МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт сервиса, туризма и дизайна (филиал) СКФУ в г. Пятигорске
Колледж института сервиса, туризма и дизайна (филиал) СКФУ в г. Пятигорске

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

ОП.08 ДИСКРЕТНАЯ МАТЕМАТИКА

Специальности СПО
09.02.01 Компьютерные системы и комплексы

Квалификация техник по компьютерным системам

Пятигорск 2020

Методические указания для практических занятий по дисциплине ОП.08 Дискретная математика составлены в соответствии с требованиями ФГОС СПО. Предназначены для студентов, обучающихся по специальности 09.02.01 Компьютерные системы и комплексы.

Рассмотрено на заседании ПЦК ИСТиД (филиал) СКФУ в г. Пятигорске.

Протокол №_8_от _12.03___2020 г.

Составитель Л.А. Плахутина

Директор З.А. Михалина

Пояснительная записка

Данные методические указания предназначены для закрепления теоретических знаний и приобретения необходимых практических навыков и умений по программе дисциплины "Дискретная математика" для специальности СПО 09.02.01Компьютерные системы и комплексы

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, законыалгебры логики;
- основные классы функций, полноту множествафункций, 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: Логические операции. Формулы логики. Таблица истинности. Дизъюнктивная и конъюнктивная нормальные формы.

Цель работы: обобщение, систематизация, углубление, закрепление полученных знаний.

Логические операции и таблицы истинности

1) Логическое умножение или конъюнкция:

Конъюнкция - это сложное логическое выражение, которое считается истинным в том и только том случае, когда оба простых выражения являются истинными, во всех остальных случаях данное сложенное выражение ложно.
Обозначение: F = A & B.

Таблица истинности для конъюнкции

A	**B**	**F**
1	1	1
1	0	0
0	1	0
0	0	0

2) Логическое сложение или дизъюнкция:

Дизъюнкция - это сложное логическое выражение, которое истинно, если хотя бы одно из простых логических выражений истинно и ложно тогда и только тогда, когда оба простых логических выраженbя ложны.

Обозначение: F = A v B.

Таблица истинности для дизъюнкции

A	**B**	**F**
1	1	1
1	0	1
0	1	1
0	0	0

3) Логическое отрицание или инверсия:

Инверсия - это сложное логическое выражение, если исходное логическое выражение истинно, то результат отрицания будет ложным, и наоборот, если исходное логическое выражение ложно, то результат отрицания будет истинным. Другими простыми слова, данная операция означает, что к исходному логическому выражению добавляется частица НЕ или слова НЕВЕРНО, ЧТО.

Обозначение: F = ¬A.

Таблица истинности для инверсии

A	**¬A**
1	0

0	1

4) Логическое следование или импликация:

Импликация - это сложное логическое выражение, которое истинно во всех случаях, кроме как из истины следует ложь. То есть данная логическая операция связывает два простых логических выражения, из которых первое является условием (А), а второе (В) является следствием.

«A → B» истинно, если из А может следовать B.

Обозначение: F = A → B.

Таблица истинности для импликации

A	**B**	**F**
1	1	1
1	0	0
0	1	1
0	0	1

5) Логическая равнозначность или эквивалентность:

Эквивалентность - это сложное логическое выражение, которое является истинным тогда и только тогда, когда оба простых логических выражения имеют одинаковую истинность.
«A ↔ B» истинно тогда и только тогда, когда А и B равны.
Обозначение: F = A ↔ B.
Таблица истинности для эквивалентности

A	**B**	**F**
1	1	1
1	0	0
0	1	0
0	0	1

6) **Операция XOR (исключающие или)**
«A ⊕ B» истинно тогда, когда истинно A или B, но не оба одновременно.
Эту операцию также называют "сложение по модулю два".
Обозначение: F = A ⊕ B.

A	**B**	**F**
1	1	0
1	0	1
0	1	1
0	0	0

Порядок выполнения логических операций в сложном логическом выражении

1. Инверсия;
2. Конъюнкция;
3. Дизъюнкция;
4. Импликация;
5. Эквивалентность.

Практическая работа № 2

Тема: Законы логики. Равносильные преобразования

Цель работы: обобщение, систематизация, углубление, закрепление полученных знаний по теме законы логики.

Знаки алгебры логики намеренно заменены на сложение и умножение.

№	Для ИЛИ, $\vee$	Для И, &	Примечание
1	$A \vee 0 = A$	$A \,\&\, 1 = A$	Ничего не меняется при действии, константы удаляются
2	$A \vee 1 = 1$	$A \,\&\, 0 = 0$	Удаляются переменные, так как их оценивание не имеет смысла
3	$A \vee B = B \vee A$	$AB = BA$	Переместительный (коммутативности)
4	$A \vee \neg A = 1$		Один из операторов всегда 1 (закон исключения третьего)
5		$A \,\&\, \neg A = 0$	Один из операторов всегда 0 (закон непротиворечия)
6	$A \vee A = A$	$A \,\&\, A = A$	Идемпотентности (NB! Вместо *A* можно подставить составное выражение!)
7	$\neg\neg A = A$		Двойное отрицание
8	$(A \vee B) \vee C = A \vee (B \vee C)$	$(A \wedge B) \wedge C = A \wedge (B \wedge C)$	Ассоциативный
9	$(A \vee B)\&C=(A\&C)\vee(B\&C)$	$(A\&B) \vee C = (A \vee C)\&(B \vee C)$	Дистрибутивный
10	$(A \vee B)\&(\neg A \vee B) = B$	$(A\&B) \vee (\neg A\&B) = B$	Склеивания
11	$\neg(A \vee B) = \neg A \,\&\neg B$	$\neg(A\&B) = \neg A \vee \neg B$	Правило де Моргана
12	$A \vee (A\&C) = A$	$A\&(A \vee C) = A$	Поглощение
13	$A{\rightarrow}B = \neg A \vee B$ и $A{\rightarrow}B = \neg B{\rightarrow}\neg A$		Снятие (замена) импликации
14	1) $A{\leftrightarrow}B = (A\&B) \vee (\neg A\&\neg B)$ 2) $A{\leftrightarrow}B = (A \vee \neg B)\&(\neg A \vee B)$		Снятие (замена) эквивалентности

Замена операций импликации и эквивалентности

Операций импликации и эквивалентности иногда нет среди логических операций конкретного компьютера или транслятора с языка программирования. Однако для решения многих задач эти операции необходимы. Существуют правила замены данных операций на последовательности операций отрицания, дизъюнкции и конъюнкции.

Так, заменить операцию импликации можно в соответствии со следующим правилом:

$A \to B = \neg A \vee B$

Для замены операции эквивалентности существует два правила:

$$A \Leftrightarrow B = (A \& B) \vee (\overline{A} \& \overline{B})$$

$$A \Leftrightarrow B = (A \vee \overline{B}) \& (\overline{A} \vee B)$$

В справедливости данных формул легко убедиться, построив таблицы истинности для правой и левой частей обоих тождеств.

Практическая работа № 3

Тема: Построение таблиц истинности логических функций.

