РОСЖЕЛДОР
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Ростовский государственный университет путей сообщения»
(ФГБОУ ВО РГУПС)

Е.В. Голубенко

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАЦИОННЫХ И КОММПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

Учебно-методическое пособие
для курсовой работы

Ростов-на-Дону
2017

УДК 681.3(07) + 06

Рецензент – доктор технических наук, профессор М.А. Бутакова

Голубенко, Е.В.

Теоретические основы информационных и компьютерных технологий: учебно-методическое пособие для курсовой работы / Е.В. Голубенко; ФГБОУ ВО РГУПС. – Ростов н/Д, 2017. – 32 с.

Рассмотрены основные требования к оформлению исследований студента в рамках курсового проектирования. Содержится обзорный материал, приведены задания и методика выполнения курсовой работы.

Предназначено для студентов и магистрантов направлений «Информатика и вычислительная техника», «Информационные системы и технологии» и «Механотроника и робототехника», а также для студентов, аспирантов и магистрантов всех специальностей, изучающих дисциплину «Теоретические основы информационных и компьютерных технологий» и смежные дисциплины.

Одобрено к изданию кафедрой «Вычислительная техника и автоматизированные системы управления».

СОДЕРЖАНИЕ

1 ОСНОВНЫЕ ПОНЯТИЯ И АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

Граф – абстрактный математический объект, представляющий собой множество вершин графа и набор рёбер, то есть соединений между парами вершин. Например, за множество вершин можно взять множество аэропортов, обслуживаемых некоторой авиакомпанией, а за множество рёбер взять регулярные рейсы этой авиакомпании между городами.

Для разных областей применения виды графов могут различаться направленностью, ограничениями на количество связей и дополнительными данными о вершинах или рёбрах. Многие структуры, представляющие практический интерес в математике и информатике, могут быть представлены графами. Например, строение справочного сайта можно смоделировать при помощи ориентированного графа, в котором вершины – это статьи, а дуги (ориентированные рёбра) – гиперссылки (тематическая карта).

Графы являются основным объектом изучения теории графов.

1.1 Определения

Теория графов не обладает устоявшейся терминологией. В различных публикациях под одними и теми же терминами понимаются разные вещи. Ниже приведены наиболее часто встречаемые определения.

Граф, или неориентированный граф G – это упорядоченная пара

$$G := (V , E),$$

где V – это непустое множество вершин или узлов, а E – множество пар (в случае неориентированного графа – неупорядоченных) вершин, называемых рёбрами.

V и E обычно считаются конечными множествами. Многие результаты, полученные для конечных графов, неверны (или каким-либо образом отличаются) для бесконечных графов, поскольку не все утверждения, имеющие место для конечных совокупностей, выполняются в случае бесконечных множеств.

Вершины и рёбра графа называются также элементами графа, число вершин в графе $| V |$ – порядком, число рёбер $| E |$ – размером графа.

Вершины u и v называются концевыми вершинами (или просто концами) ребра $e = \{ u, v \}$. Ребро, в свою очередь, соединяет эти вершины. Две концевые вершины одного и того же ребра называются соседними.

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

Два ребра называются кратными, если множества их концевых вершин совпадают.

Ребро называется петлёй, если его концы совпадают, то есть $e = \{v , v\}$. Степенью $\deg V$ вершины V называют количество инцидентных ей рёбер (при этом петли считают дважды).

Вершина называется изолированной, если она не является концом ни для одного ребра; висячей (или листом), если она является концом ровно одного ребра.

Ориентированный граф (сокращённо орграф) G – это упорядоченная пара

$G := (V , A)$,

где V – непустое множество вершин или узлов, и A – множество (упорядоченных) пар различных вершин, называемых дугами или ориентированными рёбрами.

Дуга – это упорядоченная пара вершин (v, w) , где вершину v называют началом, а w – концом дуги. Можно сказать, что дуга $v \rightarrow w$ ведёт от вершины v к вершине w .

Смешанный граф G – это граф, в котором некоторые рёбра могут быть ориентированными, а некоторые – неориентированными. Записывается упорядоченной тройкой $G := (V , E , A)$, где V , E и A определены так же, как выше.

Ориентированный и неориентированный графы являются частными случаями смешанного.

Маршрутом в графе называют конечную последовательность вершин, в которой каждая вершина (кроме последней) соединена со следующей в последовательности вершиной ребром. Цепью называется маршрут без повторяющихся рёбер. Простой цепью называется маршрут без повторяющихся вершин (откуда следует, что в простой цепи нет повторяющихся рёбер).

Ориентированным маршрутом (или путём) в орграфе называют конечную последовательность вершин и дуг, в которой каждый элемент инцидентен предыдущему и последующему.

Циклом называют цепь, в которой первая и последняя вершины совпадают. При этом длиной пути (или цикла) называют число составляющих его рёбер. Заметим, что если вершины u и v являются концами некоторого ребра, то согласно данному определению, последовательность (u , v , u) является циклом. Чтобы избежать таких «вырожденных» случаев, вводят следующие понятия.

Путь (или цикл) называют простым, если рёбра в нём не повторяются; элементарным, если он простой и вершины в нём не повторяются.

Простейшие свойства путей и циклов:

- всякий путь, соединяющий две вершины, содержит элементарный путь, соединяющий те же две вершины;
- всякий простой неэлементарный путь содержит элементарный цикл;

- всякий простой цикл, проходящий через некоторую вершину (или ребро), содержит элементарный (под-)цикл, проходящий через ту же вершину (или ребро);
- петля – элементарный цикл.

Бинарное отношение на множестве вершин графа, заданное как «существует путь из u в v », является отношением эквивалентности и, следовательно, разбивает это множество на классы эквивалентности, называемые компонентами связности графа. Если у графа ровно одна компонента связности, то граф связный. На компоненте связности можно ввести понятие расстояния между вершинами как минимальную длину пути, соединяющего эти вершины.

Дополнительные характеристики графов:

Граф называется:

- связным, если для любых вершин u , v есть путь из u в v .
- сильно связным или ориентированно связным, если он ориентированный, и из любой вершины в любую другую имеется ориентированный путь.
- деревом, если он связный и не содержит нетривиальных циклов.
- полным, если любые его две (различные, если не допускаются петли) вершины соединены ребром.
- двудольным, если его вершины можно разбить на два непересекающихся подмножества V_1 и V_2 так, что всякое ребро соединяет вершину из V_1 с вершиной из V_2 .
- k-дольным, если его вершины можно разбить на k непересекающихся подмножества V_1, V2,…,V_k так, что не будет рёбер, соединяющих вершины одного и того же подмножества.
- полным двудольным, если каждая вершина одного подмножества соединена ребром с каждой вершиной другого подмножества.
- планарным, если граф можно изобразить диаграммой на плоскости без пересечений рёбер.
- взвешенным, если каждому ребру графа поставлено в соответствие некоторое число, называемое весом ребра.
- хордальным, если граф не содержит индуцированных циклов с длиной больше трёх.

