равны ли множества:

- а) $\{2, 4, 5\}$ и $\{2, 4, 2, 5\}$; б) $\{1, 2\}$ и $\{\{1, 2\}\}$; в) $\{1, 2, 3\}$ и $\{3, 1, 2, 1\}$;
 г) $\{1, 2, 3\}$ и $\{\{1\}, \{2\}, \{3\}\}$.

3. Перечислить подмножества следующих множеств:

- а) $\{\{1, 2\}, \{3\}, 4\}$; б) $\{\{5, 2\}\}$; в) $\{\{2\}, \{6\}, \{3\}\}$; г) $\{\{4, 6\}, \{1\}, 1\}$.

4. Вставить между множествами символ $\hat{=}$ или $\neq$ так, чтобы получилось истинное

высказывание:

- а) $\{1\} \hat{=} \{1, \{1, 2\}\}$; б) $\mathbb{A} \hat{=} \{\mathbb{A}\}$; в) $\mathbb{A} \hat{=} \{\{\mathbb{A}\}\}$; г) $\{1, 2\} \hat{=} \{1, 2, \{1, 2\}\}$.

5. Найти $A \cap B$, $A \cup B$, $A \setminus B$, $B \setminus A$, $\bar{A}$, $\bar{B}$.

а) $A = \{1, 2, 3\}$, $B = \{2, 3, 5, 4\}$, $U = \{1, 2, \dots, 9\}$,

б) $A = \{x | x \text{ делится на } 2\}$, $B = \{x | x \text{ делится на } 3\}$, $U = \mathbb{N}$.

6. Доказать тождества:

а) $(A \cap B) \cap (A \cap B) = (A \cap B)$ е) $A \cap B = A \cap (A \cup B)$;

$= (A \cap B) \cap (A \cap B)$; б) $(A \cap B) \cap A = A \cap B$ ж) $(A \cap B) \setminus (A \cap B) =$

B ; $(A \setminus B) \cap (B \setminus A)$;

в) $(A \setminus B) \cap C = (A \cap C) \setminus (B \cap C)$; з) $(A \setminus B) \cap (A \setminus B) = (A \cap C) \setminus (B \cap C)$;

г) $A \setminus (B \cap C) = (A \setminus B) \cap (A \setminus C)$ и) $(A \setminus B) \cap (B \setminus A) = (A \cap B) \setminus (A \cap B)$;

д) $A \setminus B = A \cap \bar{B}$; к) $A \setminus (B \cap C) = (A \setminus B) \cap (A \setminus C)$.

7. Показать с помощью диаграмм Эйлера-Венна, какие из следующих утверждений верны:

а) $(A \cap B) \cap C = A \cap (B \cap C)$; в) $(A \setminus B) \cap C = (A \cap C) \setminus (B \cap C)$; б) $(A \setminus B) \cap B = \emptyset$; г) $(A \cap B) \cap (B \cap A) = A \cap B$.

8. Найти мощность множеств, мощность их булеанов и $|A \cup B|$ - мощность объединения множеств:

а) $A = \{4, 5, 6\}$, $B = \{4, 5, 6, 7, 8\}$; б) $A = \{1, 3, 5, 6\}$, $B = \{4, 5, 6, 7\}$;

в) $A = \{3, 5, 7\}$, $B = \{2, 5, 6, 7, 9\}$; г) $A = \{1, 2, 3, 5\}$, $B = \{2, 4, 5, 6, 7\}$.

9. Вставить между множествами символ $\hat{=}$ или $\neq$ так, чтобы получилось истинное высказывание:

а) $\{1\} \hat{=} \{1, \{1, 2\}\}$; г) $\mathcal{A} \hat{=} \{1, 2, \{1\}, \{\mathcal{A}\}\}$;

б) $\{1, 2\} \hat{=} \{1, 2, \{1\}, \{2\}\}$; д) $\mathcal{A} \hat{=} \{\{\mathcal{A}\}\}$; в) $\{1, 2\}$

$\{1, 2, \{1, 2\}\}$; е) $\mathcal{A} \hat{=} \{\mathcal{A}\}$.

10. Привести пример множеств A , B , C , таких, чтобы выполнялись условия: а) $A \hat{=} B$, $B \hat{=} C$, $A \neq C$;

б) $A \hat{=} B$, $A \hat{=} C$, $C \neq B$;

в) $A \hat{=} B$, $B \hat{=} C$, $A \neq C$.

Практическая работа № 9

Тема: Определение маршрутов, цепей и циклов в неориентированных графах

Цель работы: усвоение таких понятий, как маршрут, цепь и цикл графа, образ, прообраз вершин графа, минимальный путь в графе, эйлеровы графы.

Теоретическая часть:

Маршруты, цепи, циклы

Маршрутом в графе $G = (V, E)$ (путем в графе $D = (V, E)$) называется последовательность вершин и рёбер (дуг) вида $v_0, e_1, v_1, e_2, \dots, v_{n-1}, e_n, v_n$, где $v_i \in V, i \in [0, n], e_i \in E, (v_{i-1}, e_i), (v_i, e_i) \in I, i \in [1, n]$. Вершины v_0, v_n называются связанными данным маршрутом (или просто связанными). Вершину v_0 называют началом, а v_n – концом маршрута. Если $v_0 = v_n$, то маршрут называют замкнутым. Число n называется длиной маршрута.

Маршрут, в котором все рёбра попарно различны, называется цепью. Замкнутый маршрут, являющийся цепью, называется циклом (контуром). Цепь, в которой все вершины попарно различны, называется простой цепью. Цикл, в котором все вершины, кроме первой и последней, попарно различны, называется простым циклом.

Примеры: 1. Последовательность $v_1, e_1, v_2, e_3, v_4, e_4, v_3$ – маршрут длины 3, соединяющий вершины v_1, v_3 в графе G (рис. 11); это простая цепь, так как все ребра и вершины попарно различны.

2. v_2, e_3, v_2 – простой контур длины 1 в ориентированном псевдографе D (рис. 10).

3. v_1, e_2, v_2, e_4, v_3 – путь из v_1 в v_3 длины 2 в ориентированном псевдографе D (рис. 10); это простая цепь, так как все дуги и вершины попарно различны.

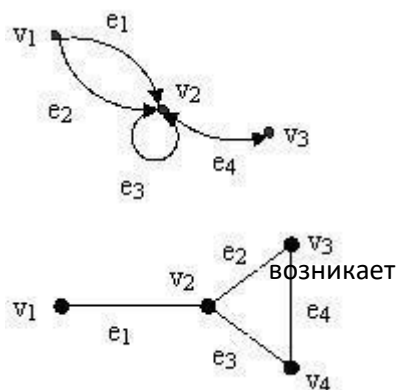


Рис. 10 Рис. 11

Поиск путей (маршрутов) с минимальным числом дуг

При

решении некоторых прикладных задач нередко

необходимо

найти минимальный маршрут, соединяющий заданные вершины в графе. Приведем алгоритм решения этой задачи. Путь в орграфе D из

вершины v в вершину w

(v^1w) называется минимальным, если он имеет минимальную

длину среди всех путей орграфа D из вершины v в вершину w .

Аналогично определяется минимальный маршрут в графе G . Алгоритм нахождения минимального маршрута состоит из следующих шагов:

Шаг 1. Помечаем вершину v индексом 0. Затем помечаем вершины, принадлежащие образу вершины v , индексом 1. Множество вершин с индексом 1 обозначаем $FW1(v)$. Полагаем $k = 1$.

Шаг 2. Если $FW_k(v) = \emptyset$ или выполняется $k = n - 1$, $w \notin FW_k(v)$, то вершина w не достижима из v и работа алгоритма на этом заканчивается. В противном случае переходим к шагу 3.

