



Факультет	Математики, физики и информатики	
Кафедра	Информатики и информационных технологий	
Направление подготовки	02.03.02 Фундаментальная информатика и информационные технологии	
Профиль	Открытые информационные системы	
Алгоритмы и анализ сложности		Б1.В.ОД.6

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тульский государственный педагогический университет им. Л. Н. Толстого»  
ФГБОУ ВО «ТГПУ им.Л.Н.Толстого»

УТВЕРЖДЕНА

на заседании Ученого совета университета

Протокол № 2

«11» февраля 2016 г.


## Учебная программа дисциплины «Алгоритмы и анализ сложности»

**Трудоемкость: 3 зачетные единицы**

**Квалификация (степень) выпускника: бакалавр**

**Форма обучения: очная**

Рассмотрена на заседании кафедры информатики и информационных технологий  
протокол № 3 от 18 ноября 2015 г.

Заведующий кафедрой  А.В.Якушин

Одобрена на заседании Ученого совета факультета  
математики, физики и информатики  
протокол № 5 от «17» декабря 2015 г.

Декан  И.Ю. Реброва

**СОДЕРЖАНИЕ**

1. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ.....	3
2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП.....	3
3. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ.....	4
4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ, СТРУКТУРИРОВАННОЕ ПО ТЕМАМ, С УКАЗАНИЕМ ОТВЕДЕННОГО НА НИХ КОЛИЧЕСТВА АКАДЕМИЧЕСКИХ ЧАСОВ И ВИДОВ УЧЕБНЫХ ЗАНЯТИЙ.....	4
5. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ «АЛГОРИТМЫ И АНАЛИЗ СЛОЖНОСТИ»...	5
6. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ .....	6
6.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы.....	6
6.2. Описание показателей, критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания .....	6
Типовые контрольные задания и иные материалы, характеризующие этапы формирования компетенций в процессе освоения образовательной программы.....	7
6.4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и/или опыта деятельности, характеризующие этапы формирования компетенций .....	11
7. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ .....	14
7.1 Основная литература: .....	14
7.2 Дополнительная литература:.....	14
8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ .....	14
9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ .	14
10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ, ВКЛЮЧАЯ ПЕРЕЧЕНЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ.....	15
11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ «АЛГОРИТМЫ И АНАЛИЗ СЛОЖНОСТИ».....	15
12. АННОТАЦИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ «АЛГОРИТМЫ И АНАЛИЗ СЛОЖНОСТИ».....	16
13. ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ К РАБОЧЕЙ ПРОГРАММЕ ДИСЦИПЛИНЫ «АЛГОРИТМЫ И АНАЛИЗ СЛОЖНОСТИ» .....	17

## 1. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Достижение планируемых результатов обучения, соотнесенных с общими целями и задачами ОПОП, является целью освоения дисциплины (модуля).

Планируемые результаты освоения образовательной программы (код и название компетенции)	Планируемые результаты обучения	Этапы формирования компетенции в процессе освоения образовательной программы
<p>способность применять в профессиональной деятельности современные языки программирования и языки баз данных, методологии системной инженерии, системы автоматизации проектирования, электронные библиотеки и коллекции, сетевые технологии, библиотеки и пакеты программ, современные профессиональные стандарты информационных технологий (ОПК-2)</p>	<p><b>Выпускник знает:</b></p> <ul style="list-style-type: none"> <li>• способы программирования нелинейных структур данных и их представление в памяти компьютера;</li> <li>• постановку и алгоритмы задач поиска и сортировки в массивах, поиска на графах (Прима-Краскала, Дейкстры и т.д.);</li> <li>• теоретические основы и приемы программирования перебора с возвратом;</li> </ul> <p><b>Умеет:</b></p> <ul style="list-style-type: none"> <li>• при решении конкретной задачи профессионально грамотно сформулировать задачу программирования, составить и оценить алгоритм решения, реализовать его в данной языковой среде, выполнить необходимое тестирование или верификацию построенной программы</li> </ul> <p><b>Владеет и (или) имеет опыт деятельности</b></p> <ul style="list-style-type: none"> <li>• в области практического программирования конкретных задач из различных предметных областей в определенной языковой среде</li> </ul>	<p style="text-align: center;">3 этап из 3 (8 семестр)</p>

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Алгоритмы и анализ сложности» относится к обязательным дисциплинам вариативной части дисциплин направления. Изучение данной дисциплины базируется на освоении студентами курсов «Основы алгоритмизации», «Вводный курс программирования», «Основы программирования», «Языки программирования», «Математическая логика и теория алгоритмов», «Дискретная математика».

К началу изучения дисциплины студенты должны знать структуру программы в языке C(C++), организацию ввода и вывода данных различного типа на экран и в поток, операцию присваивания значений, программирование основных алгоритмических конструкций и функций пользователя. Очень важно владение аппаратом указателей на уровне работы с массивами, строками и функциями пользователя. Студенты должны знать алгоритмы работы машин Тьюринга, доказывать примитивную рекурсивность функций, использовать графы для моделирования и решения задач в различных областях математики.