Способы представления графа в информатике

Матрица смежности – таблица, где как столбцы, так и строки соответствуют вершинам графа. В каждой ячейке этой матрицы записывается число, определяющее наличие связи от вершины-строки к вершине-столбцу

(либо наоборот). Это наиболее удобный способ представления плотных графов.

Недостатком являются требования к памяти, прямо пропорциональные квадрату количества вершин.

Матрица инцидентности – таблица, где строки соответствуют вершинам графа, а столбцы соответствуют связям (рёбрам) графа. В ячейку матрицы на пересечении строки i со столбцом j записывается: 1в случае, если связь j «выходит» из вершины i , −1, если связь «входит» в вершину, 0 во всех остальных случаях (то есть если связь является петлёй или связь не инцидентна вершине)

Данный способ является самым ёмким (размер пропорционален |V| |E|) для хранения, поэтому применяется очень редко, в особых случаях (например, для быстрого нахождения циклов в графе).

Список смежности – список, где каждой вершине графа соответствует строка, в которой хранится список смежных вершин. Такая структура данных не является таблицей в обычном понимании, а представляет собой «список списков».

Размер занимаемой памяти: O (| V | + | E |) .

Это наиболее удобный способ для представления разреженных графов, а также при реализации базовых алгоритмов обхода графа в ширину или глубину, где нужно быстро получать «соседей» текущей просматриваемой вершины.

Список рёбер – список, где каждому ребру графа соответствует строка, в которой хранятся две вершины, инцидентные ребру.

Размер занимаемой памяти: O (| E |) .

Это наиболее компактный способ представления графов, поэтому часто применяется для внешнего хранения или обмена данными.

1.2 Некоторые задачи и алгоритмы теории графов

1.2.1 Построение семейства непополнимых внутренне устойчивых подмножеств графа

Подмножества вершин некоторого графа G(X, U), несмежных между собой, называют *внутренне устойчивыми.* Внутренне устойчивое подмножество считают непополнимым (НВУП), если его нельзя дополнить ни одной другой вершиной $x_j \in X$ графа G(X, *U)* без потери свойств внутренней устойчивости.

Произвольный граф G(X, *U)* может иметь несколько НВУП, для конкретной же задачи бывает необходимо выделить НВУП с определенными свойствами, например те, которые содержат наибольшее число вершин. Поэтому возникает необходимость перечисления всех НВУП (построения семейства НВУП). Это семейство можно выделить с помощью различных алгоритмов. Выбор алгоритма зависит от требований к его быстродействию и

объему памяти, предназначенному для размещения программы и ее входных и выходных данных.

Быстродействие алгоритма для заданной структуры входных, рабочих и выходных массивов, предназначенных для хранения данных и промежуточных результатов, определяется суммарным временем выполнения отдельных операторов. Задавшись числом вершин графа, структурой массивов, используемых в программе, можно оценить быстродействие алгоритма, а следовательно, и объем памяти.

Изменяя структуру массивов, можно оценить предельное быстродействие различных алгоритмов и выбрать тот, который наиболее полно удовлетворяет требованиям по быстродействию и объему используемой памяти.

Рассмотрим основные алгоритмы построения НВУП: алгоритмы, основанные на методе Магу и алгоритм, реализующий поэлементную дизъюнкцию строк матрицы смежности.

Алгоритмы, использующие метод Магу, основаны на вычислении составленного по матрице инциденций $S = \|s_{ij}\|_{n\times r}$ (n– число вершин графа; r – число ребер графа) произведения $\Pi_G = \prod_{j=1}^{r}\sum_{i-1}^{n} s_{ij}x_i$ в котором j-й сомножитель есть сумма вершин, инцидентных ребру $u_j \in U$ графа G(X, U). В произведении раскрывают скобки и для полученной суммы выполняют минимизацию по правилам булевой алгебры:

$x_i^n = x_i$; $x_i + x_i + \ldots + x_i = x_i$;
$x_i + 1 = 1$; $x_i\, x_i \ldots x_i = x_i$; $x_i \cdot 1 = x_i$.

Известно, что *q-е* слагаемое суммы K_q, преобразованное по правилам булевой алгебры, в качестве сомножителей содержит вершины из $X \setminus F_q$.

Поэтому для нахождения всех F_q необходимо относительно каждого слагаемого суммы вычислить $F_q = X \setminus K_q$, где q=1..m – число слагаемых. Для графа, заданного матрицей инциденций вида:

S =		u_1	u_2	u_3	u_4
	x_1	0	1	1	0
	x_2	0	1	0	1
	x_3	1	0	1	0
	x_4	1	0	0	1

Произведение $\Pi_G = (x_3+x_4)\,(x_1+x_2)\,(x_1+x_3)\,(x_2+x_4)$. Раскрывая скобки, получим:

$\Pi_G = x_1x_2x_3 + x_1x_3x_4 + x_1x_2x_3 + x_1x_3x_4 + x_1x_2x_3 + x_1x_2x_3x_4 + x_2x_3 + x_2x_3x_4 + x_1x_2x_4 + x_1x_4 + x_1x_2x_3x_4 + x_1x_3x_4 + x_1x_2x_4 + x_1x_2x_4 + x_2x_3x_4 + x_2x_3x_4$.

Применяя к слагаемым суммы правила булевой алгебры, в частности

$x_{i_1} x_{i_2} x_{i_3} + x_{i_1} x_{i_2} x_{i_3} = x_{i_1} x_{i_2} x_{i_3}$, найдем

$\Pi_G = x_1x_2x_3 + x_1x_3x_4 + x_1x_2x_3x_4 + x_2x_3 + x_2x_3x_4 + x_1x_2x_4 + x$

Используя операцию поглощения $((x_{i_1} x_{i_2} x_{i_3} + x_{i_1} x_{i_2}) = x_{i_1} x_{i_2} (x_{i_3} + 1) = x_{i_1} x_{i_2}$,
окончательно определим

$\Pi_G = x_2x_3 + x_1x_4 = K_1 + K$

Следовательно,

$F_1 = X \setminus K_1 = \{ x_1, x_2, x_3, x_4 \} \setminus \{ x_2, x_3 \} = \{ x_1, x_4 \}$

$F_2 = X \setminus K_2 = \{ x_1, x_2, x_3, x_4 \} \setminus \{ x_1, x_4 \} = \{ x_2, x_3 \}$.

