

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»

Кафедра вычислительной техники

“УТВЕРЖДАЮ”
ДЕКАН АВТФ

к.т.н. Рева И. Л.

“ ____ ” _____ г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
Алгоритмы и структуры данных

Образовательная программа: 09.03.04 Программная инженерия, профиль: Технологии разработки программного обеспечения

Курс: 3, семестр : 5

Факультет автоматики и вычислительной техники,

		Семестр
№	Вид деятельности	5
1	Всего зачетных единиц (кредитов)	3
2	Всего часов	108
3	Всего занятий в контактной форме, час.	61
4	Лекции, час.	36
5	Практические занятия, час.	0
6	Лабораторные занятия, час.	18
7	из них в активной и интерактивной форме, час.	18
8	Аттестация, час.	2
9	Консультации, час.	5
10	Самостоятельная работа, час.	47
11	Виды самостоятельной работы (курсовой проект, курсовая работа, РГЗ, подготовка к контрольной работе)	РГЗ
12	Вид аттестации	ДЗ

Рабочая программа составлена на основании федерального государственного образовательного стандарта высшего образования по направлению (специальности): 09.03.04 Программная инженерия

ФГОС введен в действие приказом №229 от 12.03.2015 г. , дата утверждения: 01.04.2015 г.

Место дисциплины в структуре учебного плана: Блок 1, вариативная

Рабочая программа разработана на основе компетентностной модели выпускника по направлению (специальности): 09.03.04 Программная инженерия

Рабочая программа обсуждена на заседании кафедры

ВТ, протокол заседания кафедры №6 от 20.06.2017

Утверждена на совете факультета автоматики и вычислительной техники, протокол № 6 от 21.06.2017

Программу разработал:

старший преподаватель, Романенко Т. А.

Заведующий кафедрой:

доцент, к.т.н. Якименко А. А.

Ответственный за образовательную программу:

доцент Романов Е. Л.

1. Внешние требования

Таблица 1.1

Компетенция ФГОС: ОПК.3 готовность применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов; в части следующих результатов обучения:
з3. знать принципы и методологии структурного и объектно-ориентированного подходов к разработке программного обеспечения
у9. владеть методами оценки трудоемкости программного проекта
Компетенция ФГОС: ПК.22 способность создавать программные интерфейсы; в части следующих результатов обучения:
з6. знать методы и средства проектирования программных интерфейсов

2. Требования НГТУ к результатам освоения дисциплины

Таблица 2.1

Результаты изучения дисциплины по уровням освоения (иметь представление, знать, уметь, владеть)	Формы организации занятий
Алгоритмы и структуры данных	
ОПК.3.з3 знать принципы и методологии структурного и объектно-ориентированного подходов к разработке программного обеспечения	
1.о назначении, формах и области применения алгоритмов обработки данных и структур данных в реальных программных системах;	Лекции; Самостоятельная работа
2.о разнообразии форм представления данных для обработки на ЭВМ	Лекции; Лабораторные работы; Самостоятельная работа
3."библиотеку" фундаментальных структур и алгоритмов обработки данных	Лекции; Лабораторные работы; Самостоятельная работа
4.способы представления структур данных в ЭВМ	Лекции; Самостоятельная работа
ОПК.3.у9 владеть методами оценки трудоемкости программного проекта	
5.о подходах к оценке сложности алгоритмов и методах построения эффективных алгоритмов	Лекции; Лабораторные работы; Самостоятельная работа
6.методы анализа и оценки вычислительной сложности алгоритмов	Лекции; Лабораторные работы; Самостоятельная работа
7.выбирать, использовать, реализовывать и применять оптимальные в конкретных условиях базовые структуры данных и алгоритмы для организации сложных управляющих и информационных структур	Лекции; Самостоятельная работа
8.проводить предварительную оценку эффективности проектируемых алгоритмов и структур данных	Лекции; Лабораторные работы; Самостоятельная работа
9.экспериментального исследования эффективности реализации алгоритмов и структур данных	Лабораторные работы; Самостоятельная работа
ПК.22.з6 знать методы и средства проектирования программных интерфейсов	
10.технологии проектирования структуры данных и интерфейсов на основе абстрактного типа данных	Лабораторные работы; Самостоятельная работа
11.формирования документального сопровождения программной разработки	Лабораторные работы; Самостоятельная работа
ОПК.3.з3 знать принципы и методологии структурного и объектно-ориентированного подходов к разработке программного обеспечения	
12.принципы и методологии структурного и объектно-ориентированного подходов к разработке программного обеспечения	Лекции; Лабораторные работы; Самостоятельная работа
ОПК.3.у9 владеть методами оценки трудоемкости программного проекта	
13. владеть методами оценки трудоемкости программного проекта	Лекции; Лабораторные работы; Самостоятельная работа

ПК.22.36 знать методы и средства проектирования программных интерфейсов	
14.методы и средства проектирования программных интерфейсов	Лабораторные работы