Освоение данной дисциплины необходимо для выработки профессионального стиля программирования, изучения основ и принципов объектно-ориентированного и логического программирования, прохождения преддипломной практики, выполнения выпускной квалификационной

работы.

### 3. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ

Вид учебной работы	Объем часов/ зачетных единиц по формам обучения
	очная
<b>Максимальная учебная нагрузка (всего)</b>	<b>108/3</b>
<b>Контактная работа обучающихся с преподавателем (всего)</b>	<b>44</b>
в том числе:	
лекции с применением мультимедийных технологий и раздаточным материалом для студентов	16
лабораторные занятия с использованием современных информационных технологий по разработке алгоритмов и программ	26
контрольные работы	2
<b>Самостоятельная работа студента (всего)</b>	<b>64</b>
в том числе:	
внеаудиторная самостоятельная работа при подготовке к лабораторным занятиям	40
подготовка к контрольной работе	4
решение задач повышенной сложности	10
Выполнение заданий для самостоятельной работы в модульной объектно-ориентированной динамической учебной среде Moodle	10
<i>Промежуточная аттестация в форме: зачета</i>	

### 4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ, СТРУКТУРИРОВАННОЕ ПО ТЕМАМ, С УКАЗАНИЕМ ОТВЕДЕННОГО НА НИХ КОЛИЧЕСТВА АКАДЕМИЧЕСКИХ ЧАСОВ И ВИДОВ УЧЕБНЫХ ЗАНЯТИЙ

Наименование тем (разделов).	Количество академических или астрономических часов по видам учебных занятий			
	Занятия лекционного типа	Занятия лабораторного типа	Другие виды учебных занятий	Самостоятельная работа обучающихся
Раздел 1. Структуры данных				
Тема 1.1. Программирование структур	4	6		8
Тема 1.2. Алгоритмы сортировки	4	6		12
Тема 1.3. Графы. Задачи поиска.	4	6		16
Раздел 2. Алгоритмы обработки данных				
Тема 2.1. Перебор с возвратом.	2	6		10
Тема 2.2. Важные алгоритмы	2	2		10
Контроль самостоятельной работы студентов			2	
Подготовка к зачету				8
<b>ИТОГО</b>	<b>16</b>	<b>26</b>	<b>4</b>	<b>64</b>

**Раздел 1. Структуры данных**

**Тема 1.1. Программирование структур.** Способы представления структур данных. Массивы. Списки. Множества. Стеки. Очереди. Классы памяти и организация программ. Локализация объектов. Глобальные объекты. Динамическая память. Внешние объекты. Деревья. Определение дерева. Корень, узлы. Трие-деревья. Прохождение деревьев.

**Тема 1.2. Алгоритмы сортировки.** Внутренние сортировки. Сортировка в массивах. Обобщение известных методов сортировки вставками, обменом, выбором. Сортировка элементов массива методом подсчета. Анализ алгоритмов сортировок массивов. Быстрая сортировка. Бинарная пирамидальная сортировка. Анализ эффективности алгоритмов. Внешние сортировки. Простое слияние. Естественное слияние. Улучшенные методы сортировки: многофазная и каскадная сортировки. Быстрый поиск: бинарный и последовательный поиски в массивах. Дихотомия.

**Тема 1.3. Графы. Задачи поиска.** Графы. Понятие графа. Представление графа в памяти компьютера. Поиск в графе. Поиск в ширину. Поиск в глубину. Деревья. Нахождение каркаса минимального веса. Задача Прима-Краскала. Поиск кратчайшего пути на графе. Алгоритм Дейкстры. Методы поиска на графах. Определение остовных деревьев.

**Раздел 2. Алгоритмы обработки данных**

**Тема 2.1. Перебор с возвратом.** Общая схема. Пример задачи о расстановке ферзей. Динамическое программирование. Примеры задач (треугольник, степень числа). Метод ветвей и границ. Метод решета.

**Тема 2.2. Важные алгоритмы.** Жадные алгоритмы. В-деревья. Хеширование. Теория сложности алгоритмов: NP-полные и NP-трудные задачи.

**5. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ  
САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ  
«АЛГОРИТМЫ И АНАЛИЗ СЛОЖНОСТИ»**

- 1) Методическая система, используемая авторами данной рабочей программы, базируется на оптимальном сочетании активных форм и методов организации учебной деятельности студентов (лекция, беседа, анализ, синтез, мозговой штурм и т.п.), приемов групповой (выполнение и защита заданий по разработке алгоритмов и программ) и самостоятельной работы (разработка и защита программ).
- 2) Все студенты являются активными пользователями ресурса системы LMS MOODLE, поскольку в нем представлены конспекты всех лекций и методические разработки к проведению каждого практического и лабораторного занятия.
- 3) В течение всего периода обучения организуется регулярная проверка и учет выполнения домашних заданий, размещенных в LMS MOODLE.
- 4) Промежуточная аттестация принимается в форме экзамена по заранее определенному перечню вопросов с обязательным самостоятельным написанием кодов базовых алгоритмов и функций, разобранных и протестированных на занятиях любого вида в течение семестра.
- 5) Ресурс LMS MOODLE поддерживается в актуальном состоянии.
- 6) Одной из важнейших задач методического обеспечения учебной деятельности студента является формирование устойчивого навыка разработки алгоритмов и работы в среде программирования посредством использования ресурсов удаленного рабочего стола, расположенного на сервере кафедры, а также контроль умений студентов читать, анализировать и использовать в работе учебную и специальную литературу.
- 7) По дисциплине используется рейтинг.