Шаг 3. Если $w \notin FW_k(v)$, то переходим к шагу 4. В противном случае существует путь из v в w длины k , и этот путь является минимальным. Последовательность вершин $v w_1 w_2 \dots w_{k-1} w$, где $w_{k-1} \in FW_{k-1}(v) \subset D^{-1}(w)$; $w_{k-2} \in FW_{k-2}(v) \subset D^{-1}(w_{k-1})$; $\dots w_1 \in FW_1(v) \subset D^{-1}(w_2)$ и есть искомым минимальный путь из v в w . На этом работа алгоритма заканчивается.

Шаг 4. Помечаем индексом $k + 1$ все непомеченные вершины, которые принадлежат образу множества вершин с индексом k . Множество вершин с индексом $k + 1$ обозначаем $FW_{k+1}(v)$. Присваиваем $k := k + 1$ и переходим к шагу 2.

Пример. Используя предложенный алгоритм, определим минимальный путь из v_1 в v_6 в орграфе D , заданном матрицей смежности, представленной в табл.

1. Таблица 1

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

Действуя согласно алгоритму, последовательно определяем $FW_1(v_1) = \{v_4, v_5\}$; $FW_2(v_1) = D(FW_1(v_1)) \setminus \{v_1, v_4, v_5\} = \{v_2, v_3\}$;

$FW_3(v_1) = D(FW_2(v_1)) \setminus \{v_1, v_2, v_3, v_4, v_5\} = \{v_6\}$. Таким образом, $v_6 \in FW_3(v_1)$, а значит (см. шаг 3) существует путь из v_1 в v_6 длины 3 и этот путь является минимальным.

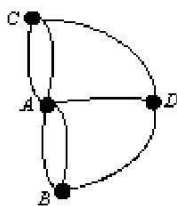
Найдем теперь минимальный путь из v_1 в v_6 . Определим множество

$$FW_2(v_1) \subset D^{-1}(v_6) = \{v_2, v_3\} \subset \{v_2, v_3\} = \{v_2, v_3\}.$$

Выберем любую вершину из найденного множества, например вершину v_3 . Определим далее множество $FW_1(v_1) \subset D^{-1}(v_3) = \{v_4, v_5\} \subset \{v_4, v_5, v_6 = \{v_4, v_5\}$. Выберем любую вершину из найденного множества, например, вершину v_5 . Тогда $v_1 v_5 v_3 v_6$ – искомым минимальный путь из v_1 в v_6 длины 3 в орграфе D .

Эйлеровы графы

Пусть G – псевдограф. Цепь (цикл) в G называется эйлеровой, если она (он) проходит по одному разу через каждое ребро псевдографа G .



Задача состоит в следующем: найти эйлеров цикл в мультиграфе G (рис. 12).

Прежде чем решать задачу о выделении эйлеровой цепи или эйлерова цикла в псевдографе, надо выяснить, существуют ли они.

Простейшее необходимое условие

их существования, очевидно, заключается в связности G . Ответ на этот вопрос дают приводимые ниже теоремы. Эйлеровым называется цикл, проходящий по каждому ребру графа ровно один раз. Граф,

Рис. 12 имеющих эйлеров цикл, тоже будем называть эйлеровым. В любом эйлеровом графе степени всех вершин чётные.

Пусть по некоторой вершине v цикл проходит k раз. Но так как перед этой вершиной и после неё цикл должен проходить по инцидентным ей рёбрам, то количество рёбер, инцидентных данной вершине, по которым проходит цикл – $2k$. Так как цикл эйлеров – рёбер, по которым цикл не проходит, нет, значит $2k$ – степень вершины v .

Если в связном графе степени всех вершин чётные, то в графе существует эйлеров цикл. В ходе доказательства мы дадим алгоритм построения такого цикла. Начнём с произвольной вершины v и будем строить из неё цепь, пока есть возможность её продолжить. Пусть в какойто момент построения мы находимся в вершине u (не совпадающей с v).

В задачах часто возникает необходимость нахождения цепи, проходящей по каждому ребру ровно один раз (снимается требование замкнутости). Такие цепи будем называть эйлеровыми цепями.

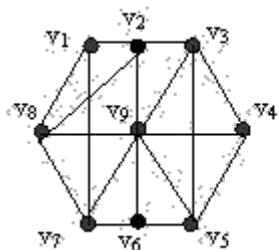
Практическая работа № 10

Тема: Минимальный путь в графе.

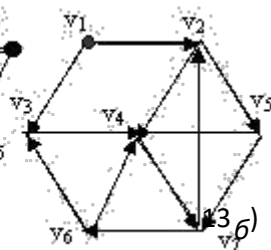
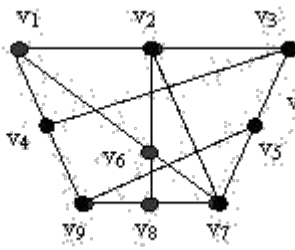
Цель работы: усвоение таких понятий, как маршрут, цепь и цикл графа, образ, прообраз вершин графа, минимальный путь в графе, эйлеровы графы

Задания Для аудиторных занятий

1. Задан граф G (рис. 13а). Найти путь с минимальным числом ребер из



вершины $v1$ в



вершину $v6$ графа G .

27

а б в а б в

Рис. 13

2. Для графа G (рис. 13а) записать матрицу смежности и найти путь с минимальным числом ребер из вершины $v2$ в вершину $v9$.

3. Для орграфа D (рис. 13б) записать матрицу смежности и найти путь с минимальным числом ребер из вершины $v1$ в вершину $v7$.

4. Определить, содержит ли граф на рис. 13а эйлеров цикл или эйлерову цепь.

5. Может ли вершина, входящая в цикл графа, иметь степень, меньшую двух?

6. Как называется путь, у которого начало первой дуги совпадает с концом последней?

7. Как называется маршрут, у которого первая вершина совпадает с последней?

8. Показать, что в любом графе количество вершин нечетной степени четно.

9. Показать, что из всякого замкнутого маршрута нечетной длины можно выделить простую цепь.

10. Показать, что ребро, входящее в цикл графа, входит в некоторый его простой цикл.

Для самостоятельной работы

1. Задан граф G (рис. 13а). Найти путь с минимальным числом ребер из вершины v_1 в вершину v_7 графа G .
2. Для графа G (рис. 13б) записать матрицу смежности и найти путь с минимальным числом ребер из вершины v_1 в вершину v_8 .
3. Для графа G (рис. 13б) записать матрицу инцидентности.
4. Для орграфа D (рис. 13в) записать матрицу смежности и найти путь с минимальным числом ребер из вершины v_1 в вершину v_6 .
5. Показать, что любая вершина, входящая в цикл, не является висячей.
6. Доказать, что если в орграфе отсутствуют вершины с нулевой полустепенью исхода (захода), то в орграфе имеется простой контур.
7. Доказать, что удаление из орграфа вершины v с $d^+(v) \leq 1$ ($d^-(v) \leq 1$) приводит к орграфу, контуры которого совпадают с контурами исходного орграфа.
8. Привести определение простой цепи:
 - а) цепь, в которой вершины и ребра повторяются;
 - б) чередование вершин и ребер;
 - в) цепь, в которой все вершины попарно различны.
9. В чем заключается понятие смежности?
 - а) смежными являются вершины, инцидентные одному ребру;
 - б) смежными являются два ребра, не имеющие общих вершин;
 - в) смежными являются вершины, соединенные некоторым маршрутом.

Задан псевдограф G . Цепь в G называется эйлеровой, если...

- а) она проходит по одному разу через каждую точку псевдографа;
- б) она проходит по одному разу через каждое ребро псевдографа;
- в) она проходит по одному разу через вершины и ребра.

