Министерство образования
Республики Беларусь

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

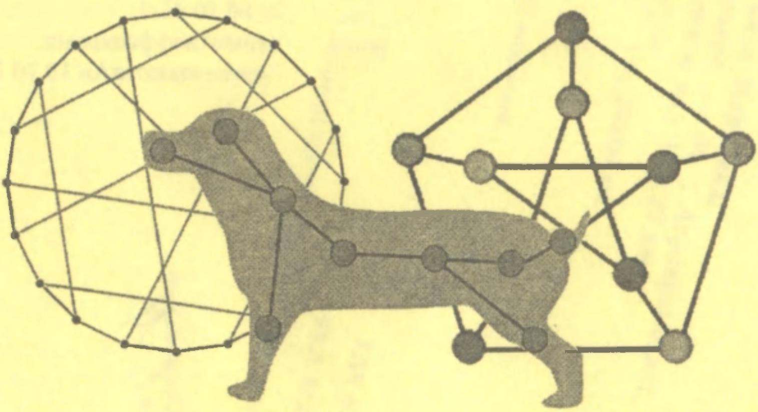
Кафедра «Теоретическая механика»

С.Г. Бохан
С.И. Пармон

ДИСКРЕТНАЯ МАТЕМАТИКА

Методическое пособие

Часть 1



Минск
БНТУ
2010

Министерство образования Республики Беларусь
БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Кафедра «Теоретическая механика»

С.Г. Бохан
С.И. Пармон

ДИСКРЕТНАЯ МАТЕМАТИКА

Методическое пособие по лабораторным работам
для студентов машиностроительного факультета специальностей
1-55 01 01 «Интеллектуальные приборы, машины и производства»,
1-55 01 03 «Компьютерная мехатроника»,
1-36 01 01 «Технология машиностроения»,
1-36 01 04 «Оборудование и технологии
высокоэффективных процессов обработки материалов»,
1-53 01 01 «Автоматизация технологических процессов и производств»

В 2 частях

Часть 1

Минск
БНТУ
2010

УДК 519.1(076.5)(075.8)
~~ББК 22.176я7~~
Б86

Рецензенты:
О.И. Чичко, Е.В. Полынкова

Бохан, С.Г.

Б86 Дискретная математика: методическое пособие по лабораторным работам для студентов машиностроительного факультета специальностей 1-55 01 01 «Интеллектуальные приборы, машины и производства», 1-55 01 03 «Компьютерная мехатроника», 1-36 01 01 «Технология машиностроения», 1-36 01 04 «Оборудование и технологии высокоэффективных процессов обработки материалов», 1-53 01 01 «Автоматизация технологических процессов и производств»: в 2 ч. / С.Г. Бохан, С.И. Пармон. – Минск: БНТУ, 2010. – Ч. 1. – 74 с.

ISBN 978-985-525-384-7 (Ч.1).

В методическом пособии излагается материал по курсу лекций «Дискретная математика», читаемых на дневном и заочном отделениях машиностроительного факультета. Первая часть пособия включает материалы курса, связанные с математической логикой, теорией множеств, комбинаторными алгоритмами и теорией графов.

УДК 519.1(076.5)(075.8)
ББК 22.176я7

ISBN 978-985-525-384-7 (Ч.1)
ISBN 978-985-525-385-4

ВВЕДЕНИЕ

Целью выполнения лабораторных работ является изучение алгоритмов решения типовых технических задач и овладение методами дискретной математики, наиболее приемлемыми для их решения.

В современной иерархии математических наук дискретная математика является промежуточным звеном между рядом дисциплин естественно-научного и технического профиля.

Задачами курса дискретной математики являются ознакомление и освоение основных моделей и методов формализованного представления: теоретико-множественных, логических, графических. Мощным фундаментом современной дискретной математики являются теория множеств, математическая логика и теория графов.

Дискретная математика была и остается одной из наиболее динамичных математических дисциплин. Она изучается почти во всех вузах технического и экономического профиля.

Основное отличие дискретной математики от классической заключается в отсутствии понятий предела и непрерывности, а основными носителями являются, например, целые числа и многочлены, векторы и матрицы, чертежи и рисунки, слова и команды и т.п.

В курсе закрепляются умения оперировать аппаратом теории множеств, в том числе с отношениями и функциями; использовать основные законы алгебры логики для преобразования логических функций, распознавать различные комбинаторные конфигурации и подсчитывать их число, осуществлять операции над графами, строить и программировать алгоритмы для обработки различных графов.

Для изучения дисциплины «Дискретная математика» требуется математическое образование, основанное на ряде математических курсов, таких, как математический анализ, аналитическая геометрия, линейная алгебра и т.д.

На материале этой дисциплины в значительной степени базируется содержание дисциплины «Математическое моделирование физических и технических процессов». Знание дискретной математики необходимо также при изучении таких дисциплин, как «Теоретические основы электротехники», «Исследование операций», «Электроника и микропроцессорная техника» и др.

При изучении дисциплины необходимо обращать внимание студентов на то, что недостаточно знать рассматриваемые алгоритмы дискретной математики, надо уметь разрабатывать реализующие их алгоритмы и компьютерные программы. В связи с этим, полезно на примере нескольких алгоритмов показать студентам, каким образом традиционное изложение алгоритмов, ориентированное на решение задач на графах «вручную», переработать применительно к обработке их матриц смежности или инцидентности, что позволит их запрограммировать.

На сегодняшний день наиболее значимым направлением развития дискретной математики являются информационные технологии. Это объясняется, прежде всего, необходимостью создания и эксплуатации персональных ЭВМ, компьютерных сетей, систем управления, а также автоматизированных средств обработки информации.

Комплекс лабораторных работ предназначен для закрепления материала курса «Дискретная математика» с помощью компьютерных технологий.

Для всех лабораторных работ предусмотрены 14 вариантов индивидуальных заданий (по количеству компьютеров в классе). Эти варианты помещены в электронное издание «Дискретная математика» и находятся в научной библиотеке БНТУ.

По количеству часов весь лекционный материал можно представить в виде следующей диаграммы (рис. 1).

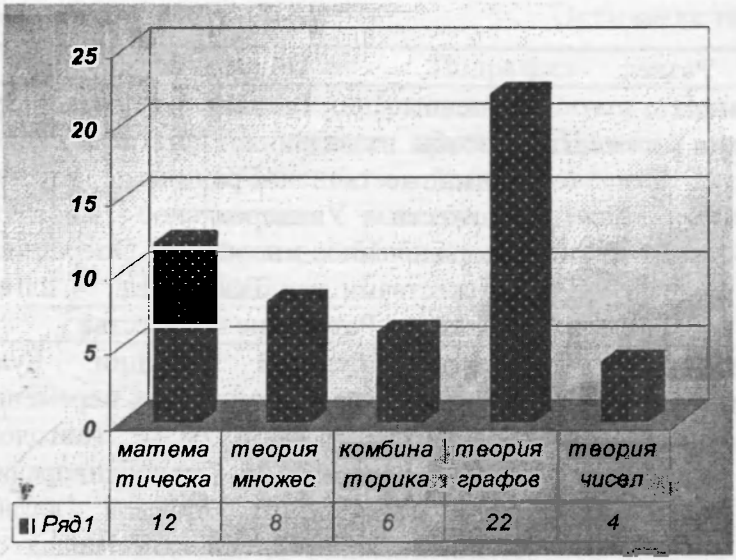


Рис. 1. Диаграмма распределения учебных часов

При чтении этого курса в технических вузах, как правило, выбираются следующие разделы и подразделы дискретной математики (табл. 1).

Таблица 1

Разделы и подразделы учебного курса

Раздел	Подраздел
Математическая логика	Логические операции. Высказывания и высказывательные формы. Элементарные и составные предложения. Конъюнкция и дизъюнкция. Импликация и эквиваленция. Язык логики высказываний. Формулы логики высказываний. Составление таблиц истинности для заданных формул. СДНФ и СКНФ. Кванторы. Переключательные схемы

Продолжение табл. 1

Раздел	Подраздел
Элементы теории множеств	Конечные и бесконечные множества, способы их задания. Понятие и свойства подмножества. Границы числовых множеств. Универсальное множество. Упорядоченное множество. Операции над множествами. Тождества алгебры множеств. Разбиение множества
Элементы теории булевых функций	Понятие булевой функции. Булевы функции нуля, одной и двух переменных. Логические формулы. Тавтологии, противоречивые и непротиворечивые формулы. Законы булевой алгебры. Конъюнктивная и дизъюнктивная нормальные формы
Комбинаторика	Комбинаторные конфигурации. Подстановки. Инверсии. Биномиальные коэффициенты. Разбиения. Принцип включения и исключения
Теория графов	Понятие графа. Неориентированные, смешанные и орграфы. Матрицы смежности и инцидентности графа. Подграф, полный граф, дополнение графа. Степень, степень входа, степень выхода вершины. Цепи, циклы, пути, контуры. Гамильтоновы и эйлеровы цепи, циклы, пути, контуры. Взвешенные графы. Вес, тонкость, ширина графа, пути, цепи. Связность графов. Деревья, корневые деревья, венки, остовы. Двудольный граф. Изоморфизм графов. Транспортная сеть. Разрез транспортной сети. Пропускная способность разреза. Поток на транспортной сети

Окончание табл. 1

Раздел	Подраздел
Теория чисел	Наибольший общий делитель. Наименьшее общее кратное. Простые числа. Сравнение, свойства сравнений. Полная система вычетов. Функция Эйлера. Функция Мебиуса. Формула обращения Мебиуса

ЛАБОРАТОРНЫЕ РАБОТЫ

Лабораторная работа № 1
МАТЕМАТИЧЕСКАЯ ЛОГИКА

Цель работы: ознакомиться с предметом математической логики. Рассмотреть и изучить основные элементы математической логики с помощью электронной таблицы Excel.

Методические указания

Загрузите программу Excel. Перейдите на новый рабочий лист. Дайте ему имя «Логика».

Введите в ячейку A1 формулу: =7>5. Она вернет значение ИСТИНА. Скопируйте содержимое A1 в A2 и исправьте в A2 формулу: =3>5. Эта формула вернет значение ЛОЖЬ. Правые части обеих формул представляют собой высказывания, т.е. утверждения, относительно которых можно заключить, верны они или нет.

Рассмотрим другой пример. Введем в ячейку A4 число 2, а в ячейку B4 формулу =A4>3. Формула возвращает значение ЛОЖЬ. Введем в A4 число 6. Формула возвращает значение ИСТИНА. В B4 записан предикат, т.е. высказывание с переменными (в данном случае переменная одна). В зависимости от значения переменных предикат может принимать значения ИСТИНА и ЛОЖЬ. В этом примере формула как бы дает ответ на вопрос: «Число (или результат вычислений по формуле), хранящееся в ячейке A4, превышает 3?» В зависимости от значения A4 ответ будет ДА (ИСТИНА) или НЕТ (ЛОЖЬ).

Сравнение двух арифметических выражений, содержащих переменные, дает предикат. В формуле =A4>3 ее составные части (A4 и 3) можно считать арифметическими выражениями, только очень простыми. Более сложный пример: =(A4 ∧ 2–1)>(2*A4+1). В этом выражении скобки можно опустить, потому что арифметические операции имеют более высокий приоритет, чем операции сравнения, но скобки придают формуле наглядность.

Высказывание и предикат имеют общее название – логическое выражение. Имеются логические операции, которые позволяют строить сложные логические выражения. Эти операции реализованы в Excel как функции.

Можно провести аналогию с арифметическими операторами: отрицанию соответствует унарный минус, конъюнкции – логическое умножение, дизъюнкции – логическое сложение (табл. 2). На самом деле, в Excel приоритет логических операций не имеет значения, так как они реализованы в виде функций.

Таблица 2

Перечень логических операций

Название	Обозначение	Функция Excel
Отрицание	$\neg$	НЕ
Конъюнкция	$\wedge$ (&)	И
Дизъюнкция	$\vee$	ИЛИ

У логических функций аргументы могут принимать только два значения: ИСТИНА и ЛОЖЬ. Поэтому логические функции можно задать таблицей, где перечислены все возможные значения аргументов и соответствующие им значения функций. Такие таблицы называются таблицами истинности (табл. 3).

Таблица 3

Таблица истинности

x	*y*	*И(x,y)*	*ИЛИ(x,y)*
ЛОЖЬ	ЛОЖЬ	ЛОЖЬ	ЛОЖЬ
ЛОЖЬ	ИСТИНА	ЛОЖЬ	ИСТИНА
ИСТИНА	ЛОЖЬ	ЛОЖЬ	ИСТИНА
ИСТИНА	ИСТИНА	ИСТИНА	ИСТИНА

x	*НЕ(x)*
ЛОЖЬ	ИСТИНА
ИСТИНА	ЛОЖЬ

Выполните свой вариант задания. Вариант задания определяется по номеру системного блока компьютера.

Пример варианта задания

Пример 1

В ячейке А2 (с именем *z*) записано число. Выяснить, принадлежит ли оно отрезку [2, 5].

Решение

Присвоим ячейке А6 имя *z*. Введем в А6 число 3. Сначала сконструируем логическое выражение, решающее задачу: $z \in [2, 5] \Leftrightarrow (z > 2) \wedge (z < 5)$. Для того чтобы *z* принадлежал отрезку [2, 5], нужно, чтобы одновременно были истинны два предиката: $z > 2$ и $z < 5$. В ячейке В6 разместим формулу =И(z>=2;z<=5). В В6 получим значение ИСТИНА.

Пример 2

В ячейке А7 (с именем *X*) записано число. Выяснить, принадлежит ли оно одному из лучей на числовой оси: $(-\infty, 2)$ или $(5, \infty)$.

Решение

Сконструируем логическое выражение, решающее задачу: $z \in (-\infty, 2)$ и $(5, \infty) \Leftrightarrow (z < 2) \vee (z > 5)$, где значок «и» обозначает операцию объединения множеств. Для того чтобы *z* принадлежал хотя бы одному из лучей, нужно, чтобы был истинным хотя бы один из предикатов: $z < 2$ или $z > 5$. В ячейке D6 разместим формулу =ИЛИ(X<2;X>5). А7 содержит число 3, поэтому формула возвращает ЛОЖЬ.