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

Вариант 1

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

Вариант 2

1. Составить таблицу истинности логического выражения $(\neg B \& A) \leftrightarrow (A \to \neg B)$ XOR B
2. Построить логическую схему функции $(A \& B) \vee ((A \vee B) \wedge \neg A)$
3. Упростить логическое выражение $\neg((\neg A \vee B) \& A) \wedge (\neg A \vee \neg B)$
4. Определить, являются ли два высказывания эквивалентными ¬(¬A&BvA& (Bv ¬C)) и ¬B& (¬AvC)
5. Определить истинность или ложность высказываний (¬(X<5) v (X<3)) & (¬(X<2) v (X<1) при X=3

Вариант 3

1. Составить таблицу истинности логического выражения $\neg(B \vee A) \leftrightarrow (\neg A \to B)$ XOR A
2. Построить логическую схему функции $(\neg A \& \neg B) \vee (\neg A \& B)$
3. Упростить логическое выражение $(\neg A \vee B) \wedge (A \vee \neg B) \wedge (B \vee A)$
4. Определить, являются ли два высказывания эквивалентными ¬Cv ¬Bv ¬(Av ¬C) и ¬A&Bv ¬C&B
5. Определить истинность или ложность высказываний X>1 & (¬(X<5) v (X<3)) при X=2

Вариант 4

1. Составить таблицу истинности логического выражения $(\neg A \to \neg B) \leftrightarrow (B \,\&\, A)\ \mathrm{XOR}\ B$

2. Построить логическую схему функции $\neg((\neg A \vee B) \wedge (\neg B \vee A))$

3. Упростить логическое выражение $\neg((\neg A \vee B) \wedge (\neg B \vee A)) \vee (A \vee B)$

4. Определить, являются ли два высказывания эквивалентными ¬(A v ¬B) v ¬B&C и ¬A& (B∨C)

5. Определить истинность или ложность высказываний ¬((X>2) v (X<2)) v (X>4) при X=1

Вариант 5

1. Составить таблицу истинности логического выражения $(A \,\&\, B) \leftrightarrow (\neg A \,\&\, B)\ \mathrm{XOR}\ B$.

2. Построить логическую схему функции $\neg(A \,\&\, B) \vee (\neg(B \vee A))$

3. Упростить логическое выражение $\neg A \vee B \vee \neg(\neg B \vee A) \vee A \,\&\, B$

4. Определить, являются ли два высказывания эквивалентными

5. Определить истинность или ложность высказываний X>1 & (¬(X<5) v (X<3)) при X=2

Практическая работа № 4

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 = “это утро ясное”, a b = “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 тро 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)·(A+C) Решение. C+B·A+B·C·A+A·A≡а) Раскроем скобки (A+B)·(A+C) , следовательно,A≡б)По закону идемпотентности A·A C+B·A+B·C·A+A≡C+B·A+B·C·A8A·A в) В высказываниях А и А·С вынесем за скобки А и используя свойство

C·B+A·B+A≡C·B+A·B+)C+1(·A≡C·B+A·B+C·1, получим A+A≡A+1 Аналогично пункту в) вынесем за скобки высказывание А.)г C·B+A≡CB+)B+1(A≡C·B+A·B+A Таким образом, мы доказали закон дистрибутивности (распределительный). Преобразования “поглощение” и “склеивание” - поглощениеA≡)B+1(A≡B·A+A.Решение. B·A+Пример 2. Упростить выражение A -A≡)+B(A≡·A+B·A Решение. ·A+B·AПример 3. Упростить выражение склеивание Всякую формулуможно 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 сложения; - 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: Y∨≡Y→X (1) ≡Y→X (2) Докажем равносильность (1) с помощью таблицы истинности: X Y Y→X Y∨ 0 0 1 1 1 0 1 1 1 1 1 0 0 0 0 1 1 1 0 1 Эквиваленция выражается через конъюнкцию и импликацию)X→Y(·)Y→X(≡Y↔X)3(Из (3) и (1) получаем Y·X∨·=Y·X∨X ·∨· Y∨·≡)X∨(·)Y∨(≡Y↔X (4) Эта равносильность выражает эквиваленцию через конъюнкцию, дизъюнкцию и отрицание. Из равносильностей (3) и (2) получаем равносильность , (5)∧= Y↔X 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.

Пример 1. Упростить формулу **(A □ B) & (A □ C)**.

Пример 2. Упростить выражения $\overline{AB}+\overline{B}$, $\overline{\overline{B}C+C}$, $\overline{\overline{A}C}+B\overline{C}$ 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 утлы 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 № 5

Тема: Выполнение бинарных алгебраических операций над множествами.

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

Теоретическое обоснование

Кортеж — это упорядоченная последовательность элементов с возможными повторениями. Элементы кортежа перечисляются в угловых скобках.

Пример 1. < a, b, c >

Порядок записи имеет значение: < a, b > не равно < b, a >

Повторы имеют значение: < a, b > не равно < a, b, a >

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.

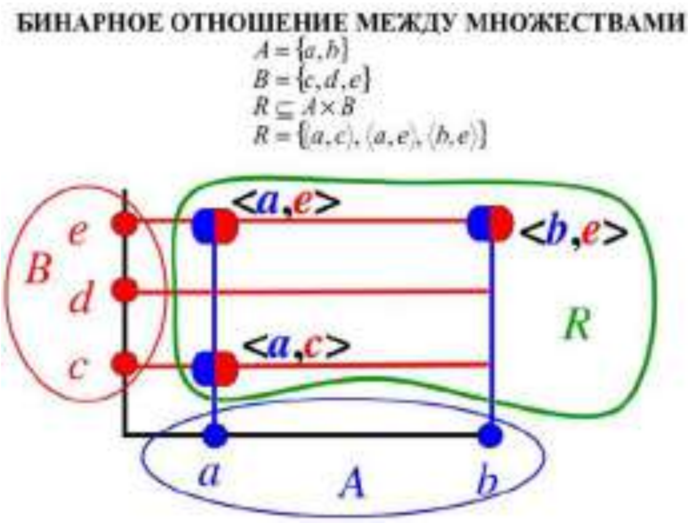
Обозначение:

А хВ — прямое произведение множеств А и В.

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

Формальное определение: $R \subseteq A \times B$.

R —это бинарное отношение между множествами А и В.



Пример 2.

Бинарное отношение <u>между</u> множествами

$A=\{a,b,c,d\}$ — множество студентов

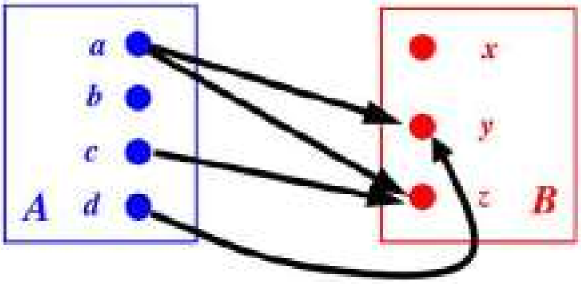
$B=\{x,y,z\}$ — множество спортивных секций

$R \subseteq A \times B$ — 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 в степени | A | | 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.