Контрольные вопросы:

1. В чем состоит алгоритм поиска путей с минимальным числом дуг?
2. Какой цикл называется эйлеровым?
3. Каков критерий существования эйлерова цикла в графе?
4. Каков критерий существования эйлеровой цепи в графе?
5. Дать определение следующих понятий:

- маршрут в графе (привести пример);
- цепь и простая цепь в графе;
- цикл и простой цикл в графе;
- эйлеровой цепь;
- матрица смежности графа; – степень вершины графа

Практическая работа № 11

Тема: Определение маршрутов, цепей и циклов в ориентированных графах

Цель: научить студентов определению маршрутов, цепей и циклов в ориентированных графах **Теоретическая часть:**

Если элементы множества EE графа $G=(V,E)G=(V,E)$ — упорядоченные пары, то граф называется **ориентированным** или **орграфом**. Можно сказать, что *орграф* — это граф, ребрам которого присвоено направление.

Ребро ee графа GG называют ориентированным, если одну вершину считают началом ребра, а другую — концом, на рисунке его изображают стрелкой между вершинами. Таким образом, граф, все ребра которого ориентированы, является ориентированным графом.

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

Степенью выхода вершины орграфа называется число выходящих из вершины ребер. **Степенью входа** вершины орграфа называется число входящих в вершину ребер.

В орграфах в зависимости от сочетания степеней входа и выхода для данной вершины рассматриваются три случая:

- вершина называется **изолированной**, если ее степени выхода и входа $\square$ равны $0;\square$
 $\square$
- вершина называется **источником**, если ее степень входа равна 0 , а степень выхода больше $0;\square$
 $\square$
- вершина называется **стоком**, если ее степень выхода равна 0 , а степень $\square$ входа больше $0.\square$

Путем в орграфе называется маршрут без повторяющихся ребер, **простой путь** — без $\square$ повторяющихся вершин. Замкнутый путь в орграфе называется ориентированным циклом или контуром.

Длиной пути называется число ребер в этом пути.

Полным ориентированным графом называется орграф, каждая пара вершин которого соединена в точности одним ориентированным ребром. Очевидно, что, если с каждого ребра полного орграфа снять направление, то получится неориентированный полный граф. 29 **Петлей** называется ребро, у которого начальная и конечная вершины совпадают. Петля обычно считается неориентированной.

Практическая работа № 12

Тема: Построение направленного графа.

Цель: научить студентов определению маршрутов, цепей и циклов в ориентированных графах.

Практическое задание

Постройте направленный граф (связанный дугами) и матрицу смежности для этого графа с информацией о родственниках, содержащейся в таблице.

Имя мужчины	Дедушка	Отец	Сыновья	Братья	Дяди	Племянники	Внуки
Лев			Андрей, Петр				Алексей, Михаил, Олег
Андрей		Лев	Алексей	Петр		Михаил, Олег	
Петр		Лев	Михаил, Олег	Андрей		Алексей	
Алексей	Лев	Андрей			Петр		
Михаил	Лев	Петр		Олег	Андрей		
Олег	Лев	Петр		Михаил	Андрей		

Контрольные вопросы:

1. Дайте определение графа (определение, пример, иллюстрируемый рисунком).
2. Что такое изоморфные графы, ориентированные графы?

3. Дайте определение потока в графах.
4. Сформулируйте принцип сохранения потока.
5. Приведите примеры систем, имеющих иерархическую структуру.

Практическая работа № 13

Тема: Пример составления алгоритмов Маркова

Цель работы: научить студентов составлять алгоритмы Маркова

Теоретическая часть: Прежде всего рассмотрим возможности реализации арифметических операций с помощью нормальных алгоритмов Маркова. Сначала обратим внимание на одно обстоятельство, связанное с работой любого НАМ: нужно либо вводить дополнительное правило остановки работы нормального алгоритма (иначе в примере увеличения числа на 1 алгоритм продолжит работу и снова будет увеличивать полученный результат еще на 1 и т.д. неограниченное число раз), либо перед началом работы нормального алгоритма добавлять к входной строке специальные символы, отличные от других символов строки, которые учитываются подстановками алгоритма в начале его работы и которые удаляются в конце работы алгоритма. Мы будем придерживаться второго

способа, как и одна из наиболее успешных реализаций нормальных алгоритмов Маркова в 30 виде языка программирования Рефал. В качестве добавляемого символа возьмем символ "@". *Пример 1.* Рассмотрим простейшую операцию увеличения десятичного числа на 1. В этом случае почти всегда необходимо увеличить последнюю цифру на 1, а последняя цифра отличается тем, что после нее идет символ "@". Поэтому первыми подстановками должны быть подстановки типа

$$\langle \text{цифра} \rangle @ \rightarrow \langle \text{цифра} + 1 \rangle.$$

Но если это цифра 9, то ее нужно заменить 0 и увеличение на 1 перенести в предыдущий разряд. Этому отвечает подстановка

$$9@ \rightarrow @0.$$

Наконец, если число начинается с 9 и перед этой цифрой нужно поставить 1, то этому будет отвечать подстановка

$$@@ \rightarrow 1,$$

а если это не так, то в конце работы алгоритма символы @ надо стереть, что выполнит подстановка

$$@ \rightarrow \lambda.$$

Таким образом, мы получаем следующий НАМ увеличения десятичного числа на 1:

1. $0@ \rightarrow 1$	9. $8@ \rightarrow 9$
2. $1@ \rightarrow 2$	10. $9@ \rightarrow @0$
3. $2@ \rightarrow 3$	11. $@@ \rightarrow 1$
.....	12. $@ \rightarrow \lambda$

Приведем работу построенного алгоритма для чисел 79 и 99:

$$\begin{aligned} @79@ &\xrightarrow{10} @7@0 \xrightarrow{8} @80 \xrightarrow{12} 80, \\ @99@ &\xrightarrow{10} @9@0 \xrightarrow{10} @@00 \xrightarrow{11} 100. \end{aligned}$$

Аналогичным образом разрабатывается нормальный алгоритм Маркова для уменьшения числа на 1 (см. упражнение 1.3.1).

$$\left. \begin{array}{l} 1. 0@ \rightarrow 0!* \\ 2. 1@ \rightarrow 1!* \end{array} \right\} \text{начальные пометки копир. разряда и копии числа}$$

$$\left. \begin{array}{l} 3. 0! \rightarrow ?0a \\ 4. 1! \rightarrow ?1b \end{array} \right\} \text{копирование разряда с заменой пометки разряда}$$

$$\left. \begin{array}{l} 5. a0 \rightarrow 0a \\ 6. a1 \rightarrow 1a \\ 7. b0 \rightarrow 0b \\ 8. b1 \rightarrow 1b \end{array} \right\} \text{передвижение скопированного разряда}$$

$$\left. \begin{array}{l} 9. a* \rightarrow *a \\ 10. b* \rightarrow *b \end{array} \right\} \text{остановка передвижения скопированного разряда}$$

$$11. ? \rightarrow ! \quad \text{обратная замена пометки разряда}$$

$$\left. \begin{array}{l} 12. a \rightarrow 0 \\ 13. b \rightarrow 1 \end{array} \right\} \text{обратная замена скопированного разряда}$$

$$14. @! \rightarrow \lambda$$

Пример 2. Прежде, чем перейти к другим арифметическим операциям, рассмотрим как довольно типичный пример, используемый часто в других алгоритмах, алгоритм копирования двоичного числа. В этом случае прежде всего исходное и скопированное числа разделим символом "*". В разрабатываемом алгоритме мы будем копировать разряды числа по очереди, начиная с младшего, но нужно решить 2 проблемы: как запоминать значение символа, который мы копируем, и как запоминать место копируемого символа. Для решения второй проблемы используем символ "!", которым мы будем определять еще не скопированный разряд числа, после которого и стоит этот символ. Для запоминания значения копируемого разряда мы будем образовывать для значения 0 символ "a", а для значения 1 - символ "b". Меняя путем подстановок эти символы "a" или "b" споследующими, мы будем передвигать разряды "a" или "b" в начало копируемого числа (после "*"), но для того, чтобы пока не происходило копирование следующего разряда справа, мы перед передвижением разряда временно символ "!" заменим на символ "?", а после передвижения сделаем обратную замену. После того как все число окажется скопированным в виде символов "a" и "b", мы заменим эти символы на 0 и 1 соответственно. В результате нормальный алгоритм копирования двоичного числа можно определить следующей

