

Вид нагрузки: КР

Для дисциплин:

Год набора	Шифр	Форма обучения (о или о/з)	Дисциплина
2016	09.03.04	о	Алгоритмы и структуры данных
2016	09.03.01	о	Алгоритмы и структуры данных
2017	09.03.04	о	Алгоритмы и структуры данных
2017	09.03.01	о	Алгоритмы и структуры данных
2018	09.03.04	о	Алгоритмы и структуры данных
2018	09.03.01	о	Алгоритмы и структуры данных
2018	09.03.02	о	Алгоритмы и структуры данных
2018	09.03.03	о	Алгоритмы и структуры данных

Предлагаемый вариант названия:

Алгоритмы и структуры данных: методические указания по выполнению курсовой работы (проекта)

## Содержание

Введение .....	3
1 Основные этапы выполнения работы (проекта) .....	4
1.1 Описание и анализ поставленной задачи.....	4
1.2 Анализ методов решения задачи .....	4
1.3 Анализ и выбор структур данных.....	5
1.4 Проектирование программного средства (для курсового проекта) .....	5
1.5 Реализация и тестирование программного средства .....	7
1.6 Оценка эффективности .....	7
2 Тематика курсовых работ (проектов).....	9
3 Требования к курсовой работе (проекту) .....	10
3.1 Объем и содержание курсовой работы (проекта) .....	10
3.2 Оформление пояснительной записки .....	11
3.3 Выразительные средства представления алгоритмов .....	11
3.4 Языки, среды программирования и пользовательский интерфейс .....	12
Приложение А (справочное) Примерный перечень тем курсовых работ (проектов) .....	13

## **Введение**

Целью курсовой работы (проекта) по дисциплине «Алгоритмы и структуры данных» является закрепление студентами теоретических знаний о структурах данных и алгоритмах их обработки и приобретение практических навыков обоснованного выбора, эффективной реализации и оценки временной и емкостной сложности.

Основными задачами выполнения курсовой работы (проекта) являются:

- изучение и практическая реализация классических форм организации данных в программах для ЭВМ;
- изучение и практическая реализация алгоритмов обработки данных, применяемых при решении задач различных классов;
- закрепление навыков анализа эффективности алгоритмов по времени и памяти;
- приобретение навыков обоснованного выбора, эффективной реализации, дополнения и модификации структур данных и алгоритмов их обработки.

В процессе выполнения курсовой работы (проекта) студенты также знакомятся на практике с основными этапами решения инженерных задач в области разработки программных систем и принципами их описания.

## **1 Основные этапы выполнения работы (проекта)**

### **1.1 Описание и анализ поставленной задачи**

Во-первых, осуществляется изучение задачи и ее детальное описание.

При этом описание может быть:

- неформальным: на естественном языке;
- формальным: математическая нотация задачи;
- полужформальным: с использованием естественного языка, математических аппарата, диаграмматических конструкций и пр.

Результатом данного этапа должен являться перечень функциональных требований к разрабатываемому программному средству.

### **1.2 Анализ методов решения задачи**

Каждая алгоритмическая задача может быть решена различными методами (с использованием различных подходов и алгоритмов). Целью данного этапа является выбор метода или методов, с помощью которых данная задача будет решена более эффективно. Основными критериями эффективности решения алгоритмической задачи является сложность алгоритма по времени и памяти. При этом на данном этапе достаточно рассматривать асимптотические оценки трудоемкости алгоритмов, которые приводятся практически во всех источниках, приведенных в списке литературы.

С целью анализа приводится укрупненное описание рассматриваемых алгоритмов, оценки их сложности, делаются выводы об их эффективности для решения поставленной задачи. Описание алгоритмов может быть приведено на достаточно абстрактном уровне, так как конкретные структуры данных и специфика алгоритмов при решении данной задачи не определены.

В случае, если алгоритмы, используемые при решении задачи определены заданием на курсовую работу (проект), то осуществляется как их анализ, так и анализ других алгоритмов данного класса и выявляются достоинства и недостатки заданных.