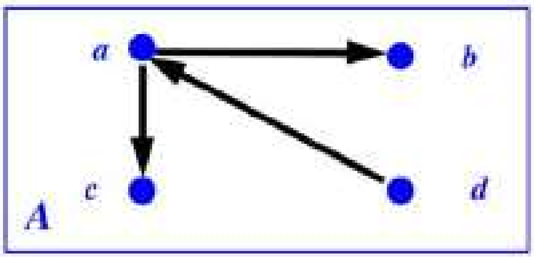
Бинарное отношение <u>на</u> множестве

$A=\{a,b,c,d\}$ — 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

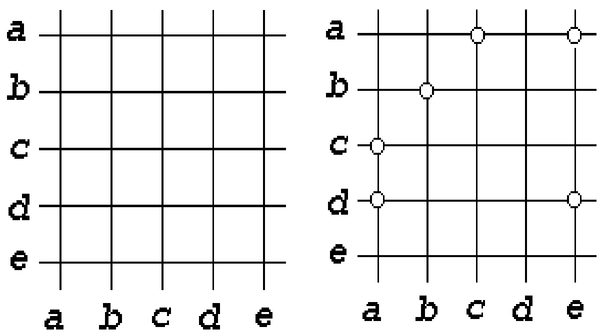
Перечисление всех пар из базового множества A и базового множества B

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

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

Графический метод задания

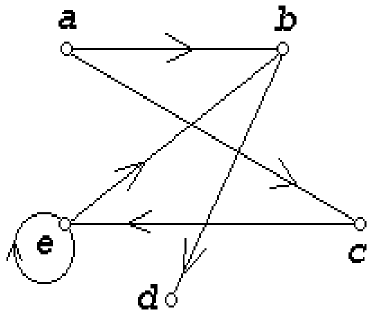
a= {(a, d), (a, c), (b, b), (c, a), (e,d), (e, a)}



Графовое представление

Граф - фигура состоящая из точек (вершин) соединенных линиями (дугами). Вершины графа соответствуют элементам множества А, 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.

А={(*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 А. Упорядочим каким-либо образом элементы множества $X = \{x_1, x_2, ..., x_n\}$ и определим **матрицу отношения**$A = [a_{ij}]$ следующим образом:

$$a_{ij} = \begin{cases} 1, \text{ если } (x_i, x_j) \text{ принадлежит } \alpha, \\ 0, \text{ если } (x_i, x_j) \text{ не принадлежит } \alpha. \end{cases}$$

$$A = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \left\| \begin{array}{ccccc} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{array} \right\| \end{array}$$

Матрица записывается в двойных вертикальных черточках

Операции над бинарными отношениями:

– пересечение двух бинарных отношений R_1 и R_2 - это отношение $R_1 \cap R_2 = \{ (x, y) \mid (x, y)\in R_1$ и $(x, y)\in R_2 \}$.

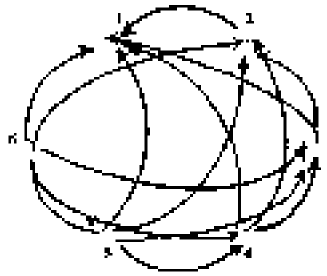
– объединение двух бинарных отношений R_1 и R_2 - это отношение $R_1 \cup R_2 = \{ (x, y) \mid (x, y)\in R_1$ или $(x, y)\in R_2 \}$.

– разностью отношений R_1 и R_2 называется такое отношение, что: $R_1 \backslash R_2 = \{ (x, y) \mid (x, y)\in R_1$ и $(x, y)\notin R_2 \}$

– дополнение к отношению $R=\{ (x, y) \mid (x, y)\in (A\times A)\backslash R\}$.

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

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



Граф указанного отношения

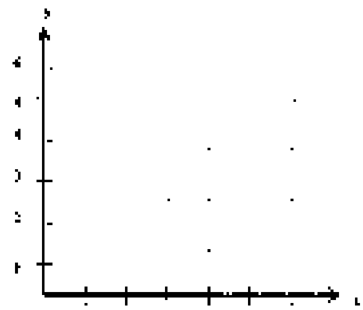


график этого отношения

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

Симметричность: Если бы это отношение было бы симметричным на множестве А, то х > у => у > х. Например, 3>2 => 2>3(ложь),. Значит, отношение « > » на А не является симметричным.

Транзитивность: Если бы это отношение было бы транзитивным на множестве А, то х > у, у >z =>х >z.Это утверждение истинно для любых натуральных чисел, т. е. и чисел из А. Значит, отношение « > » на А является транзитивным.

Асимметричность: Ни для каких чисел А не может быть одновременно истинным $\begin{cases} x > y \\ y > x \end{cases}$, т. е. отношение “>” на А асимметрично. Отношение “>” на множестве А является отношением строгого порядка т. к. оно асимметрично и транзитивно.

Связность: Для любых двух элементов $x,\ y \in A$ верно: $\left[\begin{array}{l} x > y \\ y > x \\ y = x \end{array}\right.$ т. е. отношение “>” на множестве А является связным. Т. к. отношение “>” на множестве А связное и является отношением строгого порядка, то оно есть отношение строгого линейного порядка.

Практическая работа № 6

Тема: Операции над 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 R1	Отношение R2
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
Вариант 3	1)Быть больше	1)Быть эквивалентным по mod 3

	2)Быть меньше	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 иR_2:
- объединение;
- пересечение;
- разность R1\R2 и R2\R1;
- симметрическая разность;
- произведение (композиция) R1·R2 и R2·R1.

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.

Множество и его элементы обозначаются следующим образом:

A = { a1, a2, a3} – множество, состоящее из трех элементов;

A = {a1, a2, …} – 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 Î A. Если элемент a не принадлежит множеству A, то записывают так: a Ï A. Если какое-либо множество A включено в другое множество B, то используется запись A Ì B. Множество, содержащее конечное число элементов, называется конечным, если множество не содержит ни одного элемента, то оно называется пустым и обозначается Æ. Принято считать, что пустое множество является подмножеством любого множества: Æ Í 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 A1 – множество простых чисел, A2 – множество целых чисел, a = 4. Тогда a Î A2, a Ï A1.

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, ..., 9} или M = {x | x – целое, 0 £ x £ 9}, где справа от вертикальной черты указывают свойство элементов этого множества. Множество M чётных чисел можно записать в виде: M = {x | x – чётное число}.

2. Если R – множество точек числовой прямой, то Rn – множество точек n-мерного арифметического пространства; в частности, R2 – множество точек плоскости, R3 – 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 1 или U. Множество M (универсальное) не должно быть у́же 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 Æ, два одноэлементных множества {1} и {2} и само множество A = {1, 2}, т. е.

B(A) = {Æ, {1}, {2}, {1, 2}}. Мы видим, что множество B(A) состоит из четырех элементов (4 = 22).

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 Í Z Í 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 È B, все элементы которого являются элементами хотя бы одного из множеств A или B:

A È B = {x ç x Î A и / или x Î B}.