Задачу можно было решить иначе с учетом того обстоятельства, что на рабочем листе есть формула проверки принадлежности числа *z* отрезку [2, 5]. Упомянутые два луча составляют на числовой оси дополнение к этому отрезку.

Введем в ячейку E6 формулу =НЕ(B6). Убедитесь, вводя в ячейку A7 различные числа, что формулы в ячейках D6 и E6 дают идентичные результаты. Мы воспользовались одним из законов Де Моргана: $\neg (a \wedge b) = \neg a \vee \neg b$.

Пример 3

На практике «в чистом виде» логические выражения, как правило, не используются. Логическое выражение служит первым аргументом функции ЕСЛИ: «ЕСЛИ(лог_выражение, значение_если_истина, значение_если_ложь)».

Во втором аргументе записывается выражение, которое будет вычислено, если лог_выражение возвращает значение ИСТИНА, а в третьем аргументе – выражение, вычисляемое, если лог_выражение возвращает ЛОЖЬ. В языках программирования высокого уровня этой функции соответствует оператор «если лог_выражение то действие 1, иначе действие 2».

Пример 4

1. Введем в ячейку B8 формулу, которая возвращает $z + 1$, если $z > 1$, и z в противном случае: =ЕСЛИ(z>1;z+1;z) (в «Мастере функций» ЕСЛИ находится в категории «Логические», так же, как функции И, ИЛИ, НЕ).

2. Если $z > 60$, то в ячейке B9 выводить сообщение «Превышено пороговое значение», в противном случае выводить z: =ЕСЛИ(z>60,"Превышено пороговое значение",z).

3. Если $z \in [10, 25]$, то возвращать z, если $z < 10$, то возвращать 10, если z > 25, то возвращать 25. Сконструируем выражение (одно из возможных): «если $z < 10$ то 10 иначе (если $z < 25$ то z иначе 25)». Запишем формулу в C9: =ЕСЛИ(z < 10; 10; ЕСЛИ(z <= 25;z; 25)).

Задача 1

Дайте ячейкам A20, B20 и C20 имена u, v, w. В самих ячейках содержатся числа. Введите в ячейки A21, A22 и так далее логические формулы, которые возвращают значение ИСТИНА тогда и только тогда, когда:

а) каждое из чисел *u*, *v*, *w* является положительным;

б) хотя бы одно из чисел *u*, *v*, *w* является положительным;

в) только одно из чисел *u*, *v*, *w* является положительным;

г) ни одно из чисел *u*, *v*, *w* не является положительным;

д) хотя бы одно из чисел *u*, *v*, *w* не является положительным.

Задача 2

Торговый агент получает процент от суммы совершенной сделки. Если объем сделки до 3000, то 5 %; если объем до 10 000, то 2 %; если выше 10 000, то 1,5 %. Введите в ячейку A10 текст «Объем сделки», в ячейку A11 – «Размер вознаграждения». В ячейку B10 введите объем сделки, а в B11 – формулу, вычисляющую размер вознаграждения.

Задача 3

В трех ячейках записаны числа. Если все они ненулевые, вернуть 1, в противном случае – 0. Решить задачу с использованием только одной функции ЕСЛИ (без вложений).

Задача 4

Экзаменатор проверяет письменную работу, состоящую из пяти задач. За каждую задачу он проставляет оценку – целое число в диапазоне от 0 до 4. Иногда (в виде исключения) он может поставить нецелое число, например 3,5. Введите в A24:E24 порядковые номера задач (от 1 до 5), в F24 – строку «Сумма». Экзаменатор вводит оценки в диапазон A25:E25. В F25 автоматически должна вычисляться сумма оценок. При переходе к ячейке подсказка не выводится, при неверном вводе выводится предупреждение.

Указание. Перед вызовом меню «Данные/ Проверка» выделите диапазон A25:E25.

Контрольные вопросы

1. Назовите известные Вам логические операции.
2. Отличие высказываний от высказывательных форм.
3. Формулы логики высказываний.
4. Конъюнкция и дизъюнкция.
5. Таблицы истинности.
6. Какие предложения являются составными, а какие элементарными?
7. Логические связки и их использование.

Лабораторная работа № 2
ЛОГИЧЕСКИЕ ОПЕРАЦИИ

Цель работы: изучить логические операции и научиться применять законы логики для решения задач с помощью таблиц Excel.

Методические указания

Загрузите программу Excel. Перейдите на новый рабочий лист. Дайте ему имя «Логические операции».

Логическая операция, соответствующая союзу «если…, то» называется импликацией. Обозначается символом $\rightarrow$.

Логическая операция, соответствующая союзу «тогда и только тогда, когда», называется эквиваленцией. Обозначается символом $\leftrightarrow$.

Преобразование формул

Формулам математической логики свойственны следующие законы:

1. $X \equiv X$ – закон тождества.

2. $X \wedge \overline{X} \equiv$ л – закон противоречия.

3. $X \vee \overline{X} \equiv$ и – закон исключенного третьего.

4. $\overline{\overline{X}} = X$ – закон двойного отрицания.

5. $X \wedge X \equiv X$;

$X \vee X \equiv X$ – законы идемпотентности.

6. $X \wedge Y \equiv Y \wedge X$;

$X \vee Y \equiv Y \vee X$ – законы коммутативности.

7. $(X \wedge Y) \wedge Z \equiv X \wedge (Y \wedge Z)$;

$(X \vee Y) \vee Z \equiv X \vee (Y \vee Z)$ – законы ассоциативности.

8. $\overline{X \wedge Y} \equiv \overline{X} \vee \overline{Y}$;

$\overline{X \vee Y} \equiv \overline{X} \wedge \overline{Y}$ – законы Де Моргана.

Переключательные схемы

Переключательная схема – это устройство из переключателей (контактов) и проводов, связывающих 2 полюса – вход и выход.

Рассмотрим схему с одним контактом (рис. 2).

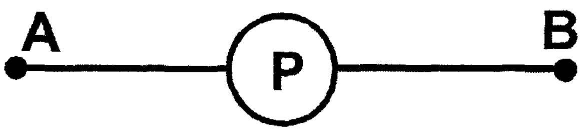


Рис. 2. Схема с одним контактом

Сопоставим контакту *P* переменную *x*, а состоянию контакта «замкнут» – значение ИСТИНА, состоянию контакта «разомкнут» – значение ЛОЖЬ.

Если взять два контакта, соединенных последовательно, то полученная схема будет выражаться формулой вида: $X_1 \wedge X_2$ (рис. 3).

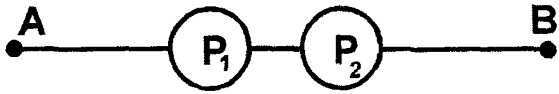


Рис. 3. Схема с последовательными контактами

Если контакты P_1 и P_2 соединены параллельно, то цепь замкнута, когда хотя бы один из контактов замкнут, и разомкнута, когда оба они разомкнуты. Такой схеме соответствует формула $X_1 \vee X_2$ (рис. 4).

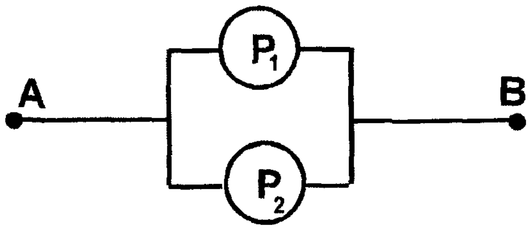


Рис. 4. Схема с параллельными контактами

Установленные соответствия дают возможность описать любую цепь с последовательно или параллельно соединенными контактами формулами логики высказываний. С другой стороны, любую формулу логики высказываний можно смоделировать в виде переключательной схемы.

Пример варианта задания

Вариант 1

1. Определить, эквивалентны ли следующие формулы $(\overline{X} \to Y)Y \vee Z$ и $(\overline{X \wedge Y}) \vee Z$. Составить таблицы истинности.

2. Доказать, что заданная формула является тавтологией: $(P \to (Q \to P))$.

3. Записать следующие высказывания словами и определить истинность их значений: $\forall x((x>2 \wedge x<1) \leftrightarrow x \neq x)$.

4. Построить переключательную схему для формулы $(\overline{X} \to Y)Y \vee Z$.

Контрольные вопросы

1. Импликация и эквиваленция.
2. Тавтология и противоречие.
3. Построение переключательных схем.
4. Построение СДНФ и СКНФ.
5. Упрощение формул.
6. Законы логики.

Лабораторная работа № 3
ТЕОРИЯ МНОЖЕСТВ

Цель работы

1. Ознакомиться с теорией множеств.
2. Рассмотреть основные элементы теории множеств с помощью программирования на различных языках или с использованием различных пакетов прикладных программ (MathCad, Maple, Mathematica и т.д.).

Методические указания

Множество – фундаментальное неопределяемое понятие. Интуитивно под множеством понимают совокупность определенных вполне различаемых объектов, рассматриваемых как единое целое.

Множество четных чисел описывается: $A = \{x|x - \text{четное}\}$.

Множество конечное, если число его элементов конечно, т.е. если существует натуральное число N, являющееся числом элементов множества.

Пустое множество – множество, не содержащее ни одного элемента, обозначается $\varnothing$, например: $\{x \in c | x2 - x + 1 = 0\} = \varnothing$.

Подмножество. Множество X является подмножеством множества $Y = X \in Y$, если любой элемент множества X принадлежит к множеству Y.

Объединение множеств X и Y – множество $X \bigcup Y$, состоящее из всех тех и только тех элементов, которые принадлежат хотя бы одному из множеств X и Y, $X \bigcup Y = \{x | x \in X;$ или $x \in Y\}$. $\bigcup_{i=1}^{n} X_i = X_1 \bigcap \ldots \bigcap X_n$.

Пересечение множеств X и Y – множество $X \bigcap Y$, состоящее из всех тех и только тех элементов, которые принадлежат как множеству X, так и множеству Y, $X \bigcup Y = \{x |$ $x \in X;$ или $x \in Y\}$. $\bigcap_{i=1}^{n} X_i = X_1 \bigcap \ldots \bigcap X_n$.

Разность множеств X и Y – множество $X \backslash Y$, состоящее из тех и только тех элементов, которые принадлежат X и не принадлежат Y, $X \backslash Y = \{x | x \in X;$ и $x \notin Y\}$.

Универсальное множество I – множество, удовлетворяющее условию $X \bigcap I = X$.

Дополнение множества $\overline{X}$ – множество $\overline{X}$, дополнявшее множество X до универсального $\overline{X} = I \setminus X$.

Тождества алгебры множеств:

1. $(X \bigcup Y) \bigcap Z = (X \bigcap Z) \bigcup (Y \bigcap Z)$.
2. $(X \bigcap Y) \bigcup Z = (X \bigcup Z) \bigcap (Y \bigcup Z)$.
3. $X \bigcap Y = Y$, $X \bigcup Y = X$, $Y \in X$.
4. $X \bigcap X = X$, $X \bigcup X = X$.
5. $\overline{X \bigcup Y} = \overline{X} \bigcap \overline{Y}$.
6. $\overline{X \bigcap Y} = \overline{X} \bigcup \overline{Y}$.

Разбиения множеств

Пусть $\chi = \{x1, \ldots, xn\}$ – система множеств, M – множество.

Система χ – разбиение множества M, если выполнено:

1) $\forall X \in \chi[X \in M]$;
2) $\forall X, Y \in \chi[X \neq Y \rightarrow X \cap Y = \varnothing]$;
3) $\bigcup\limits_{X \in \chi} X = M$.

Кортеж – последовательность элементов или совокупность элементов множества, в которой каждый элемент занимает определенное место.

Прямое произведение множеств X и Y – множество $X{\times}Y$, состоящее из всех тех и только тех упорядоченных пар, первая компонента которых принадлежит множеству X , а вторая – множеству Y, $X{\times}Y=\{(x,y)|x \in X, y \in Y\}$.

Пример варианта задания

Задача 1

Пусть $X = \{x \in R|0 \le x \le 1\}$,

$Y = \{y \in R|0 \le y \le 2\}$.

Что представляет собой множества $X \cup Y$; $X \cap Y$; $X \setminus Y$.

Задача 2

Изобразить множества $A = \{(x,y) \in R2 \mid x2 + y2 \le 1\}$,

$B = \{(x,y) \in R2 \mid x2 + (y-1)2 \le 1\}$.

Задача 3

Составить композицию.

Изделие	Рабочий						Станок			Смена	
	1	2	3	4	5	6	1	2	3	1	2
1	*						*			*	
2		*	*					*		*	
3	*			*	*		1				*
4			*			*			*		*

Задача 4

Геометрические множества

$X = \{ 0;1;4;9 \}$, $Y = \{1;4;9;16 \}$,

$Z = \{ 1;2;5;8;9;10;13;16;17;18;20;25\}$.

Найти: $X \cup Y$, $X \cap Y$, $X \cup Z$, $X \cap Z$, $X \cup Z \cup Y$.

Задача 5

Найти прямое декартовое произведение множеств:

$X = \{0,1\}$,

$Y = \{0,1,4,9\}$.

Задача 6*

$A = \{a,b,c\}$, $B = \{1,2,3,4\}$, $P1 \subseteq A \times B$, $P2 \subseteq B2$.

Изобразите $P1$, $P2$ графически. Найдите [$(P1 * P2) - 1$]. Проверьте, является ли отношение $P2$ рефлексивным, антисимметричным, транзитивным?

Контрольные вопросы

1. Способы задания множеств.
2. Парадокс Рассела.
3. Подмножества и надмножества.
4. Диаграммы Эйлера-Венна.
5. Операции над множествами.
6. Алгебраические преобразования множеств.
7. Упорядоченные множества.
8. Свойства отношений.
9. Отношения эквивалентности, порядка, доминирования.
10. Задание соответствий.
11. Обратное соответствие и композиция соответствий.
12. Отображения и их свойства.

Лабораторная работа № 4
КОМБИНАТОРИКА

Цель работы: изучить комбинаторные формулы и научиться решать задачи с применением комбинаторных алгоритмов, а также программировать заданные алгоритмы.