## 6. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

### 6.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы представлен в таблице пункта 1 рабочей программы.

Формирование компетенции «Способность применять в профессиональной деятельности современные языки программирования и языки баз данных, методологии системной инженерии, системы автоматизации проектирования, электронные библиотеки и коллекции, сетевые технологии, библиотеки и пакеты программ, современные профессиональные стандарты информационных технологий» (ОПК-2) осуществляется в течение трех этапов освоения основной профессиональной образовательной программы.

Первый этап формирования компетенции осуществляется в процессе освоения дисциплины «Основы алгоритмизации». Второй этап формирования компетенции осуществляется в процессе освоения дисциплины «Математическая логика и теория алгоритмов» и одной из дисциплин по выбору «Технологии программирования» или «Экономическая статистика». Третий этап формирования компетенции осуществляется в процессе освоения дисциплины «Алгоритмы и анализ сложности».

### 6.2. Описание показателей, критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Дескриптор компетенций	Показатели оценивания	Критерии оценивания
Знания	способы программирования нелинейных структур данных и их представление в памяти компьютера; постановку и алгоритмы задач поиска и сортировки в массивах, поиска на графах (Прима-Краскала, Дейкстры и т.д.); теоретические основы и приемы программирования перебора с возвратом;	Отметка «зачтено» выставляется, если студент в целом за семестр набрал от 41 до 100 баллов (с учетом баллов, набранных на промежуточной аттестации (зачете)).  Отметка «не зачтено» выставляется, если студент в целом за семестр набрал менее 41 балла (с учетом баллов, набранных на промежуточной аттестации (зачете)).
Умения	при решении конкретной задачи профессионально грамотно сформулировать задачу программирования, составить и оценить алгоритм решения, реализовать его в данной языковой среде, выполнить необходимое тестирование или верификацию построенной программы	
Навыки и опыт деятельности	навыками: практического программирования конкретных задач из различных предметных областей в определенной языковой среде	

Критерии оценивания компетенций формируются на основе балльно-рейтинговой системы с помощью всего комплекса методических материалов, определяющих процедуры оценивания знаний, умений, навыков и опыта деятельности, характеризующих данный этап формирования компетенций.

Баллы, набранные	Баллы за промежу-	Общая сумма бал-	Отметка
г. Тула			Стр. 6 из 18

студентом в течение семестра	точную аттестацию (зачет)	лов за модуль в семестр	
11 – 80	0 – 20	41-100	Зачтено
0 – 20	0 – 20	0 – 40	Не зачтено

Знания, умения, навыки и компетенции студентов по дисциплине оцениваются по двухбалльной шкале с отметками: «зачтено»; «не зачтено». Как правило, при двухбалльной системе преподавателями используются следующие показатели, при условии успешного прохождения текущего контроля успеваемости по дисциплине:

Оценка «зачтено» ставится, если студент освоил программный материал всех разделов, последователен в изложении программного материала, достаточно последовательно и логически стройно его излагает, умеет увязывать теорию с практикой, успешно прошел текущий контроль успеваемости по дисциплине, продемонстрировал индивидуальные знания, умениями и навыки практической работы.

Оценка «не зачтено» ставится, если студент не знает значительной части программного материала, допускает существенные ошибки, непоследователен в его изложении, не прошел текущий контроль успеваемости, не в полной мере владеет необходимыми знаниями, умениями и навыками при выполнении практических заданий, то есть студент не может продолжить обучение без дополнительной подготовки по соответствующей дисциплине.

### **Типовые контрольные задания и иные материалы, характеризующие этапы формирования компетенций в процессе освоения образовательной программы**

#### **Вопросы к зачету**

1. Способы представления структур данных. Массивы.
2. Способы представления структур данных. Списки.
3. Способы представления структур данных. Множества.
4. Способы представления структур данных. Стеки.
5. Способы представления структур данных. Очереди.
6. Способы представления структур данных. Деревья: определение, составляющие, виды, способы представления в памяти компьютера.
7. Способы представления структур данных. Деревья: определение, прохождение деревьев, внутренние и внешние итераторы, порядки обхода.
8. Способы представления структур данных. Деревья: определение, обходы деревьев, бинарные деревья поиска.
9. Способы представления структур данных. Графы: определение, составляющие, виды, способы представления в памяти компьютера.
10. Способы представления структур данных. Графы: определение, обходы в графах.
11. Способы представления структур данных. Графы: определение, поиск кратчайших путей.
12. Способы представления структур данных. Графы: определение, нахождение остовных деревьев.
13. Алгоритмы компьютерной обработки данных. Перебор с возвратом: понятие метода, общая схема решения задачи о расстановке ферзей.
14. Алгоритмы компьютерной обработки данных. Перебор с возвратом: понятие метода, вспомогательные массивы и функции для решения задачи о расстановке ферзей.
15. Алгоритмы компьютерной обработки данных. Перебор с возвратом: метод ветвей и границ.
16. Алгоритмы компьютерной обработки данных. Перебор с возвратом: метод решета.