Достоинством этих алгоритмов является то, что их реализация основана на использовании булевой алгебры, что способствует ускорению разработки программы. Однако этим алгоритмам присущи такие недостатки, как низкое быстродействие вследствие многократного применения операции поглощения, реализуемой попарным сравнением слагаемых Π_G значительный объем памяти, необходимой для размещения промежуточных слагаемых Π_G, число которых трудно предсказать. Это приводит к отказам программы по допустимым времени решения и объему памяти. По этой причине такие алгоритмы могут быть использованы для графов с небольшим числом ребер.

Алгоритм, реализующий поэлементную дизъюнкцию строк матрицы смежности, имеет следующий вид:

1. В матрице смежности A нулевые элементы главной диагонали заменить на единицы.

2. Из строки x_i $i \in I = \{1, 2, ..., n\}$, где n – число вершин графа, выбрать элемент $c_{ij}=0$, $j \in I$, $i<j$, который определяет строку x_j, и перейти к п. 3. Если все $c_{ij}=1$, то получить подмножество вида $F_\beta=\{x_i\}$ и перейти к п. 9.

3. Осуществить поэлементную дизъюнкцию строк x_i и x_j, определить $M_{ij}=x_i \cup x_j$.

4. Из строки M_{ij} выбрать элемент $c_k=0$ $k \in I$ и перейти к п. 5. Если $c_к = 1$, то перейти к п. 6.

5. Произвести поэлементную дизъюнкцию строк M_{ij} и $x_к$, найти $M_{i, j, k} = M_{i, j} \cup x_k$. Заменяя индексы (i, j) на i, a k на j, перейти к п. 4.

6. Определить очередное НВУП: $F_y = \{x_i, x_j, x_k, ...\}$.

7. Из строки x_i выбрать элемент $c_l=0$, $l \in I$, причем $x_l \notin Fy$, и перейти к п.8. Если $x_l=1$, то перейти к п. 9.

8. Заменить в выражении $M_{i,j}= x_i \cup x_j$ индекс j на l и перейти к п. 3.

9. Заменяя индекс i на *i+1* перейти к п. 2. Если *i=n*, то перейти к п. 10.

10. Семейство *F* построено.

1.2.2 Алгоритмы раскраски вершин графа

При решении ряда практических задач часто необходимо вершины неориентированного простого, без петель графа, разбить на непересекающиеся подмножества несмежных вершин. Разбиение множества вершин графа G(X, *U)* на *k* непересекающихся подмножеств (классов)

$$X_1,\ X_2,\ \ldots,\ X_k;\ \ X=\bigcup_{i=1}^{k} X_i; X_i \bigcap X_j=\varnothing; i, j \in I=\{1,2,...,k\} \qquad (1)$$

при котором каждое X_i подмножество является внутренне устойчивым подмножеством (подмножеством, у которого любые две вершины не смежны), называют *раскраской вершин графа.* Если каждому подмножеству X_t поставить в соответствие определенный цвет, то вершины, например, подмножества X_1 можно раскрасить в один цвет, вершины подмножества X_2 – в другой и т.д. Наименьшее число подмножеств, на которое разбивается граф при раскраске (1), называют *хроматическим числом k(G).*

Задачу о раскраске вершин графа можно сформулировать таким образом. Дан неориентированный простой без петель граф G(X, *U).* Найти такой алгоритм (метод) раскрашивания вершин графа, который позволял бы наименьшим числом цветов раскрасить вершины графа так, чтобы никакие смежные вершины не были раскрашены одним цветом.

Решение этой задачи осуществляется с помощью точных и приближенных методов. В первом случае для получения оптимальной раскраски требуется большое время, во втором достаточно «хорошую» раскраску вершин графа можно найти за более короткое время. Примерами алгоритмов первой группы являются алгоритм, использующий метод Магу; алгоритм, основанный на рассмотрении r-подграфов и др.

Алгоритм, использующий метод Магу, состоит из следующих операций:

1. Для графа G(X, *U)* построить семейство *F'* НВУП.

2. Составить матрицу $V||m_{ij}||_{n\times m}$, где $n=|X|$ m =|F'|, каждый элемент которой

$$v_{ij}=\{1, если\_x_i \in F_j; 0, если\_x_i \notin F_j\}$$

3. Для каждой вершины графа G(X, *U)* по матрице V найти суммы тех $F_j \in F'$, в которые она входит, и записать произведение этих сумм.

4. Раскрыть скобки по правилам булевой алгебры и выбрать слагаемое, состоящее из наименьшего числа сомножителей .

Алгоритм, основанный на рассмотрении максимальных r-подграфов, включает в себя следующие операции:

1. Пусть r=1. Найти множество вершин F_r^j (j=l, q_r) максимальных r-подграфов графа G(q_r– число r-подграфов; для $r=1$ выделяют семейство НВУП). Положить j=1.

2. Для подграфа G^j (X_r^j, Γ_x), у которого $X'_r=X\backslash F^J_r$; $x \in X_r^j$, выделить НВУП $F_1[G^J]$. Если такое множество существует, то переходят к п. 3. Если такие множества определены, то перейти к п. 6.

3. Образовать $F= F_r^j \cup F_1[G^j]$.

4. Если $F = X$, то разбиение графа на максимальные r-подграфы получено. Число (r+1) есть хроматическое число графа G. Если $F \neq X$, то перейти к п. 5.

5. Если $F \in F^1$, где F^1–ранее полученное разбиение графа $G(F^1 \in Q$; Q – семейство разбиений), то переходят к п. 2. Если$F \supset F^1$, то положить $Q=Q\backslash F^1$ по всем разбиениям из семейства Q. Определить $Q=Q \cup F$ и перейти к п. 2. Если не выполняется ни одно из предыдущих условий, то положить $Q=Q \cup F$ и перейти к п. 2.

6. Если $j<q_r$, то положить j=j+1 и перейти к п. 2. Если $i=q_r$, то положить j=1, r=r+1, где q_r – число разбиений в семействе Q, и перейти к п. 2.

1.2.3 Построение минимального покрывающего дерева

Если взвешенный орграф содержит циклы, то из него можно получить дерево, которое содержит все вершины исходного графа. Такое дерево называется покрывающим или остовным деревом. Для одного и того же графа можно, как правило, построить множество покрывающих деревьев. Покрывающее дерево, у которого сумма весов входящих рёбер минимальна, называется минимальным покрывающим деревом.

Подобная задача возникает, например, если надо соединить сетью дорог множество населённых пунктов с минимальными затратами.

Один из простых алгоритмов получения минимального покрывающего дерева состоит в следующем:

1. Упорядочим рёбра графа по порядку возрастания весовых коэффициентов.