31

последовательностью подстановок: Продемонстрируем работу алгоритма для числа 10:

$$\begin{aligned} @10@ &\xrightarrow{1} @10!* \xrightarrow{3} @1?0a* \xrightarrow{9} @1?0 * a \xrightarrow{11} @!10 * a \xrightarrow{4} \\ @?1b0 * a &\xrightarrow{7} @?10b*a \xrightarrow{10} @?10 * ba \xrightarrow{11} @!10 * ba \xrightarrow{13} \\ @!10 * 10 &\xrightarrow{12} @!10 * 10 \xrightarrow{14} 10 * 10 \end{aligned}$$

Для построения алгоритма сложения двух чисел можно использовать идею уменьшения одного числа на 1 с последующим увеличением другого числа на 1 и повторением этого до тех пор, пока уменьшаемое число не исчезнет после того, как станет равным 0. Но можно использовать и более сложную идею поразрядного сложения с переносом 1 в разряд слева. Построение этих алгоритмов, а также алгоритмов вычитания, умножения и деления выделено для самостоятельной работы в упражнениях 2 - 9 (см. 1.3).

Приведенные примеры показывают также возможности аппарата нормальных алгоритмов Маркова по организации ветвления и циклических процессов вычисления. Это показывает, что **всякий алгоритм может быть нормализован** т. е. задан нормальным алгоритмом Маркова. В этом и состоит тезис Маркова, который следует понимать как определение алгоритма.

Вместе с тем построение алгоритма в последнем приведенном примере подсказывает следующую методику разработки НАМ:

1. Произвести декомпозицию (разбиение на части) строящегося алгоритма.

В примере это следующие части:

1. разделение исходного числа и копии;
 2. копирование разряда; 32
 3. повторение предыдущей части до полного копирования всех разрядов.
2. Решение проблем реализации каждой части. В примере это следующие проблемы:
 1. запоминание копируемого разряда - разряд 1 запоминается как символ "a", а разряд 0 - как символ "b"; 2. запоминание места копируемого разряда - пометка еще не скопированного символа дополнительным символом "!" с заменой его на символ "?" при передвижении копируемого разряда и обратной заменой после передвижения.
 3. Если часть для реализации является сложной, то она также подвергается декомпозиции.
 4. Сборка реализации в единый алгоритм.

Практическая работа № 14

Тема: Решения задач с использованием алгоритмов Маркова

Цель работы: научить студентов составлять алгоритмы Маркова

Упражнения

1. Определите нормальный алгоритм, который уменьшает число на единицу.
2. Определите нормальный алгоритм сложения двух двоичных чисел методом уменьшения одного числа на 1 и увеличением другого числа на 1 до тех пор,

пока уменьшаемое число не станет равным 0.

3. Определите нормальный алгоритм логического сложения двух двоичных чисел.

4. Определите нормальный алгоритм логического умножения двух двоичных чисел.

5. Определите нормальный алгоритм сложения по модулю 2 двух двоичных чисел.

6. Определите нормальный алгоритм поразрядного сложения двух двоичных чисел.

7. Определите нормальный алгоритм вычитания двоичных чисел.

8. Определите нормальный алгоритм умножения двух двоичных чисел столбиком.

9. Определите нормальный алгоритм деления двух двоичных чисел с определением частного и остатка.

10. Определите нормальный алгоритм вычисления наибольшего общего делителя двух двоичных чисел.

11. Определите нормальный алгоритм вычисления наименьшего общего кратного двух двоичных чисел. **Контрольные вопросы:**

1. Выпишите понятие нормального алгоритма, приведите примеры.

2. Примените алгоритм Маркова для преобразования строки: Я купил кг Аов в Т М. Алфавит: {а...я, А...Я, пробел, точка} Правила:

1. А ☐ апельсин

2. кг ☐ килограмм

3. М ☐ магазинчике

4. Т ☐ том

5. магазинчике ☐ ларьке (заключительная подстановка) 6. в том ларьке ☐ на том рынке

Практическая работа № 15

Тема: Нахождение кратчайшего пути на графе методом Форда

Цель работы: научить студентов нахождению кратчайшего пути на графе методом

Форда

Теоретическая часть: Алгоритм нахождения расстояния от источника до всех вершин - метод Форда - Беллмана

Данные: Ориентированный граф $\langle V, E \rangle$ с n вершинами с выделенным источником $s \in V$, матрица дуг $A[u, v]$, $u, v \in V$ (граф не содержит контуров отрицательной длины). Результаты: Расстояния от источника до всех вершин графа: $D[v] = d(s, v)$, $v \in V$.

```
1 begin
2 for  $v \in V$  do  $D[v] := A[s, v]$ ;  $D[s] := 0$ ;
3 for  $k := 1$  to  $n - 2$  do
4 for  $v \in V \setminus \{s\}$  do
5 for  $u \in V$  do  $D[v] := \min(D[v], D[u] + A[u, v])$  6 end
```

Докажем правильность этого алгоритма. Для этого обозначим через $d(m)(v)$ длину кратчайшего из путей из s в v , содержащих не более m дуг ($d(m)(v) = \infty$, если не существует ни одного такого пути). Тогда имеем $d(m+1)(v) = \min\{d(m)(u) + a[u, v] : u \in V\}$, $v \in V$. (1)

В самом деле, для каждого $u \in V$ очевидно имеем $d(m+1)(v) \leq d(m)(u) + a[u, v]$, причем равенство появляется, когда u является предпоследней вершиной произвольного кратчайшего пути из s в v . Покажем теперь, что если при входе в очередную итерацию цикла

3 $d(s, v) \leq D[v] \leq d(m)(v)$ для всех $v \in V$,

(2) то по окончании этой итерации

$d(s, v) \leq D[v] \leq d(m+1)(v)$ для всех $v \in V$. (3)

Действительно, предполагая выполнение условия (2) и анализируя действие оператора в строке 5, мы видим, что по окончании итерации цикла 3 имеем $d(s, v) \leq D[v] \leq \min\{d(m)(u) + a[u, v] : u \in V\}$, что, принимая во внимание равенство (1) дает условие (3).

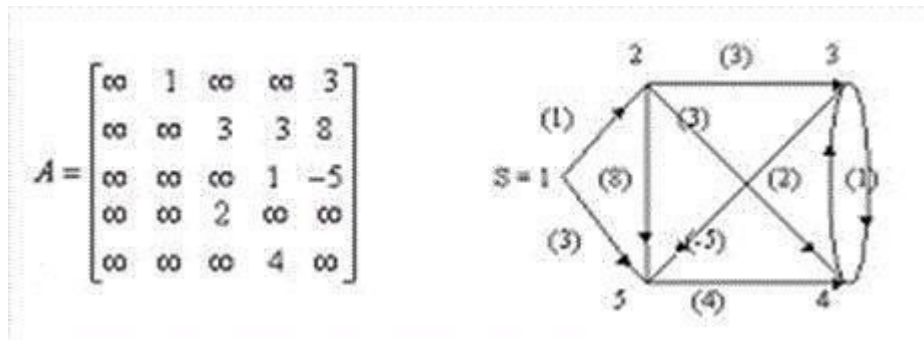
Отметим, что при входе в цикл 3 имеем $D[v] = d(1)(v)$, $v \in V$, следовательно, после выполнения $n - 2$ итераций этого цикла будут выполняться неравенства $d(s, v) \leq D[v] \leq d(n-1)(v)$, $v \in V$.