17. Алгоритмы компьютерной обработки данных. Внутренние сортировки: постановка задачи, сортировка простыми обментами, сортировка простым выбором.
18. Алгоритмы компьютерной обработки данных. Внутренние сортировки: постановка задачи, сортировка простыми включениями.
19. Алгоритмы компьютерной обработки данных. Внутренние сортировки: постановка задачи, сортировка простыми слияниями.
20. Алгоритмы компьютерной обработки данных. Внутренние сортировки: постановка задачи, сортировка естественными слияниями.
21. Алгоритмы компьютерной обработки данных. Внутренние сортировки: постановка задачи, быстрая сортировка.
22. Алгоритмы компьютерной обработки данных. Внутренние сортировки: постановка задачи, бинарная пирамидальная сортировка.
23. Алгоритмы компьютерной обработки данных. Внешние сортировки: постановка задачи, метод слияний, простое слияние.
24. Алгоритмы компьютерной обработки данных. Внешние сортировки: постановка задачи, метод слияний, естественное слияние.
25. Алгоритмы компьютерной обработки данных. Внешние сортировки: постановка задачи, метод слияний, сортировка методом поглощения.
26. Алгоритмы компьютерной обработки данных. Алгоритмы поиска: постановка задачи, поиск элемента с заданным свойством (задачи 1 и 2).
27. Алгоритмы компьютерной обработки данных. Алгоритмы поиска: постановка задачи, поиск элемента с заданным свойством (задачи 3 и 4).
28. Алгоритмы компьютерной обработки данных. Алгоритмы поиска: хэширование: постановка вопроса, понятие о хэшировании, хэш-функции, коллизии.
29. Алгоритмы компьютерной обработки данных. Алгоритмы поиска: хэширование: постановка вопроса, общие понятия, универсальные функции расстановки.
30. Алгоритмы компьютерной обработки данных. Алгоритмы поиска: хэширование: постановка вопроса, общие понятия, методы разрешения коллизий.
31. Алгоритмы компьютерной обработки данных. В-деревья.
32. Алгоритмы компьютерной обработки данных. Жадные алгоритмы.

### Для реализации «жадного» алгоритма разбирается задача «РЕКЛАМА».

В супермаркете решили время от времени транслировать рекламу новых товаров. Для того, чтобы составить оптимальное расписание трансляции рекламы, руководство супермаркета провело следующее исследование: в течение дня для каждого покупателя, посетившего супермаркет, было зафиксировано время, когда он пришел в супермаркет, и когда он из него ушел.

Менеджер по рекламе предположил, что такое расписание прихода-ухода покупателей сохранится и в последующие дни. Он хочет составить расписание трансляции рекламных роликов, чтобы каждый покупатель услышал не меньше двух рекламных объявлений. В то же время, он выдвинул условие, чтобы два рекламных объявления не транслировались одновременно, и, поскольку продавцам все время приходится выслушивать эту рекламу, общее число рекламных объявлений за день должно быть минимальным.

**Требуется** написать программу, которая при заданных исходных данных составит такое расписание трансляции рекламных роликов. Рекламные объявления можно начинать транслировать только в целые моменты времени. Считается, что каждое рекламное объявление заканчивается до наступления следующего целого момента времени. Если рекламное объявление транслируется в тот момент времени, когда покупатель входит в супермаркет или уходит из него, покупатель это объявление услышать успеет.

### Технические требования:

Имя входного файла: *INPUT.TXT*

Имя выходного файла: *OUTPUT.TXT*



Ограничение по времени тестирования: 5 секунд на один тест.

Ограничение на объем используемой памяти: 4 мегабайта.

**Формат входных данных:**

Входной файл *INPUT.TXT* содержит сначала целое число  $N$  – количество покупателей, посетивших супермаркет за день ( $1 \leq N \leq 3000$ ). Затем идет  $N$  пар натуральных чисел  $A_i, B_i$ , задающих соответственно время прихода и время ухода покупателей из супермаркета ( $0 < A_i < B_i < 10^6$ ).

**Формат выходных данных:**

Выходной файл *OUTPUT.TXT* должен содержать в первой строке количество рекламных объявлений, которое будет транслироваться за день. Во второй строке в возрастающем порядке содержатся моменты времени, в которые нужно транслировать рекламные объявления. Если решений несколько, вывести любое из них.