Методические указания

Во многих практических случаях возникает необходимость знать возможные наборы объектов множества, удовлетворяющие определенным условиям. Такие задачи называют комбинаторными, а возможные наборы объектов – выборками.

Комбинаторные задачи оценивают с точки зрения числа выборок, а алгоритм этого выбора – с точки зрения его сложности. При этом различают сложность алгоритма по времени, т.е. числу шагов алгоритма, и по памяти, т.е. объему памяти для выполнения поставленной задачи.

Множества элементов, состоящие из одних и тех же различных элементов и отличающиеся друг от друга только их порядком, называются перестановками этих элементов. Число всевозможных перестановок из n элементов обозначают через P_n, это число равно $n!$:

$$P_n = n!. \tag{1}$$

Замечание 1. Для пустого множества принимается соглашение: пустое множество можно упорядочить только одним способом; по определению полагают $0! = 1$.

Размещениями называют множества, составленные из n различных элементов по m элементов, которые отличаются либо составом элементов, либо их порядком. Число всех возможных размещений определяется формулой

$$A_n^m = n\cdot(n-1)\cdot(n-2)\cdot\ldots\cdot(n-m+1)\,. \qquad (2)$$

Сочетаниями из n различных элементов по m называются множества, содержащие m элементов из числа n заданных, и которые отличаются хотя бы одним элементом. Число сочетаний из n элементов по m обозначают: C_n^m или $\binom{m}{n}$. Это число выражается формулой

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

Замечание 2. По определению полагают $C_n^0 = 1$. Для числа сочетаний справедливы равенства:

$$C_n^m = C_n^{n-m}\,,\; C_{n+1}^{m+1} = C_n^m + C_n^{m+1}\,. \qquad (4)$$

Отметим, что числа перестановок, размещений и сочетаний связаны:

$$C_n^m = \frac{A_n^m}{P_m}\,. \qquad (5)$$

Замечание 3. Выше предполагалось, что все n элементов различны. Если же некоторые элементы повторяются, то в этом случае множества с повторениями вычисляют по другим формулам.

Например, если среди n элементов есть n_1 элементов одного вида, n_2 элементов другого вида и т.д., то число перестановок с повторениями определяется следующей формулой:

$$P_n(n_1, n_2, \ldots, n_k) = \frac{n!}{n_1!\cdot n_2!\cdot\ldots\cdot n_k!}\,, \qquad (6)$$

где $n_1 + n_2 + \ldots + n_k = n$.

Число размещений по m элементов с повторениями из n элементов равно n^m, т.е.

$$(A_n^m)_{\text{повтор}} = n^m. \tag{7}$$

Число сочетаний с повторениями из n элементов по m элементов равно числу сочетаний без повторений из $n+m-1$ элементов по m элементов, т.е.

$$(C_n^m)_{\text{повтор}} = C_{n+m-1}^m. \tag{8}$$

При решении задач комбинаторики используют следующие правила.

Правило суммы. Если некоторый объект A может быть выбран из множества объектов m способами, а другой объект B может быть выбран n способами, то выбрать либо A, либо B можно $m + n$ способами.

Правило произведения. Если объект A можно выбрать из множества объектов m способами и после каждого такого выбора объект B можно выбрать n способами, то пара объектов (A, B) в указанном порядке может быть выбрана $m \cdot n$ способами.

Пример варианта задания

Пример решения

Сколько экзаменационных комиссий, состоящих из 7 членов, можно образовать из 14 преподавателей?

Решение

Загрузите программу Excel. Перейдите на новый рабочий лист. Дайте ему имя «Комбинаторика».

Для решения используем формулу (3):

$$C_n^m = \frac{n!}{m! \cdot (n-m)!}.$$

Очевидно столько, сколько существует семиэлементных подмножеств у четырнадцатиэлементного множества.

$$C_7^{14} = \frac{14!}{7! \cdot (14-7)!} = 3432.$$

Присваиваем значения ячеек в Excel соответственно каждому сомножителю данной формулы, т.е. например присваиваем:

A1:= 7;
B1:=14;
C1:= ФАКТР(B1–A1);
A2:= ФАКТР(A1);
B2:= ФАКТР(B1);
C2:=B2/(C1*A2).

В результате в ячейке C2 появиться значение соответствующее ответу: **3432.**

Задача 1

На 5 одинаковых карточках написаны буквы М, И, Н, С, К. Эти карточки наудачу разложены в ряд. Какова вероятность того, что получится слово МИНСК?

Задача 2

Из десяти билетов выигрышными являются два. Чему равна вероятность того, что среди взятых наудачу пяти билетов один выигрышный?

Контрольные вопросы

1. Что называют перестановками?
2. По какой форме вычисляют число перестановок из *n* различных элементов?

3. Что называют размещениями?

4. По какой формуле вычисляют число размещений из n различных элементов по m элементов?

5. Что называют сочетаниями?

6. По какой формуле вычисляют число сочетаний из n элементов по m элементов?

7. Каким равенством связаны числа перестановок, размещений и сочетаний?

8. По какой формуле вычисляется число перестановок из n элементов, если некоторые элементы повторяются?

9. Какой формулой определяется число размещений по m элементов с повторениями из n элементов?

10. Какой формулой определяется число сочетаний с повторениями из n элементов по m элементов

Лабораторная работа № 5
ПРЕДСТАВЛЕНИЕ ГРАФОВ В ВИДЕ МАТРИЦ

Цель работы: изучить основные понятия теории графов и научиться использовать матрицы графов для решения задач и применять их при программировании.

Методические указания

Представление графов в ЭВМ в виде матриц смежности, инцидентности и весов.

Пусть N – мощность множества X вершин графа G.

Построим матрицу M_{ij}, строки и столбцы которой соответственно равны вершинам графа, а элементы определяются следующим образом:

$$M_{ij} = \begin{cases} 1, \text{ если } \exists U = \{X_i, X_j\}; \\ 0, \text{ если не } \exists U = \{X_i, X_j\}. \end{cases}$$

Такая матрица M называется матрицей смежности (рис. 5).

Примеры

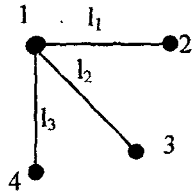


Рис. 5. Неограф

Матрица смежности для графа на рис. 5:

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

Для описания графов используется матрица инцидентности. Каждой вершине ставится в соответствие каждая строка, а ребру или дуге – столбец матрицы (рис. 6).

Для ориентированных графов:

$$R_{ij} = \begin{cases} 1, \text{ если } V_i \text{ – вершина дуги } e_i; \\ -1, \text{ если } V_i \text{ – конец дуги } e_i; \\ 0, \text{ если } V_i \text{ и } e_i \text{ неинцидентны.} \end{cases}$$

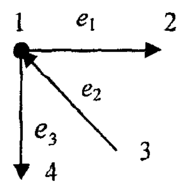


Рис. 6. Орграф

Матрица инцидентности для графа на рис. 6:

$$R_{ij} = \begin{pmatrix} 1 & -1 & 1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}.$$

Матрица весов реберно-взвешенного графа есть квадратная матрица, число строк и столбцов которой равно числу вершин графа (рис. 7). Позиции матрицы весов (x_i, x_j) при $i \neq j$ определяются соотношением

$$P_{ij} = \begin{cases} \infty, \text{ если } i = j; \\ l_{ij}, \text{ если вершина } x_i \text{ смежна вершине } x_j \text{ и вес ребра } l_{ij}. \end{cases}$$

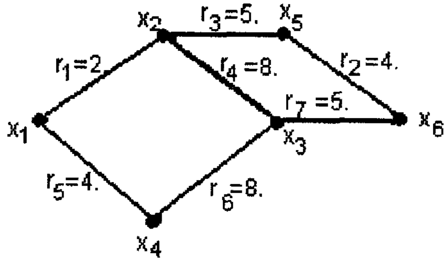


Рис. 7. Реберно-взвешенный граф

Матрица весов для графа на рис. 7.

P	x1	x2	x3	x4	x5	x6
x1	∝	2		4		
x2	2	∝	8		5	
x3		8	∝	8		5
x4	4		8	∝		
x5		5			∝	4
x6			5		4	∝

Пример варианта задания

Построить для заданных графов следующие матрицы:
- смежности для неографа графа *G*1 (рис. 8);
- инцидентности для орграфа *G*2 (рис. 9);
- весов для реберно-взвешенного графа *G*3 (рис. 10).

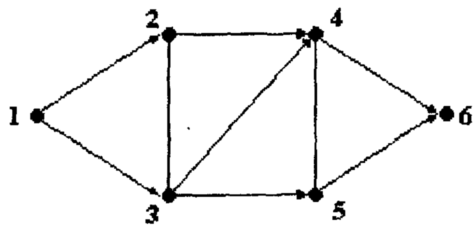


Рис. 8. Граф $G1$

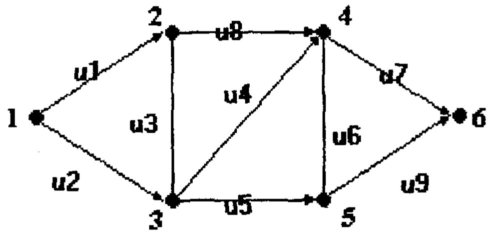


Рис. 9. Граф $G2$

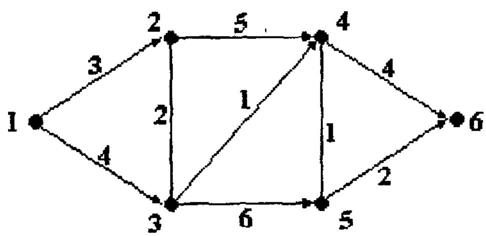


Рис. 10. Граф $G3$

Контрольные вопросы

1. Определение графа, его вершин и ребер.
2. Связность графов.
3. Определение пути в графе и связных компонент графа.
4. Определить цепи, простые цепи, циклы, простые циклы.
5. Неографы и орграфы.
6. Степени вершин в графе.

Лабораторная работа № 6
ПУТИ В ГРАФАХ

Цель работы: научиться находить кратчайшие, длиннейшие, тончайшие и широчайшие пути в графах при решении различных задач.

Методические указания

Путем в орграфах *G* называется такая последовательность дуг (*M*), в которой конец каждой предыдущей совпадает с каждой последующей. Путь, в котором ни какая дуга не встречается дважды, называется простым. Путь, в котором никакая вершина не встречается дважды, называется элементарным.

Контур – это путь, у которого начальная вершина совпадает с конечной вершиной.

Контур, который проходит через все вершины графа, причем только по одному разу, называется **гамильтоновым.**

Контур, который включает все дуги графа, причем только по одному разу, называется **эйлеровым.**

Весом графа называется сумма составляющих его ребер или дуг.

Тонкость пути – вес самого тяжелого, ширина пути – вес самого легкого из его ребер и дуг.

Длиной пути *M* невзвешенного графа называется число *L*, равное числу дуг, составляющих этот путь.

Длина пути взвешенного графа определяется как сумма длин дуг, входящих в этот путь.

Тончайший путь – путь минимальной тонкости.

Широчайший путь – путь максимальной ширины.

Алгоритм Дейкстры

Поиск **кратчайшего** пути в графе от одной вершины к другой.

Шаг 1. Ставим метки:

$d(x_0) = 0$, окрашиваем;

$d(x_i) = +\infty$.

Шаг 2. Для каждой неокрашенной вершины, связанной с последней окрашенной вершиной y дугой (y, x_i), пересчитываем ее метку по формуле $d(x_i) = \min\{d(x_i), d(y)+l(y, x_i)\}$.

Шаг 3. Среди неокрашенных вершин находим вершину x_S с минимальной меткой и окрашиваем ее и ведущую в нее из одной из ранее окрашенных вершин y_X дугу (y_X, x_S), удовлетворяющую условию $d(x_S) = d(y_X) + l(y_X, x_S)$. Если при этом закрашенной оказалась конечная вершина, то задача решена, иначе нужно перейти к шагу 2.

Получится дерево закрашенных дуг с корнем x_0, отображающее кратчайшие пути из этой вершины в другие.

Поиск **длиннейшего** пути в графе от одной вершины к другой.

Шаг 1. Ставим метки:

$d(x_0) = 0$, окрашиваем;

$d(x_i) = -\infty$.

Шаг 2. Для каждой неокрашенной вершины, связанной с последней окрашенной вершиной у дугой (y, x_i), пересчитываем ее метку по формуле $d(x_i) = \max\{d(x_i), d(y) + l(y, x_i)\}$. Если при этом изменяется метка y ранее окрашенной вершины, то краска с нее и инцидентной ей дуги снимается. Если все вершины окрашены, то задача решена, иначе переходим к шагу 3.

Шаг 3. Среди неокрашенных вершин находим вершину x_S и дугу, ведущую в нее из ранее окрашенной вершины, чтобы удовлетворялось условие: $d(x_S) = d(y_r) + l(y_r, x_S)$.

Закрашиваем их. Если при этом закрашенной оказалась конечная вершина, то задача решена, иначе нужно перейти к шагу 2.

Поиск **тончайшего** пути в графе от одной вершины к другой.

Шаг 1. Ставим метки:

$d(x_0) = -\infty$, окрашиваем;

$d(x_i) = +\infty$.

Шаг 2. Для каждой неокрашенной вершины, связанной с последней окрашенной вершиной y дугой (y, x_i), пересчитываемее метку по формуле $d(x_i) = \min\{\max\{d(y), l(y, x_i)\}, d(x_i)\}$.

Шаг 3. Среди неокрашенных вершин находим вершину x_S с максимальной меткой и такую дугу, ведущую в нее из ранее окрашенной вершины, чтобы удовлетворялось условие: $d(x_S) = \max\{d(y_r), l(y_r, x_S)\}$. Значение $d(x_{кон})$ будет искомым значением тонкости тончайшего пути до $x_{кон}$.

Поиск **широчайшего** пути в графе от одной вершины к другой.

Шаг 1. Ставим метки:

$d(x_0) = -\infty$, окрашиваем;

$d(x_i) = +\infty$.

Шаг 2. Для каждой неокрашенной вершины, связанной с последней окрашенной вершиной y дугой (y, x_i), пересчитываем ее метку по формуле $d(x_i) = \max\{\min\{d(y), l(y, x_i)\}, d(x_i)\}$.