Теперь достаточно показать, что $d(n-1)(v) = d(s, v)$. Это справедливо, поскольку каждый путь более чем с $n - 1$ дугами содержит контур, устранение которого не увеличивает длины пути (ведь мы предполагаем отсутствие контуров отрицательной длины). Тем самым закончено

34 доказательство корректности алгоритма.

Очевидно, что временная сложность алгоритма есть $O(n^3)$. Мы можем, конечно, закончить вычисления, когда выполнение цикла 4 не вызывает изменения ни одной из переменных $D[v]$, $v \in V$. Это может наступить для $k < n - 2$, однако такая модификация алгоритма не изменяет существенным образом его сложности. Для редких графов ($m \ll n^2$) удобнее представлять граф списками инцидентности ПРЕДШ[v], $v \in V$. Заменяя строку 5 на **for** u ОПРЕДШ[v] **do** $D[v] := \min(D[v], D[u] + A[u, v])$, получаем алгоритм со сложностью $O(nm)$.

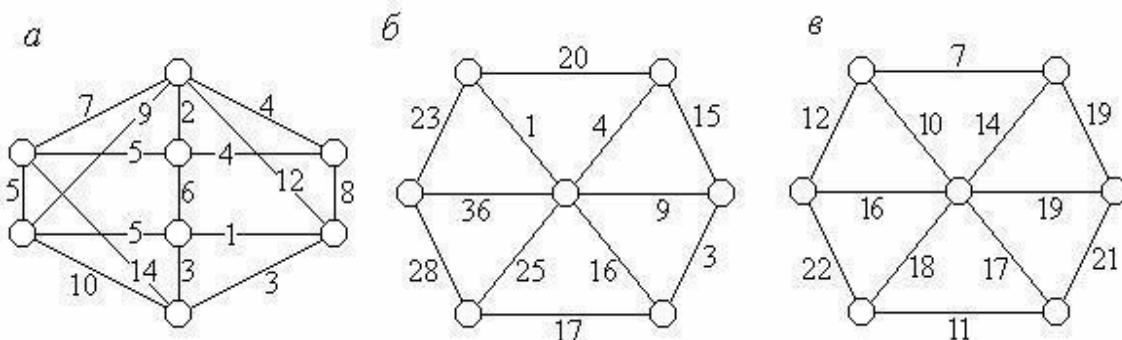
Работа алгоритма Форда - Беллмана проиллюстрирована на следующем рисунке ($V = \{1, \dots, 5\}$, веса дуг даны числами в скобках, циклы 4 и 5 выполняются в порядке возрастания номеров вершин).



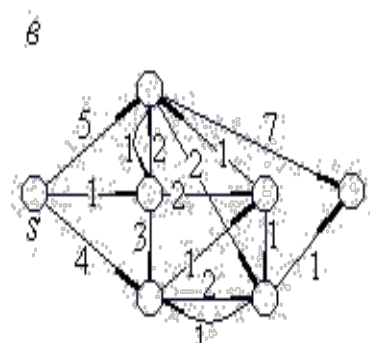
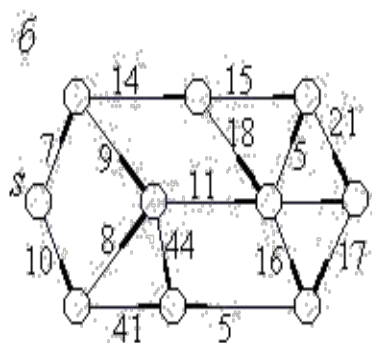
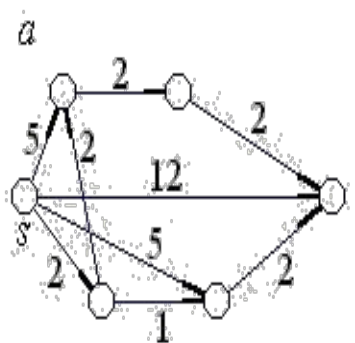
k	D[1]	D[2]	D[3]	D[4]	D[5]
	0	1	∞	∞	3
1	0	1	4	4	-1
2	0	1	4	3	-1
3	0	1	4	3	-1

Задачи для самостоятельного решения

Задание Найти минимальное остовное дерево, используя алгоритм 1) Крускала, 2) Прима.



Задание Найти кратчайшие пути от вершины-источника s до всех остальных вершин как методом Форда-Беллмана, так и методом Дейкстры.



Задание на занятие:

1. Определить кратчайшие пути между заданными вершинами S и T для графа методами - Форда-Беллмана - Дейкстры

1	2	3
4	5	6
7	8	9

6 2

0

0 4 5

0 2 6

2 0 4 1

3 2 0 4

0 5 5

1 3 0 4

2 4 0 1

1 0

Задание 1: S= 1; T = 9.

Задание 2: S= 9; T = 1.

Практическая работа № 16

Тема: Алгоритм построение максимального потока

Цель работы: научить студентов построению максимального потока на сетях

Теоретическая часть: Введем граф приращений $I(N, \varphi)$ следующим образом:

Вершины в графе те же, что и в сети, а дуг в два раза больше, причем $\forall a \in A$ соответствуют нормальной дуге a и обращенная дуга a' , длины которых, определены

если $\lambda(a) = \begin{cases} 0, \varphi(a) < \beta(a) \\ \infty, \varphi(a) = \beta(a) \end{cases}$ $\lambda(a') = \begin{cases} 0, \varphi(a) > \alpha(a) \\ \infty, \varphi(a) = \alpha(a) \end{cases}$ правилом: 36 будет максимальным, в $I(N, \varphi)$ нет путей из V_1 в V_n Поток φ нулевой

длины. Алгоритм построения

максимального потока

1. Задание начального допустимого потока φ_0 .

2. $i = 0$.

3. Построение графа приращений $I(N, \varphi_i)$.

4. Определение кратчайшего расстояния в $I(N, \varphi_i)$ и

соответствующей ему цепи C_i кратчайшего пути.

5. Если $\lambda_i < \infty$, то выполнение п.6, иначе выполнение п.10.

σ_i по цепи C_i . $\varphi_{i+1} = \varphi_i + \sigma_i$
 $\varphi_{i+1} = \varphi_i + t\sigma_i$ $i = i + 1$.

6. Строим простой поток

7. (в общем случае:).

8.

9. Переходим к пункту 3. 10.

Окончание алгоритма.

Алгоритм отыскания максимального потока в произвольных сетях с ограниченными пропускными способностями

Если нам известен некий допустимый поток в сети V , то берем его в качестве начального и используем предыдущий алгоритм.

Рассмотрим вопрос нахождения начального допустимого потока:

Для нахождения начального потока строим вспомогательную сеть $N'(V', A')$: , в которой множество вершин V' включает, кроме V , еще 2 дополнительные вершины v_0, v_{n+1} , а множество дуг A' определяется следующим образом: каждой дуге $a \approx (v, w)$ будет соответствовать 3 дуги:

$$a' \approx (v, w), \quad a'' \approx (v_0, w), \quad a''' \approx (v_0, v_{n+1}), \\ b' \approx (v_1, v_n)$$

Кроме того добавляем дугу

Пропускные способности определяются следующим образом:

43

$$\alpha'(a') = \alpha'(a'') = \alpha'(a''') = \alpha'(b') = 0. \\ \beta'(a') = \beta(a) - \alpha(a) \\ \beta'(a'') = \beta'(a''') = \alpha(a) \\ \beta''(b') = K, \text{ где } K$$

Тогда сеть $N'(V', A')$
Определение. Допустимый поток

- очень большое целое число.

- транспортная.

φ из нового источника v_0 в новый сток

$$v_0 \quad \varphi'(a'') = \beta'(a'') \quad (\text{или} \quad \varphi'(a''') = \beta'(a'''))$$

$$V' \quad \varphi(a) = \varphi'(a') + \alpha(a)$$

, то поток $\varphi(b')$, а его величина равняется $\varphi(b')$.