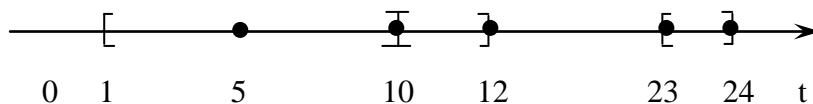
**Пример файлов входных и выходных данных:**

<i>INPUT.TXT</i>	<i>OUTPUT.TXT</i>
5	5
1 10	5 10 12 23 24
10 12	
1 10	
1 10	
23 24	

**Указания к решению.**

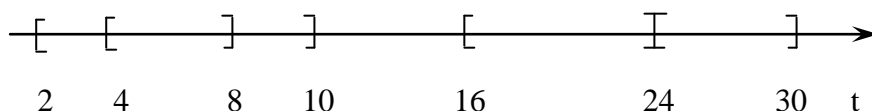
Отобразим все время работы магазина временной осью, а время прихода и ухода покупателей – отрезками на этой оси. При такой интерпретации задача формулируется следующим образом: поставить на оси минимальное количество точек с целочисленными координатами (моменты начала трансляции очередного ролика) так, чтобы в каждом отрезке (для каждого покупателя во время его нахождения в магазине) содержалось не менее двух точек (два рекламных ролика).

Приведенный пример можно изобразить так:

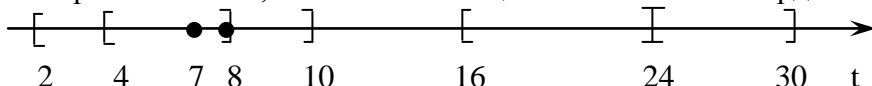


Для решения задачи необходимо отсортировать все отрезки по возрастанию правых границ. В случае если правые границы отрезков совпадают, сортируем по убыванию координат левых границ. Ограничение задачи (количество отрезков  $\leq 3000$ ) позволяет применить любой из известных алгоритмов сортировки без учета трудоемкости (простыми обменов, простым выбором, быструю и т.п.).

Рассмотрим следующий пример. Пусть дано четыре отсортированных отрезка: [2, 8], [4, 10], [16, 24], [24, 30].



В отрезке [2, 8] должны содержаться две точки, которые необходимо поставить как можно правее, т.к. раньше данного отрезка другие отрезки не могли закончиться, однако могли начаться другие отрезки. Значит, ставим точки с целочисленными координатами 7 и 8.



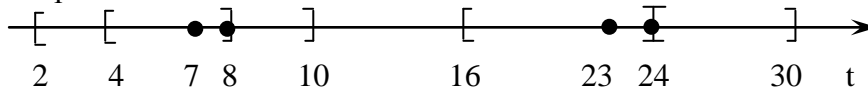
Отметим, что поставленные две точки сразу попали и в отрезок  $[4, 10]$ , при том, что если бы были проставлены точки, например, в координатах 1 и 3, то такая расстановка была бы неэффективной – мы покрыли бы ей только один отрезок, а не два.

Пусть точка 8 называется последней расставленной точкой (pt), а точка 7 – предпоследней (ppt). Если  $r$  – это правая граница текущего отрезка, то координаты последней и предпоследней расставленной точек вычисляются следующим образом:

$pt := r$ ;  $ppt := r - 1$ ;

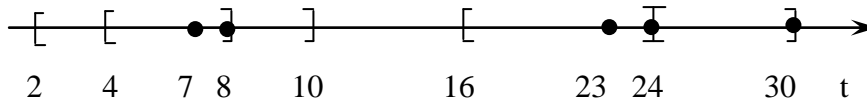
Внутри следующего отрезка  $[4, 10]$  уже находятся две точки, т.к. левая граница отрезка ( $le$ ) меньше, чем предпоследняя расставленная точка. Таким образом, ничего расставлять не нужно, если  $le \leq ppt$ .

В отрезке  $[16, 24]$  не стоит еще ни одной точки, так как  $le > pt$ . Ставим в нем две точки как можно правее.



В последнем отрезке  $[24, 30]$  уже содержится одна из поставленных точек ( $pt = le$ ). Еще одну точку поставим в правую границу отрезка, т.е. в точку с координатой 30. В результате предыдущей точкой станет точка 24:

$ppt := pt$ ;  $pt := r$ ;



Выполненные действия можно описать с помощью следующего алгоритма:

1. определить, сколько точек ( $h$ ) нужно поставить в текущий отрезок;
2. если  $h=2$ , то ставим точки как можно правее (граница и граница-1), иначе, если  $h=1$ , то ставим точку в правую границу отрезка.

## КОНТРОЛЬНАЯ РАБОТА

### 1 вариант

1. Опишите функцию, реализующую сортировку вставкой и сдвигом по убыванию для одномерного массива из 20 целых чисел.
2. Опишите функцию вычисления суммы элементов квадратной матрицы, расположенных под побочной и под главной диагоналями (элементы, расположенные на самих диагоналях, включить в сумму).
3. Опишите программу, определяющую, является ли заданное натуральное  $n$  число числом Армстронга (число Армстронга равно сумме своих цифр, каждая из которых взята в степени, соответствующей количеству цифр. Например,  $153 = 1^3 + 5^3 + 3^3$ ).
4. Вставьте пропущенные понятия:

Простейший  представляет собой линейную последовательность элементов. Для каждого из них, кроме последнего, имеется  элемент, и для каждого, кроме первого – .