Из определения следует, что A Í A È B и B Í A È B. Аналогично определяется объединение нескольких множеств.

Пример. 1. Пусть A = {4, 5, 6}, B = {2, 4, 6}. Тогда A È 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, …}, B = {3, 6, 9, …}. Тогда A È B – множество чисел, которые делятся на 2 или на 3: A È B = {2, 3, 4, 6, 8, 9, 10, …}.

Пересечением множеств A и B называется множество A Ç B, все элементы которого являются элементами обоих множеств A и B: A Ç B = {x ç x Î A и x Î B}. Из определения следует, что A Ç B Í A, A Ç B Í B и A Ç B Í A È B. Аналогично определяется пересечение нескольких множеств.

Пример. 1. Пусть A = {4, 5, 6}, B = {2, 4, 6}. Тогда A Ç B = {4, 6}.

2. Пусть A – множество чисел, которые делятся на 2, а B – множество чисел, которые делятся на 3: A = {2, 4, 6, …}, B = {3, 6, 9, …}. Тогда A Ç B – множество чисел, которые делятся и на 2, и на 3: A È B = {6, 12, 18, …}.

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 Ç B Ç C = Æ.

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

A \ B = {x ç x Î A и x Ï B}.

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

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

Симметрической разностью множеств A и B называется множество A + B: A + B = (A \ B) È (B \ A).

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

2. A = {2, 4, 6, …}, B = {3, 6, 9, …}, A \ B = {2, 4, 8, 10, 14, …}.

B \ A = {3, 9, 15, 21, 27, …}, A + B = {2, 3, 4, 8, 9, …}.

Дополнением`M множества M является множество

`M = {mi | mi Ï 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`A,`B.

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

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

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

Также справедливы следующие формулы: для любых множеств А и В Þ |А È В| = |А| + |В| – |А Ç В|, то есть учитываются общие для обоих множеств элементы.

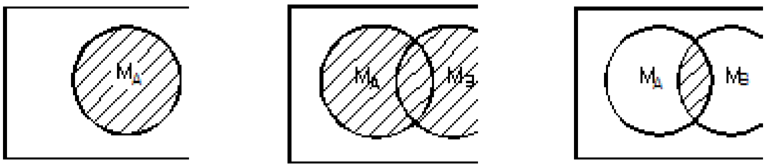
Пример. А = {1, 2, 3} |А| = 3; В = {1, 2, 3, 4, 5} |В| = 5, тогда А È В = {1, 2, 3, 4, 5} |А È В| = 5; А Ç В = {1, 2, 3} |А Ç В| = 3, то есть получим равенство: |А È В| = |А| + |В| – |А Ç В| 5 = 3 + 5 – 3.

Для конечного множества М мощность его булеана |В(М)| равна 2|М|.

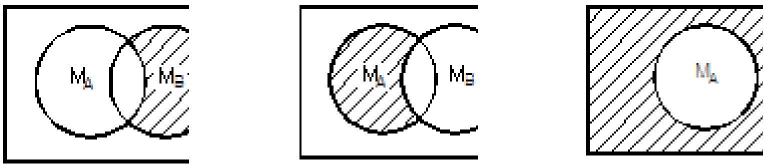
1.3. Применение диаграмм Эйлера-Венна при решении практических задач

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.



MA MAÈMB MAÇMB



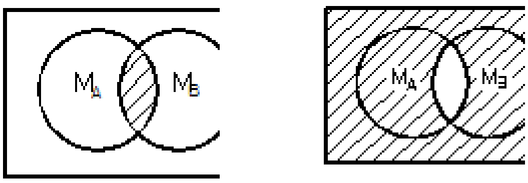
MB\MA MA\MB `MA

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

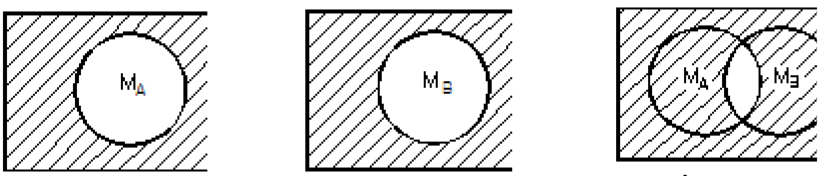
Пример. Рассмотрим операцию дополнения множества, являющегося пересечением множеств MA и MB. Необходимо доказать, что её результат совпадает с объединением дополнений этих множеств:

Решение. М = $\overline{\mathbf{M_A \cap M_B}}$ =`MA È`MB.

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



MA Ç MB $\overline{M_A \cap M_B}$



`MA `MB `MA È`MB

Рис. 2

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

1. A È B = B È A (коммутативность È).

1¢. A Ç B = B Ç A (коммутативность Ç).

2. A È (B È C) = (A È B) È C (ассоциативность È).

2¢. A Ç (B Ç C) = (A Ç B) Ç C (ассоциативность Ç).

3. A È (B Ç C) = (A È B) Ç (A È C) (дистрибутивность È относительно Ç).

3¢. A Ç (B È C) = (A Ç B) È (A Ç C) (дистрибутивность Ç относительно È).

4. A È Æ = A (свойство нуля).

4¢. A Ç Æ = Æ.

5. A È`A = U (свойство дополнения).

5¢. A Ç`A = Æ.

6. A È A = A.

6¢. A Ç A= A.

7. A È U = U (свойство единицы).

7¢. A Ç U = A.

8. $\overline{A \cup B}$ =`A Ç`B (закон де Моргана).

8¢. $\overline{A \cap B}$ =`A È`B (закон де Моргана).

9. A È (A Ç B) = A (закон поглощения).

9¢. A Ç (A È B) = A (закон поглощения).

10. $\overline{\overline{A}}$ = A (инволюция).

11. A \ B = A Ç $\overline{B}$.

Примеры. 1. Рассмотрим предположение о том, что произвольные множества A и B попарно эквивалентны: 1) A Ì B; 2) A Ç B = A; 3) A È B = B.

Решение. Докажем, что из первого предположения следует второе. Действительно, так как A Ç B Ì A, то достаточно показать, что в этом случае A Ì A Ç B. Но если x Î A, то x Î B, т. к. A Ì B и, следовательно, x Î A Ç B.

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

Докажем, что из третьего предположения следует первое. Так как A Ì A È B, а по условию третьего предположения A È B = B, тоA Ì 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 £ x £ 15. Записать двумя способами.

2.Записать множествоA целых чисел x, которые делятся на 2 и на 3 и находятся в интервале 20 £ x £ 25. Записать двумя способами.

3.Принадлежит ли x множеству M, если:

а) M = {2, 6, 8, …, 50}; x = 35;

б) M = {2, 3, 4, 6, 8, 9, 10, …, 100}; x = 23;

в) M = {-2, 2, -4, 4, …, 120}; x = -30.

Записать приведенные множества с указанием свойств их элементов.

4.Доказать неравенство: {{1, 2}, { 2, 3}} ¹ {1, 2, 3}.

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

а) Æ Î {{Æ}}; б) 1 Î {{1, 2}}; в) {1, 2} Î {{1, 2}}; г) {1, 2} Î {{1, 2}, {1, 3}, 1, 2}.