### **1.3 Анализ и выбор структур данных**

На предыдущем этапе алгоритмы могут описываться в терминах абстрактных типов данных. Это означает, что для реализации описанных операторов могут использоваться различные виды организации данных (принципы структурирования). При этом зачастую выбор той или иной структуры данных существенно влияет на эффективность реализации как по времени, так и по памяти.

Таким образом, задача данного этапа заключается в выборе конкретной структуры данных, обеспечивающей эффективную реализацию операторов абстрактного типа данных. Также должен рассматриваться вопрос выбора вида представления структуры данных на физическом уровне (в последовательной или связной памяти) в зависимости от сформулированных на первом этапе функциональных требований к программному средству и с учетом показателей эффективности.

В целях анализа и выбора приводится описание возможных структур на абстрактном и физическом уровне. Затем производится оценка и сравнение эффективности по времени и памяти. Результатом данного этапа является выбор одной или нескольких структур данных и их полное описание, в том числе на логическом уровне.

### **1.4 Проектирование программного средства (для курсового проекта)**

Этап проектирования осуществляется при выполнении курсового проекта и включает в себя две обязательные ступени:

- предварительное проектирование;
- детальное проектирование.

Предварительное проектирование формирует абстракции архитектурного уровня, детальное проектирование уточняет эти абстракции, добавляет подробности алгоритмического уровня. Кроме того, во многих случаях выделяют интерфейсное проектирование, цель которого – сформировать графический интерфейс пользователя.

Предварительное проектирование включает три типа деятельности:

1. Структурирование системы. Система структурируется на несколько подсистем, где под подсистемой понимается независимый программный компонент. Определяются взаимодействия подсистем.
2. Моделирование управления. Определяется модель связей управления между частями системы.
3. Декомпозиция подсистем на модули. Каждая подсистема разбивается на модули, определяются типы модулей и межмодульные связи.

В рамках курсового проекта реализуются простые программные системы, которые в терминах проектирования можно отнести к уровню подсистем. Поэтому на этапе предварительного проектирования в рамках курсового проекта выполняется только декомпозиция программной системы на модули.

При выделении модулей следует руководствоваться принципом информационной закрытости, который утверждает: содержание модулей должно быть скрыто друг от друга. То есть модуль должен определяться и проектироваться так, чтобы его содержимое (процедуры и данные) было недоступно тем модулям, которые не нуждаются в такой информации (клиентам). Для обеспечения таких возможностей система внутренних и внешних связей модуля должна отвечать особым требованиям, описываемым следующими характеристиками:

1. Связность модуля – мера зависимости его частей. Является внутренней характеристикой модуля. Для получения качественного результата проектирования связность модуля должна быть максимальной. Наилучшим случаем является функциональная связность, когда части модуля вместе реализуют одну функцию. Наихудшим случаем является связность по совпадению, когда элементы модуля вообще не имеют никаких отношений друг с другом.
2. Сцепление – мера взаимозависимости модулей по данным. Является внешней характеристикой модуля, которую желательно уменьшать. Наилучшим случаем является сцепление по данным, когда процедуры одного модуля вызываются из другого, причем все входные и выходные параметры вызываемых процедур – простые элементы данных. Наихудшим случаем является сцеп-

ление по содержанию, когда один модуль прямо ссылается на содержание другого модуля (не через его точку входа), например, коды их команд перемежаются друг с другом. Также в большинстве случаев недопустимым является сцепление по общей области, когда модули разделяют одну и ту же глобальную структуру данных.

Результатом предварительного проектирования является модульная структура программной системы, описание модулей (наборы процедур, локальных структур данных, точки входа, внутренние связи) и принципов их взаимодействия.

На основании результатов предварительного проектирования и предыдущих этапов курсового проекта осуществляется детальное проектирование и, при необходимости, проектирование интерфейса. Результатом детального проектирования являются детализированное описание основных структур данных и алгоритмов процедур выделенных ранее модулей (с указанием точек входа, наборов и типов передаваемых и возвращаемых данных). Результатом проектирования интерфейса пользователя является схема его состояний.