– это непустое множество элементов , в котором выделен один элемент, называемый , а все остальные элементы разбиты на несколько непересекающихся подмножеств, называемых  исходного .

Трудоёмкость медленных алгоритмов сортировок составляет  операций сравнений и обменов.

предполагает поиск одной или нескольких подзадач, которые могут быть решены непосредственно, без рекурсивных вызовов.

**2 вариант**

1. Опишите функцию, реализующую быструю сортировку по убыванию одномерного массива из 20 целых чисел.
2. Опишите функцию вычисления суммы элементов квадратной матрицы, расположенных над главной и под побочной диагоналями (элементы, расположенные на самих диагоналях, включить в сумму).
3. Опишите программу, определяющую, является ли заданное натуральное  $n$  число Мерсена (число Мерсена, являясь простым, представимо в виде  $2^p - 1$ , где  $p$  – простое).
4. Вставьте пропущенные понятия:

Математическим понятием, которое привело к появлению в языках программирования понятия «массив», является [ ] и частные случаи этого понятия: [ ] или [ ].

[ ] – это тип структуры данных, организованной по принципу «последним пришел – первым ушел».

[ ] – это максимальное число узлов, которое может встретиться на пути из корня дерева в некоторый узел, при условии, что этот путь проходит только по связанным между собой узлам и никогда не проходит дважды через один и тот же узел.

Сортировка слияниями и [ ] сортировка имеют трудоемкость [ ] операций сравнений и обменов.

Вычисления по рекурсивным функциям носят название [ ].

**3 вариант**

1. Опишите функцию, реализующую бинарную пирамидальную сортировку по убыванию одномерного целочисленного массива из 20 элементов.
2. Опишите функцию вычисления суммы элементов квадратной матрицы, расположенных над побочной и под главной диагоналями (элементы, расположенные на самих диагоналях, включить в сумму).
3. Опишите программу, определяющую, является ли заданное натуральное число  $n$  автоморфным (автоморфным называется натуральное число, записью которого заканчивается его квадрат).
4. Вставьте пропущенные понятия:

[ ] – это тип структуры данных, организованной по принципу «первым пришел – первым ушел».

Если сложный объект состоит из некоторого множества элементов одного и того же типа, между которыми существуют определенные связи (отношения), то такой объект в технике принято называть [ ]. Для моделирования [ ] в математике служат объекты, называемые [ ].

Быстрая сортировка в худшем случае имеет трудоемкость [ ] операций сравнений и обменов, однако в большинстве реальных задач эта трудоемкость сравнима с трудоемкостью быстрых алгоритмов и может быть равна [ ].

При решении задачи рекурсивным способом текущие состояния программы хранятся в области памяти, называемой [ ]. Максимальное количество слоев [ ], заполняемых при конкретном вычислении значения рекурсивной функции, носит название [ ].

**6.4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и/или опыта деятельности, характеризующие этапы формирования компетенций****Рейтинг по дисциплине «Алгоритмы и анализ сложности»**

Максимальная сумма баллов – 100.

Промежуточная аттестация – 70 баллов, зачет – 30 баллов.

Вид работы	Количество единиц работы	Количество баллов на единицу вида работы	Максимальная сумма баллов по виду работы
Посещение занятий в период семестра	42	0,5	21
Выполнение заданий для самостоятельной работы в течение выполнения лабораторных работ	12	2	24
Отчёт по индивидуальным заданиям	1	10	10
Решение задач повышенной сложности	2	5	10
Контрольная работа	1	5	5
Зачет	1	30	30

*способность использовать знания методов проектирования и производства программного продукта, принципов построения, структуры и приемов работы с инструментальными средствами, поддерживающими создание программного обеспечения (ПО) (ОПК 8)*

Планируемые результаты обучения	Критерии оценивания с весовым коэффициентом	Показатели оценивания				
		1	2	3	4	5
Выпускник знает способы программирования нелинейных структур данных и их представление в памяти компьютера;	когнитивный – 0.2	Знает о существовании нелинейных структур данных и может привести примеры	Знает о принципах и способах представления нелинейных структур данных в памяти компьютера	Знает нелинейные структуры данных и может описать их средствами языка для обработки в программе	Знает семантику команд и сущность процессов, происходящих в среде при обработке в программе объектов нелинейной структуры	Знает тонкости процессов, происходящих в среде программирования при обработке в программе объектов различной сложности, имеющих нелинейную структуру
Выпускник знает постановку и алгоритмы задач поиска и сортировки в массивах, поиска на графах	когнитивный – 0.2	Знает понятия массива, графа, может описать идеи методов поиска в массиве и на графе и сортировок массива	Знает основные методы поиска в массиве и на графах и основные методы сортировок массива, может реализовать простые методы в виде алгоритмов	Знает методы сортировок массива, умеет реализовать их в виде программных кодов, знает идеи и методы реализации поиска на графах	Знает отличия внутренних и внешних сортировок массивов, реализованных в виде программных модулей, умеет реализовать какой-либо алгоритм поиска на	Знает тонкости реализации каждого метода сортировки и поиска в массиве и на графе, может выбрать оптимальный способ представления графа для конкретной задачи