3. Содержание и структура учебной дисциплины

Таблица 3.1

Темы лекций	Активные формы, час.	Часы	Ссылки на результаты обучения	Учебная деятельность
Семестр: 5				
Дидактическая единица: Вычислительная сложность алгоритмов				
1. Анализ вычислительной сложности алгоритмов. Методы оценки вычислительной сложности.	0	1	13, 5, 6, 8	Лекция
2. Классификация алгоритмов по вычислительной сложности. NP-сложные и трудно решаемые задачи	0	1	13, 5, 6, 8	Лекция
Дидактическая единица: Вычислительная сложность алгоритмов сортировки				
3. Быстрый бинарный поиск. Задача сортировки данных. Внутренняя сортировка. Внешняя сортировка.	0	1	3, 5, 6, 7	Лекция
4. Обзор базовых и усовершенствованных методов сортировки. Анализ вычислительной сложности алгоритмов сортировки.	0	4	3, 5, 6, 7	Лекция
Дидактическая единица: Элементарные структуры данных				
5. Позиционные множества и структуры данных. Формы представления структур данных в памяти ЭВМ. Стек, очередь, дек, список. Операции и запросы. и их трудоемкость.	0	2	1, 2, 3, 4	Лекция
Дидактическая единица: Ассоциативные структуры данных.				
6. Ассоциативные структуры данных с доступом по ключу. Разновидности ассоциативных структур и области применения. Словари. Запросы и операции. Формы представления в памяти ЭВМ.	0	1	1, 2, 3, 4, 7	Лекция
7. Бинарное дерево поиска. Схемы обхода дерева поиска. Рекурсивные и итеративные алгоритмы основных операций. Случайное и вырожденное дерево поиска. Анализ эффективности операций.	0	3	1, 3, 4, 5, 7	Лекция

8. Сбалансированные деревья поиска. Рандомизированное дерево, AVL-дерево, 2-3 дерево, 2-3-4 - дерево, красно-черное дерево. Оценка эффективности операций для сбалансированных деревьев.	0	4	1, 2, 3, 4, 7, 8	Лекция
9. Хеш-таблицы. Основные методы хеширования. Методы разрешения коллизий, линейное, квадратичное и двойное повторное хеширование. Анализ эффективности методов хеширования.	0	3	1, 2, 3, 4, 7, 8	Лекция
Дидактическая единица: Внешние структуры поиска				
10. Структуры данных и алгоритмы для внешней памяти. Индексированные, хешированные файлы. Внешние деревья поиска: В - дерево. Оценка вычислительной сложности операций для структур внешнего поиска.	0	2	1, 12, 2, 3	Лекция
Дидактическая единица: Графы.				
11. Терминология и основные определения. Неориентированные графы и орграфы, взвешенные графы. Формы программного представления графов: L- граф, M- граф. Генераторы графов.	0	1	1, 12, 2, 3, 4	Лекция.
12. Обходы вершин графа. Алгоритм поиска в ширину (BFS) и в глубину (DFS).	0	2	1, 12, 3	Лекция
13. Обзор задач для графов, основанных на алгоритмах обхода вершин.	0	2	1, 12, 3, 7	Лекция
14. Обзор алгоритмов решения задач для взвешенных графов.	0	3	1, 12, 3, 7	Лекция
15. Обзор алгоритмов решения задач для ориентированных графов.	0	3	1, 12, 3, 7	Лекция
Дидактическая единица: Поиск оптимальных решений.				
16. Поиск оптимального решения методом полного перебора. Исчерпывающий поиск, альфа - бета отсечение, метод ветвей и границ.	0	1,5	1, 12, 3, 7	Лекция
17. Метод динамического программирования. Метод жадного выбора.	0	1,5	1, 12, 3, 7	Лекция

Таблица 3.2

Темы лабораторных работ	Активные формы, час.	Часы	Ссылки на результаты обучения	Учебная деятельность
-------------------------	----------------------	------	-------------------------------	----------------------

Семестр: 5				
Дидактическая единица: Элементарные структуры данных				
1. Проектирование и реализация коллекции данных - список	4	4	10, 11, 12, 13, 14, 3, 8, 9	Лабораторная работа
Дидактическая единица: Ассоциативные структуры данных.				
2. Проектирование и реализация коллекции данных - бинарное дерево поиска.	4	4	10, 11, 12, 13, 14, 3, 5, 6, 9	Лабораторная работа
3. Проектирование и реализация коллекции данных - сбалансированное дерево поиска.	4	4	10, 11, 12, 3, 6, 8, 9	Лабораторная работа
4. Проектирование и реализация коллекции данных - хеш-таблица.	6	6	10, 11, 12, 13, 14, 2, 3, 6, 8, 9	Лабораторная работа

Таблица 3.3

Темы для самостоятельного изучения	Активные формы, час.	Часы	Ссылки на результаты обучения	Учебная деятельность
Семестр: 5				
Дидактическая единица: Вычислительная сложность алгоритмов сортировки				
1. Изучение алгоритмов сортировки с квадратичной $O(n^2)$ и линейной $O(n \log n)$ трудоемкостью.	0	2	13, 3, 5, 6, 7	Самостоятельное изучение теоретического материала
Дидактическая единица: Ассоциативные структуры данных.				
2. Изучение алгоритмов для бинарного дерева поиска	0	2	3, 4, 7, 8	Самостоятельное изучение теоретического материала
3. Изучение алгоритмов для сбалансированных деревьев поиска	0	4	2, 3, 7, 8	Самостоятельное изучение теоретического материала
Дидактическая единица: Внешние структуры поиска				
4. Изучение разновидностей форм и алгоритмов для внешней структуры поиска - В- дерево.	0	3	1, 12, 2, 3	Самостоятельное изучение теоретического материала
Дидактическая единица: Поиск оптимальных решений.				
5. Изучение примеров алгоритмов поиска оптимального решения: построение оптимального бинарного дерева поиска, построение кодового дерева Хаффмена.	0	4	1, 13, 3, 5, 6, 7, 8	Самостоятельное изучение теоретического материала

4. Самостоятельная работа обучающегося

№	Виды самостоятельной работы	Ссылки на результаты обучения	Часы на выполнение	Часы на консультации
Семестр: 5				
1	РГЗ	1, 10, 12, 3, 4	10	5
Проектирование и реализация универсальной программной коллекции "Граф" и использование её для решения различных задач для неориентированных, ориентированных и взвешенных графов: Романенко Т. А. Структуры и алгоритмы обработки данных (7 семестр) [Электронный ресурс] : электронный учебно-методический комплекс / Т. А. Романенко ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://courses.edu.nstu.ru/index.php?show=155&curs=305 . - Загл. с экрана.				