6. Привести примеры таких множествA, B, C, чтобы были истинными следующие высказывания:

а) A Î B Ù B Î C Ù A Ï C;

б) A Î B Ù B Î C Ù A Î C.

7. Какие из следующих множеств конечны и какие бесконечны:

а) {x Î Rçx2 - 5x + 4 = 0};

б) {x Î Nçx2 - 5x + 4 > 0};

в) {x Î Nçx2 / 24}.

8. Равны ли множества:

а) {x Î Rçx2 - 2x – 2 = 0} и {x Î Qçx2 - 2x – 2 = 0};

б) {x Î Zç4/x Ù 15/x} и {x Î Zç4/x Ù 15/x}{x Î Zç20/x Ù 30/x}.

9. Перечислить элементы следующих множеств:

а) {x Î Rçx2 - 3x + 2 = 0};

б) {x Î Rçx2 + 1 = 0};

в) множество всех корней уравнения x2 + 6x + 9 = 0.

10. Перечислить все элементы каждого из следующих множеств:

а) {xçx Í{ 1}}; в) {xçx Í {1, 2, 3}};

б) {xçx Í {1, 2}}; г) {xçx Í Æ}.

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 Î или Ì так, чтобы получилось истинное высказывание:

а) {1} {1, {1, 2}}; б) Æ {Æ}; в) Æ {{Æ}}; г) {1, 2} {1, 2, {1, 2}}.

5. Найти A È B, A C B, A\B, B\A, `A, `B.

а) A={1,2,3}, B={2,3,5,4}, U={1,2,…….,9},

б) A={x|x делиться на 2}, B={x|x делиться на 3}, U=N.

6. Доказать тождества:

а) (A Ç B) È (A Ç`B) = (A È B) Ç (A È`B);

б) (A È B) Ç A = A Ç B;

в) (A \ B) \ C = (A \

е) A Ç B = A Ç (A È B);

ж) (A È B) \ (A Ç B) = (A \ B) È (B \ A);

з) (A \ B) È (A \`B) = (A

С) \(В \ С);

г) А \ (В È С) = (А \ В) \ С;

д) А \ В = А Ç`В;

È В) \ (А Ç В);

и) (А \`В) È (В \`А) = (А È В) \ (А Ç В);

к) А \ (В È С) = (А \ В) Ç (А \ С).

1. Показать с помощь диаграмм Эйлера-Венна, какие из следующих утверждений верны:

а) (А È В) Ç С = А È (В Ç С);

б) (А \ В) È В = А;

в) (А \ В) È С = (А È С) \ (В È С);

г) (А Ç`В) È (В Ç`А) Ì В.

2. Найти мощность множеств, мощность их булеанов и |A È B| - мощность объединения множеств:

а) А={4,5,6}, В={4,5,6,7,8}; б) А={1,3,5,6}, В={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 Î или Í так, чтобы получилось истинное высказывание:

а) {1} {1, {1, 2}}; г) Æ {1, 2, {1}, {Æ}};

б) {1, 2} {1, 2, {1},{2}}; д) Æ {{Æ}};

в) {1, 2} {1, 2,{1, 2}}; е) Æ {Æ}.

10. Привести пример множеств *A, B, C*, таких, чтобы выполнялись условия:

а) *A* Î *B, B* Ï*C, A* Í *C*;

б) *A* Î *B, A* Ï *C, C* Í *B*;

в) *A* Í *B, B* Î *C, A* Ï *C*.

Практическая работа № 9

Тема: Определение маршрутов 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, ..., *v*n-1, *e*n, *v*n, где *v*i Î *V*, *i* Î [0, *n*], *e*i Î *E*, (*v*i-1, *e*i), (*v*i, *e*i) Î *I*, *i* Î [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.

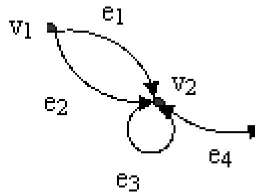
Маршрут, в котором все рёбра попарно различны, называется цепью. Замкнутый маршрут, являющийся цепью, называется циклом (контуром). Цепь, в которой все вершины попарно различны, называется простой цепью. Цикл, в котором все вершины, кроме первой и последней, попарно различны, называется простым циклом.

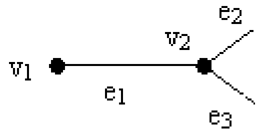
Примеры: 1. Последовательность *v*1, *e*1, *v*2, *e*3, *v*4, *e*4, *v*3 – маршрут длины 3, соединяющий вершины *v*1, *v*3 в графе *G* (рис. 11); это простая цепь, так как все ребра и вершины попарно различны.

2. *v*2, *e*3, *v*2 – простой контур длины 1 в ориентированномпсевдографе*D* (рис. 10).

3. *v*1, *e*2, *v*2, *e*4, *v*3 – путь из *v*1 в *v*3 длины 2 в ориентированномпсевдографе*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 возникает необходимость 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 FWk(*v*) = Æ или выполняется *k* = *n* – 1, w Ï FWk(*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* Ï FWk(*v*), то переходим к шагу 4. В противном случае существует путь из *v* в *w* длины *k*, и этот путь является минимальным. Последовательность вершин

vw1w2 …wk-1w, где *wk-1* Î FWk-1(*v*) Ç D-1(*w*); *wk-2* Î FWk-2(*v*) Ç D-1(*wл-1*); … *w1* Î FW1(*v*) Ç D-1(*w2*) и есть искомый 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 FWk+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 FW1(*v1*) = {*v4*, *v5*}; FW2(*v1*) = D(FW1(*v1*))\{*v1*,*v4*,*v5*} ={ *v2*,*v3*};

FW3(*v1*) = D(FW2(*v1*))\{*v1*, *v2*, *v3*, *v4*, *v5*} = {*v6*}. Таким образом, *v6Î* FW3(*v1*), а значит (см. шаг 3) существует путь из *v1* в *v6* длины 3 и этот путь является минимальным. Найдем теперь минимальный путь из *v1* в *v6*. Определим множество

FW2(*v1*) Ç D-1(*v6*) = {*v2*,*v3*} Ç {*v2*,*v3*} = {*v2*,*v3*}.

Выберем любую вершину из найденного множества, например вершину *v3*. Определим далее множество FW1(*v1*) Ç D-1(*v3*) = { *v4*,*v5*} Ç {*v4*,*v5*, *v6* = {*v4*,*v5*}. Выберем любую вершину из найденного множества, например, вершину *v5*. Тогда *v1v5v3v6* – искомый минимальный путь из *v1* в *v6* 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*.

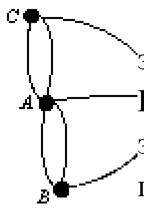


Рис. 12

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, имеющихэйлеров 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 – 2 *k*. Так как цикл эйлеров – рёбер, по которым цикл не проходит, нет, значит 2 *k* – степень вершины *v*.