### **1.5 Реализация и тестирование программного средства**

На этапе реализации осуществляется выбор инструментальных средств (среды программирования), реализация структур данных на языке программирования, кодирование алгоритмов и проверка функционирования программной системы на тестовых задачах.

### **1.6 Оценка эффективности**

В рамках данной курсовой работы (проекта) осуществляется оценка эффективности основных алгоритмов по времени и, при необходимости, памяти. При этом необходимыми и достаточными считаются асимптотические оценки. Нахождение функций трудоемкости может осуществляться для достаточно простых алгоритмов.

В процессе анализа сложности по времени может использоваться либо выведение асимптотической оценки путем рассмотрения записи алгоритма на языке программирования или псеводокоде, либо экспериментальный анализ,

либо их комбинация. Первый вид анализа допускается не использовать в случае сложных алгоритмов, однако, аналитические оценки их сложности должны быть приведены (со ссылкой на источник). В процессе анализа сложности должен быть описан и обоснован выбор значимой операции или группы операций и выбор конкретных операций в алгоритме. В случае экспериментального анализа должен быть сформулирован и обоснован план эксперимента, описаны входные данные, получаемые результаты, дана их интерпретация. По результатам анализа должны быть сделаны выводы.

Оценка сложности по памяти обычно является тривиальной задачей. Исключение составляют рекурсивные алгоритмы, где необходимо учитывать память, выделяемую для организации рекурсивных вызовов. Сложность заключается в определении зависимости количества рекурсивных вызовов от значащих характеристик соответствующего класса алгоритмов по виду функции трудоемкости, например, в случае количественно-зависимых алгоритмов – от размера входного набора данных. Однако, эта зависимость получается в ходе анализа сложности алгоритма по времени, и получение оценки по памяти заключается лишь в умножении правой части найденной зависимости на объем памяти, необходимой для реализации одного вызова.

Теоретические основы и практические инструменты анализа подробно описаны в учебном пособии.



## **2 Тематика курсовых работ (проектов)**

Задания на курсовую работу в основном базируются на лекционном материале дисциплины «Алгоритмы и структуры данных». Часть тем несколько расширяет границы рассматриваемого на лекциях материала. Тематика также может быть связана с материалом, изучаемым в рамках других дисциплин, предусмотренных учебным планом, предшествующих данной дисциплине. В отдельных случаях возможны темы исследовательского характера.

Выделяются следующие основные направления тематики курсовых работ:

1. Углубленное изучение, реализация и оценка эффективности заданных структур данных и (или) алгоритмов, используемых при решении алгоритмических задач определенного класса.

2. Углубленное изучение, реализация и оценка эффективности заданных структур данных и (или) алгоритмов, реализующих один из известных подходов к решению алгоритмических задач (не обязательно одного класса).

3. Решение практической задачи с выбором структур данных и алгоритмов, обеспечивающих более эффективную реализацию.

4. Решение типовой алгоритмической задачи на основе типовых структур данных и алгоритмов, дополненных и модифицированных соответственно для достижения цели и повышения эффективности.

5. Темы, предложенные студентами или основанные на материале других дисциплин учебного плана специальности, с соответствующим приведением их к одной из указанных выше групп.

Задание на курсовую работу (проект) является индивидуальным. Приблизительный перечень тем курсовых работ приведен в Приложении А.

### **3 Требования к курсовой работе (проекту)**

#### **3.1 Объем и содержание курсовой работы (проекта)**

Общий объем пояснительной записки (без приложений): 20-25 страниц.

Пояснительная записка к курсовой работе должна давать достаточно полное представление обо всех этапах работы, выполненной студентом. Записка иллюстрируется схемами, таблицами, графиками и пр.

Структура пояснительной записки:

1. **Титульный лист**, оформленный в соответствии с установленной формой, размещенной в информационно-коммуникационной среде университета.