Шаг 3. Среди неокрашенных вершин находим вершину (назовем ее x_S) с максимальной меткой и такую дугу, ведущую в нее из ранее окрашенной вершины, чтобы удовлетворялось условие: $d(x_S) = \min\{d(y_r), l(y_r, x_S)\}$. Окрашиваем найденные вершину и дугу. Если конечная вершина окрашена, то задача решена, иначе переходим к шагу 2. Значение $d(x_{кон})$ будет искомым значением ширины широчайшего пути до $x_{кон}$.

Пример

В графе (рис. 11) найти длину кратчайшего пути из x_4 в x_1.

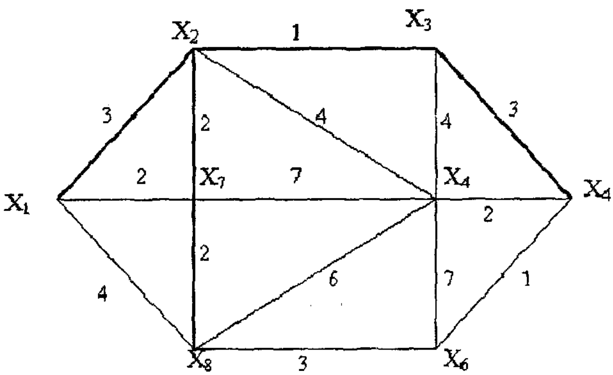


Рис. 11. Граф $G1$

Решение

Составим таблицу пошагово.

		1	2	3	4	5	6	7	8
S	x_1	∞	∞	∞	∞	∞	$7,x_2$	$7,x_2$	$7,x_4$
	x_2	∞	∞	∞	$6,x_5$	$4,x_3$			
	x_3	∞	$3,x_4$	$3,x_4$	$3,x_4$				
t	x_4	$0,x_4$							
	x_5	∞	$2,x_4$	$2,x_4$					
	x_6	∞	$1,x_4$						
	x_7	∞	∞	∞	$9,x_5$	$9,x_5$	$6,x_2$	$6,x_2$	
	x_8	∞	∞	$4,x_6$	$4,x_6$	$4,x_6$	$4,x_6$		

На первом шаге вершине s присваиваем метку $(0, s)$, а остальным (∞). Выбираем из всех меток наименьшую (y, x_4) и делаем ее постоянной. Из этой «постоянной» вершины пытаемся пометить достижимые из нее вершины и присваиваем метку: метка = min(старая метка, метка постоянной вершины + длина ребра). Метки недостижимых вершин сохраняются. У постоянных вершин метки не меняют точки. Процесс продолжаем, пока постоянной меткой не будет помечена вершина t. Метка вершины t равна длине кратчайшего пути. В нашем случае $L = 7$. Маршрут восстанавливаем из второй части меток, которые содержат индекс вершины, из которых они помечены. Тогда кратчайший маршрут: (x_1, x_2, x_3, x_4) или (x_4, x_3, x_2, x_1).

Пример варианта задания

Найти кратчайший и длиннейший пути из вершины $x_н$ в вершину $x_к$ для графа, представленного на рис. 12.

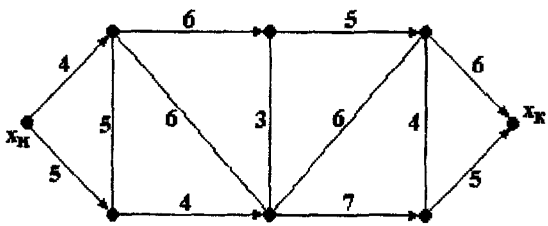


Рис. 12. Реберно-взвешенный граф

Контрольные вопросы

1. Гамильтонов контур.
2. Эйлеров контур.
3. Основные характеристики графа.
4. Тонкость и ширина графа.
5. Поиск кратчайшего пути в графах.

Лабораторная работа № 7
ПОТОКИ В ВЫЧИСЛИТЕЛЬНЫХ СЕТЯХ И ТРАНСПОРТНЫХ СИСТЕМАХ

Цель работы: изучить основные положения теории о транспортных потоках. Научиться решать задачи поиска наибольшего потока с помощью алгоритма Форда–Фалкерсона.

Методические указания

Подобные задачи возникают при выборе маршрутов в вычислительных сетях и транспортных системах. В вычислительных сетях решение подобных задач определяет состав и структуру сетевых протоколов для обслуживания движения пакетов информации. По данным о загрузке каналов, направлении и протяженности вычисляется наиболее оптимальный маршрут прохождения пакета информации.

В транспортных системах решение подобных задач определяет наиболее экономный по стоимости или по времени маршрут движения транспортной единицы.

При работе транспортной системы, когда осуществляется обмен транспортными единицами между узлами сети, возникает задача передачи максимального числа транспортных единиц в заданный отрезок времени.

При передаче энергии в электрических сетях, жидкости в трубопроводных системах возникает задача распределения и передачи максимального объема энергии или вещества в заданный отрезок времени.

Особенностью сети является наличие вершины-истока и вершины-стока, ориентация всех отрезков линий в графе и отсутствие петель и кратных дуг.

Объем информации, энергии или вещества, передаваемый в сети от узла x_i к узлу x_j, называют потоком и обозначают φ_{ij}.

Формулировка задачи

При заданной конфигурации транспортной сети и известной пропускной способности ребер найти наибольшее значение потока, который может пропустить транспортная сеть, а также распределение этого потока по ребрам.

Алгоритм Форда–Фалкерсона

Шаг 1. Всем вершинам транспортной сети присваиваем обозначения $x_0, x_1, x_2, ..., x_n$, где x_1 – обозначение источника, x_n – стока.

Шаг 2. Заполняем таблицу размером $n \times n$ пропускными способностями $d_{ij} = d(x_i, x_j)$ ребер транспортной сети. При этом имеем в виду, что неориентированное ребро может пропускать поток в обоих направлениях. Числа 0, 1, 2, ..., n будем называть номерами столбцов и строк.

Шаг 3. Помечаем столбцы таблицы номерами строк. Нулевой столбец помечаем номером 0. Далее последовательно просматриваем те строки, номера которых совпадают с номерами уже помеченных столбцов. Если при этом просматриваемая строка содержит число, большее нуля, а столбец, в котором расположено это число, еще не помечен, то последний помечаем номером просматриваемой строки. Пометив столбцы, мы тем самым пометили вершины транспортной сети номерами других вершин, из которых в них входят ненасыщенные ребра. При этом, если последний столбец остался непомеченным, то переходим к заключительному шагу 7.

Шаг 4. По пометкам столбцов находим ненасыщенный путь из источника в сток по следующей схеме: $x_n \leftarrow$ вершина, номер которой равен метке столбца

$n \leftarrow$ вершина, номер которой равен метке столбца последующей вершины $\leftarrow \ldots \leftarrow x_0$.

Шаг 5. Помечаем клетки таблицы. Если ребро (x_i, x_j) вошло в найденный на предыдущем шаге путь, то клетку ij помечаем символом «–», а симметричную ей клетку ji – символом «+».

Шаг 6. Строим новую таблицу. Среди чисел, помеченных символом «–», находим минимальное, вычитаем его из клеток, помеченных символом «–», и добавляем в клетки, помеченные символом «+». Остальные клетки оставляем без изменений. Переходим к шагу 3.

Шаг 7. Заключительный. Из таблицы, полученной на шаге 2, вычитаем поклеточно таблицу, полученную на последней итерации на шаге 5. Клетки результирующей таблицы будут содержать пропускные способности соответствующих ребер. Величина искомого максимального потока равна сумме чисел последнего столбца или первой строки.

Пример

Найти максимальный поток на транспортной сети, представленной на рис. 13.

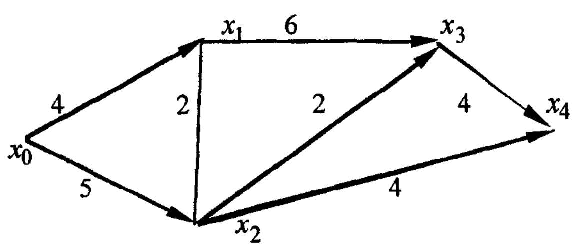


Рис. 13. Ориентированный граф, представляющий транспортную сеть

Решение

Шаг 1. Вершины пронумерованы в нужном порядке.

Шаг 2. Строим таблицу и заполняем ее пропускными способностями ребер.

Шаг 3. Помечаем столбцы таблицы. Последний столбец помечен, значит переходим к шагу 4.

Шаг 4. Метки столбцов определяют следующий путь $x_4 \leftarrow x_2 \leftarrow x_0$.

Шаг 5. Помечаем клетки таблицы. Из помеченных минусом минимальным является число 4.

Шаг 6. Строим новую таблицу, вычитая число 4 из клеток (0,2) и (2,4) и добавляя его в клетки (2,0) и (4,2).

Шаг 3. Помечаем столбцы.

Шаг 4. Выбираем ненасыщенный путь $x_4 \leftarrow x_3 \leftarrow x_1 \leftarrow x_0$.

Шаг 5. Помечаем клетки таблицы.

Шаг 6. Переходим к новой таблице.

Шаг 5. Помечаем клетки таблицы. Так как при этом последний столбец остался не помеченным, то переходим к заключительному шагу.

Шаг 7. Из первой таблицы вычитаем последнюю.

Величина максимального потока равна 8.

Примечание. Клетки таблицы содержат распределение потока по ребрам.

	0	0	0	1	2
	x_0	x_1	x_2	x_3	x_4
x_0	*	4-	1		
x_1	+	*	2	6-	
x_2	4	2	*	2	4-
x_3		+	2	*	4
x_4			4	+	*

	0	0	0	1	3
	x_0	x_1	x_2	x_3	x_4
x_0	*	4	5-		
x_1		*	2	6	
x_2	+	2	*	2	
x_3			2	*	4-
	x_4		+		*

	0	2	0	2	
	x_0	x_1	x_2	x_3	x_4
x_0	*		1		
x_1	4	*	2	2	
x_2	4	2	*	2	
x_3		4	2	*	
x_4			4	4	*

	x_0	x_1	x_2	x_3	x_4
x_0	*	4	4		
x_1	-4	*		4	
x_2	-4		*		4
x_3		-4		*	4
x_4			-4	-4	*

Пример варианта задания

Найти наибольший поток на заданной транспортной сети (рис. 14).

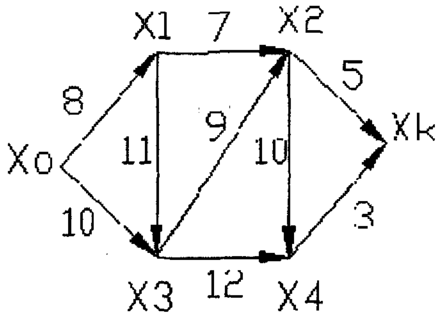


Рисунок 14. Граф, представляющий транспортную сеть

Контрольные вопросы

1. Определение транспортной сети.
2. Основные характеристики транспортной сети.
3. Пропускная способность ребер и дуг.
4. Потоки на транспортной сети.
5. Поиск максимального потока.
6. Разрезы на транспортной сети.

Лабораторная работа № 8
ПОТОКИ МИНИМАЛЬНОЙ СТОИМОСТИ

Цель работы: научиться использовать поиски потоков минимальной стоимости для решения транспортных задач.

Методические указания

Задача о потоке минимальной стоимости формулируется следующим образом: при заданной конфигурации транспортной сети и известной пропускной способности ребер найти поток минимальной стоимости, который может пропустить транспортная сеть. Решить поставленную задачу можно с помощью алгоритма Басакера–Гоуэна.

Алгоритм Басакера–Гоуэна

Шаг 1. В исходной сети величину потока, а также все реберные потоки полагаем равными нулю, т.е. $\varphi_z = 0$, $\varphi_{ij} = 0 \;\; \forall i, j = \overline{1, n}$, где $n = |x|$.

Шаг 2. Находим путь μ минимальной стоимости из источника x_0 в сток z по алгоритму поиска кратчайшего пути в графе, используя в качестве длин ребер дуговые стоимости c_{ij} на первой итерации и модифицированные дуговые стоимости на последующих.

Шаг 3. Определяем пропускную способность θ найденного пути μ: $\theta = \min\limits_{(x_i, x_j) \in \mu} \{d_{ij} - \varphi_{ij}\}$.

Шаг 4. Увеличиваем величину потока φ_z и поток φ_{ij} на ребрах, составляющих путь μ на величину $\theta' = \min\{\theta, B - \varphi_z\}$: $\quad \varphi_z = \varphi_z + \theta'$, $\quad \varphi_{ij} = \varphi_{ij} + \theta'$,

$\forall(x_i,x_j)\in\mu$. Если после этого $\varphi_z = B$, то задача решена и переходим к шагу 7, если $\varphi_z < B$, то переходим к шагу 5.

Шаг 5. Для ребер пути μ вводим симметричные ориентированные ребра (если их еще нет). Пропускные способности симметричных ребер полагаем равными величинам соответствующих реберных потоков, т.е. $d_{ij}^c := \varphi_{ij}$, а реберные стоимости $c_{ij}^c = -c_{ij}$.

Шаг 6. Модифицируем реберные стоимости: $c_{ij}^m = c_{ij}$, если $0 \le \varphi_{ij} \le d_{ij}$, и $c_{ij}^m = \infty$, если $\varphi_{ij} = d_{ij}$. Переходим на шаг 2.

Шаг 7. Определяем стоимость найденного потока. При этом поток ϕ_{ji}^c на симметричном ребре означает, что результирующий поток на исходном ребре (x_i,x_j) равен $\phi_{ij} - \phi_{ji}^c$.

Замечание 1. Если на шаге 2 найдено несколько путей минимальной стоимости, то желательно их обрабатывать одновременно. При этом, если такие пути содержат общие ребра, необходимо сравнивать сумму пропускных способностей путей с пропускными способностями общих ребер. Если эта сумма больше минимальной пропускной способности общих ребер, то суммарный поток необходимо уменьшить до этой минимальной пропускной способности.

Замечание 2. При поиске пути минимальной стоимости необходимо принимать во внимание и симметричные ребра с учетом их ориентации.