2. Выбираем ребро с минимальным весом и включаем его в дерево.

3. Выбираем следующее ребро из упорядоченного списка. Если при включении ребра не образуется цикла, включаем его в дерево, иначе отбрасываем.

4. Если количество рёбер в дереве достигло n–1, где n – количество вершин в исходном графе, то минимальное покрывающее дерево построено, иначе перейти к п. 3.

2 ОБЗОР НЕКОТОРЫХ АЛГОРИТМОВ ДИСКРЕТНОЙ МАТЕМАТИКИ

2. 1 Операции в конечных полях

Обработка цифровой информации в виде кодов и шифров производится с помощью операций конечных полей (полей Галуа), которые являются совершенно точными, так как не имеют погрешностей округления и ограничения, свойственных позиционным арифметикам в бесконечных числовых полях.

Конечные поля имеют $q = p^m$ элементов, где $p \geq 2$ – простое число, называемое характеристикой, $m \geq 2$ – степень расширения (целое положительное число) и обозначаются – $GF(p^m)$.

При $m = 1$ конечное поле называется простым – $GF(p)$. Это поле изоморфно кольцу вычетов целых чисел по модулю p:

$$GF(p) \cong Z_p = \langle \{0, 1, \ldots, p\text{-}1\}, \oplus, \otimes \rangle .$$

Арифметика простых полей – модулярная:

$$a \oplus b = (a + b) \bmod p = rest\left[\frac{a+b}{p}\right],$$

$$a \otimes b = ab \bmod p = rest\left[\frac{ab}{p}\right],$$

где $a, b \in GF(p)$ – целочисленные элементы поля; *rest* – остаток; все операции в правых частях формул – обычные целочисленные.

ПРИМЕР 1

Аддитивные и мультипликативные таблицы Кэли для поля $GF(7)$ имеют вид:

$\oplus$	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	2	3	4	5	6	0
2	2	3	4	5	6	0	1
3	3	4	5	6	0	1	2

4	4	5	6	0	1	2	3
5	5	6	0	1	2	3	4
6	6	0	1	2	3	4	5

$\otimes$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

Для расширений простых полей $GF(p^m)$, $m \geq 2$, существуют и другие формы представления элементов поля (табл. 1).

Таблица 1

Представления элемента конечного поля

Форма представления	Соотношение
Целочисленная, N	$N \in \{0, 1, 2, \ldots, q-1\}$
Степенная, α^{N-1}	$\alpha^{N-1} \in \{\alpha^{-\infty}, \alpha^0, \alpha^1, \ldots, \alpha^{q-2}\}$
Полиномиальная, $GF_N(x)$	$GF_N(x) = x^{N-1} \bmod \pi(x) = GF_N[m]x^{m-1} \oplus GF_N[m-1]x^{m-2} \oplus$ $\oplus \ldots \oplus GF_N[1]$
Векторная, GF_N	$GF_N = (GF_N[m], GF_N[m-1], \ldots, GF_N[1])$

В данной таблице α – примитивный элемент поля, через степени которого выражаются все элементы поля (ненулевые), а $\pi(x)$ – примитивный

полином поля, порождающий полиномиальные представления тех же элементов по закону

$$x^{N-1} \bmod \pi(x).$$

Примитивный элемент α является корнем примитивного полинома $\pi(x)$.

Для обработки цифровой информации применяются в основном поля Галуа характеристики 2, т. е. $GF(2^m)$, поскольку их арифметика является наиболее простой (минимальной по вычислительной сложности).

Коэффициенты примитивных полиномов для таких полей приведены в табл. 18.

Таблица 2

Коэффициенты примитивных полиномов для полей $GF(2^m)$

m	Коэффициенты $\pi(x)$
2	1 1 1
3	1 0 1 1
4	1 0 0 1 1
5	1 0 0 1 0 1
6	1 0 0 0 0 1 1
7	1 0 0 0 1 0 0 1
8	1 0 0 0 1 1 1 0 1
9	1 0 0 0 0 1 0 0 0 1
10	1 0 0 0 0 0 0 1 0 0 1

Из табл. 1 следует, что целочисленная форма представления N изоморфна степенной, поскольку

$$N = \log_{\alpha} \alpha^{N-1} + 1.$$

Поэтому N называют модифицированным логарифмом, а α^{N-1} – антилогарифмом.

Полиномиальная и векторная формы просто идентичны, поэтому поля Галуа описываются зависимостью $GF_N(N)$, называемой таблицей логарифмов – антилогарифмов.

ПРИМЕР 2

Для поля $GF(2^3)$, порождённого примитивным полиномом $\pi(x) = x^3 + x + 1$, таблица логарифмов – антилогарифмов (табл. 3) представлена ниже.

Таблица 3

Зависимость $GF_N(N)$ для поля $GF(2^3)$

N	GF_N
0	0 0 0
1	0 0 1
2	0 1 0
3	1 0 0
4	0 1 1
5	1 1 0
6	1 1 1
7	1 0 1

Однако для вычисления векторного представления GF_N не требуется приведения по $\mathrm{mod}\,\pi(x)$, поскольку существует алгоритм, сводящийся к битовым операциям сдвига и сложения по $\mathrm{mod}\,2$, что минимизирует вычислительные затраты.

АЛГОРИТМ 1

1 В регистр 1 ($P1$) записать бинарную комбинацию $\pi[m], \pi[m-1], \ldots, \pi[0]$, соответствующую коэффициентам примитивного полинома.

2 В регистр 2 ($P2$) записать комбинацию из m компонент 0…01, соответствующую единичному элементу.

3 Текущий номер j элемента поля принять равным 1. Текущий адрес A элемента принять равным начальному адресу $A1$ массива элементов поля в памяти.

4 Содержимое $P2$ переписать в память по адресу A.

5 Осуществить сдвиг содержимого $P2$ влево.

6 Если в $P2$ значение разряда переполнения $P2[m] = 0$, то перейти к п. 7, иначе выполнить сложение по mod 2: $P2 = P2 \oplus P1$.

7 Если текущий номер $j = q - 1 = 2^m - 1$, то перейти к п. 8; иначе увеличить на единицу текущий номер j и текущий адрес A; вернуться к п. 4.

8 Конец алгоритма.

Этот алгоритм сводит вычисление $x^{N-1} \bmod \pi(x)$ к сдвигу влево на каждом шаге и сложению результата с коэффициентами $\pi(x)$ по mod 2, если появляется 1 в m-м разряде.