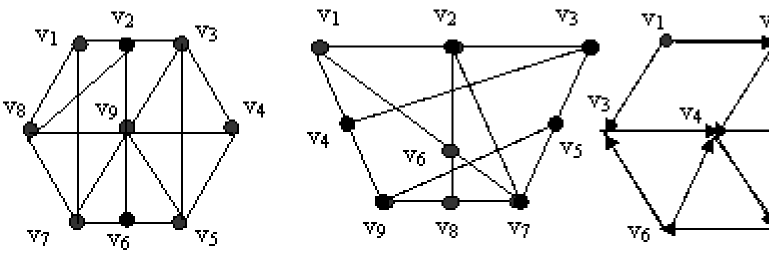
Если в связном графе степени всех вершин чётные, то в графе существует эйлеров цикл. В ходе доказательства мы дадим алгоритм построения такого цикла. Начнём с произвольной вершины *v* и будем строить из неё цепь, пока есть возможность её продолжить. Пусть в какой-то момент построения мы находимся в вершине *u* (не совпадающей с *v*). Тогда цепь, которую мы построили, проходит по нечётному числу рёбер, инцидентных данной вершине. Степень вершины *u* чётная, следовательно, есть хотя бы одно ребро, инцидентное вершине *u*, по которому цепь не прошла – значит её можно продолжить. Следовательно, цепь может закончится только в вершине *v*. Получим цикл.

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

Задания

Для аудиторных занятий

1. Задан граф *G* (рис. 13*а*). Найти путь с минимальным числом ребер из вершины *v*1 в вершину *v*6 графа *G*.



а б в

а б в

Рис. 13

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

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

4. Определить, содержит ли граф на рис. 13*а*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. 13*а*). Найти путь с минимальным числом ребер из вершины $v1$ в вершину $v7$ графа G.

2. Для графа G (рис. 13*б*) записать матрицу смежности и найти путь с минимальным числом ребер из вершины $v1$ в вершину $v8$.

3. Для графа G (рис. 13*б*) записать матрицу инцидентности.

4. Для орграфа D (рис. 13*в*) записать матрицу смежности и найти путь с минимальным числом ребер из вершины $v1$ в вершину $v6$.

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 $d+(v) \le 1$ ($d-(v) \le 1$) 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) сежными являются вершины, инцидентные одному ребру;

б) смежными являются два ребра, не имеющие общих вершин;

в) смежными являются вершины, соединенные некоторым маршрутом.

10. ЗаданпсевдографG. 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 № 10

Тема: Определение цепей и циклов в ориентированных графах

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

Теоретическая часть:

Если элементы множества *E*E 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 *e*e графа *G*G 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 равны 0;
- вершина называется **источником**, если ее степень входа равна 0, а степень выхода больше 0;
- вершина называется **стоком**, если ее степень выхода равна 0, а степень входа больше 0.

Путем в орграфе называется маршрут без повторяющихся ребер, **простой путь** — без 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.

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

Практическая работа № 11

Тема: Определение маршрутов в ориентированных графах

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.

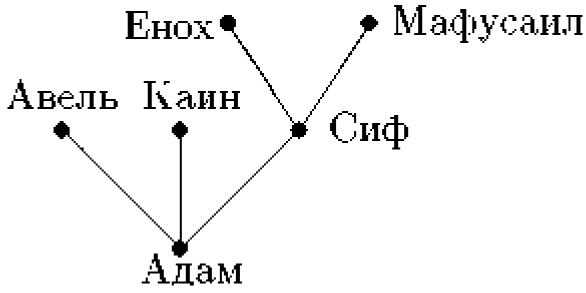
Практическая работа № 12

Тема: Определение цепей и циклов в ориентированных графах.

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

Практическое задание

Пример 1 (генеологическое дерево). На рисунке показано библейское генеологическое дерево.



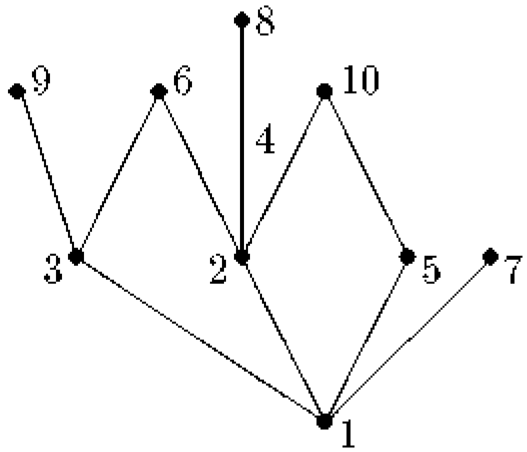
□

1. Докажите, что граф является деревом тогда и только тогда, когда любая пара различных вершин соединена единственной цепью.

2. Докажите, что граф является деревом тогда и только тогда, когда он связен, но после удаления любого ребра становится несвязным.

2.3 Докажите, что количество рёбер дерева на единицу меньше количества вершин

Пример 2(граф отношения делимости). Построим граф, изображающее отношение делимости на множестве {1,2,3,4,5,6,7,8,9,10}. Принцип такой: если от одного числа до другого есть цепь, ведущая вверх, тогда второе число делится на первое.



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 строкеспециальные 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 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

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

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

$$9@\ \to\ @0.$$

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

$$@@\ \to\ 1,$$

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

$$@\ \to\ \lambda.$$

Таким образом, мы получаем следующийНАМ увеличения десятичного числа на 1:

$$\begin{array}{ll}
1.\ 0@ \to 1 & 9.\ 8@ \to 9 \\
2.\ 1@ \to 2 & 10.\ 9@ \to @0 \\
3.\ 2@ \to 3 & 11.\ @@ \to 1 \\
\ldots\ldots\ldots\ldots\ldots & 12.\ @ \to \lambda
\end{array}$$

Приведем работу построенного алгоритма для чисел 79 и 99:

$$@7\underline{9@} \xrightarrow{10} @\underline{7@}0 \xrightarrow{8} \underline{@}80 \xrightarrow{12} 80,$$
$$@9\underline{9@} \xrightarrow{10} @\underline{9@}0 \xrightarrow{10} \underline{@@}00 \xrightarrow{11} 100.$$

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

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 последовательностью подстановок:

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

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

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

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

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

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

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

Продемонстрируем работу алгоритма для числа 10:

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

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 цикличных 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;
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: Решение задач на составления 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 алгоритмвычитания 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 <V, E> с n вершинами с выделенным источником s O V, матрица дуг A[u, v], u, v OV (граф не содержит контуров отрицательной длины).