2	Оформление отчета по лабораторной работе, подготовка к защите.	10, 11, 9	5	0
Оформление результатов лабораторных работ в соответствии с требованиями, предъявляемыми к отчету по лабораторной работе. Дополнительное изучение теоретического материала, формулирование ответов на контрольные вопросы к лабораторным работам.: Романенко Т. А. Структуры и алгоритмы обработки данных (6 семестр) [Электронный ресурс] : электронный учебно-методический комплекс / Т. А. Романенко ; Новосиб. гос. техн. ун-т. - Новосибирск, [2010]. - Режим доступа: http://courses.edu.nstu.ru/index.php?show=155&curs=70. - Загл. с экрана.				
3	Экспериментальное исследование трудоемкости основных операций коллекции данных.	6, 8, 9	9	0
Измерение трудоемкости основных операций коллекций данных, разработанных в ходе лабораторных работ, с помощью программ тестирования. : Романенко Т. А. Структуры и алгоритмы обработки данных (7 семестр) [Электронный ресурс] : электронный учебно-методический комплекс / Т. А. Романенко ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://courses.edu.nstu.ru/index.php?show=155&curs=305. - Загл. с экрана.				
4	Подготовка к теоретическому зачету в форме тестирования.	1, 2, 3, 4	8	0
Повторение теоретического курса лекций для сдачи теоретического зачета. Изучение демо-версии задач, используемых на теоретическом зачете. Пробное решение тестов.: Романенко Т. А. Структуры и алгоритмы обработки данных (7 семестр) [Электронный ресурс] : электронный учебно-методический комплекс / Т. А. Романенко ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://courses.edu.nstu.ru/index.php?show=155&curs=305. - Загл. с экрана.				
5	Самостоятельное изучение теоретического материала	1, 12, 13, 2, 3, 4, 5, 6, 7, 8	15	0
Студент изучает темы, приведенные в таблице 3.3 : Романенко Т. А. Структуры и алгоритмы обработки данных (6 семестр) [Электронный ресурс] : электронный учебно-методический комплекс / Т. А. Романенко ; Новосиб. гос. техн. ун-т. - Новосибирск, [2010]. - Режим доступа: http://courses.edu.nstu.ru/index.php?show=155&curs=70. - Загл. с экрана. Романенко Т. А. Структуры и алгоритмы обработки данных (7 семестр) [Электронный ресурс] : электронный учебно-методический комплекс / Т. А. Романенко ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://courses.edu.nstu.ru/index.php?show=155&curs=305. - Загл. с экрана.				

5. Технология обучения

Для организации и контроля самостоятельной работы обучающихся, а также проведения консультаций применяются информационно-коммуникационные технологии (табл. 5.1).

Таблица 5.1

Деятельность	Информационно-коммуникационные технологии
Информирование	Личный типовой сайт: ftp://tkvt.cs.nstu.ru:2112/Ermak/ASOD
Размещение учебных материалов	ЭБС:Романенко Т. А. Проектирование и реализация коллекций данных : учебно-методическое пособие / Т. А. Романенко ; Новосиб. гос. тех

6. Правила аттестации обучающихся по учебной дисциплине

Для аттестации обучающихся по дисциплине используется балльно-рейтинговая система (БРС), позволяющая выставять оценки по традиционной шкале и 15-уровневой ECTS. Краткая информация о БРС приведена в табл. 6.1.

Таблица 6.1

Оцениваемые виды деятельности обучающихся	Мин. балл	Максимальный балл

Семестр: 5		
Лабораторная №2:	4	8
Контролирующие материалы приводятся в "Романенко Т. А. Структуры и алгоритмы обработки данных (6 семестр) [Электронный ресурс] : электронный учебно-методический комплекс / Т. А. Романенко ; Новосиб. гос. техн. ун-т. - Новосибирск, [2010]. - Режим доступа: http://courses.edu.nstu.ru/index.php?show=155&curs=70 . - Загл. с экрана."		
Лабораторная №3:	6	12
Контролирующие материалы приводятся в "Романенко Т. А. Структуры и алгоритмы обработки данных (7 семестр) [Электронный ресурс] : электронный учебно-методический комплекс / Т. А. Романенко ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://courses.edu.nstu.ru/index.php?show=155&curs=305 . - Загл. с экрана."		
Лабораторная №4:	7	14
Контролирующие материалы приводятся в "Романенко Т. А. Структуры и алгоритмы обработки данных (7 семестр) [Электронный ресурс] : электронный учебно-методический комплекс / Т. А. Романенко ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://courses.edu.nstu.ru/index.php?show=155&curs=305 . - Загл. с экрана."		
Лабораторная №5:	8	16
Контролирующие материалы приводятся в "Романенко Т. А. Структуры и алгоритмы обработки данных (7 семестр) [Электронный ресурс] : электронный учебно-методический комплекс / Т. А. Романенко ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://courses.edu.nstu.ru/index.php?show=155&curs=305 . - Загл. с экрана."		
РГЗ:	15	30
Зачет:	10	20
Контролирующие материалы приводятся в "Романенко Т. А. Структуры и алгоритмы обработки данных (7 семестр) [Электронный ресурс] : электронный учебно-методический комплекс / Т. А. Романенко ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://courses.edu.nstu.ru/index.php?show=155&curs=305 . - Загл. с экрана."		