2. **Задание на курсовую работу**, оформленное в соответствии с установленной формой, размещенной в информационно-коммуникационной среде университета.

3. **Содержание** включает наименование всех разделов курсовой работы (проекта), а также подразделов и пунктов, если они имеют наименование, с указанием номера страницы, на которой размещается начало материала раздела, подраздела, пункта.

4. **Введение** содержит общую постановку задачи, ее позиционирование в классификации алгоритмических задач, обоснование актуальности и примеры практического применения рассматриваемых структур данных и (или) алгоритмов. Также формулируется цель курсовой работы (проекта) и перечень задач, подлежащих решению в процессе ее выполнения.

5. **Основная часть** состоит из разделов, перечень и порядок следования которых должен приблизительно соответствовать перечню и порядку этапов выполнения курсовой работы (проекта), изложенным в первом разделе настоящих методических указаний. В целях структуризации материала пояснительной записки рекомендуется выделять подразделы и, в случае необходимости, пункты. Наименования структурных элементов основной части должны быть осмысленными и явно характеризовать сущность рассматриваемого в данном разделе (подразделе или пункте) вопроса. Содержание разделов должно

описывать выполненные на соответствующем этапе работы (проекта), четко обозначать полученные результаты.

6. **Заключение** должно содержать краткое описание основных результатов, полученных в процессе работы (проекта), качественные и количественные оценки эффективности алгоритмов и их реализации в программном средстве, выводы по результатам работы (проекта), достоинства и недостатки рассмотренных структур данных и алгоритмов. Если в ходе этапов анализа или проектирования были выбраны неоптимальные структуры и (или) алгоритмы данных или приняты нерациональные технические решения, то следует указать причины, обусловившие это, а также нерешенные вопросы, рекомендации по возможному усовершенствованию программного средства.

7. **Список использованной литературы** содержит перечень источников, использованных при выполнении курсовой работы (проекта). Указывают только те источники, на которые имеются ссылки в тексте пояснительной записки.

8. **Приложения** при необходимости могут содержать вспомогательный материал (листинги программ, инструкции по пользованию программами, примеры экранных форм, результаты экспериментов и т.д.).

### **3.2 Оформление пояснительной записки**

Оформление пояснительной записки должно соответствовать требованиям, установленным ГОСТ 19.701 – 90, ГОСТ 2.105 – 95, ГОСТ 7.1 – 2003.

### **3.3 Выразительные средства представления алгоритмов**

Для представления алгоритмов могут использоваться следующие методы:

- графический (в виде блок-схем или диаграмм состояний в случае событийного управления);
- словесное описание;
- описание на достаточно общем псевдокоде.

Выбор вида описания для каждого конкретного алгоритма или группы алгоритмов осуществляется автором с учетом сложности алгоритма, его специфики и уровня абстракции. При этом на этапах анализа и детального проектирования совпадение видов описания алгоритмов обязательным не является. Однако, в рамках одного этапа требуется единообразие описания.

### **3.4 Языки, среды программирования и пользовательский интерфейс**

Автор курсовой работы (проекта) вправе как на основании некоторых критериев эффективности, так и на основании личных предпочтений, выбрать:

1. Любой императивный язык высокого уровня (ассемблерные вставки допускаются), реализующий как процедурный, так и объектно-ориентированный подход.
2. Любую среду программирования, реализующую выбранный язык, в том числе визуальную.
3. Способ реализации пользовательского интерфейса (в случае его необходимости), процедур ввода и вывода данных. Если другое не установлено заданием на курсовую работу, то единственным требованием является достаточность интерфейса для решения поставленной задачи.

В ходе реализации приложения запрещается использовать встроенные в среду программирования объекты, представляющие собой уже реализованные структуры данных, относящиеся к классу производных (такие как стеки, очереди, списки и т.д.).

## **Приложение А**

### **(справочное)**

#### **Примерный перечень тем курсовых работ (проектов)**

##### **1. Тема курсовой работы (проекта): Алгоритмы сортировки во внешней памяти: прямое и естественное слияние.**

Исходные данные: Файл содержит записи фиксированного размера (200 байт). Одно из полей записи содержит ключ (ключи могут повторяться). Количество записей не допускает их размещение в оперативной памяти.