называется насыщенным, если для всех дуг из

).

Теорема. Если φ насыщенный поток из

в сети N

$\forall a \in A$ является допустимым из V_1 в V_n

Если ϕ - максимальный, но ненасыщенный в сети N , то в N нет допустимых потоков вообще.

Практическая работа № 17

Тема: Построение максимального потока на сетях

Цель работы: научить студентов построению максимального потока на сетях

Упражнения

1. Добавьте стоимости в класс допустимых потоков из упражнения 22.74. Для решения задачи о допустимых потоках минимальной стоимости воспользуйтесь классом MINCOST.

2. Приведите более точную верхнюю границу, чем ЕСМ, для стоимости максимального потока в транспортной сети, не все ребра которой имеют

максимальную пропускную способность и максимальную стоимость.

3. Докажите, что если все пропускные способности и стоимости выражены целыми числами, то задача о минимальной стоимости имеет решение, в котором все

потоки имеют целочисленные значения.

4. Реализуйте функцию `negsuc()` для программы 22.9, воспользовавшись

алгоритмом Беллмана-Форда (см. упражнение 21.134).

5. Замените в программе 22.9 инициализацию вычислением потока на

инициализацию потоком в фиктивном ребре.

6. Приведите все возможные последовательности расширяющих циклов,

которые могли бы быть показаны на [рис. 22.41](#).

7. Приведите все возможные последовательности расширяющих циклов,

которые могли бы быть показаны на [рис. 22.42](#).

8. Покажите в стиле [рис. 22.41](#) поток и остаточные сети после каждого

расширения при использовании программы 22.9 (реализация алгоритма вычеркивания циклов) для отыскания потока минимальной стоимости в транспортной сети, изображенной на [рис. 22.10](#). Ребра 0-2 и 0-3 имеют стоимость 2, ребра 2-5 и 3-5 - стоимость 3, ребро 1-4 - стоимость 4, а все остальные ребра - единичную стоимость. Предполагается, что максимальный поток вычисляется с помощью алгоритма, который использует кратчайшие расширяющие пути.

9. Выполните упражнение 22.112 в предположении, что программа

начинает работу с максимальным потоком в фиктивном ребре из истока в сток, как показано на [рис. 22.42](#).

10. Добавьте в решения упражнений 22.6 и 22.7 обработку стоимостей
в

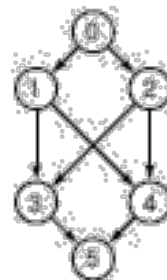
транспортных сетях.

11. Добавьте в решения упражнений 22.9-22.11 обработку стоимостей в сетях. Назначьте каждому ребру стоимость, приблизительно пропорциональную евклидовому расстоянию между его вершинами.

пр. от.	стои-мость	поток
0-1	3	0
0-2	3	0
1-3	2	0
1-4	2	0
2-3	1	0
2-4	1	0
3-5	2	0
4-5	2	0



пр. от.	стои-мость	поток
0-1	3	0
0-2	3	0
1-3	2	0
1-4	2	0
2-3	1	0
2-4	1	0
3-5	2	0
4-5	2	0

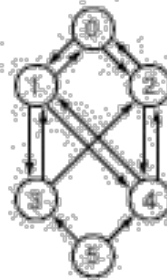


Начальный максимальный поток

пр. от.	стои-мость	поток
0-1	3	2
0-2	3	2
1-3	2	1
1-4	2	1
2-3	1	1
2-4	1	1
3-5	2	2
4-5	2	2



пр. от.	стои-мость	поток
0-1	3	2
0-2	3	2
1-3	2	1
1-4	2	1
2-3	1	1
2-4	1	1
3-5	2	2
4-5	2	2



общая стоимость: 22

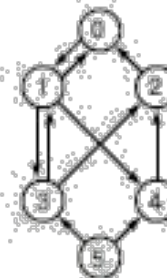
отрицательные циклы:
1-1-0-2-4
3-2-0-1-3
3-2-4-1-3

Увеличиваем на +1 по 1-1-0-2-4 (стоимость -1)

пр. от.	стои-мость	поток
0-1	3	1
0-2	3	3
1-3	2	1
1-4	2	0
2-3	1	1
2-4	2	2
3-5	2	2
4-5	2	1



пр. от.	стои-мость	поток
0-1	3	1
0-2	3	3
1-3	2	1
1-4	2	0
2-3	1	1
2-4	2	2
3-5	2	2
4-5	2	1



общая стоимость: 21

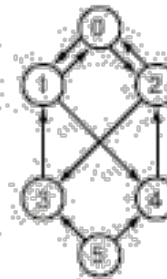
отрицательные циклы: 3-2-0-1-3

Увеличиваем на +1 по 3-2-0-1-3 (стоимость -1)

пр. от.	стои-мость	поток
0-1	3	2
0-2	3	2
1-3	2	2
1-4	2	0
2-3	1	0
2-4	2	2
3-5	2	2
4-5	2	1



пр. от.	стои-мость	поток
0-1	3	2
0-2	3	2
1-3	2	2
1-4	2	0
2-3	1	0
2-4	2	2
3-5	2	2
4-5	2	1



общая стоимость: 20

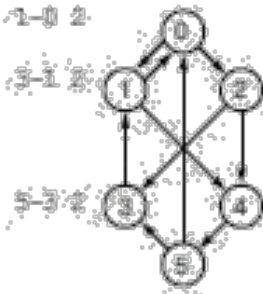
Рис. 22.41. Остаточные сети (вычеркивание циклов)

Увеличение на 2 по 0-1-3-5-0 (прямая +5)

	пр.	стат.	стат.	пото.
0-1	3	3	2	0
0-2	3	1	0	0
1-3	2	1	0	0
1-4	2	1	0	0
2-3	1	4	0	0
2-4	2	2	0	0
3-5	2	2	2	2
4-5	4	1	0	0
5-0	0	0	0	0



0-1	1	1-0	2
0-2	3	0-1	2
1-4	2	1-4	2
2-3	1	2-3	1
2-4	2	2-4	2
4-5	2	4-5	2
5-0	4	5-0	4



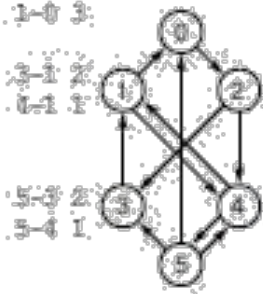
общая стоимость: 12

Увеличение на 1 по 0-1-4-5-0 (прямая +5)

	пр.	стат.	стат.	пото.
0-1	3	3	3	0
0-2	3	1	0	0
1-3	2	1	2	0
1-4	2	1	1	0
2-3	1	4	0	0
2-4	2	2	0	0
3-5	2	2	2	2
4-5	2	1	1	0
5-0	0	0	0	0



1-0	3	1-0	3
0-2	3	0-2	3
1-4	1	1-4	1
2-3	1	2-3	1
2-4	2	2-4	2
4-5	2	4-5	2
5-0	1	5-0	1



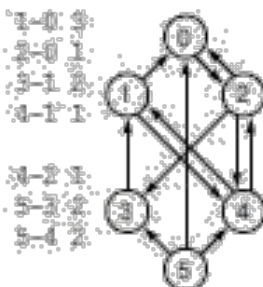
общая стоимость: 17

Увеличение на 1 по 0-2-4-5-0 (прямая +4)

	пр.	стат.	стат.	пото.
0-1	3	3	3	0
0-2	3	1	1	0
1-3	2	1	2	0
1-4	2	1	1	0
2-3	1	4	0	0
2-4	2	2	1	0
3-5	2	2	2	2
4-5	2	1	1	0
5-0	0	0	0	0