Наиболее просто операция сложения (вычитание в полях $GF(2^m)$ совпадает со сложением) выполняется для полиномиальных (векторных) элементов:

$$GF_{N1} \boxplus GF_{N2} = (GF_{N1}[m] \oplus GF_{N2}[m], GF_{N1}[m-1] \oplus GF_{N2}[m-1], \ldots, GF_{N1}[1] \oplus$$

$$\oplus GF_{N2}[1],$$

где $\oplus$ означает сложение по $\mathrm{mod}\,2$.

Данный алгоритм сводится к битовым операциям покомпонентного (поразрядного) сложения по $\mathrm{mod}\,2$.

ПРИМЕР 3

Построим таблицу сложения ненулевых элементов поля $GF(2^3)$. Используя зависимость GF_N от N, складываем элементы как векторы GF_{N1} и GF_{N2}, затем переходим к целочисленному эквиваленту суммы. Достаточно

вычислить верхнюю (нижнюю) треугольную матрицу результатов, так как операция сложения коммутативна.

⊞	1	2	3	4	5	6	7
1	0	4	7	2	6	5	3
2		0	5	1	3	7	6
3			0	6	2	4	1
4				0	7	3	5
5					0	1	4
6						0	2
7							0

На самом деле достаточно вычислить строку 1, т. е. суммы вида $1 \boxplus N$, называемые функциями Якоби–Зеча по следующему алгоритму формирования таблицы сложения.

АЛГОРИТМ 2

1 Обнуляем главную диагональ, так как $N \boxplus N = 0$:

$$a_{ii} = 0, \quad 1 \le i \le q-1.$$

2 Вычисляем элементы первой строки и первого столбца $a_{1,j} = 1 \boxplus j$; $a_{j,1} = a_{1,j}$, $2 \le j \le q-1$, используя таблицу $GF_N(N)$ или таблицу функций Зеча, т. е. векторное представление для элемента j.

3 Находим остальные элементы:

$$k = a_{i-1,\,j-1} + 1;$$

$$a_{ij} = \begin{cases} k, & k \le q-1; \\ k-(q-1), & k > q-1; \end{cases}$$

$$a_{ji} = a_{ij},$$

где $2 \le i \le q-2$, $i+1 \le j \le q-1$, а все операции целочисленные.

На шаге 2 выполняется $q-2$ сложений вектора 00 … 01 с векторами GF_j, а остальные элементы получаются из элементов первой строки путём

увеличения на 1 вдоль каждой побочной диагонали с приведением в случае необходимости по $\mathrm{mod}\ q-1$, которое сводится к вычитанию $q-1$.

Таким образом, алгоритм сводится к битовым операциям поразрядного сложения векторов и аддитивным целочисленным операциям с константами 1 и $q-1$.

Разумеется, достаточно использовать верхнюю треугольную матрицу сумм, т. е. исключить из алгоритма операции $a_{ji} = a_{ij}$.

С другой стороны, мультипликативные операции (умножение и деление) проще выполняются для элементов в целочисленной форме.

АЛГОРИТМ 3

Произведение двух целочисленных элементов $N_1 \boxtimes N_2$ выполняется по соотношению:

$$N_1 \boxtimes N_2 = \begin{cases} 0, \ N_1 = 0 \text{ или } N_2 = 0; \\ N_1 + N_2 - 1, \ N_1 + N_2 \le q; \\ N_1 + N_2 - q, \ N_1 + N_2 > q, \end{cases}$$

использующему обычные целочисленные аддитивные операции.

ПРИМЕР 4

Для поля $GF(2^3)$ получаем по алгоритму 5 следующую таблицу (верхнюю треугольную часть):

$\boxtimes$	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7
2		3	4	5	6	7	1
3			5	6	7	1	2
4				7	1	2	3
5					2	3	4
6						4	5
7							6

На самом деле можно исключить и целочисленные аддитивные операции, поскольку матрица произведений представляет собой ганкелев

циркулянт, в котором строки получаются из первой путём циклического сдвига влево.

АЛГОРИТМ 4

1 Формируем первую строку матрицы произведений:

$$m_{1,j} = j, \quad 1 \le j \le q-1.$$

2 Остальные элементы получаются по соотношению:

$$m_{i,j} = \begin{cases} m_{i-1,j+1}, & j \ne q-1; \\ m_{i-1,1}, & j = q-1, \end{cases}$$

где $2 \le i, \ j \le q-1.$

Деление целочисленных элементов производится по следующему алгоритму.

АЛГОРИТМ 5

Пусть $c, b \in GF(q)$ и $b \ne 0$, тогда

$$c \boxdot b = \begin{cases} 0, & c = 0; \\ 1 + c\text{-}b, & c \ge b; \\ q + c\text{-}b, & c < b, \end{cases}$$

где в правой части равенства используются обычные аддитивные целочисленные операции.

ПРИМЕР 5

Таблица деления для поля $GF(2^3)$ имеет вид:

⊡	1	2	3	4	5	6	7
1	1	7	6	5	4	3	2
2	2	1	7	6	5	4	3
3	3	2	1	7	6	5	4
4	4	3	2	1	7	6	5
5	5	4	3	2	1	7	6
6	6	5	4	3	2	1	7
7	7	6	5	4	3	2	1

Поскольку таблица деления является теплицевым циркулянтом, у которого элементы вдоль каждой диагонали, параллельной главной, одинаковы, можно использовать алгоритм без арифметических операций.

АЛГОРИТМ 6

1 Строим первый столбец:

$$d_{i,1} = i, \quad 1 \le i \le q-1.$$

2 Остальные элементы получаются из соотношения:

$$d_{j,i} = \begin{cases} d_{q-1,i-1}, & j = 1; \\ d_{j-1,i-1}, & j \ne 1, \end{cases}$$

где $2 \le i \le q-1, \quad 1 \le i \le q-1.$

Рассмотренные алгоритмы позволяют сводить все полевые операции к простейшим битовым или целочисленным, но с использованием таблицы элементов поля (таблицы логарифмов-антилогарифмов). Такой подход используется в известных компьютерных пакетах Communications Toolbox (расширение системы MATLAB) и The Art of ECC.

С другой стороны, можно использовать только полиномиальное представление элементов поля, т. е. отказаться от таблиц логарифмов-антилогарифмов, так как существуют эффективные алгоритмы для реализации мультипликативных полевых операций, сводящихся к элементарным битовым.

Приведём такой алгоритм для умножения полиномиальных элементов.

АЛГОРИТМ 7

1 В регистры $P1$ и $P2$ записать сомножители в виде векторов GF_{N1} и GF_{N2}, составленных из коэффициентов полиномов $GF_{N1}(x)$ и $GF_{N2}(x)$. В регистр $P3$ записать комбинацию π из коэффициентов примитивного полинома $\pi(x)$. Обнулить разряды регистра $P4$, используемого для формирования произведения векторных элементов GF_{N1} и GF_{N2}.