Задание: Разработать программы, осуществляющие сортировку предложенными методами. Оценить сложность алгоритмов, проверить полученные результаты экспериментально.

##### **2. Тема курсовой работы (проекта): Алгоритмы сортировки во внешней памяти: многофазное слияние.**

Исходные данные: Файл содержит записи фиксированного размера (200 байт). Одно из полей записи содержит ключ (ключи могут повторяться). Количество записей не допускает их размещение в оперативной памяти. Допускается использование начальных отрезков фиксированного размера.

Задание: Разработать программу, осуществляющую сортировку многофазным слиянием. Оценить эффективность алгоритма на одном случайном наборе входных данных в зависимости от количества путей слияния и размера начальных отрезков.

##### **3. Тема курсовой работы (проекта): Алгоритмы сортировки во внешней памяти: формирование начальных отрезков.**

Исходные данные: Файл содержит записи фиксированного размера (200 байт). Одно из полей записи содержит ключ (ключи могут повторяться). Количество записей не допускает их размещение в оперативной памяти.

Задание: Реализовать процедуры формирования начальных отрезков и программу внешней сортировки сбалансированным многопутевым слиянием. Оценить эффективность алгоритма при различных стратегиях формирования начальных отрезков (без предварительной подготовки, сортировкой методом

разделения и просеиванием через пирамиду). Сравнить вычислительную сложность алгоритмов подготовки начальных отрезков с получаемым эффектом при сортировке.

**4. Тема курсовой работы (проекта): Алгоритмы сортировки во внутренней памяти: сортировка стопками, быстрая и пирамидальная сортировка.**

Исходные данные: Структура данных содержит записи фиксированного размера (200 байт). Одно из полей записи содержит ключ (ключи могут повторяться). Количество записей допускает их размещение в оперативной памяти.

Задание: Разработать программы, осуществляющие сортировку предложенными методами. Оценить сложность алгоритмов по количеству операций сравнения и перестановки записей, проверить полученные результаты экспериментально.

**5. Тема курсовой работы (проекта): Алгоритмы сортировки за линейное время.**

Исходные данные: Структура данных содержит записи фиксированного размера (200 байт). Одно из полей записи содержит ключ (ключи могут повторяться). Количество записей допускает их размещение в оперативной памяти.

Задание: Разработать программы, осуществляющие сортировку следующими методами: сортировка подсчетом, сортировка стопками, цифровая сортировка, сортировка вычерпыванием. Оценить сложность алгоритмов по количеству операций перестановки записей, проверить полученные результаты экспериментально.

**6. Тема курсовой работы (проекта): Порядковые статистики.**

Исходные данные: Структура данных содержит записи фиксированного размера (200 байт). Одно из полей записи содержит ключ (ключи могут повторяться). Количество записей допускает их размещение в оперативной памяти.

Задание: Рассмотреть алгоритмы нахождения  $i$ -ой порядковой статистики, оценить их эффективность. Разработать программы, реализующие решение задачи на основе рассмотренных алгоритмов. Оценить сложность алгоритмов.

**7. Тема курсовой работы (проекта): Задача о расстановке ферзей: рекурсивный алгоритм и вероятностный алгоритм Лас Вегаса.**

Исходные данные: Восемь ферзей необходимо расставить на шахматной доске так, чтобы ни один из них не угрожал другому.

Задание: Разработать программы, решающие данную задачу на основе предложенных алгоритмов. Провести статистический анализ работы (проекта) вероятностного алгоритма и сравнить его сложность со сложностью рекурсивного алгоритма.

**8. Тема курсовой работы (проекта): Алгоритм поиска по АВЛ сбалансированному дереву.**

Исходные данные: В узле дерева храниться запись фиксированного размера (200 байт). Одно из полей записи содержит ключ (ключи не повторяются). Количество записей допускает их размещение в оперативной памяти.