В таблице 6.2 представлено соответствие форм контроля заявляемым требованиям к результатам освоения дисциплины.

Таблица 6.2

Коды компетенций ФГОС	Результаты обучения	Формы контроля		
		Защита ЛР	Защита РГЗ	Зачет
ОПК.3	з3. знать принципы и методологии структурного и объектно-ориентированного подходов к разработке программного обеспечения			+
	у9. владеть методами оценки трудоемкости программного проекта	+	+	
ПК.22	зб. знать методы и средства проектирования программных интерфейсов	+	+	

Фонд оценочных средств по дисциплине представлен в приложении № 1 к рабочей программе.

7. Литература

Основная литература

1. Вирт Н. Алгоритмы и структуры данных : с примерами на Паскале / Никлаус Вирт ; [пер. с англ. Д.Б. Подшивалова]. - СПб., 2007. - 350, [1] с. : ил.
2. Ландовский В. В. Структуры данных : учебное пособие / В. В. Ландовский ; Новосиб. гос. техн. ун-т. - Новосибирск, 2016. - 65, [2] с. : ил., табл.. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000233660
3. Кубенский А. А. Структуры и алгоритмы обработки данных: объектно-ориентированный подход и реализация на C++ : учебное пособие по специальности "Математическое обеспечение и администрирование информационных систем" - 351500 / А. А. Кубенский. - Санкт-Петербург, 2004. - 464 с. : ил. + 1 CD-ROM.
4. Кнут Д. Э. Искусство программирования. Т. 1 : пер. с англ. / Дональд Э. Кнут ; под общ. ред. Ю. В. Козаченко. - М. [и др.], 2007. - 712 с. : ил.

Дополнительная литература

1. Ахо А. В. Структуры данных и алгоритмы / Альфред В. Ахо, Джон Э. Хопкрофт, Джеффри Д. Ульман ; пер. с англ. и ред. А. А. Минько. - М., 2001. - 382 с. : ил.

2. Седжвик Р. Фундаментальные алгоритмы на C++. Ч. 5 : [пер. с англ.] / Роберт Седжвик. - М. [и др.], 2002. - 484 с. : ил.
3. Седжвик Р. Фундаментальные алгоритмы на C++. Ч. 1-4 : [пер. с англ.] / Роберт Седжвик. - Москва [др.], 2002. - 687 с. : ил.. - Парал. тит. л. англ..
4. Кормен Т. Алгоритмы: построение и анализ / Т. Кормен, Ч. Лейзерсон, Р. Ривест ; [пер. с англ. К. Белова и др. ; под ред. А. Шеня]. - Москва, 2001. - 955 с. : ил.
5. Каррано Ф. М. Абстракция данных и решение задач на C++. Стены и зеркала / Фрэнк М. Каррано, Джанет Дж. Причард ; [пер. с англ. и ред. Д. А. Ключина]. - Москва [и др.], 2003. - 847 с.. - Парал. тит. л. англ..

Интернет-ресурсы

1. ЭБС НГТУ : <http://elibrary.nstu.ru/>
2. ЭБС «Издательство Лань» : <https://e.lanbook.com/>
3. ЭБС IPRbooks : <http://www.iprbookshop.ru/>
4. ЭБС "Znanium.com" : <http://znanium.com/>
5. :

8. Методическое и программное обеспечение

8.1 Методическое обеспечение

1. Романенко Т. А. Проектирование и реализация коллекций данных : учебно-методическое пособие / Т. А. Романенко ; Новосиб. гос. техн. ун-т. - Новосибирск, 2005. - 110, [1] с. : ил.. - Режим доступа: http://www.ciu.nstu.ru/fulltext/textbooks/2005/05_romanenko.rar
2. Романенко Т. А. Структуры и алгоритмы обработки данных (6 семестр) [Электронный ресурс] : электронный учебно-методический комплекс / Т. А. Романенко ; Новосиб. гос. техн. ун-т. - Новосибирск, [2010]. - Режим доступа: <http://courses.edu.nstu.ru/index.php?show=155&curs=70>. - Загл. с экрана.
3. Романенко Т. А. Структуры и алгоритмы обработки данных (7 семестр) [Электронный ресурс] : электронный учебно-методический комплекс / Т. А. Романенко ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: <http://courses.edu.nstu.ru/index.php?show=155&curs=305>. - Загл. с экрана.

8.2 Специализированное программное обеспечение

- 1 Visual Studio 2010

9. Материально-техническое обеспечение

Компьютерный класс

№	Наименование	Назначение
1	Компьютерный класс (Компьютеры объединены в локальную сеть с выходом в Internet)	Компьютеры объединены в локальную сеть с выходом в Internet

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»

Кафедра вычислительной техники

“УТВЕРЖДАЮ”
ДЕКАН АВТФ
к.т.н., доцент И.Л. Рева
“ ” _____ Г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

УЧЕБНОЙ ДИСЦИПЛИНЫ

Алгоритмы и структуры данных

Образовательная программа: 09.03.04 Программная инженерия, профиль: Технологии разработки программного обеспечения

1. Обобщенная структура фонда оценочных средств учебной дисциплины

Обобщенная структура фонда оценочных средств по дисциплине Алгоритмы и структуры данных приведена в Таблице.