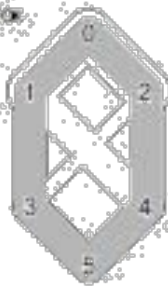
3-0	4	3-0	4
3-5	1	3-5	1
4-1	1	4-1	1
2-3	1	2-3	1
2-4	1	2-4	1
5-0	0	5-0	0



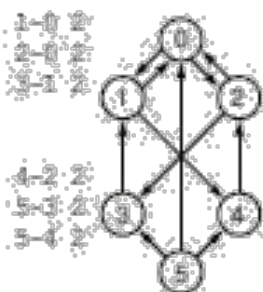
общая стоимость: 21

Увеличение на 3 по 0-1-0-2-4 (прямая +1)

	пр.	стат.	стат.	пото.
0-1	3	3	2	0
0-2	3	1	2	0
1-3	2	1	2	0
1-4	2	1	0	0
2-3	1	4	0	0
2-4	2	2	2	2
3-5	2	2	2	2
4-5	2	1	2	2
5-0	0	0	0	0



0-1	1	1-0	2
0-2	1	0-2	1
1-4	2	1-4	2
2-3	1	2-3	1
4-5	2	4-5	2
5-0	0	5-0	0



общая стоимость: 20

Тема: Построение остовного дерева

Цель работы: научить студентов построению остовного дерева.

Теоретическая часть: Предположим, что необходимо принять решение, связанное с организацией сети компьютеров в различных территориальных пунктах. Это решение является довольно сложным и зависит от большого количества факторов, которые включают в себя вычислительные ресурсы, доступные в каждом пункте, соответствующие уровни потребностей, пиковые нагрузки на систему, возможное неэффективное использование основного ресурса в системе и, кроме того, стоимость предлагаемой сети. В эту стоимость входят приобретение оборудования, прокладка линий связи, обслуживание системы и т.д. Необходимо определить стоимость такой сети.

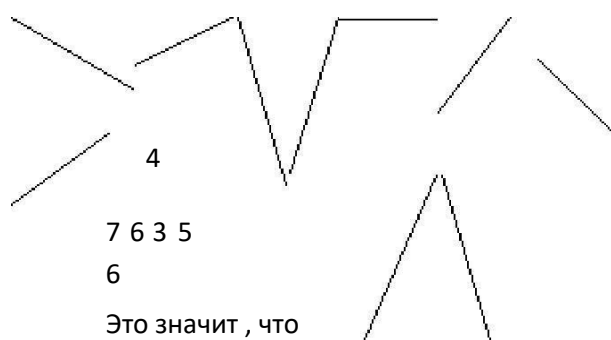
Нетрудно видеть, что сформулированная здесь задача имеет много других аналогов. Например, требуется соединить несколько населенных пунктов линиями телефонной связи таким образом, чтобы все эти пункты были связаны в сеть, и чтобы стоимость прокладки коммуникаций была минимальной. Вместо телефонных линий можно говорить о прокладке водопроводных коммуникаций, о строительстве дорог и т. д. Решение подобных задач возможно с использованием теории графов (сетей).

Пусть $G=(V,E)$ - связный неориентированный граф, содержащий циклы, т.е. замкнутые маршруты, где V – множество вершин, а E – множество ребер. Остовным (покрывающим) деревом называется подграф, не содержащий циклов, включающий все вершины исходного графа, для которого сумма весов ребер минимальна.

Введем понятие цикломатического числа (γ) показывающего, сколько ребер на графе нужно удалить, чтобы в нём не осталось ни одного цикла. Цикломатическое число равно, увеличенной на единицу разности между количеством ребер и количеством вершин графа. $\gamma = n - m + 1$, где n – количество ребер, m – количество вершин,

Например, для графа, изображенного на рисунке, цикломатическое число равно

$$\gamma = n - m + 1 = 7 - 5 + 1 = 3$$



3 5

Это значит, что
три ребра, то в

если на графе убрать 2 4

нем не останется ни

одного цикла, а суммарный вес ребер равен 14.

Для построения остовного дерева графа используются алгоритмы Крускала и Прима.

На рис.1 показана схема алгоритма Крускала.

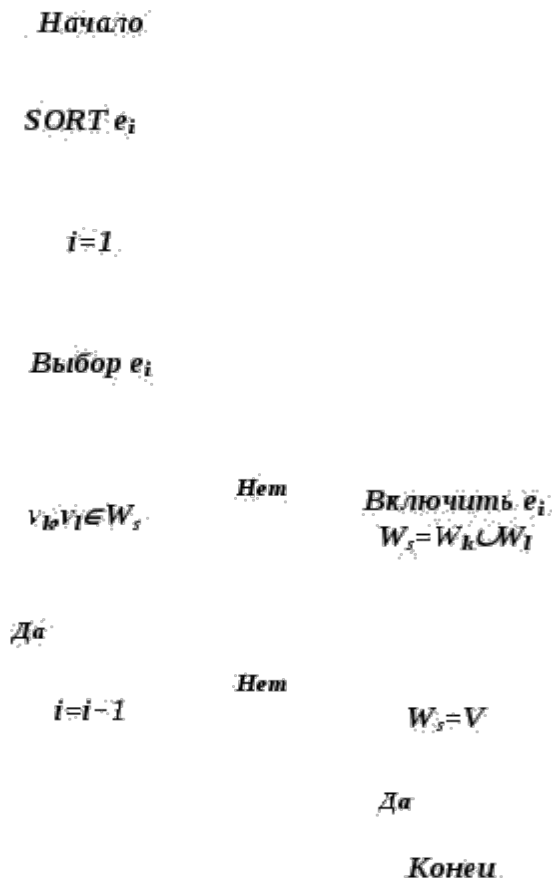


Рис.1. Схема алгоритма Крускала

Блок *Sorte* предполагает предварительную сортировку весов ребер в порядке их возрастания. В начале работы алгоритма предполагается, что в искомом остове не проведено ни одно ребро (т.е. остов состоит из изолированного множества вершин $v_1, v_2, \dots, v_m$, где m -

количество вершин графа). Поэтому считается, что множество W_s имеет вид: $W_s = \{\{v_1\}, \{v_2\}, \dots, \{v_m\}\}$, где $\{v_i\}$ обозначает множество, состоящее из единственной изолированной вершины v_i . Проверка предполагает установление факта: входят ли вершины во множество как изолированные, или они сами входят в подмножества постепенно увеличивающихся связанных вершин, каждое из которых имеет вид: $W_k = \{\dots, v_k, \dots\}$ $W_l = \{\dots, v_l, \dots\}$

вид:

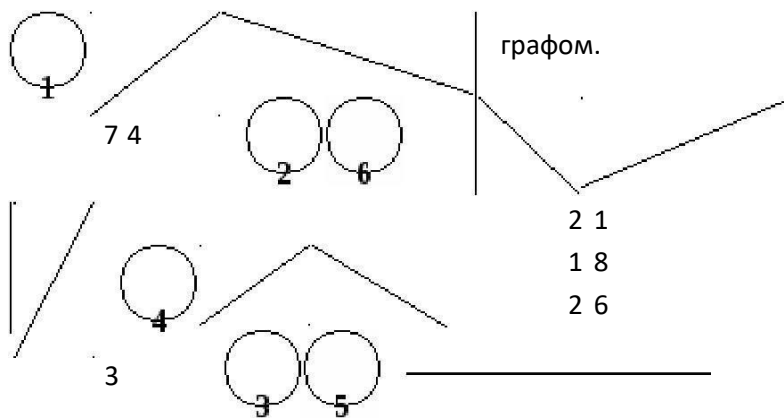
и

Если обе вершины содержатся в одном из подмножеств, то ребро (k, l) в остов не включается, в противном случае данное ребро включается в остов, а множества объединяются. Работа алгоритма Крускала заканчивается, когда множество совпадет по мощности с множеством V - множеством всех вершин графа. Нетрудно видеть, что это произойдет, когда все вершины графа окажутся связными.