Пример решения

Найти поток минимальной стоимости на транспортной сети, приведенной на рис. 15, если величина потока $B = 3$.

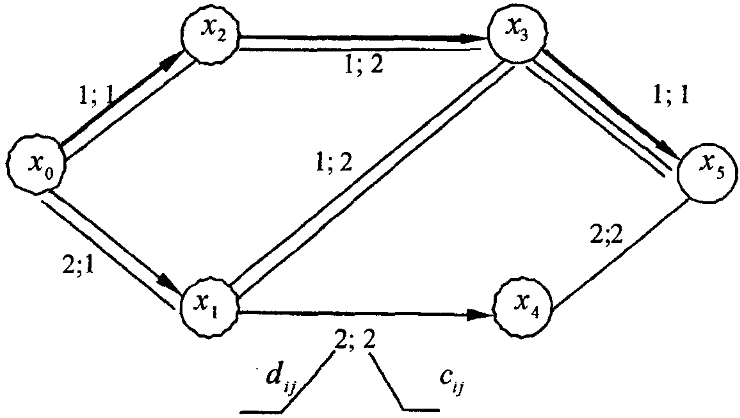


Рис. 15. Транспортная сеть

Шаг 1. $\forall i, \forall j, \varphi_{ij} = 0$, $\varphi_z = 0$.

Шаг 2. Находим путь минимальной стоимости.
$\mu_1 = (x_0, x_1, x_3, x_5)$, $\mu_2 = (x_0, x_2, x_3, x_5)$.

Шаг 3. Находим пропускные способности путей: $\theta_1 = 1$, $\theta_2 = 1$. Так как общее ребро (x_3, x_5) имеет пропускную способность 1, то из двух путей вынуждены выбрать один, например, μ_1.

Шаг 4. Увеличиваем величину потока $\varphi_z := \varphi_z + \theta_1 = 1 < B$ и реберные потоки $\varphi_{ij} := \varphi_{ij} + \theta_1$ на ребрах из μ_1.

Шаг 5. Добавляем симметричные ориентированные ребра, устанавливаем их пропускные способности и реберные стоимости (рис. 16).

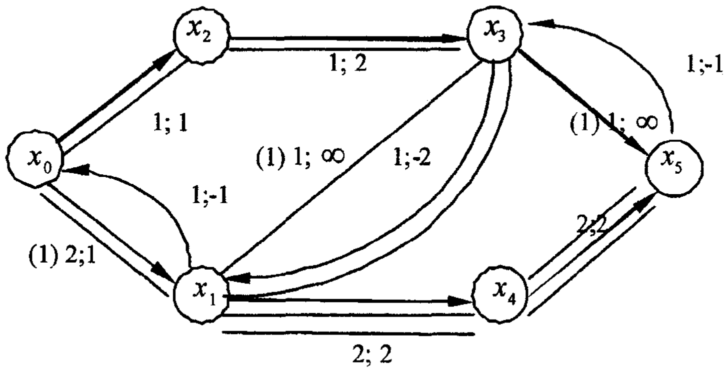


Рис. 16. Модификация ребер на транспортной сети

Шаг 6. Модифицируем реберные стоимости: c_{13}, $c_{35} = \infty$.

Шаг 2. Находим путь минимальной стоимости, при этом принимаем во внимание и симметричные ребра: $\mu_1 = (x_0, x_2, x_3, x_1, x_4, x_5)$, $\mu_2 = (x_0,\ x_1,\ x_4,\ x_5)$.

Шаг 3. $\theta_1 = 1$, $\theta_2 = 1$. Учитываем оба найденных пути, так как общие ребра имеют пропускные способности, равные двум.

Шаг 4. $\theta' = \min\{2, 3-1\} = 2$, значит $\varphi_z := 1 + 2 = 3$, реберные потоки $\varphi_{02}, \varphi_{23}, \varphi_{01}$ увеличиваем на единицу, реберные потоки φ_{14} и φ_{45} увеличиваем на 2. Так как после этого $\varphi_z = 3 = B$, то задача решена, поток минимальной стоимости отмечен на сети. Стоимость найденного потока равна $c_\varphi = 1 \cdot 1 + 1 \times$ $\times\, 2 + 1 \cdot 1 + 2\,(1 + 2 + 2) = 14$.

Замечание 1. Если на шаге 2 найдено несколько путей минимальной стоимости, то желательно их обрабатывать одновременно. При этом, если такие пути содержат общие ребра, необходимо сравнивать сумму пропускных способностей путей с пропускными способностями общих ребер.

Если эта сумма больше минимальной пропускной способности общих ребер, то суммарный поток необходимо уменьшить до этой минимальной пропускной способности.

Замечание 2. При поиске пути минимальной стоимости необходимо принимать во внимание и симметричные ребра с учетом их ориентации.

Пример варианта задания

Найти максимальный поток минимальной стоимости для графа, представленного на рис. 17. Первая цифра – величина потока, вторая – стоимость провоза единицы вещества по данной дуге.

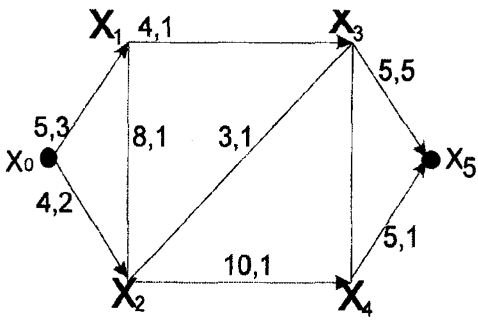


Рис. 17. Транспортная сеть, нагруженная величиной потока и стоимостью

Контрольные вопросы

1. Определение величины потока на заданной транспортной сети.
2. Определение пропускной способности ребер на транспортной сети.
3. Определение минимальной стоимости единицы потока.
4. Понятие минимального разреза на транспортной сети.
5. Постановка задачи о потоке минимальной стоимости.
6. Постановка задачи о максимальном потоке минимальной стоимости.

Лабораторная работа № 9
ОПЕРАЦИИ НАД ГРАФАМИ. РАДИУС И ДИАМЕТР ГРАФА

Цель работы: изучение операций, выполняемых над графами; изучение метрических параметров графов.

Методические указания

Операции над графами

1. Дополнение графа (рис. 18).

Дополнение графа $G(X,E)$ до полного графа

$$\overline{G} = (X, \overline{E} = \{e \in X \times X \mid e \notin E\}).$$

Обратите внимание на стрелки.

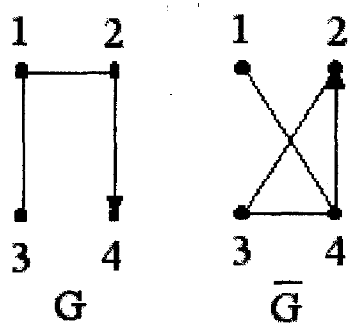


Рис. 18. Дополнение графа

2. Объединение (рис. 19).

$$G_1 \cup G_2 = G(X_1 \cup X_2, E_1 \cup E_2).$$

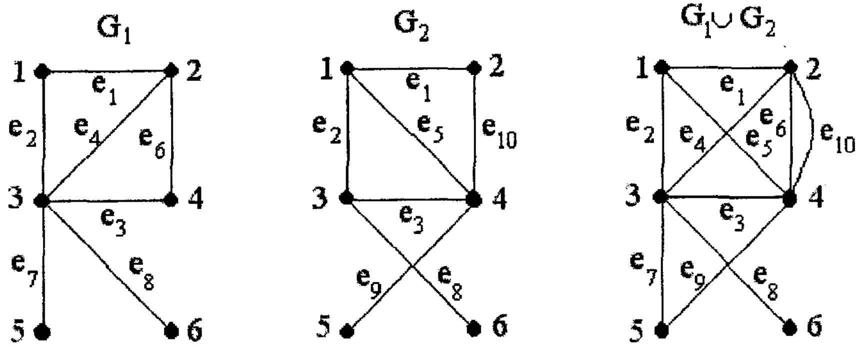


Рис. 19. Объединение графов

Обратите внимание – ребра E_6 и E_{10} – это разные связи вершин 2 и 4 (разные дороги между пунктами 2 и 4).

3. Пересечение (рис. 20) при условии
$X_1 \cap X_2 \neq \varnothing, E_1 \cap E_2 \neq \varnothing$.

$$G_1 \cap G_2 = G(X_1 \cap X_2, E_1 \cap E_2).$$

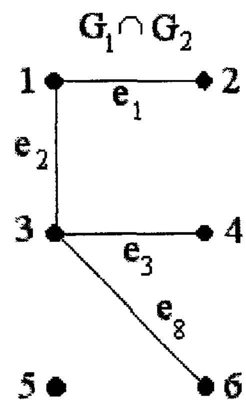


Рис. 20. Пересечение графов

4. Кольцевая сумма (рис. 21).

$$G_1 \oplus G_2 = G(X = X_1 \cup X_2,\ E = E_1 \oplus E_2 = E_1 \backslash E_2 \cup E_2 \backslash E_1).$$

Замечание. Операции 2–4 коммутативные бинарные операции, но могут быть расширены на большее число графов.

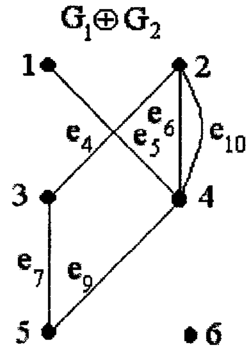


Рис. 21. Кольцевая сумма

5. Произведение (рис. 22).

$$G_1 \times G_2 = G(X, E),$$

где $X = X_1 \times X_2$,

$E = \{(a_1 b_1,\ a_2 b_2) \in E, \forall (\,a_1 = a_2) \wedge (b_1,\ b_2) \in$

$\in E_2 \vee \forall\, (b_1 = b_2) \wedge (a_1,\ a_2) \in E_1\}).$

В произведении графов вершины обозначаются парами ab, где символы a и b – обозначения вершин в G_1 и G_2 соответственно.

Пример

Ребро $(1x,\ 1y) \subset$ E, так как первые символы совпадают (1=1), а в G_2 есть ребро $(x,\ y)$. Аналогично и для других ребер.

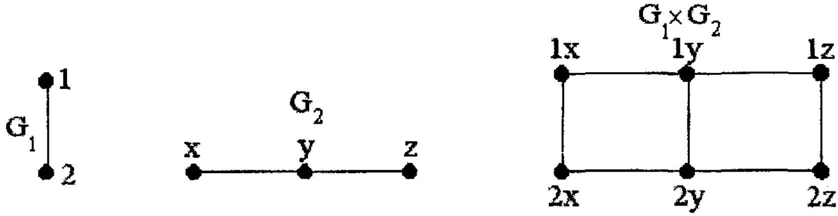


Рис. 22. Пример произведения графов

Произведение $G_1 \times G_2$ означает, что каждая вершина G_1 заменяется на копию $G_a = G_2$, а каждая вершина G_2 заменяется на копию $G_b = G_1$.

6. Композиция (рис. 23).

$$G_1[G_2] = G(X = X_1 \times X_2, E = \{(a_1 b_1, a_2 b_2) \in E, \forall (a_1 = a_2) \wedge (b_1, b_2) \in$$
$$\in E_2 \vee \forall \ (a_1, a_2) \in E_1\}).$$

В композиции графов, как и в произведении графов, вершины обозначаются парами ab, где символы a и b – обозначения вершин в G_1 и G_2 соответственно.

Пример

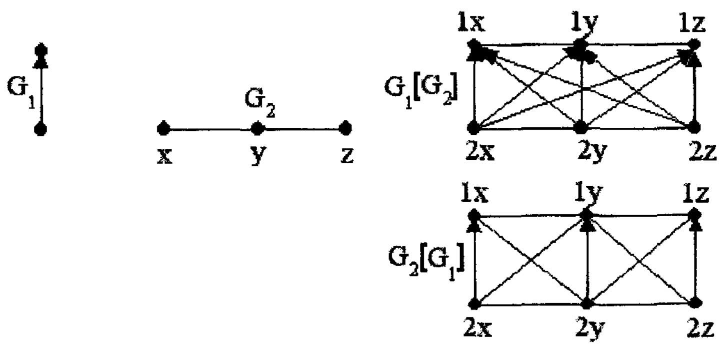


Рис. 23. Композиция графов

Композиция $G_1[G_2]$ означает, что каждая вершина G_1 заменяется на копию $G_a = G_2$, а затем, если $(a_1, a_2) \in E_1$, то между любыми вершинами b_1 из G_{a1} и b_2 из G_{a2} проводится ребро (дуга) (b_1, b_2).

7. Разность.

$$G_1 \backslash G_2 = G(X_1 \backslash X_2, E),$$

где $E = \{[x_1, x_2] | x_1, x_2 \in X_1 \backslash X_2 \wedge [x_1, x_2] \in E_1 \wedge [x_1, x_2] \notin E_2\}$.

8. Удаление вершины.

$$G(X, E) \backslash \{x_i\}.$$

В результате получается подграф, содержащий все ребра, инцидентные множеству $X \backslash \{x_i\}$.

9. Удаление ребра (рис. 24).

$$G \backslash \{e_i\}.$$

Удаляется ребро, но при этом сохраняются концевые вершины, получается частный подграф.

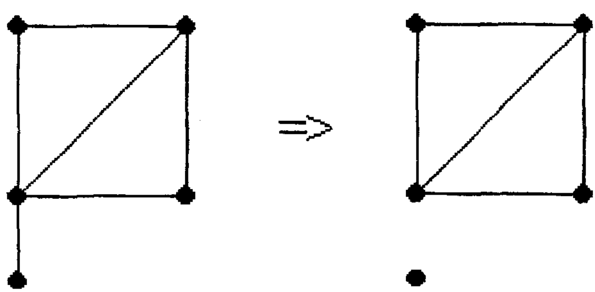


Рис. 24. Удаление ребра графа

10. Добавление вершины.

$$G(X_1, E_1) + \{x\} = G(X_1 \cup \{x\}, E = E_1), \{x\} \notin X_1.$$

11. Добавление ребра.

$$G(X_1, E_1) + \{e\} = G(X, E = E_1 \cup \{e\}), \{e\} \notin E_1.$$

Радиус и диаметр графа

Обозначим через $\mathrm{d}(a, b)$ длину (число ребер или дуг) кратчайшего маршрута между вершинами a и b.