Таблица

Формируемые компетенции	Показатели сформированности компетенций (знания, умения, навыки)	Темы	Этапы оценки компетенций	
			Мероприятия текущего контроля (курсовой проект, РГЗ(Р) и др.)	Промежуточная аттестация (экзамен, зачет)
ОПК.3 готовность применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	з3. знать принципы и методологии структурного и объектно-ориентированного подходов к разработке программного обеспечения	Обзор базовых и усовершенствованных методов сортировки. Анализ вычислительной сложности алгоритмов сортировки. Изучение алгоритмов сортировки с квадратичной $O(n^2)$ и линейной $O(n \log n)$ трудоемкостью.		Зачет, тест № 2
		Позиционные множества и структуры данных. Формы представления структур данных в памяти ЭВМ. Стек, очередь, дек, список. Операции и запросы. и их трудоемкость.	Отчёт по лабораторной работе № 1	Зачет, тест № 3
		Бинарное дерево поиска. Схемы обхода дерева поиска. Рекурсивные и итеративные алгоритмы основных операций. Случайное и вырожденное дерево поиска. Анализ эффективности операций. Изучение алгоритмов для бинарного дерева поиска	Отчёт по лабораторной работе № 2	Зачет, тест № 4
		Изучение алгоритмов для сбалансированных деревьев поиска. Сбалансированные деревья поиска. Рандомизированное дерево, АВЛ-дерево, 2-3 дерево, 2-3-4 - дерево, красно-черное дерево. Оценка эффективности операций для сбалансированных деревьев.	Отчёт по лабораторной работе № 3	Зачет, тест № 5

		Хеш-таблицы. Основные методы хеширования. Методы разрешения коллизий, линейное, квадратичное и двойное повторное хеширование. Анализ эффективности методов хеширования.	Отчёт по лабораторной работе № 4	Зачет, тест № 6
		Структуры данных и алгоритмы для внешней памяти. Индексированные, хешированные файлы. Внешние деревья поиска: В - дерево. Оценка вычислительной сложности операций для структур внешнего поиска. Изучение разновидностей форм и алгоритмов для внешней структуры поиска - В- дерево.		Зачет, тест № 7
		Обзор задач для графов, основанных на алгоритмах обхода вершин. Обходы вершин графа. Алгоритм поиска в ширину (BFS) и в глубину (DFS). Обзор алгоритмов решения задач для ориентированных графов.	Отчёт по РГЗ (задача 1)	Зачет, тест № 8
		Обзор алгоритмов решения задач для взвешенных графов.	Отчёт по РГЗ (задача 2)	Зачет, тест № 9
		Поиск оптимального решения методом полного перебора. Исчерпывающий поиск, альфа - бета отсечение, метод ветвей и границ. Метод динамического программирования. Метод жадного выбора.		Зачет, тест № 10

ОПК.3	у9. владеть методами оценки трудоемкости программного проекта	Обзор задач для графов, основанных на алгоритмах обхода вершин. Обзор алгоритмов решения задач для взвешенных графов. Обзор алгоритмов решения задач для ориентированных графов.	Отчет по РГЗ (задачи1 и 2)	
		Поиск оптимального решения методом полного перебора. Исчерпывающий поиск, альфа - бета отсечение, метод ветвей и границ. Метод динамического программирования. Метод жадного выбора.	Отчет по РГЗ (задачи1 и 2)	
		Проектирование и реализация коллекции данных - список	Отчёт по лабораторной работе № 1	
		Проектирование и реализация коллекции данных - бинарное дерево поиска.	Отчёт по лабораторной работе № 2	
		Проектирование и реализация коллекции данных - сбалансированное дерево поиска.	Отчёт по лабораторной работе № 3	
		Проектирование и реализация коллекции данных - хеш-таблица.	Отчёт по лабораторной работе №4	
ПК.22/П способность создавать программные интерфейсы	з6. знать методы и средства проектирования программных интерфейсов	Проектирование и реализация коллекции данных - список	Отчёт по лабораторной работе № 1	
		Проектирование и реализация коллекции данных - бинарное дерево поиска.	Отчёт по лабораторной работе № 2	
		Проектирование и реализация коллекции данных - сбалансированное дерево поиска.	Отчёт по лабораторной работе № 3	
		Проектирование и реализация коллекции данных - хеш-таблица.	Отчёт по лабораторной работе №4	

2. Методика оценки этапов формирования компетенций в рамках дисциплины.

Промежуточная аттестация по дисциплине проводится в 5 семестре - в форме дифференцированного зачета, который направлен на оценку сформированности компетенций ОПК.3, ПК.22/П.

Зачет проводится в письменной форме, по тестам).

Зачет проводится в форме письменного тестирования, варианты теста составляются из вопросов, приведенных в паспорте зачета, позволяющих оценить показатели сформированности соответствующих компетенций

Кроме того, сформированность компетенций проверяется при проведении мероприятий текущего контроля, указанных в таблице раздела 1.