Пример 1. Пусть дана схема микрорайона.

Необходимо соединить дома телефонным кабелем, таким образом, чтобы его длина 42 была минимальной.

Схему микрорайона представим взвешенным



2

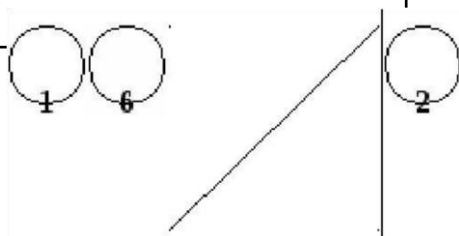
Определим цикломатическое число графа $\gamma = n - m + 1 = 10 - 6 + 1 = 5$,

т.е. на графе необходимо удалить 5 ребер.

Первоначально каждая вершина исходного графа помещается в одноэлементное подмножество, то есть считаем, что все вершины изолированы, т.е не связаны.

Ребро включается в остоное дерево, если оно связывает вершины, принадлежащие разным подмножествам, при этом вершины объединяются в новое подмножество.

Ребро	ес	Множества вершин	Операция
{V1},{V2},{V3},{V4} ,{V5},{V6}			
{V2,V3}		{V2,V3},{V1},{V4},{V5},{V6}	Вкл.
{V4,V6}		{V2,V3},{V4,V6},{V1},{V5}	Вкл.
{V1,V4}		{V2,V3},{V1,V4,V6},{V5}	Вкл.



$\{V3, V4\}$		$\{V1, V2, V3, V4, V6\}, \{V5\}$	Вкл.
$\{V2, V4\}$			Искл.
$\{V3, V5\}$		$\{V1, V2, V3, V4, V5, V6\}$	Вкл.

В таблицу последовательно включаются ребра в порядке возрастания их весов.

Ребро $\{V2, V3\}$ связывает две вершины, принадлежащие разным подмножествам $\{V2\}$ и $\{V3\}$.

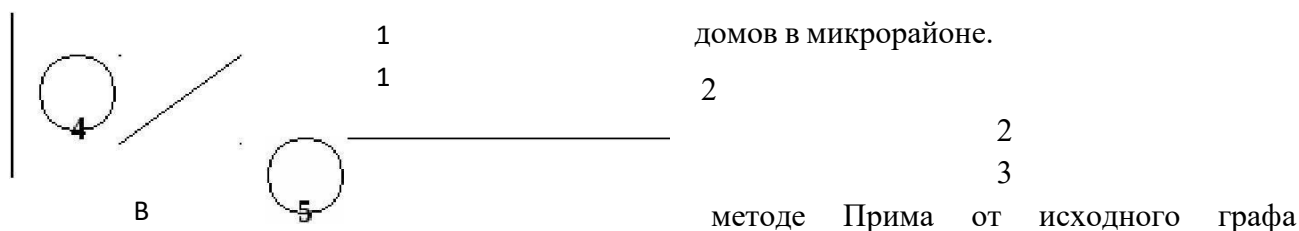
Поэтому ребро включается в остовное дерево, а вершины объединяются в одно подмножество $\{V2, V3\}$.

Ребро $\{V4, V6\}$ также связывает вершины из разных подмножеств, оно включается в остовное дерево, а вершины образуют подмножество $\{V4, V6\}$.

Вершины $V2$ и $V4$ находятся в одном подмножестве, поэтому ребро $\{V2, V4\}$ исключается из рассмотрения.

Алгоритм заканчивает работу, когда все вершины объединяются в одно множество, при этом оставшиеся ребра не включаются в остовное дерево.

Последовательно просматривая таблицу, получим схему соединения телефонным кабелем



переходим к его представлению

в виде матрицы смежности. На графе выбирается ребро минимального веса. Выбранное ребро вместе с вершинами образует первоначальный фрагмент остовного дерева. Затем анализируются длины ребер от каждой вершины фрагмента до оставшихся невыбранных вершин. Выбирается минимальное ребро и присоединяется к первоначальному фрагменту, и т. д.

Процесс продолжается до тех пор, пока в остовное дерево не будут включены все вершины исходного графа.

Пример 2. Пусть дана схема компьютерной сети.

Необходимо соединить компьютеры таким образом, чтобы длина проводки была минимальной.



Определим цикломатическое число графа:

$$\gamma = n -$$

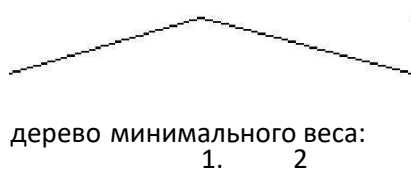
$$m + 1$$

$$= 8 - 5 + 1 = 4$$

Выбранные вершины	Невыбранные вершины			
	V1	V2	V3	V4
V5	10	<u>6</u>	7	11
V2	<u>1</u>	*	7	3
V1	*	*	<u>2</u>	3
V3	*	*	*	<u>3</u>

При просмотре строки таблицы находим минимальную величину веса ребра, выделяем её и больше этот столбец, в котором находится эта величина, в рассмотрении не участвует.

Для построения остовного дерева необходимо просмотреть столбцы таблицы снизу



вверх и
зафиксировать первое
появление минимальной величины.

В итоге получим
остовное

Для проведения работы необходимо выполнить следующие действия.

1. Изучить основные понятия теории графов

Путь к файлу: D:\ИПОВС\АиСД\OSTOV\Теория графов

2. Вызвать систему **Ostov1**, включающую в себя изучение методов Крускала и Прима. Путь к файлу: D:\ИПОВС\АиСД\OSTOV\Ostov1

Система работает в диалоговом режиме с использованием “меню”. Вся необходимая информация во время работы системы отображается на экране дисплея и не требует специальных пояснений.

Для графов, содержащих 10 и 12 вершин построить остовные деревья двумя методами.

По окончании работы выставляется оценка.

3. Вызвать системы **Ostov2** и **Ostov3** для исследования методов Крускала и Прима. Путь к файлу: D:\ИПОВС\АиСД\OSTOV\Ostov2 (Ostov3) Результаты записать в тетрадь.

~~Требования к отчёту~~ Отчёт должен

содержать:

1. конспект лабораторной работы;
2. примеры работы методов Крускала и Прима;
3. результаты выполнения работы;
4. выводы по работе;

Контрольные вопросы

1. Что понимается под остовным деревом?
2. Каковы особенности методов Крускала и Прима?
3. В чём состоит методика анализа сложности алгоритмов построения остовного дерева графа?

Основная литература:

1. Хусаинов, А. А. Дискретная математика : учебное пособие для СПО / А. А. Хусаинов. — Саратов : Профобразование, 2019. — 77 с. — ISBN 978-5-4488-0281-2. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/86136.html>. — Режим доступа: для авторизир. Пользователей
2. Веретенников, Б. М. Дискретная математика : учебное пособие для СПО / Б. М. Веретенников, В. И. Белоусова ; под редакцией Н. В. Чуксиной. — 2-е изд. — Саратов, Екатеринбург : Профобразование, Уральский федеральный университет, 2019. — 131 с. — ISBN 978-5-4488-0404-5, 978-5-7996-2858-1. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/87799.html>. — Режим доступа: для авторизир. Пользователей
3. Хусаинов, А. А. Дискретная математика : учебное пособие / А. А. Хусаинов. — 2-е изд. — Саратов : Ай Пи Ар Медиа, 2019. — 77 с. — ISBN 978-5-4497-0057-5. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/85811.html>. — Режим доступа: для авторизир. Пользователей

Дополнительная литература:

1. Спирина М.С., Спирин П.А. Дискретная математика. Сборник задач с алгоритмами решений. —М.: ОИЦ «Академия», 2016.