Результаты: Расстояния от источника до всех вершин графа: D[v]=d(s, v), vO V.

```
1 begin
2 for v O V do D [v] := A[s, v]; D [s]:= 0;
3 for k  := 1 to n - 2 do
4 for v O V \ {s} do
5 for u O 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) = ╔ , если не существует ни одного такого пути). Тогда имеем

d(m+1)(v) = min{d(m)(u) + a[u, v]: u O V}, v O V. (1)

В самом деле, для каждого uO V очевидно имеем d(m+1)(v) ёd(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) ё D[v] ё d(m)(v) для всех v O V, (2)

то по окончании этой итерации

d (s, v) ё D[v] ё d(m+1)(v) для всех v O 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) ё D[v] ё min{d(m)(u) + a[u, v]: u O V},

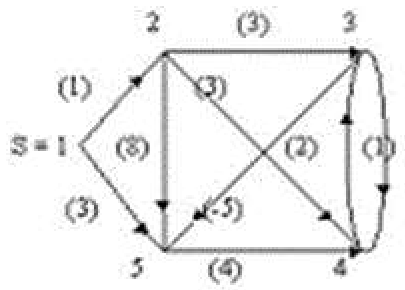
что, принимая во внимание равенство (1) дает условие (3).

Отметим, что при входе в цикл 3 имеем D[v] = d 1 (v), v O V, следовательно, после выполнения n - 2 итераций этого цикла будут выполняться неравенства d (s, v) ё D[v] ёd(n-1)(v), v O 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 доказательство корректности алгоритма.

Очевидно, что временная сложность алгоритма есть O(n3). Мы можем, конечно, закончить вычисления, когда выполнение цикла 4 не вызывает изменения ни одной из переменных D[v], v О V. Это может наступить для k < n - 2, однако такая модификация алгоритма не изменяет существенным образом его сложности. Для редких графов (m << n2) удобнее представлять граф списками инцидентности ПРЕДШ[v], v О 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, ..., 5}, веса дуг даны числами в скобках, циклы 4 и 5 выполняются в порядке возрастания номеров вершин).

$$A = \begin{bmatrix} \infty & 1 & \infty & \infty & 3 \\ \infty & \infty & 3 & 3 & 8 \\ \infty & \infty & \infty & 1 & -5 \\ \infty & \infty & 2 & \infty & \infty \\ \infty & \infty & \infty & 4 & \infty \end{bmatrix}$$

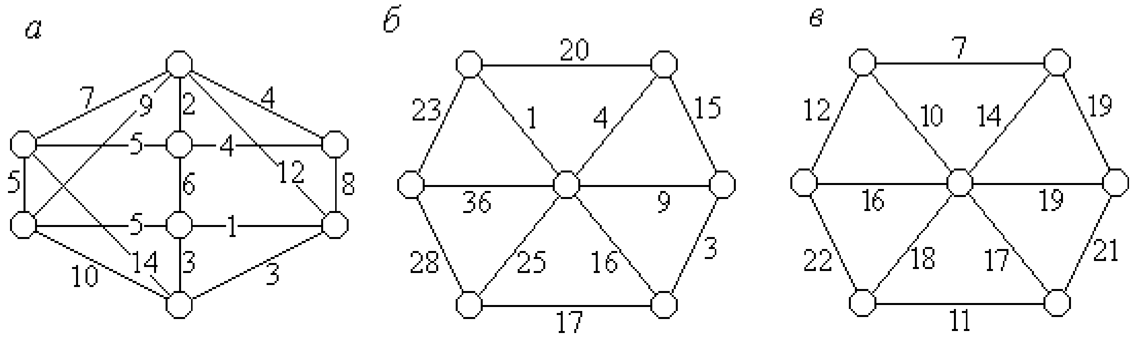


k	D[1]	D[2]	D[3]	D[4]	D[5]
	0	1	╔	╔	3
1	0	1	4	4	-1
2	0	1	4	3	-1
3	0	1	4	3	-1

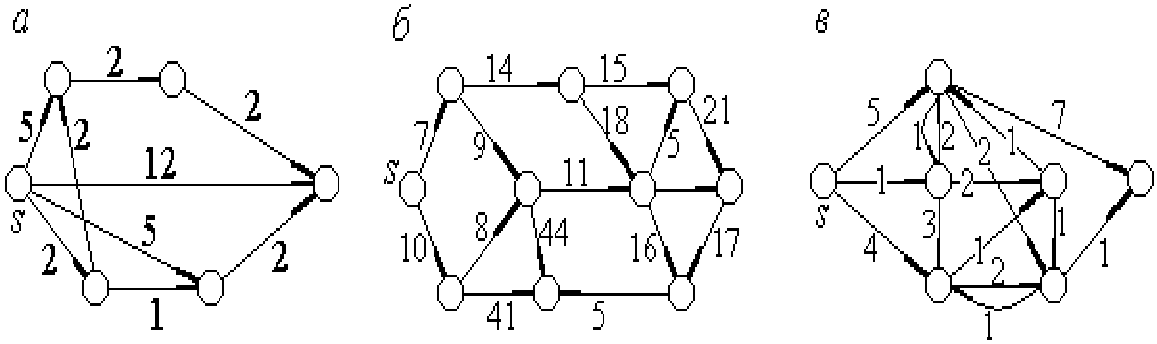
Задачи для самостоятельного решения

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

6 2

7 8 9

0

2

0	4	5					
	0		2	6			
		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: Построение максимального потока на сетях

Цель работы: научить студентов построению максимального потока на сетях

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

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

$$\lambda(a)=\begin{cases}0, \varphi(a)<\beta(a)\\ \infty, \varphi(a)=\beta(a)\end{cases} \quad \lambda(a')=\begin{cases}0, \varphi(a)>\alpha(a)\\ \infty, \varphi(a)=\alpha(a)\end{cases}$$

Поток φ будет максимальным, если в $I(N,\varphi)$ нет путей из V_1 в V_n 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 λ_i 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.
6. Строим простой поток σ_i по цепи C_i.
7. $\varphi_{i+1}=\varphi_i+\sigma_i$ (в общем случае: $\varphi_{i+1}=\varphi_i+t\sigma_i$).
8. $i=i+1$.
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 дополнитльные вершины v_0 и v_{n+1}, а множество дуг A' определяется следующим образом: каждой дуге $a\approx(v,w)$ будет соответствовать 3 дуги:

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

Кроме того добавляем дугу $b'\approx(v_1,v_n)$

Пропускные способности определяются следующим образом:

$\alpha'(a')=\alpha'(a'')=\alpha'(a''')=\alpha'(b')=0.$

$\beta'(a')=\beta(a)-\alpha(a)$

$\beta'(a'')=\beta'(a''')=\alpha(a)$

$\beta''(b')=K$, где K - очень большое целое число.