Задание: Реализовать процедуры поиска, вставки и удаления записей из АВЛ сбалансированного дерева. Произвести оценку сложности данных операций. Сравнить полученные результаты с поиском по 2-3 сбалансированному дереву (тема №9), сделать выводы об эффективности данных структур данных для решения различных задач поиска.

**9. Тема курсовой работы (проекта): Алгоритмы поиска по 2-3 сбалансированному дереву.**

Исходные данные: В узле дерева храниться запись фиксированного размера (200 байт). Одно из полей записи содержит ключ (ключи не повторяются). Количество записей допускает их размещение в оперативной памяти.

Задание: Реализовать процедуры поиска, вставки и удаления записей из 2-3 сбалансированного дерева. Произвести оценку сложности данных операций. Сравнить полученные результаты с поиском по АВЛ сбалансированному дереву (тема №8), сделать выводы об эффективности данных структур данных для решения различных задач поиска.

**10. Тема курсовой работы (проекта): Реализация сканера транслятора на основе оптимального и полностью сбалансированного деревьев поиска.**

Исходные данные: Программа написана на языке программирования высокого уровня (язык программирования может быть выбран произвольно). Язык программирования определяет набор ключевых (зарезервированных) слов, которые необходимо выделять из текста программы.

Задание: На основе алгоритма поиска по полностью сбалансированному дереву разработать программу, анализирующую тексты программ с целью накопления статистической информации о частотах использования ключевых слов. Данные анализа сохранять в файле. На основе алгоритма поиска по оптимальному дереву разработать программу сканера транслятора, определяющего, относится ли каждое слово (идентификатор) к ключевым словам. Сравнить сложность этих алгоритмов и процедур построения соответствующих деревьев.

**11. Тема курсовой работы (проекта): Задачи планирования передачи данных в компьютерной сети.**

Исходные данные: Структура компьютерной сети задается пользователем: количество узловых точек, наличие и пропускная способность каналов связи. Возможны ситуации добавления, и удаления каналов передачи данных и узлов (моделируется выход из строя одного или нескольких объектов и их добавление).

Задание: Разработать программу, решающую задачи определения оптимальных маршрутов передачи информации из произвольной узловой точки во все остальные, в одну или несколько точек. При выходе из строя одного из элементов сети программа должна проверять ее целостность.

**12. Тема курсовой работы (проекта): Алгоритмы сравнения с образцом.**

Исходные данные: Текстовые файлы, не содержащие форматирования.



Задание: Проанализировать алгоритмы поиска сравнением с образцом. Разработать программы, реализующие эти алгоритмы. Оценить и сравнить сложность реализованных алгоритмов.

### **13. Тема курсовой работы (проекта): Проверка орфографии текстовых документов приблизительным сравнением с образцом.**

Исходные данные: Текстовые файлы, не содержащие форматирования.

Задание: Разработать программу, осуществляющую накопление слов в словаре и производящую проверку орфографии текста, вводимого с клавиатуры или из файла, с выдачей вариантов правильного написания на основе алгоритма приблизительного сравнения с образцом (алгоритм нечеткого поиска Ландау-Вишкина).

### **14. Тема курсовой работы (проекта): Численные вероятностные алгоритмы.**

Исходные данные: 1) Задача вычисления площади фигуры, ограниченной графиком произвольной полиномиальной функции и прямыми  $x=a$  и  $x=b$ . 2) Задача о совпадающих днях рождения.

Задание: Решить задачи вероятностными численными методами. Оценить точность и сложность алгоритма вычисления площади по сравнению с методами численного интегрирования.

### **15. Тема курсовой работы (проекта): Динамическое программирование.**

Исходные данные: 1) Задачи повышения эффективности рекурсивных алгоритмов на примере подсчета чисел Фибоначчи и вычисления биномиальных коэффициентов. 2) Динамическое умножение матриц.