В 5 семестре обязательным этапом текущей аттестации являются лабораторные работы и расчетно-графическое задание (работа) (РГЗ(Р)). Требования к выполнению лабораторных работ и РГЗ(Р), состав и правила оценки сформулированы в паспорте РГЗ(Р).

Общие правила выставления оценки по дисциплине определяются балльно-рейтинговой системой, приведенной в рабочей программе учебной дисциплины.

На основании приведенных далее критериев можно сделать общий вывод о сформированности компетенций ОПК.3, ПК.22/П, за которые отвечает дисциплина, на разных уровнях.

Общая характеристика уровней освоения компетенций.

Ниже порогового. Уровень выполнения работ не отвечает большинству основных требований, теоретическое содержание курса освоено частично, пробелы могут носить существенный характер, необходимые практические навыки работы с освоенным материалом сформированы не достаточно, большинство предусмотренных программой обучения учебных заданий не выполнены или выполнены с существенными ошибками.

Пороговый. Уровень выполнения работ отвечает большинству основных требований, теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые виды заданий выполнены с ошибками.

Базовый. Уровень выполнения работ отвечает всем основным требованиям, теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые из выполненных заданий, возможно, содержат ошибки.

Продвинутый. Уровень выполнения работ отвечает всем требованиям, теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному.

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»
Кафедра вычислительной техники

Паспорт зачета

по дисциплине «Алгоритмы и структуры данных», 5 семестр

1. Методика оценки

Для аттестации студентов используется балльно-рейтинговая система. Рейтинг студента по дисциплине определяется как сумма баллов за работу в семестре (текущий рейтинг) и баллов, полученных в результате итоговой аттестации (дифференцированный зачет).

Текущий рейтинг определяется суммой баллов, полученных в результате выполнения четырёх лабораторных работ и РГЗ в течение 5 –го семестра. Дифференцированный зачёт проводится на зачётной неделе и направлен на проверку знаний, полученных в результате изучения теоретического курса дисциплины.

Максимальная сумма баллов за 5 семестр составляет 100 (текущий рейтинг - 80 баллов, дифференцированный зачет - 20 баллов). Минимальная сумма баллов за 5 семестр - 50 баллов.

Дифференцированный зачет проводится в письменной форме, по тест - билетам. Билет включает 10 тестов.

Каждый тест является задачей, представляющей отдельную тему теоретического курса дисциплины. В зависимости от сложности задачи тест оценивается от 2 до 5 баллов.

	Тема теоретического курса	Оценочный балл теста
тест 1	Вычислительная сложность алгоритмов	2
тест 2	Вычислительная сложность алгоритмов сортировки	2
тест 3	Элементарные структуры данных	3
тест 4	Ассоциативные структуры данных. Бинарное дерево поиска.	3
тест 5	Ассоциативные структуры данных. Сбалансированные деревья поиска.	5
тест 6	Ассоциативные структуры данных. Хеш-таблица.	4
тест 7	Внешние структуры поиска. В – дерево.	3
тест 8	Графы. Задачи для графов, основанные на алгоритмах обхода вершин.	3
тест 9	Графы. Задачи для взвешенных графов.	3
тест 10	Алгоритмы поиска оптимальных решений	4

Форма тест - билета для зачета

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Факультет АВТФ

Утверждаю: зав. кафедрой _____ должность, ФИО
(подпись)
(дата)

Тест - билет № _____
к зачету по дисциплине «Алгоритмы и структуры данных»

- Задача 1.
Поле для ответа (2 балла).
Задача 2.
Поле для ответа (2 балла).
Задача 3.
Поле для ответа (2 балла).
Задача 4.
Поле для ответа (3 балла).
Задача 5.
Поле для ответа (5 баллов).
Задача 6.
Поле для ответа 4 балла).
Задача 7.
Поле для ответа (3 балла).
Задача 8.
Поле для ответа (3 балла).
Задача 9.
Поле для ответа (3 балла).
Задача 10.
Поле для ответа (4 балла).

Общее число первичных баллов за тест: _____
Общее число баллов за тест по 20 бальной шкале: _____
Общее число баллов за 5 семестр по 100 бальной шкале _____
Оценка за диф. зачёт за 5 семестр по 5- бальной шкале: _____

Пример теста для зачета

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ Факультет АВТФ

Утверждаю: зав. кафедрой _____ должность, ФИО
(подпись)
(дата)

Тест - билет № 1 к зачету по дисциплине «Алгоритмы и структуры данных»

1. Методы сортировки, имеющие нотацию трудоемкости $O(n \log_2 n)$? Выберите номера ответов:

Варианты ответа:

1. Пирамидальная сортировка
2. Метод разделения
3. Метод выбора
4. Шейкер сортировка
5. Метод слияния
6. Обменная сортировка
7. Дерево выбора
8. Поразрядная сортировка
9. Метод вставки

Ответ: _____ (2 балла)

2. Дан неупорядоченный массив: 8 1 4 4 5 9 6 2. Выберите перестановку элементов в массиве после первого шага сортировки по возрастанию методом выбора для значений в массиве (номер ответа).