Тогда сеть $N'(V',A')$ - транспортная.

<u>Определение.</u> Допустимый поток φ' из нового источника v_0 в новый сток v_{n+1} называется насыщенным, если для всех дуг из v_0 $\varphi'(a'')=\beta'(a'')$ (или $\varphi'(a''')=\beta'(a''')$).

<u>Теорема.</u> Если φ' насыщенный поток из V', то поток $\varphi(a)=\varphi'(a')+\alpha(a)$ в сети N $\forall a\in A$ является допустимым из V_1 в V_n, а его величина равняется $\varphi'(b')$.

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

Практическая работа № 17

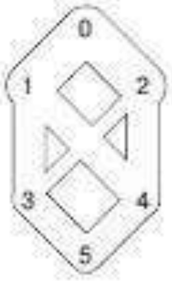
Тема: Алгоритмически неразрешимые проблемы

Цель работы: научить студентов построению максимального потока на сетях

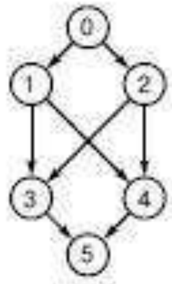
Упражнения

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 ECM, 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 negcyc() 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. сп.	стои-мость	поток
0-1	3	3	0
0-2	3	1	0
1-3	2	1	0
1-4	2	1	0
2-3	1	4	0
2-4	2	2	0
3-5	2	2	0
4-5	2	1	0



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

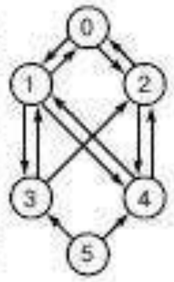


Начальный максимальный поток

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

общая стоимость: 22

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



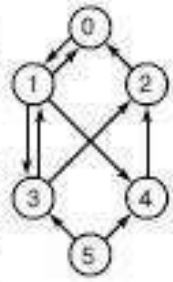
отрицательные циклы: 4-1-0-2-4
3-2-0-1-3
3-2-4-1-3

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

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

общая стоимость: 21

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



отрицательные циклы: 3-2-0-1-3

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

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

общая стоимость: 20

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

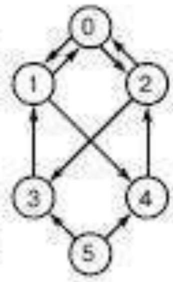


Рис. Остаточные сети (вычеркивание циклов)

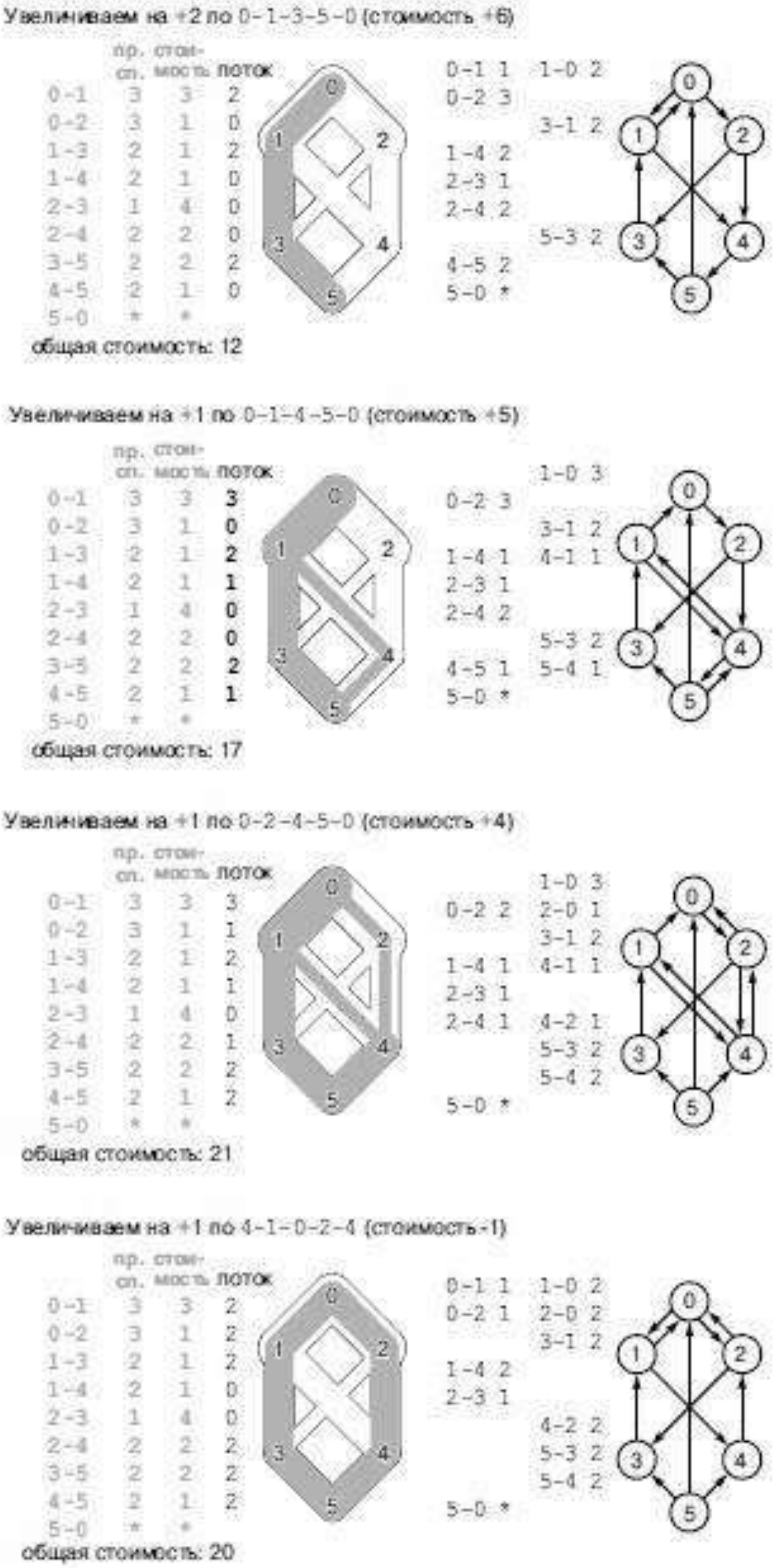


Рис. Вычеркивание циклов без начального максимального потока

Рекомендуемая литература:

Основная литература:

1. Седова Н.А. Дискретная математика [Электронный ресурс] : учебное пособие / Н.А. Седова. — Электрон. текстовые данные. — Саратов: Ай Пи Эр Медиа, 2018. — 67 c. — 978-5-4486-0069-2. — Режим доступа: http://www.iprbookshop.ru/69316.html

2. Хусаинов, А. А. Дискретная математика [Электронный ресурс] : учебное пособие для СПО / А. А. Хусаинов. — Электрон. текстовые данные. — Саратов : Профобразование, 2019. — 77 c. — 978-5-4488-0281-2. — Режим доступа: http://www.iprbookshop.ru/86136.html

3. Рогова Н.В. Дискретная математика [Электронный ресурс] : учебное пособие / Н.В. Рогова. — Электрон. текстовые данные. — Самара: Поволжский государственный университет телекоммуникаций и информатики, 2017. — 143 c. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/75372.html

Дополнительная литература:

1. Бережной В.В. Дискретная математика [Электронный ресурс] : учебное пособие / В.В. Бережной, А.В. Шапошников. — Электрон. текстовые данные. — Ставрополь: Северо-Кавказский федеральный университет, 2016. — 199 c. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/69380.html

2.Математика. Дискретная математика [Электронный ресурс] : учебник / В.Ф. Золотухин [и др.]. — Электрон. текстовые данные. — Ростов-на-Дону: Институт водного транспорта имени Г.Я. Седова – филиал «Государственный морской университет имени адмирала Ф.Ф. Ушакова», 2016. — 129 c. — 2227-8397. — Режим доступа: ЭБС http://www.iprbookshop.ru/57348.html