Для $\mathrm{d}(a, b)$ справедливы следующие утверждения:

1) $\mathrm{d}(a, a) = 0$;

2) $\mathrm{d}(a, b) \geq 0$;

3) $\mathrm{d}(a, b) = 0 \Rightarrow a = b$.

Для неорграфа расстояния симметричны:

4) $\mathrm{d}(a, b) = \mathrm{d}(b, a)$;

5) $\mathrm{d}(a, b) \leq \mathrm{d}(a, c) + \mathrm{d}(c, b)$.

Пример (рис. 25)

Рассмотрим вершину x_1:

$\mathrm{d}(x_1, x_2) = \mathrm{d}(x_1, x_4) = \mathrm{d}(x_1, x_5) = 1$,

$\mathrm{d}(x_1, x_3) = 2$,

$\max\limits_{x_i \in X} \mathrm{d}(x_1, x_i) = \mathrm{d}(x_1, x_3) = 2$.

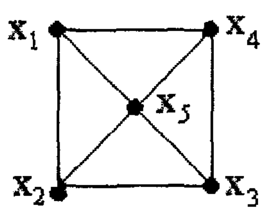


Рис. 25. Пример графа

Для каждой вершины x_i существует максимальный кратчайший маршрут до некоторой вершины x_j, он называется **эксцентриситетом** вершины и обозначается $e(x_i)$.

Максимальный из всех эксцентриситетов графа – это **диаметр** графа:

$$D(G) = \max e(x_i).$$

Пример варианта задания

Рассчитать радиус и диаметр для графа, построенного на основе карты Республики Беларусь (рис. 26) и таблицы расстояний (табл. 4).

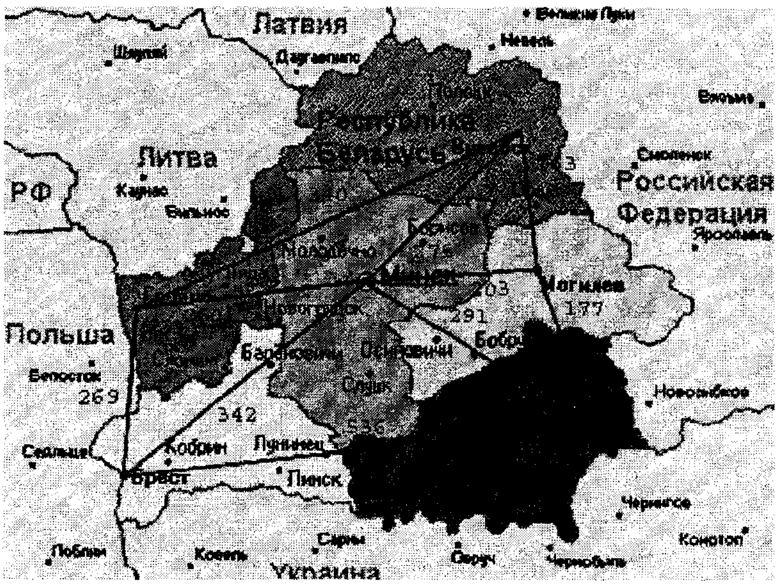


Рис. 26. Карта Республики Беларусь

По карте можно построить граф (рис. 27), а по графу можно построить матрицу смежности, а затем рассчитать нужные параметры (рис. 28).

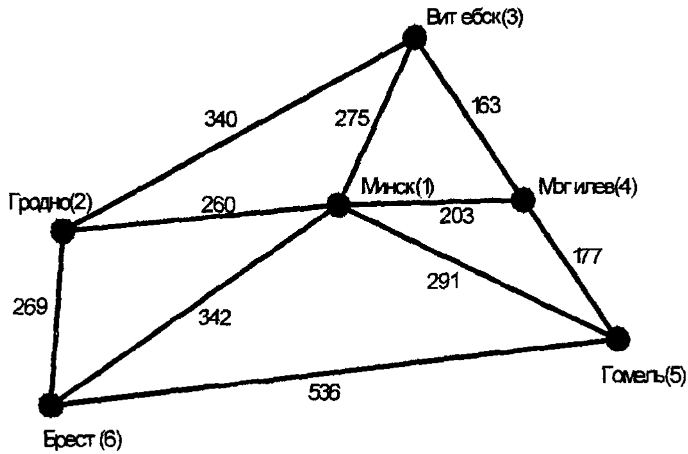


Рис. 27. Граф, построенный по карте Республики Беларусь

Таблица 4

Таблица расстояний

Название города	Минск	Гродно	Витебск	Могилев	Гомель	Брест
Минск	∞	260	275	203	291	342
Гродно	260	∞	340	463	551	269
Витебск	275	340	∞	163	340	617
Могилев	203	463	163	∞	177	545
Гомель	291	551	340	177	∞	536
Брест	342	269	617	545	536	∞

$$\begin{pmatrix} \infty & 260 & 275 & 203 & 291 & 342 \\ 260 & \infty & 340 & 463 & 551 & 269 \\ 275 & 340 & \infty & 163 & 340 & 617 \\ 203 & 463 & 163 & \infty & 177 & 545 \\ 291 & 551 & 340 & 177 & \infty & 536 \\ 342 & 269 & 617 & 536 & 536 & \infty \end{pmatrix}$$

$e(1) = 342;$

$e(2) = 551;$

$e(3) = 617;$ $\quad d(G) = 617;$

$e(4) = 545;$ $\quad r(G) = 342;$

$e(5) = 551;$

$e(6) = 617;$

Рис. 28. Матрица расстояний

Контрольные вопросы

1. Радиус графа.
2. Диаметр графа.
3. Операции над графами.
4. Операции удаления вершины, удаления ребра, разбиения ребра.

Лабораторная работа № 10
ПАРОСОЧЕТАНИЯ МАКСИМАЛЬНОЙ МОЩНОСТИ

Цель работы: приобрести и закрепить навыки построения паросочетания максимальной мощности при помощи алгоритма Эдмансона.

Методические указания

Алгоритм ЭДМАНСОНА построения паросочетания максимальной мощности

Шаг 1. Определение начальных условий. Обозначить рассматриваемый граф через G_0. Выбрать любое паросочетание M_0 графа G_0. Все открытые вершины считать неисследованными. Положить $i: = 0$.

Шаг 2. Исследование открытой вершины. Если в графе G_i имеется только одна неисследованная открытая вершина, то перейти к шагу 6, в противном случае выбрать любую неисследованную открытую для паросочетания M_i вершину v и построить чередующееся дерево с корнем в этой вершине. Если при построении этого дерева найдена увеличивающаяся чередующаяся цепь, то перейти к шагу 3, если обнаружен нечетный цикл, перейти к шагу 4, если алгоритм привел к построению венгерского дерева, то перейти к шагу 5.

Шаг 3. Обработка увеличивающейся цепи. Этот шаг выполняется только после того, как с помощью алгоритма построения чередующегося дерева получена увеличивающаяся чередующаяся цепь. Исключить из паросочетания M_i ребра, входящие в найденную цепь, и включить ранее не входившие в него ребра этой цепи. В результате мощность паросочетания возрастет на единицу, а вершина v становится вершиной паросочетания.

Шаг 4. Обработка нечетного цикла. Этот шаг выполняется только после того, как с помощью алгоритма построения чередующегося дерева получен нечетный цикл. Положить $i := i + 1$. Найденный нечетный цикл обозначить G_i. Построить новый граф G_i, в котором нечетный цикл G_i заменен одной вершиной ai, а все ребра, инцидентные вершинам цикла G_i графа $G_{i\text{-}1}$, становятся инцидентными вершине a_i графа G_i. В остальном графы G_i и $G_{i\text{-}1}$ совпадают. Переходим к шагу 2, выбирая для исследования в качестве открытой вершины отображение вершины v в графе G_i.

Шаг 5. Обработка венгерского дерева. Этот шаг выполняется только после того, как с помощью алгоритма построения чередующегося дерева построено венгерское дерево. Считать вершину v (корень дерева) просмотренной, вернуться к шагу 2.

Шаг 6. Восстановление стянутых в вершину нечетных циклов. Этот шаг выполняется только после того, как все открытые вершины (кроме одной) графа G_i просмотрены или стали инцидентными ребрам паросочетания. При этом в результате повторения шага 4 получена последовательность графов G_1,

$G_2,\ldots,G_t$ и последовательность фиктивных вершин a_1, a_2, ..., a_t. Для каждого графа G_i, $i = 1$, t было получено паросочетание M_i, $i = 1$, t.

Обозначим через M^*i паросочетание максимальной мощности для графа G_i.

Для графа G_t паросочетанием максимальной мощности является Mt, т.е. M^*t – Mt. Для графа $G_{t\text{-}1}$ паросочетание максимальной мощности M^*t–1 строим следующим образом:

а) если фиктивная вершина at не является вершиной паросочетания M^*t, то M^*t–1 включает все ребра паросочетания M^*t плюс любое сочетание несмежных между собой ребер из G_t с учетом условия, чтобы открытой осталась лишь одна вершина этого цикла;

б) если фиктивная вершина яг является вершиной паросочетания M^*t, то M^*t–1 включает все ребра паросочетания M^*t плюс любое сочетание несмежных между собой ребер из Gt, которые инцидентны всем открытым относительно M^*t вершинам этого цикла.

Паросочетания максимальной мощности M^*t–2, M^*t–3, M^*0 строятся последовательно, аналогично построению паросочетания M^*t–1.

Пример решения

Найти паросочетания максимальной мощности для графа, представленного на рис. 29.

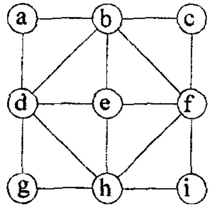


Рис. 29. Неограф G_0

Шаг 1. Граф обозначим G_0, выбираем паросочетание M_0 (ребра этого паросочетания на рис. 29 обозначены буквой M), $i := 0$.

Шаг 2. Для исследования выбираем открытую вершину a. Строим чередующееся дерево: помечаем a как внешнюю; окрашиваем в оранжевый цвет ребра (a, d) и (d, b). Вершину d помечаем как внутреннюю, b – как внешнюю. Окрашиваем в оранжевый цвет ребро (a, b), в результате чего получаем нечетный оранжевый цикл $\{(a, d), (d, b), (a, b)\}$. Переходим к шагу 4.

Шаг 4. $i := i + 1$. Строим граф G_1 (рис. 30). Переходим к шагу 2.

Шаг 2. Для исследования выбираем вершину a_1 как отображение в графе G_1 рассматриваемой на предыдущем шаге вершины а графа G_0. В соответствии с алгоритмом построения чередующегося дерева помечаем a_1 как внешнюю, выбираем ребро (a_1, c), c – открытая непомеченная, следовательно, найдена увеличивающаяся цепь из одного ребра (a_1, c). Переходим к шагу 3.

Шаг 3. К паросочетанию M_1 графа G_1 добавляем ребро (a_1, c). Переходим к шагу 2.

Шаг 2. Для просмотра выбираем вершину g и строим чередующееся дерево. Помечаем g как внешнюю, окрашиваем в оранжевый цвет ребра (g, h) и (h, f). Вершину h помечаем как внутреннюю, f – как внешнюю. Выбираем и окрашиваем в оранжевый цвет ребро (f, i). В результате найдена чередующаяся цепь $\{(g, h), (h, f), (f, i)\}$. Переходим к шагу 3.

Шаг 3. Из паросочетания исключаем ребро (h, f) и включаем в него ребра (g, h)и (f, i). Переходим к шагу 2.

Шаг 2. Открытой является единственная вершина e. Переходим к шагу 6.

Шаг 6. Паросочетание максимальной мощности M^*1 для графа G_1 показано на рис. 30, *д*. Преобразовываем вершину a_1 в исходный для нее нечетный цикл. Из трех вершин a, b и d вершина b является вершиной паросочетания. Следовательно, в искомое паросочетание включаем ребро (a, d).

Задача решена.

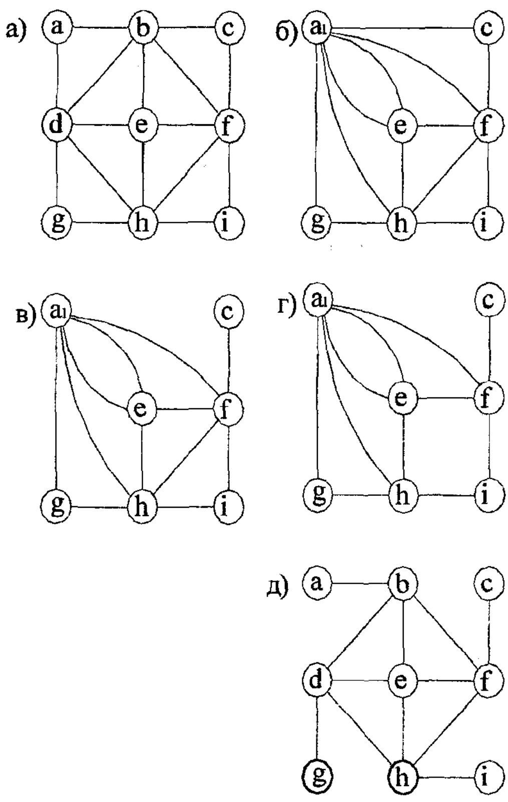


Рис. 30. Графическое решение задачи. Граф G_1

Пример варианта задания

Найти паросочетание максимальной мощности с помощью алгоритма Эдмонса.

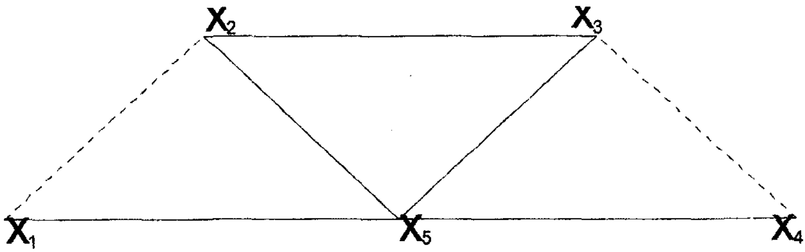


Рис. 31. Паросочетание

Примечание. Паросочетание *M* – выделено штрихпунктирной линией.