Варианты ответа:

1. 5 1 4 2 8 9 6 4
2. 2 1 4 4 5 9 6 8
3. 1 8 2 4 4 5 9 6
4. 8 1 4 2 5 9 6 4
5. 2 4 4 8 5 9 6 1
6. 1 8 4 4 5 9 6 2
7. 1 8 4 9 5 4 2 6
8. 8 1 4 4 5 9 6 2

Ответ: _____ (2 балла)

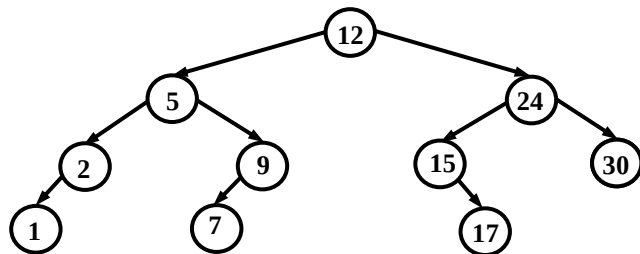
3. Приведите схему стека на базе массива после серии операций: вставка (5), вставка (1), вставка (2), удаление, удаление, вставка (8), вставка (1), удаление, вставка (4). Размер массива равен 8. Стек первоначально пуст.

Ответ: (2 балла)

4. Приведите изображение изначально пустого **BST**-дерева после вставки в корень дерева последовательности ключей 6, 2, 8, 3, 9.

Ответ: (3 балла)

5. Приведите изображение структуры AVL – дерева после выполнения операции вставки ключа 18:

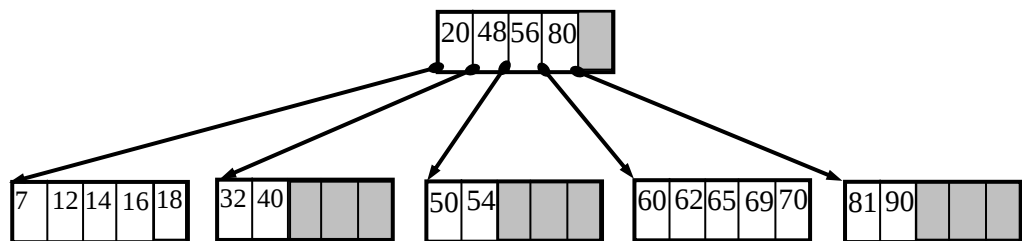


Ответ: (5 баллов)

6. Приведите вид первоначально пустой хеш-таблицы с цепочками коллизий после вставки ключей 20, 3, 51, 19, 57, 25. Размер хеш-таблицы равен $m = 8$. Для хеширования используется мультипликативное хеширование с хеш-функцией $h(k) = \lfloor m \times (k \times A \bmod 1) \rfloor$, где $k \times A \bmod 1$ – дробная часть произведения $k \times A$, $A = 0,6180339887$.

Ответ: (4 балла)

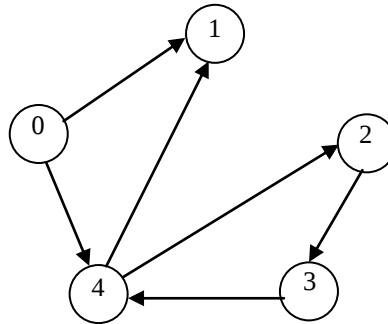
7. Приведите вид структуры В-дерева после вставки элемента с ключом 15. Минимальная степень В-дерева $t = 3$.



Ответ: (3 балла)

--

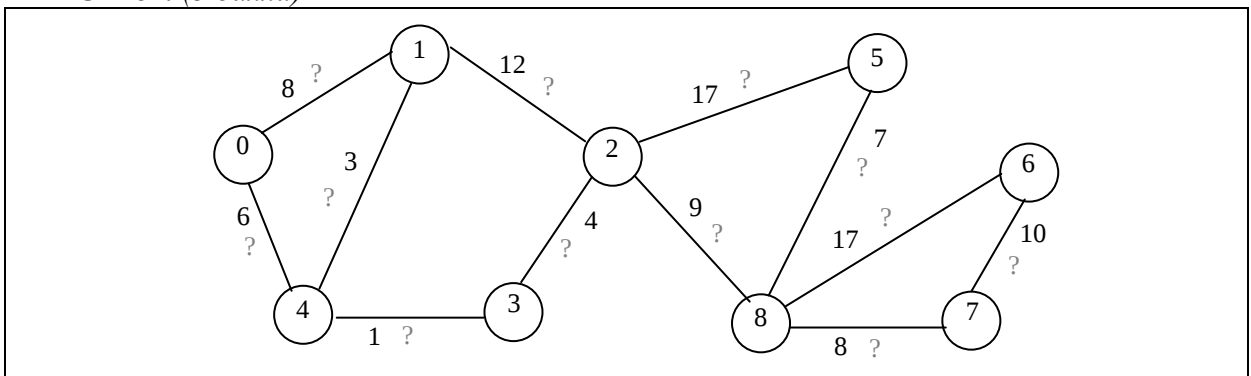
8. Дан ориентированный граф (орграф). Составьте матрицу транзитивного замыкания



Ответ: (3 балла)

9. Дан неориентированный взвешенный граф. Постройте минимальное остовное дерево методом Крускала. Ребра остоного дерева выделите другим цветом на чертеже графа. Укажите, в каком порядке алгоритм Крускала выбирает ребра остова.

Ответ: (3 балла)



10. Составьте таблицу соответствия между номерами задач и номерами соответствующих методов поиска решения задач.