Задание: 1) Разработать нерекурсивные алгоритмы решения предложенных задач, сравнить их эффективность с альтернативными рекурсивными алгоритмами. 2) Разработать алгоритм динамического перемножения матриц, определяющий оптимальную последовательность перемножения  $N$  матриц. Для пе-

умножения матриц использовать алгоритм делающий менее  $N^3$  операций умножения и менее  $N^3 - N^2$  операций сложения.

**16. Тема курсовой работы (проекта): Динамические множества с сохранением предыдущих версий.**

Исходные данные: Множество представляется AVL-сбалансированным деревом поиска деревом. Добавление и удаление элементов множества (в общем случае нескольких) осуществляется пользователем.

Задание: Разработать алгоритм построения множества с возможностью сохранения предыдущих версий, минимизирующий затраты памяти. Сравнить сложность алгоритма по памяти и времени с методом полного копирования множества при его изменении.

**17. Тема курсовой работы (проекта): Пополнение структур данных: динамические порядковые статистики.**

Исходные данные: Данные представлены красно-черным деревом.

Задание: Пополнить структуру данных типа красно-черное дерево с целью решения задач определения  $i$ -ой порядковой статистики и определения порядкового номера заданного элемента.

**18. Тема курсовой работы (проекта): Пополнение структур данных: деревья промежутков.**

Исходные данные: В структуре данных храниться информация о протяженных во времени событиях в виде промежутков времени, которые они занимают.

Задание: На основе пополнения структуры данных типа красно-черное дерево разработать алгоритм определения всех событий, которые пересекаются во времени с заданным промежутком.

**19. Тема курсовой работы (проекта): Ориентированные ациклические графы. Топологическая сортировка.**

Исходные данные: Перечень дисциплин, изучаемых в ВУЗе на определенной специальности, с указанием связей, обеспечивающих последовательное

изложение материала, и длительности дисциплины в семестрах (данные задаются пользователем).

Задание: Разработать программу, осуществляющую формирование списка дисциплин, изучаемых в каждом семестре. При вводе данных осуществлять проверку ацикличности графа. Оценить сложность использованных алгоритмов.

## **20. Тема курсовой работы (проекта): Алгоритмы вычислительной геометрии: поиск пересекающихся отрезков.**

Исходные данные: Точками в декартовой системе координат пользователем задается  $N$  отрезков.

Задание: Разработать программу определения факта пересечения двух отрезков из заданного множества методом движущейся прямой. Оценить сложность алгоритма.

## **21. Тема курсовой работы (проекта): Алгоритмы вычислительной геометрии: построение выпуклой оболочки.**

Исходные данные: Пользователем задается  $N$  точек в декартовой системе координат.

Задание: Разработать программу построения выпуклой оболочки на заданном множестве точек. Оценить сложность алгоритма.

## **22. Тема курсовой работы (проекта): Приближенные алгоритмы решения недетерминированных полиномиальных задач.**

Исходные данные: 1) Задача о сумме элементов подмножества. 2) Задача о раскраске графа.

Задание: Разработать программы решения задач на основе приближенных полиномиальных алгоритмов. Оценить сложность алгоритмов и точность по сравнению с детерминированными алгоритмами.

## **23. Тема курсовой работы (проекта): Приоритетные очереди: возможные реализации и сравнение подходов.**

Исходные данные: Поток заявок на обслуживание, характеризующихся длительностью обслуживания. Наивысший приоритет имеют те заявки, которые требуют наименьшего времени на обслуживании.

Задание: Рассмотреть подходы к построению и поддержанию приоритетных очередей, оценить их эффективность. На основе анализа выбрать структуру данных для реализации приоритетной очереди. Реализовать программу, моделирующую работу обслуживающего устройства. Произвести экспериментальную оценку сложности по времени.

#### **24. Тема курсовой работы (проекта): Цифровой (политерный) поиск.**

Исходные данные: Большое число словарных статей (например, статьи толкового словаря русского языка).

Задание: Проанализировать задачу поиска словарной статьи и возможные методы ее решения. На основе метода цифрового поиска реализовать возможности вставки, хранения, и поиска словарных статей.