Контрольные вопросы

1. Определение паросочетаний в графе и их разновидностей.
2. Определение двудольного графа.
3. Алгоритм выбора наибольшего паросочетания в двудольном графе.

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А

Индивидуальное задание

1. Составить три множества из букв «Фамилии Имени Отчества» студента.
2. Представить полученные множества в виде кругов Эйлера.
3. Представить буквы множеств «Фамилия Имя Отчество» в двоичной системе.
4. Представить диаграмму Вена.
5. Построить СНДФ из букв множеств «Фамилия Имя Отчество».
6. Определить классы толерантности.
7. Представить карту и матрицу толерантности.
8. Построить граф отношения.

Пример решения

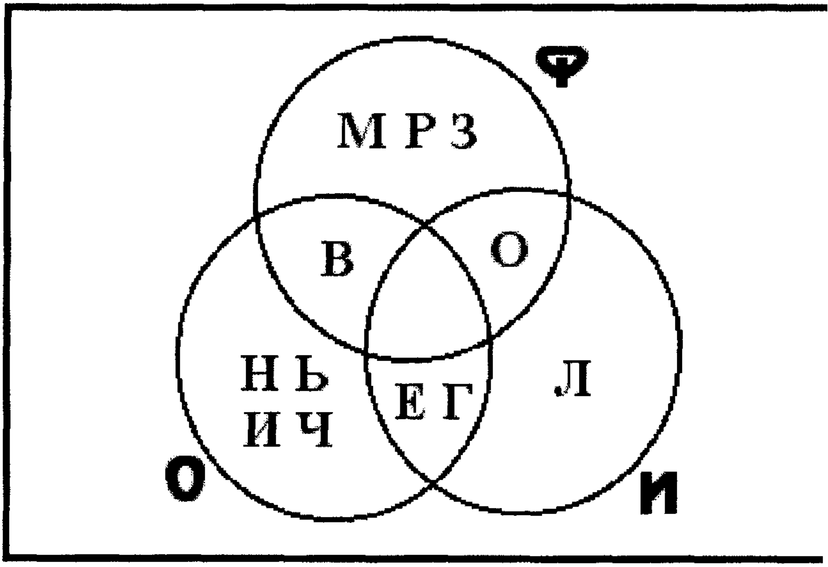
Составить множества из букв «Фамилии Имени Отчества».

МОРОЗОВ ОЛЕГ ЕВГЕНЬЕВИЧ

Ф = { **М, О, Р, З, В** };

И = { **О, Л, Е, Г** };

О = { **Е, В, Г, Н, Ь, И, Ч** };



Буквы алфавита в двоичной системе

А	00001	Л	01011	Ц	10110
Б	00010	М	01100	Ч	10111
В	00011	Н	01101	Ш	11000
Г	00100	О	01110	Щ	11001
Д	00101	П	01111	Ъ	11010
Е	00110	Р	10000	Ы	11011
Ж	00111	С	10001	Ь	11100
З	01000	Т	10010	Э	11101
И	01001	У	10011	Ю	11110
К	01010	Ф	10100	Я	11111
		Х	10101		

Буквы ФИО в двоичной системе

	М	О	Р	З	В	О	Л	Е	Г	Е	В	Г	Н	Ь	И	Ч	
	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	**3**
*F*1	1	1	0	1	0	1	1	0	0	0	0	0	1	1	1	0	**8**
*F*2	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	**10**
*F*3	0	1	0	0	1	1	1	1	0	1	1	0	0	0	0	1	**8**
	0	0	0	0	1	0	1	0	0	0	1	0	1	0	1	1	**6**

СНДФ из букв ФИО

	*x*1 *x*2 *x*3 *x*4	*F*1	*F*2	*F*3
0.	0 0 0 0	1	1	0
1.	0 0 0 1	1	1	1
2.	0 0 1 0	0	0	0
3.	0 0 1 1	1	0	0
4.	0 1 0 0	0	0	1
5.	0 1 0 1	1	1	1
6.	0 1 1 0	1	0	1
7.	0 1 1 1	0	1	1
8.	1 0 0 0	0	1	0
9.	1 0 0 1	0	1	1
10.	1 0 1 0	0	0	1
11.	1 0 1 1	0	1	0
12.	1 1 0 0	1	1	0
13.	1 1 0 1	1	1	0
14.	1 1 1 0	1	0	0
15.	1 1 1 1	0	1	1

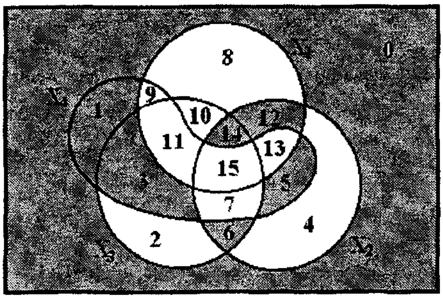
$\boldsymbol{F1} = x1x2x3x4 \cup x1x2x3x4 \cup x1x2x3x4 \cup x1x2x3x4 \cup$
$\cup\ x1x2x3x4 \cup x1x2x3x4 \cup x1x2x3x4 \cup x1x2x3x4$
$\boldsymbol{F2} = x1x2x3x4 \cup x1x2x3x4 \cup x1x2x3x4 \cup x1x2x3x4 \cup$

$\cup\ x1x2x3x4 \cup x1x2x3x4 \cup x1x2x3x4 \cup x1x2x3x4 \cup$
$\cup\ x1x2x3x4 \cup x1x2x3x4$

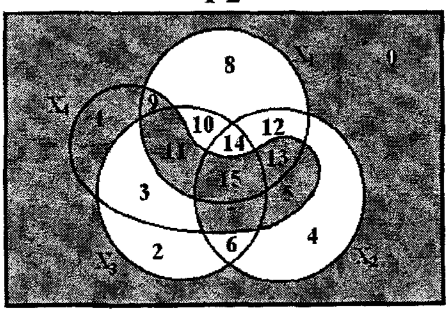
$\boldsymbol{F3} = x1x2x3x4 \cup x1x2x3x4 \cup x1x2x3x4 \cup x1x2x3x4 \cup$
$\cup\ x1x2x3x4 \cup x1x2x3x4 \cup x1x2x3x4 \cup x1x2x3x4$

Диаграммы Вена

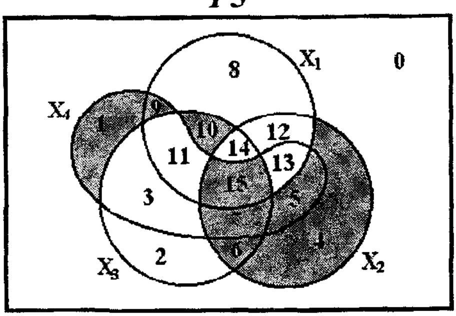
***F*1**



***F*2**



***F*3**



Анализ F на толерантность и эквивалентность

	$x1$	$x2$	$x3$	$x4$	$x5$	$x6$	$x7$	$x8$	$x9$	$x10$	$x11$	$x12$	$x13$	$x14$	$x15$
$\alpha 1$	1	1	0	1	0	1	1	0	0	0	0	0	1	1	1
$\alpha 2$	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0
$\alpha 3$	0	1	0	0	1	1	1	1	0	1	1	0	0	0	0
	6	7	$\varnothing$	4	1	7	5	3	2	3	1	2	6	6	4

$H1 = \{ \alpha 1, \alpha 2 \}$

$H2 = \{ \alpha 1, \alpha 2, \alpha 3 \}$

$H3 = \{ \varnothing \}$

$H4 = \{ \alpha 1 \}$

$H5 = \{ \alpha 3 \}$

$H6 = \{ \alpha 1, \alpha 2, \alpha 3 \}$

$H7 = \{ \alpha 1, \alpha 3 \}$

$H8 = \{ \alpha 2, \alpha 3 \}$

$H9 = \{ \alpha 2 \}$

$H10 = \{ \alpha 2, \alpha 3 \}$

$H11 = \{ \alpha 3 \}$

$H12 = \{ \alpha 2 \}$

$H13 = \{ \alpha 1, \alpha 2 \}$

$H14 = \{ \alpha 1, \alpha 2 \}$

$H15 = \{ \alpha 1 \}$

$L1 = \{ x1, x2, x4, x6, x7, x13, x14, x15 \}$
$L2 = \{ x1, x2, x8, x9, x10, x12, x13, x14 \}$
$L3 = \{ x2, x5, x6, x7, x8, x10, x11 \}$

Классы эквивалентности

$M1 = \{ x1, x13, x14 \}$

$M2 = \{ x2, x6 \}$

$M3 = \{ x4, x15 \}$

$M4 = \{ x5, x11 \}$

$M5 = \{ x7 \}$

$M6 = \{ x8, x10 \}$

$M7 = \{ x9, x12 \}$

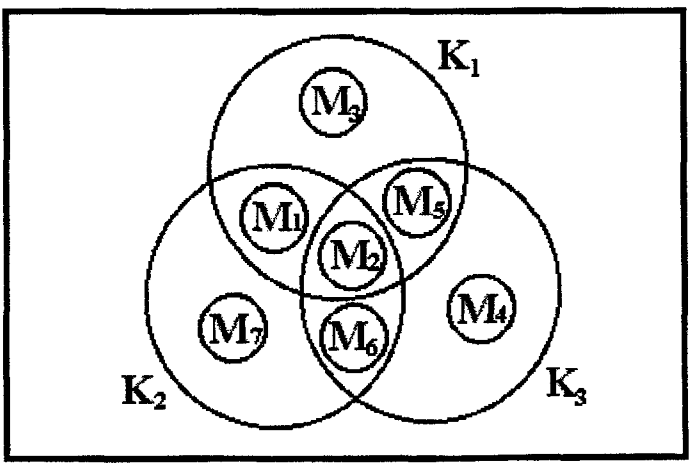
Классы толерантности

$K1 = M1 \cup M2 \cup M3 \cup M5$
$K2 = M1 \cup M2 \cup M6 \cup M7$
$K3 = M2 \cup M4 \cup M5 \cup M6$

Матрица эквивалентности

	x_1	x_{13}	x_{14}	x_2	x_6	x_4	x_{15}	x_5	x_{11}	x_7	x_8	x_{10}	x_9	x_{12}	
x_1	1	1	1												
x_{13}	1	1	1												M_1
x_{14}	1	1	1												
x_2				1	1										M_2
x_6				1	1										
x_4						1	1								M_3
x_{15}						1	1								
x_5								1	1						M_4
x_{11}								1	1						
x_7										1					M_5
x_8											1	1			M_6
x_{10}											1	1			
x_9													1	1	M_7
x_{12}													1	1	

Карта отношения толерантности



Граф отношения толерантности

$x1$ •		
$x2$ •		
$x3$ •		• $\alpha1$
$x4$ •		
$x5$ •		
$x6$ •		
$x7$ •		
$x8$ •		• $\alpha2$
$x9$ •		
$x10$ •		
$x11$ •		
$x12$ •		
$x13$ •		• $\alpha3$
$x14$ •		
$x15$ •		

ПРИЛОЖЕНИЕ Б

Основные определения теории графов

Часть 1. Для всех специальностей

1. **Теория графов** – математический язык для формализованного определения понятий, связанных с анализом и синтезом структур, систем и процессов.

2. **Граф** – пара $G = (V, E)$, где V – непустое множество, задаваемое вершинами, а E – множество пар $e_i = (v_{i1}, v_{i2})$, которые задают ребра. V называется множеством вершин, а E – множеством ребер.

3. **Вершина** графа – элемент множества вершин графа.

4. **Ребро** графа – элемент множества ребер графа.

5. **Дуга** – ориентированное ребро.

6. **Число вершин** – мощность множества вершин $p = |V|$.

7. **Число ребер** – мощность множества ребер $q = |E|$

8. **Смежными** называются две **вершины**, если существует соединяющее их ребро.

9. **Смежными** называются **ребра**, если они опираются на общую вершину.

10. **Инцидентными** называются вершина графа v и некоторое его ребро e, если $e = (v, w)$ или $e = (w, v)$, где w – некоторая вершина графа.

11. **Петля** – ребро, инцидентное одной вершине (единственной). Может быть несколько петель.

12. **Орграф** (ориентированный граф) – граф, ребра которого имеют некоторое направление.

13. **Неограф** (неориентированный граф) – граф, ребра которого не имеют направления (двусторонние).

14. **Мультиграф** – граф, содержащий мультиребра или петли.

15. **Висячее ребро** – ребро, инцидентное висячей вершине.

16. **Мультиребро** – множество ребер, инцидентных одной и той же паре вершин (u, v).

17. **Степень вершины** – число инцидентных ей ребер (обозначается $\deg(v)$).

18. **Вес вершины** – любое число (действительное, целое или рациональное), которое устанавливается в соответствие данной вершине по каким-либо логическим соображениям.

19. **Вес ребра** – любое число (действительное, целое или рациональное), которое устанавливается в соответствие данной вершине по каким-либо логическим соображениям.

20. **Регулярным** называется граф, если степени всех его вершин одинаковы.

21. **Цепь** в графе $G = \{V, E\}$ – последовательность вершин $v_0, v_1, ...v_n$ – такая, что $n > 0$ и v_i, v_j соединены ребром. Если вершины, входящие в цепь, различны, то цепь – **простая**, иначе – **составная**.

22. **Цикл** – замкнутая цепь.

23. **Обход графа** – цикл, проходящий через все вершины графа по одному разу.

24. **Связный граф** – граф, в котором из любой вершины можно найти цепь в любую другую вершину. Несвязный граф распадается на компоненты связности (максимальные связные подграфы).

25. **Корень** – обычно так называют специально выделенную по тем или иным причинам вершину дерева.

26. **Обыкновенный граф** – граф, не содержащий мультиребер и петель.

27. **Полный граф** – это обыкновенный граф $G = (V, E)$, в котором любая пара вершин смежна. Обозначается K_p, где $p = |V|$, при этом $|E| = p(p-1)/2$.

28. **Ациклический** граф (лес) – обыкновенный граф без циклов.

29. **Дерево** – связный ациклический граф.