					графе	
Выпускник знает теоретические основы и приемы программирования перебора с возвратом	когнитивный – 0.2	Знает основные понятия: полное решение, частичное решение, откат и т.п.	Знает основные механизмы перебора с возвратом, может привести примеры	Знает общую схему метода перебора с возвратом, может реализовать ее в виде алгоритма для конкретной задачи	Знает способы реализации перебора с возвратом	Способен выбрать оптимальную схему реализации метода перебора с возвратом для решения конкретной задачи
Выпускник умеет при решении конкретной задачи профессионально грамотно сформулировать задачу программирования, составить и оценить алгоритм решения, реализовать его в данной языковой среде, выполнить необходимое тестирование или верификацию построенной программы;	деятельностный – 0.2	Способен понять правильно написанные программы по предложенным задачам	Способен понять и объяснить основные шаги алгоритма правильно написанных программ для решения предложенных задач	Способен модифицировать алгоритм правильно написанной программы при изменении условия конкретной задачи	Способен самостоятельно написать алгоритм и программу для решения предложенной задачи	Способен самостоятельно написать алгоритм, программу и разработать систему тестов при решении предложенной задачи
Выпускник имеет навыки и опыт деятельности практического программирования конкретных задач из различных предметных областей в определенной языковой среде	деятельностный – 0.2	Владеет приемами программирования в конкретной программной среде	Способен выбрать программный способ реализации предложенного алгоритма для решения конкретной прикладной задачи	Способен разработать алгоритм и реализовать его программными средствами для решения предложенной задачи	Способен выбрать наиболее эффективные средства для реализации разработанного самостоятельно алгоритма решения предложенной задачи	Способен составить оптимальный алгоритм решения предложенной задачи, выбрать и обосновать свой выбор его программной реализации, оценить полученные результаты и предложить систему тестов

## 7. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

### 7.1 Основная литература:

1. Бородихин, В.Н. Языки программирования (Си/Си++): учебно-методическое пособие – Омск: Омский государственный университет, 2013 – 200с. Доступ по ссылке: [http://biblioclub.ru/index.php?page=book\\_view&book\\_id=237519](http://biblioclub.ru/index.php?page=book_view&book_id=237519)
2. Математическое программирование: Учебное пособие – М.: Издательство Лань, 2014. – 432с. Доступ по ссылке: <http://e.lanbook.com/view/book/68470/>

### 7.2 Дополнительная литература:

1. Вирт, Н. Алгоритмы и структуры данных/ Н.Вирт. – М: ДМК-пресс, 2013.
2. Кнут, Дональд Э. Искусство программирования, 3-е изд. = The Art of Computer Programming, 3-ed. /Дональд Э. Кнут. — М: Вильямс, 2007. — ISBN 978-5-8459-0081-4

## 8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1. [www.intuit.ru](http://www.intuit.ru)
2. <http://microsoft.com/>
3. <http://moodle.tsput.ru/course/view.php?id=15773>

Для качественной организации занятий в соответствии с разработанным курсом необходимо иметь возможность:

- 1) чтения лекций, в том числе с использованием мультимедийных средств обучения;
- 2) работы в компьютерном классе. На компьютерах должно быть установлено программное обеспечение:
  - a. интегрированная среда программирования;
  - b. текстовый редактор MicroSoftWord
- 3) выхода студентов в образовательную систему MOODLE

## 9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Дисциплина «Алгоритмы и анализ сложности» направлена на формирование у студентов знаний, умений и владений в области разработки алгоритмов и программирования на одном из языков процедурной парадигмы. В результате изучения данной дисциплины студенты должны научиться решать сложные задачи с помощью компьютера, определять и реализовывать необходимые при решении структуры данных, раскладывать сложные задачи на независимые подзадачи, выбирать оптимальные алгоритмы для решения прикладных задач. Курс призван обобщить и систематизировать теоретические знания и практические умения студентов, полученные ими ранее при изучении дисциплин модуля «Программирование».

Преподавание дисциплины должно включать в себя следующие образовательные технологии:

- 1) Организация лекций с использованием при необходимости мультимедийных технологий;
- 2) Использование в ходе лабораторных работ дидактических материалов в виде: опорных конспектов по теоретической составляющей занятий, файлов с примерами программ и т.п.
- 3) Использование ресурсов LMS MOODLE с целью организации процесса систематизации, приобретения и контроля знаний;
- 4) Организация выполнения индивидуальных заданий с целью усиления приобретения навыков самостоятельного построения моделей, разработки алгоритмов и написания программ;

- 5) Организация презентации индивидуальных заданий с целью формирования навыка владения профессиональной терминологией, грамотного выстраивания цепочки высказываний для доказательства обоснованности построенных моделей предметной области и правильности отражения реальных свойств объектов;
- б) Формирование у студентов убежденности в необходимости последовательного освоения следующих этапов в образовательной деятельности:
- ознакомься с содержанием и теоретическими основами изучаемой темы;
  - рассмотри, обсуди с другом и протестируй задачу, решенную кем-то;
  - реши самостоятельно задачу, подобную рассмотренной ранее;
  - реши самостоятельно задачу по изучаемой теме.