2 Положить текущий номер разряда $j = m-1$.

3 Если $P2[j] = 0$, то перейти к п. 4; иначе выполнить сложение по mod 2: $P4 = P4 \oplus P1$.

4 Если $j = 0$, то перейти к п. 7; иначе осуществить сдвиг содержимого $P4$ влево.

5 Если $P4[m] = 0$, то перейти к п. 6; иначе выполнить $P4 = P4 \oplus P3$.

6 Уменьшить текущий номер j на единицу; перейти к п. 3.

7 Конец алгоритма.

Поскольку все полевые операции сводятся к битовым и (или) целочисленным, то к таким же операциям сводятся и вычисления с полиномами и матрицами над конечными полями.

В частности, существует эффективный алгоритм для формирования матриц дискретного преобразования Фурье–Галуа (ДПФГ), применяемого при обработке кодов и шифров.

Матрица ДПФГ определяется соотношением:

$$W_{ij} = \alpha^{ij \bmod n}, \quad 0 \le i,j \le n-1,$$

где α – примитивный элемент поля, $n \mid (q-1)$ (n делит $(q-1)$ нацело).

Для вычисления W_{ij} без использования полевых операций используем следующий алгоритм.

АЛГОРИТМ 8

Элементы матрицы ДПФГ над полем $GF(2^m)$ получают из соотношения:

$$W_{ij} = 1 + \frac{q-1}{n} ij \bmod (q-1), \quad 0 \le i, j \le n-1,$$

где все операции выполняются в обычной целочисленной арифметике.

ПРИМЕР 6

Найдём матрицы ДПФГ над полем $GF(2^4)$.

Поскольку $q-1=15$, существуют матрицы при $n = 3, 5, 15.$

Приведём матрицы размера 3 и 5:

$$W_3 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 6 & 11 \\ 1 & 11 & 6 \end{bmatrix}, \qquad W_5 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 4 & 7 & 10 & 13 \\ 1 & 7 & 13 & 4 & 10 \\ 1 & 10 & 4 & 13 & 7 \\ 1 & 13 & 10 & 7 & 4 \end{bmatrix}.$$

Рекуррентный вариант позволяет избавиться от операции приведения по $\bmod (q-1)$.

АЛГОРИТМ 9

Элементы матрицы ДПФГ могут вычисляться по следующему рекуррентному соотношению:

$$W_{ij} = \begin{cases} 1, \ i = 0 \text{ или } j = 0; \\ k = W_{i,j-1} + \dfrac{q-1}{n} i, \ \ k \le q-1; \\ k - (q-1), \ \ k > q-1, \end{cases}$$

где $0 \le i, j \le q-1$.

В частном, но важном случае, когда $n = q-1$, в последнем алгоритме остаются только аддитивные целочисленные операции.

Для вычисления спектра, т. е. произведения матрицы ДПФГ на вектор-оригинал, используются аналоги быстрого преобразования Фурье над бесконечными полями.

3 ТРЕБОВАНИЯ К НАПИСАНИЮ И ОФОРМЛЕНИЮ КУРСОВОЙ РАБОТЫ

Курсовая работа – это более глубокое и объемное исследование избранной проблемы учебного курса, чем реферат, доклад и контрольная работа. Общий объем курсовой работы ***20-25 страниц***. При этом ***приложения не входят в объем работы***, что позволяет исследователю уложиться в установленные рамки.

3.1 Цель курсового проектирования

Целью курсовой работы является формирование у студентов навыков программирования вычислительных алгоритмов на основе теоретических и практических знаний, полученных в курсах «Теоретические основы информационных и компьютерных технологий», «Информатика и программирование» В процессе выполнения курсовой работы необходимо составит схему алгоритма выполнить проектирование программы, ее реализацию, представить пример использования программы.

3.2 Структура курсовой работы

Курсовая работа должна включать в себя следующие разделы:
а) титульный лист;
б) лист задания;
б) оглавление ;
в) введение;
г) главы основной части;
д) заключение (выводы);
е) список использованных источников;
ж) приложения – при необходимости;

Введение

Обосновывая тему, студент должен определить ее место и значимость изучения. Необходимо обозначить цель своей работы, четко выделить конкретные задачи, с помощью которых будет достигаться цель исследования. Введение не должно превышать 1/10 части общего объема работы (в среднем 1-2 страницы).

Автор вправе переставить местами названные элементы введения, исходя из принципа оптимальной подачи материала курсовой работы.

Введение состоит из следующих элементов:

1) *обоснование (актуальность) темы* – это степень ее важности в определенный момент и в конкретной ситуации для решения данных проблемы, вопроса или задачи. Освещение актуальности не должно быть многословным.

2) *описание степени разработанности проблемы* – перечисление основных точек зрения, подходов и методологических основ исследований;

3) *указание предмета и объекта работы:* объект исследования – это процесс или явление, порождающее проблемную ситуацию и избранное для изучения.

4) *постановка цели и задач исследования.* Цель – это результат, который необходимо получить при проведения исследования, некоторый образ будущего. Задачи исследования – это те исследовательские действия, которые необходимо выполнить для достижения поставленной в работе цели, решения проблемы;

Основная часть

Основная часть курсовой работы обычно состоит из двух теоретических и практических или экспериментальных глав, при этом каждая глава – из двух-трёх параграфов и выводов по главе. Формулировка глав и параграфов должна быть четкой, краткой и в последовательной форме раскрывать содержание работы. После каждой главы делается вывод по рассмотренному материалу (2-3 предложения).

Первая глава представляет собой аналитический (теоретический) обзор по проблеме, рассматриваемой в работе. На основе изучения литературных источников отечественных и зарубежных авторов рассматривается сущность исследуемой проблемы, анализируются различные подходы их решения, дается их критический анализ, излагается собственная позиция исследователя.

Необходимо выбрать среду разработки, описать ее преимущества и недостатки, грамотно, аргументировано обосновать свой выбор. Разработать структуру будущей программы, выделить ее основные части и продумать правила взаимодействия. Определиться с внешним видом будущего программного средства.

Вторая глава посвящена описанию разработанного Вами программного приложения. Здесь приводится интерфейс программы, руководство по использованию программы . Описываются графические составляющие, приводится контрольный пример использования программы.

Заключение

Заключение объемом 1 – 2 страницы должно содержать в концентрированном (тезисном) виде без какой-либо аргументации ранее обоснованные студентом в тексте работы наиболее важные выводы. Автор курсовой работы должен выделить собственный вклад в разработку темы, подчеркнуть значимость своих выводов и наблюдений. Качество работы увеличится, если ее студент сумеет не только грамотно и профессионально подвести итоги, но и определить перспективность направлений дальнейшего

исследования темы на новом уровне. Не стоит включать в заключение цитаты и примеры.