30. **Полный двудольный граф** – двудольный граф $G = (A_1, A_2, E)$, у которого любая пара вершин $x \in A_1$ и $y \in A_2$ смежна. Обозначается $K_{m,n}$, где $m = |A_1|$, $n = |A_2|$, $|E| = m \times n$.

31. **Граф-звезда** – полный двудольный граф $K_{1,n}$.

32. **Эйлеров граф** – связный граф, не содержащий вершин нечетной степени.

33. **Матрица расстояний (кратчайших цепей) графа** G – квадратная матрица $D = \{d_{ij}\}$, в которой элемент $d_{ij} = r(x_i, x_j)$, т.е. численно равен расстоянию от вершины x_i до вершины x_j в графе G.

34. **Изоморфизм графов.** Два графа $G = (X, U)$ и $L = (X', U')$ изоморфны, если между парами множеств их вершин, ребер и дуг существуют взаимно однозначные соответствия, сохраняющие смежность и ориентацию для дуг.

35. **Остов графа (остовное дерево, покрывающее дерево, каркас)** – суграф графа G, не содержащий циклов и имеющий столько же компонент связности, что и G.

36. **Сильная компонента орграфа** G – максимальный подграф орграфа G, в котором любая пара вершин сильно связна.

37. **Маршрут или путь** – чередующаяся последовательность $x_1, u_1, x_2, u_2, \ldots, x_k$ вершин x_i и ребер u_i, обладающая тем свойством, что любая пара соседних элементов инцидентна.

38. **Цепь** – маршрут, в котором все ребра различны.

39. **Простая цепь** – цепь, в которой все вершины различны.

40. **Цикл** – цепь, у которой начальная и конечная вершины совпадают.

41. **Простой цикл** – простая цепь, у которой концевые вершины совпадают.

42. **Ориентированная цепь** – цепь орграфа G, в которой ориентация дуг (ребер) совпадает.

43. **Контур** – ориентированная цепь, у которой начальная и конечная вершины совпадают.

44. **Эйлерова цепь** – цепь, содержащая все ребра графа в точности один раз.

45. **Эйлеров цикл** – эйлерова цепь, у которой начальная и конечная вершины совпадают.

46. **Гамильтонова цепь** – простая цепь, которая содержит все вершины графа.

47. **Гамильтонов цикл** – гамильтонова цепь, у которой начальная и конечная вершины совпадают.

48. **Взвешенный граф** – граф, вершинам и/или (обычно) ребрам которого поставлены в соответствие целые или вещественные числа (веса).

49. **Вес (длина)** – значение числа, поставленного в соответствие взвешенному элементу.

50. **Планарный граф** – граф, для которого можно построить укладку на плоскости без пересечений ребер.

Часть 2. Для специальностей 1-55 01 01 «Интеллектуальные системы» и 1-55 01 03 «Компьютерная мехатроника»

51. **Эксцентриситет** вершины – максимальное расстояние от v до других вершин графа.

52. **Диаметр** графа – максимальный эксцентриситет его вершин.

53. **Радиус графа** – наименьший из эксцентриситетов его вершин.

54. **Центральная вершина** – вершина, эксцентриситет которой равен радиусу.

55. **Периферийная вершина** – вершина, эксцентриситет которой равен диаметру.

56. **Цикломатическое число графа** $\eta(G)$ – наименьшее число ребер, удаление которых оставляет граф без циклов, образуя остов (каркас) графа.

57. **Удаление вершины**. Из графа удаляется вершина вместе с инцидентными ребрами.

58. **Добавление вершины.** К заданному множеству вершин $(x_1, x_2, ..., x_k)$ добавляется новая вершина y, смежная с $x_1, x_2, ..., x_k$.

59. **Стягивание вершин.** Заданное множество вершин объединяется в одну вершину, а полученные петли удаляются.

60. **Удаление (ребра) дуги.** В графе удаляется ребро без инцидентных вершин.

61. **Добавление ребра (дуги).** Для заданной пары вершин x, y добавляется ребро (x, y).

62. **Стягивание ребра (дуги)** – вершины x и y, инцидентные заданному ребру, стягиваются.

63. ***n*-раскрашиваемый граф** – граф G, у которого хроматическое число $\lambda(G) = n$.

64. **Сильносвязный орграф (сильный орграф) (strongly connected graph)** – орграф, у которого любые две вершины взаимодостижимы.

65. **Односторонне связный орграф (односторонний орграф)** – орграф, у которого любая пара вершин односторонне связна.

66. **Слабосвязный орграф (слабый орграф)** – орграф, который после дезориентации дуг будет связным.

67. **Однородный или регулярный (*k*-регулярный) граф** – граф, все вершины которого имеют одну и ту же степень, равную k.

68. **Число вершинной связности графа** G – наименьшее число вершин, удаление которых приводит к несвязному графу.

69. **Число реберной связности графа** G – наименьшее число ребер, удаление которых приводит к несвязному графу.

70. **Хроматическое число (chromatic number)** $\lambda(G)$ – наименьшее число n, для которого граф G имеет n-раскраску.

71. **Укладка графа на поверхность** S в трехмерном эвклидовом пространстве – построение изоморфного G графа L, лежащего на S вершинами (точками), ребрами (непрерывными линиями конечной длины) и дугами (ориентированными линиями конечной длины), причем линии на поверхности S не пересекаются.

72. **Раскраска (вершинная)** – приписывание цветов (натуральных чисел) вершинам графа, такое, что никакие две смежные вершины не получают одинаковые цвета (числа).

73. ***n*-раскраска** – раскраска, в которой используется n цветов.

74. **Раскраска реберная** – приписывание цветов ребрам графа, такое, что никакие два смежных ребра не получают одинакового цвета.

75. ***n*-раскраска реберная** – раскраска ребер, использующая точно n цветов.

76. **Полная раскраска** – раскраска вершин графа, такая, что для любых двух цветов найдутся две смежные вершины, окрашенные в эти цвета.

77. **Генерация графов** – процедура построения графов.

78. **Объединение графов (помеченных графов)** $G_1 = (X_1, U_1)$ и $G_2 = (X_2, U_2)$ – граф $G_1 \cup G_2$, множеством вершин которого является $X = X_1 \cup X_2$, а множеством ребер $U = U_1 \cup U_2$.

79. **Пересечение графов (помеченных графов)** $G_1 = (X_1, U_1)$ и $G_2 = (X_2, U_2)$ – граф $G_1 \cap G_2$, множеством вершин которого является $X = X_1 \cap X_2$, а множеством ребер $U = U_1 \cap U_2$, где U_1 и U_2 – множество неупорядоченных пар вершин.

80. **Соединение графов** $G_1 = (X_1, U_1)$ и $G_2 = (X_2, U_2)$, таких, что $X_1 \cap X_2 =$ Ж, $U_1 \cap U_2 =$ Ж – граф $G_1 + G_2$, который состоит из $G_1 \cup G_2 \cup K(X_1, X_2)$, где $K(X_1, X_2)$ – полный двудольный граф с множеством вершин X_1 и X_2 в долях.

81. **Сумма графов** $G_1 = (X_1, U_1)$ и $G_2 = (X_2, U_2)$ – граф $G_1 \oplus G_2$, состоящий из множества вершин $X = X_1 \oplus X_2$ и две вершины $s = (x_1, x_2)$ и $t = (y_1, y_2)$ $(s, t \in X, x_1, y_1 \in X_1, x_2, y_2 \in X_2)$ смежны в $G_1 \oplus G_2$ тогда и только тогда, когда $x_1 = y_1$ и x_2 смежна с y_2 или $x_2 = y_2$ и x_2 смежна с y_1.

82. **Симметрическая разность графов** G_1 и G_2 – граф $G_1 \nabla G_2 = G_1 \cup G_2 - G_1 \cap G_2$.

83. **Разность графов (помеченных графов)** – граф $G_1 – G_2$, который получается из G_1 удалением элементов, соответствующих графу G_2.

84. **Композиция графов (лексикографическое произведение графов)** $G_1 = (X_1, U_1)$ и $G_2 = (X_2, U_2)$ – граф $G = G_1[G_2]$, состоящий из множества вершин $X = X_1 \oplus X_2$, и две вершины

$s = (x_1, x_2)$ и $t = (y_1, y_2)$ смежны в G тогда и только тогда, когда x_1 смежна с y_1 или $x_1 = y_1$ и x_2 смежна с y_2.

85. **Разрез (сечение графа)** – множество элементов, удаление которых увеличивает число компонент (односторонних компонент, слабых компонент) связности графа.

86. **Простой разрез** – разрез, у которого любое собственное подмножество элементов не является разрезом.

87. **Смешанный разрез** – разрез, элементами которого выступают как ребра (дуги), так и вершины.

88. **Реберный разрез** – все элементы разреза ребра (дуги).

89. **Вершинный разрез.** Все элементы разреза – вершины.

90. **(*s*, *t*)-разрез** – разрез, при удалении которого вершина t не достижима из s.

91. **Минимальный разрез** – разрез с наименьшим числом элементов.

92. **Минимальная цепь** – (s, t)-цепь с минимальным числом ребер.

93. **Максимальный разрез** – разрез с наибольшим числом элементов среди всех простых разрезов графа.

94. **Максимальная цепь** – простая (s, t)-цепь с максимально возможным числом ребер.

95. **Максимальное паросочетание** – паросочетание с максимально возможным числом ребер в паросочетании данного графа (числом реберной независимости).

96. **Фундаментальный цикл (контур)** – цикл, определяемый некоторым остовом и хордой графа (орграфа).

97. **Одноцветный класс** – множество всех вершин одного цвета.

98. **Наибольшее паросочетание** – паросочетание с наибольшей величиной.

99. **Хорда** – ребро графа, не принадлежащее остову.

ЛИТЕРАТУРА

Основная

1. Басакер, Р. Конечные графы и сети / Р. Басакер, Т. Саати. – М.: Наука, 1997.
2. Иванов, Б.Н. Дискретная математика / Б.Н. Иванов. – М.: Лаборатория Базовых Знаний, 2001.
3. Кузнецов, О.М. Дискретная математика для инженера / О.М. Кузнецов, Г.М. Адельсон-Вельский. – М.: Энергоатомиздат, 1988. – 480 с.
4. Лихтарников, Л.М. Математическая логика: курс лекций / Л.М. Лихтарников, В.М. Сукачева. – СПб.: Лань, 1998. – 288 с.
5. Никольская, И.Л. Знакомство с математической логикой / И.Л. Никольская. – М.: Флита, 1998.
6. Новиков, Ф.А. Дискретная математика для программистов / Ф.А. Новиков. – СПб.: Питер, 2000. – 304 с.
7. Пономарев, В.Ф. Дискретная математика для информатиков-экономистов: учебное пособие / В.Ф. Пономарев. – Калининград: КГТУ и КИМБ, 2002. – 239 с.
8. Редькин, Н.П. Дискретная математика: курс лекций для студентов-механиков / Н.П. Редькин. – СПб.: Лань, 2003. – 96 с.
9. Сачков, В.Н. Введение в комбинаторные методы дискретной математики / В.Н. Сачков. – М.: Наука, 1982. – 384 с.
10. Судоплатов, С.В. Элементы дискретной математики: учебник / С.В. Судоплатов, Е.В. Овчинникова. – М.: ИНФРА-М; Изд-во НГТУ, 2002. – 280 с.
11. Уилсон, Р. Введение в теорию графов / Р. Уилсон. – М.: Мир, 1977.
12. Яблонский, С.В. Введение в дискретную математику / С.В. Яблонский. – М.: Наука, 1986.

Дополнительная

1. Грей, П. Логика, алгебра и базы данных / П. Грей. – М.: Машиностроение, 1989. – 360 с.
2. Иванов, Б.Н. Дискретная математика. Алгоритмы и программы / Б.Н. Иванов. – 2-е изд. – М.: Едиториал УРСС, 2003. – 296 с.
3. Кириллов, В.И. Логика / В.И. Кириллов, А.А. Старченко. – М.: Высшая школа, 1987.
4. Кук, Д. Компьютерная математика / Д. Кук, Г. Бейз. – М.: Наука, 1990. – 384 с.
5. Лавров, И.А. Задачи по теории множеств, математической логике и теории алгоритмов / И.А. Лавров, Л.Л. Максимова. – М.: Физматлиз, 2001. – 255 с.
6. Мендельсон, Э. Введение в математическую логику / Э. Мендельсон. – М.: Наука, 1984.
7. Непейвода, Н.Н. Прикладная логика: учебное пособие / Н.Н. Непейвода. – 2-е изд., испр. и доп. – Новосибирск: Изд-во Новосиб. ун-та, 2000. – 521 с.
8. Новиков, Ф.А. Дискретная математика / Ф.А. Новиков. – СПб.: Питер, 2001.
9. Сачков, В.Н. Введение в комбинаторные методы дискретной математики / В.Н. Сачков. – М.: Наука, 1982.
10. Харари, Ф. Теория графов / Ф. Харари; под ред. Г.П. Гаврилова. – М.: Мир, 1973.

СОДЕРЖАНИЕ

Учебное издание

БОХАН Сергей Гавриилович
ПАРМОН Светлана Ивановна

ДИСКРЕТНАЯ МАТЕМАТИКА

Методическое пособие по лабораторным работам
для студентов машиностроительного факультета специальностей
1-55 01 01 «Интеллектуальные приборы, машины и производства»,
1-55 01 03 «Компьютерная мехатроника»,
1-36 01 01 «Технология машиностроения»,
1-36 01 04 «Оборудование и технологии
высокоэффективных процессов обработки материалов»,
1-53 01 01 «Автоматизация технологических процессов и производств»

В 2 частях

Часть 1

Редактор Т.А. Подолякова
Компьютерная верстка Д.А. Исаева

Подписано в печать 27.10.2010.
Формат 60×84 $^{1}/_{16}$. Бумага офсетная.
Отпечатано на ризографе. Гарнитура Таймс.
Усл. печ. л. 4,30. Уч.-изд. л. 3,37. Тираж 100. Заказ 399.

Издатель и полиграфическое исполнение:
Белорусский национальный технический университет.
ЛИ № 02330/0494349 от 16.03.2009.
Проспект Независимости, 65. 220013, Минск.