#### **10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ, ВКЛЮЧАЯ ПЕРЕЧЕНЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ**

- Подписка Microsoft DreamSpark Premium - Сублицензионный договор № S-2042626/M18 от 04.06.2013:
  - Средства для разработки и проектирования Visual Studio 2008, 2010, 2012 и 2013 Professional Editions;
  - Интегрированная среда разработки Visual Studio Express;
  - Операционная система Windows Server 2008 Standard Edition 32-bit;
  - Операционная система Windows 8.1 Pro;
  - Отдельные программы из Office 2007, Office 2010, Office 2013;
- Операционная система Microsoft Windows XP Professional Russian – Лицензия № 16698685 от 08.08.2003 г.;
- Программное обеспечение Microsoft Office XP Professional Win32 Russian– Лицензия № 16698685 от 08.08.2003 г.;
- Веб-браузеры.
- Доступ студентов через личные кабинеты к электронным библиотечным системам.
- Возможность работы студентов на удаленном рабочем столе кафедры информатики и информационных технологий.

#### **11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ «АЛГОРИТМЫ И АНАЛИЗ СЛОЖНОСТИ»**

Компьютерные аудитории, подключение к локальной сети университета и сети Интернет, права доступа к перечисленному выше программному обеспечению.

## 12. АННОТАЦИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ «АЛГОРИТМЫ И АНАЛИЗ СЛОЖНОСТИ»

1. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы

**Компетенция** *способность применять в профессиональной деятельности современные языки программирования и языки баз данных, методологии системной инженерии, системы автоматизации проектирования, электронные библиотеки и коллекции, сетевые технологии, библиотеки и пакеты программ, современные профессиональные стандарты информационных технологий (ОПК-2).*

**Выпускник знает:**

- способы программирования нелинейных структур данных и их представление в памяти компьютера;
- постановку и алгоритмы задач поиска и сортировки в массивах, поиска на графах;
- теоретические основы и приемы программирования перебора с возвратом;

**Умеет:**

- при решении конкретной задачи профессионально грамотно сформулировать задачу программирования, составить и оценить алгоритм решения, реализовать его в данной языковой среде, выполнить необходимое тестирование или верификацию построенной программы;

**Владеет и (или) имеет опыт деятельности:**

- в области практического программирования конкретных задач из различных предметных областей в определенной языковой среде.

2. Место дисциплины «Алгоритмы и анализ сложности» в структуре ОПОП

Дисциплина «Алгоритмы и анализ сложности» относится к обязательным дисциплинам вариативной части дисциплин направления. Изучение данной дисциплины базируется на освоении студентами курсов «Основы алгоритмизации», «Вводный курс программирования», «Основы программирования», «Языки программирования», «Математическая логика и теория алгоритмов», «Дискретная математика».

К началу изучения дисциплины студенты должны знать структуру программы в языке C(C++), организацию ввода и вывода данных различного типа на экран и в поток, операцию присваивания значений, программирование основных алгоритмических конструкций и функций пользователя. Очень важно владение аппаратом указателей на уровне работы с массивами, строками и функциями пользователя. Студенты должны знать алгоритмы работы машин Тьюринга, доказывать примитивную рекурсивность функций, использовать графы для моделирования и решения задач в различных областях математики.

Освоение данной дисциплины необходимо для выработки профессионального стиля программирования, изучения основ и принципов объектно-ориентированного и логического программирования, прохождения преддипломной практики, выполнения выпускной квалификационной работы.

3. Объем дисциплины – 3 зачетные единицы.

4. Образовательный процесс осуществляется на русском языке.

5. Разработчики:

Мартынюк Юлия Михайловна, кандидат педагогических наук, доцент кафедры информатики и вычислительной техники;

Ванькова Валентина Сергеевна, кандидат физико-математических наук, доцент кафедры информатики и вычислительной техники.



**13. ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ К РАБОЧЕЙ ПРОГРАММЕ  
ДИСЦИПЛИНЫ «АЛГОРИТМЫ И АНАЛИЗ СЛОЖНОСТИ»**

Изменения к рабочей программе дисциплины отсутствуют.

Заведующий кафедрой

информатики и информационных технологий  А.В. Якушин,

«24» декабря 2015 г.

Программа составлена в соответствии с требованиями ФГОС ВО.

Разработчики:

Фамилия, имя, отчество	Учёная степень	Учёное звание	Должность	Дата разработки	Подпись
Мартынюк Юлия Михайловна,	кандидат педагогических наук,	доцент	доцент кафедры информатики и информационных технологий	24.12.2015	
Ванькова Валентина Сергеевна	кандидат физико-математических наук	доцент	доцент кафедры информатики и информационных технологий;	24.12.2015	