Список использованных источников.

В список использованных источников и литературы включаются все изученные или использованные автором книги, статьи, нормативные акты и другие источники, имеющие отношение к избранной теме, независимо от того, цитируются ли они в работе. Обязательно включение в список литературы всех цитируемых либо упомянутых в тексте публикаций.

Список сокращений составляется в алфавитном порядке. Точки между буквами, обозначающие сокращенные слова, не ставятся.

Приложения

В Приложение выносится дополнительный материал, который может нарушить связность изложения основного содержания и препятствовать его целостному восприятию. В контексте данной работы, в приложение выносится текст программы.

3.3 Оформление работы

Оформление – одна из важнейших стадий работы над курсовой работой. Причем определенные элементы оформления нельзя откладывать «на потом» – на то время, когда текст в своей основе уже будет написан.

Работа должна быть оформлена аккуратно с соблюдением ряда требований.

Объем работы зависит от многих факторов: масштабности и сложности темы, хронологических рамок исследования, количества привлеченных источников, стиля изложения. Рекомендуемый объем курсовой работы 20-25 страниц (без приложения).

Общие требования

Текст пояснительной записки оформляют в соответствии с требованиями СТП РГУПС 2-07.

Курсовая работа выполняется на листах белой бумаги плотностью 80 г/см3 и форматом 210×297 мм (ГОСТ 9327). На листах оставляются поля по всем четырём сторонам, вычерчивается внутренняя рамка. В правом нижнем углу проставляется номер страницы. Размер левого поля 30 мм, правого – 10 мм, верхнего – 15 мм, нижнего – 20 мм.

Курсовая работа печатается с применением ЭВМ на одной стороне листа. Для печати основного текста используется шрифт Times New Roman кегля 14. Межстрочный интервал – полуторный. Размер абзацного отступа – 1,25 см. Текст должен быть выровнен ***по ширине*** страницы; каждый абзац рекомендуется начинать с красной строки (устанавливается опцией ***Формат / Абзац / Отступ***). Страницы должны быть пронумерованы , номер проставляют в нижнем поле по центру, при нумерации учтите, что первой

страницей является титульный лист, второй – лист задания и оглавление, на которых номер страницы не ставится. Номера страниц проставляются с введения.

Каждая глава, введение, заключение, список используемой литературы, приложение (но не пункты и параграфы) должны начинаться с новой страницы.

Необходимо правильно оформлять общепринятые условные сокращения. После перечисления пишут т.е. (то есть), и т.д. (и так далее), и т.п. (и тому подобное), и др. (и другие), и пр. (и прочие); при ссылках: см. (смотри), ср. (сравни); при цифровом обозначении веков и годов: в. (век), вв. (века), г. (год), гг. (годы).

Текст должен быть написан грамотно, с соблюдением всех требований русского языка. Язык пояснительной записки должен быть сжатым и точным, свойственным научно-техническим документам. Не следует злоупотреблять описаниями устройств или программного обеспечения, известными из литературы. Достаточно коротко перечислить их существенные особенности и дать библиографическую ссылку. Не должны использоваться жаргонные технические выражения. При необходимости сокращенного обозначения выражений или слов принятые сокращения приводятся в перечне сокращений, символов, специальных терминов.

Изложение текста курсовой работы даётся от первого или третьего лица множественного числа или в безличной форме, например: "значение коэффициента принимаем...", "принимают...", "принято".

Заголовки разделов, подразделов, основной текст курсовой работы

Текст работы должен быть разделён на разделы, которые могут включать в себя подразделы.

Каждый раздел имеет порядковый номер, обозначенный арабской цифрой без точки. Каждый раздел работы следует начинать с нового листа. Реферат, содержание, перечень сокращений, введение, заключение, список использованных источников не нумеруются. Подразделы нумеруются арабскими цифрами в пределах каждого раздела. Номер подраздела состоит из номера раздела и подраздела, разделённых точкой. В конце номера подраздела точка не ставится. Подразделы состоят из пунктов. Номер пункта включает номер раздела, порядковый номер подраздела в разделе и порядковый номер пункта в подразделе, разделенные точками (например, «2.5.3» – третий пункт пятого подраздела второго раздела).

Текст заголовка разделов, подразделов рекомендуется выделять полужирным шрифтом:

Пример: **1.3 (третий подраздел первого раздела).**

Текст заголовков пунктов и подпунктов выделять полужирным шрифтом не следует. Единственный подраздел в разделе (а также единственный пункт в подразделе) не допускается.

Заголовки разделов, подразделов и пунктов записывают строчными буквами (кроме первой прописной) с абзацного отступа. Переносы слов в заголовках не допускаются. Точку в конце заголовков раздела, подраздела или

пункта не ставят. Если заголовок состоит из двух предложений, их разделяют точкой.

Заголовки должны быть ясными и четкими, исключающими неопределенность их толкования. Заголовок каждого раздела и подраздела должен отражать не только объект исследования, но и раскрывать содержание изложенного материала в разделе и подразделе.

Иллюстрации

К иллюстрациям относятся рисунки, схемы, фотографии, графики, номограммы, диаграммы, все виды чертежей. Они размещаются сразу после ссылки на них в тексте работы и называются рисунками. Нумеруются арабскими цифрами в пределах соответствующего раздела, например: Рисунок 2.1 (первый рисунок второго раздела).

Если рисунок один, то он обозначается: «Рисунок 1». Иллюстрации каждого приложения обозначают отдельной нумерацией арабскими цифрами с добавлением перед цифрой обозначения приложения. Например: «Рисунок А.5» – пятый рисунок Приложения А.

При ссылках на иллюстрации следует писать:

« … в соответствии с рисунком 2.1» при нумерации в пределах раздела и « … в соответствии с рисунком 2» при сквозной нумерации.

Название иллюстрации помещают под рисунком.

Слово «Рисунок», номер рисунка и название иллюстрации помещают под рисунком через тире, например:

Рисунок 1 – Составные элементы интерфейса пользователя

Располагают симметрично тексту.

В конце номера рисунка и после названия рисунка точки не ставятся, номер от названия отделяется дефисом.

В тексте обязательно должны быть ссылки на все иллюстрации. Иллюстрации размещаются по тексту после первой ссылки на них в тексте. Можно располагать иллюстрации на отдельных листах.

Список использованных источников

Сведения об источниках, включенных в список, приводятся в соответствии с требованиями ГОСТ 7.1 – 2003. Цитирование из теоретических источников должно быть взято в кавычки с указанием источника и страницы. Например: /16, с. 25/.

Нумерация источников даётся в порядке упоминания их в тексте дипломного проекта (работы) или в алфавитном порядке.

Например:

1 **Голицына, О.Л.** Системы управления базами данных: учебное пособие для вузов / О.Л. Голицына, И.И. Попов, Т.Л. Партыка. – М.: Инфра-М, 2006.

– 432 с.

2 **Харламов, А.И.** Общая теория статистики: Статистическая методология в изучении коммерческой деятельности: учебник / А.И. Харламов, О.Э. Башина, В.Т. Батурин и [др.] / Под ред. А.А. Спирина, О.Э. Башиной. – М.: Финансы и статистика, 1994. – 295 с.

При библиографическом описании электронного документа, полученного с web-страницы, необходимо включить следующие обязательные элементы:

Автор. Заглавие страницы.[Указание типа документа]. (Электронный адрес (URL)). Дата обращения.

Пример.

1 **Травин, А.** Три поисковика Рунета, не считая Google. [Электронный документ]. (http://www.netoscop.ru/theme/2001/06/21/2662.html). Проверенно 21.06.2017

Использование чужого материала без ссылки на автора и источник заимствования является плагиатом.

Приложения

Приложения оформляют на листах формата А4, допустимы и другие размеры, кратные формату А4.

Каждое приложение следует начинать с новой страницы с указанием наверху посередине страницы слова «Приложение» и его обозначения в виде заглавной буквы русского алфавита.

Приложения обозначают заглавными буквами русского алфавита, начиная с А, за исключением букв Ё, З, И, Й, О, Ч, Щ, Ь, Ы, Ъ, после буквы Я приложения обозначаются арабскими цифрами. Если в проекте одно приложение, оно обозначается Приложение А.

Ниже обозначения в скобках указывается его характеристика: справочное, рекомендуемое или обязательное.

В тексте на все приложения должны быть ссылки. Приложения располагают в порядке ссылок на них в тексте проекта, они должны иметь общую с остальной частью документа сквозную нумерацию страниц.

Ссылки на приложение оформляются следующим образом: «...смотри приложение А».

Чистовой вариант курсовой работы надо тщательно выверить. В нём должны быть исправлены все ошибки, опечатки, внесены необходимые поправки, тщательно сверены фамилии, цитаты, названия.

4 Примерная тематика курсовой работы

1 программная реализация аддитивной операции в простых полях;
2 программная реализация мультипликативной операции в простых полях;
3 программная реализация перехода от целочисленного представления элементов поля расширения GF(2m) к полиномиальному;
4 алгоритм перехода от полиномиального представления элементов поля GF(2m) к целочисленному;
5 программная реализация таблицы Кэли для мультипликативной операции в поле расширения GF(2m);
6 программная реализация таблицы Кэли для аддитивной операции в поле расширения GF(2m);
7 программная реализация операции деления в поле расширения GF(2m);
8 формирование матрицы дискретного преобразования Фурье в конечном поле GF(2m);
9 быстрый алгоритм для дискретного преобразования Фурье в конечном поле GF(2m);
10 алгоритм кодирования для циклических помехоустойчивых кодов;
11 алгоритм декодирования для циклических помехоустойчивых кодов;
12 алгоритм кодирования для БЧХ-кодов;
13 алгоритм декодирования для БЧХ-кодов;
14 алгоритм кодирования для кодов Рида-Соломона;
15 алгоритм декодирования для кодов Рида-Соломона.
16 Выделить методом Магу внутренне устойчивые подмножества графа. Информация о графе задана в виде матрицы смежности, двухстрочной или треугольной матрицы. Число вершин графа не превосходит *n*.
17 Определить методом Магу хроматическое число графа. Информация о графе задана в виде матрицы смежности, двухстрочной или треугольной матрицы. Число вершин графа равно или не превосходит *n*.
18 Найти толщину графа. Информация о графе задана в виде матрицы смежности, двухстрочной, треугольной матрицы или подмножеств смежностей. Число вершин графа равно или не превосходит *n*.
19 Выделить внутренне полные подмножества графа путем объединения его смежных вершин. Информация о графе задана в виде матрицы инциденций или подмножеств смежности. Число вершин графа равно или не превосходит *n*.
20 Определить хроматическое число графа путем объединения несмежных вершин. Информация о графе задана в виде подмножеств смежности или матрицы инциденций.
21 Построить дерево минимальной длины без ограничения числа инциденций (алгоритм Вайнберга - Лобермана или алгоритм Прима).

Информация о графе задана в виде матрицы смежности, двухстрочной или треугольной матрицы.

22 Построить дерево минимальной длины с ограничением числа инциденций (модифицированный алгоритм Прима). Информация о графе задана в виде матрицы смежности, двухстрочной или треугольной матрицы.

23 Построить гамильтонов цикл или цепь минимальной длины методом ветвей и границ. Информация о графе задана в виде матрицы смежности или двухстрочной матрицы.

24 Построить гамильтонов цикл или цепь минимальной длины итерационным методом, используя парные или циклические перестановки или перестановки с разворотом фрагментов. Информация о графе задана в виде матрицы смежности или двухстрочной матрицы.

25 Построить гамильтонов цикл или цепь минимальной длины последовательным методом. Информация о графе задана в виде матрицы смежности или двухстрочной матрицы.

Возможна реализация курсовой работы по теме, предложенной студентом и своевременно согласованной с преподавателем.

Библиографический список

1 **Дергачева, И. В.** Компьютерные технологии и информатика [Текст] : учеб. пособие / И. В. Дергачева ; ФГБОУ ВО РГУПС. - Ростов н/Д : [б. и.], 2016. - 51 с. + э.р. НТБ

2 **Самсонов, Б.Б**. Алгоритмы цифровой обработки информационных последовательностей: монография / Б.Б.Самсонов, А.И.Филоненков.-Ростов-на-Дону: ФГБОУ ВПО РГУПС, 2012.-218с. + э.р. НТБ

3 **Самсонов, Б.Б.** Дискретная математика, тесты, упражнения и задачи: учеб. пособие / Б.Б. Самсонов, А.И. Филоненков .-2008.-272с.

Учебное издание

Голубенко Евгений Владимирович

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАЦИОННЫХ И КОММПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

Печатается в авторской редакции

Технический редактор Н.С. Федорова

Подписано в печать 30.11.17. Формат 60×84/16.
Бумага газетная. Ризография. Усл. печ. л. 1,86.
Тираж экз. Изд. № 90899. Заказ .

Редакционно-издательский центр ФГБОУ ВО РГУПС.

Адрес университета: 344038, г. Ростов н/Д, пл. Ростовского Стрелкового Полка Народного Ополчения, д. 2.