Задача	Метод решения
1. Алгоритм транзитивного замыкания графа	1. Поиск с возвратом
2. Построение минимального остова во взвешенном графе (алгоритм Крускала)	2. Метод динамического программирования
3. Поиск кратчайших путей (алгоритм Дейкстры)	3. Жадный выбор
4. Поиск минимального гамильтонова цикла в графе	4. Исчерпывающий поиск с возвратом
5. Обход графа в глубину	
6. Определение всех кратчайших путей во взвешенном графе (алгоритм Флойда)	

Ответ: (4 балла)

Номер задачи	Номер метода
1	
2	
3	
4	
5	
6	

Общее число первичных баллов за тест: _____

Общее число баллов за тест по 20 бальной шкале: _____

Общее число баллов за 5 семестр по 100 бальной шкале: _____

Оценка за диф. зачёт за 5 семестр по 5- бальной шкале: _____

подпись

дата

2. Критерии оценки

- Ответ на билет (тест) для зачета считается **неудовлетворительным**, если общее число баллов за тест по 20 бальной шкале меньше 10 баллов.
- Ответ на билет (тест) для зачета засчитывается на **пороговом** уровне если общее число баллов за тест по 20 бальной шкале равно 10 баллам.
- Ответ на билет (тест) для зачета билет засчитывается на **базовом** уровне, если общее число баллов за тест по 20 бальной шкале находится в пределах 11 – 18 баллов
- Ответ на билет (тест) для зачета билет засчитывается на **продвинутом** уровне если общее число баллов за тест по 20 бальной шкале находится в пределах 19 – 20 баллов

3. Шкала оценки

Для аттестации студентов используется балльно-рейтинговая система. Рейтинг студента по дисциплине определяется как сумма баллов за работу в семестре (текущий рейтинг) и баллов, полученных в результате итоговой аттестации (дифференцированный зачет).

Максимальная сумма баллов за 5 семестр составляет 100 (текущий рейтинг - 80 баллов, дифференцированный зачет - 20 баллов). Минимальная сумма баллов за 5 семестр - 50 баллов.

К дифференцированному зачету допускаются студенты, сдавшие лабораторные работы, РГЗ, и набравшие не менее 25 баллов по результатам текущего рейтинга.

Зачет проводится в форме письменного теста, охватывающего все тематические модули курса. Зачет считается сданным, если сумма баллов по всем заданиям билета оставляет не менее 10 баллов (из 20 возможных).

По всем видам аттестации студент получает суммарный балл (от 50 до 100 баллов), соответствующий балл в системе ECTS и оценку по 5 - бальной системе.

В таблице 1 приведено максимальное количество баллов, которое может набрать студент по разным видам учебной деятельности в течение 5 семестра и диапазоны баллов, соответствующие минимальному и максимальному количеству баллов. Максимальная сумма баллов за 5 семестр составляет 100 (текущий рейтинг – 80 баллов, зачет – 20).

Таблица 1 Сроки и диапазоны баллов для всех видов учебной работы

№п/п	Вид учебной работы (учебной деятельности)	Максимальное количество баллов	Диапазоны баллов	Срок представления и защиты (неделя семестра)
Пятый семестр:				
1	Лабораторная работа № 1	6	4-8	7
2	Лабораторная работа № 2	8	6 - 12	11
3	Лабораторная работа № 3	12	7 - 14	15
4	Лабораторная работа № 4	14	8- 16	17
5	Расчетно-графическое задание	30	10 - 30	17
Итого по текущему рейтингу:		80	40 - 80	
6	Зачет	20	10-20	
Итого за пятый семестр:		100	50-72 (удовл.) 73-87 (хор.) 88-100 (отл.)	

Полученные баллы после зачёта переводятся в оценки по системе ECTS и по 5-бальной шкале согласно таблице 2:

Таблица 2 Сроки и диапазоны баллов для всех видов учебной работы

90 - 100	A+ (99 - 100)	отлично
	A (93 - 98)	
	A- (90 - 92)	
80 - 89	B+(87 - 89)	хорошо
	B (83 - 86)	
	B-(80 - 82)	
70 - 79	C+(77- 79)	удовлетворительно
	C (73 -76)	
	C-(70 - 72)	
60 - 69	D+(67 - 69)	не удовлетворительно
	D (63 -66)	
	D-(60 - 62)	
50 - 59	E (50 -59)	
25 - 49	FX	
0 - 24	F	

4. Вопросы к зачету по дисциплине «Алгоритмы и структуры данных»

Форма аттестации – тестирование.

Общее количество заданий – 41.

В вариант тест – билета включаются 10 заданий.

Схема варианта экзаменационного билета:

№ вопроса в билете	№ задания	Число заданий	Балл правильного ответа
1.	одно из (1 - 4)	4	2
2.	одно из (5 - 14)	10	2
3.	одно из (15 - 24)	10	2
4.	одно из (25 - 27)	3	3
5.	одно из (28 - 31)	4	5
6.	одно из (32 - 33)	2	4
7.	34	1	3
8.	одно из (35 – 37)	3	3
9.	одно из (38- 40)	2	3
10.	41	1	4

1. Нотации трудоемкости P – алгоритмов. Выберите номера ответов:

Варианты ответа:

1. $O(1)$
2. $O(n)$
3. $O(n!)$
4. $O(n^2)$
5. $O(2^n)$
6. $O(\log n)$
7. $O(n^3)$
8. $O(n \log n)$

Ответ: _____ (2 балла)

Эталонный ответ: 1, 2, 4, 6, 7, 8

2. Нотации трудоемкости NP – алгоритмов? Выберите номера ответов:

Варианты ответа:

1. $O(1)$
2. $O(n)$
3. $O(n!)$
4. $O(n^2)$
5. $O(2^n)$
6. $O(\log n)$
7. $O(n^3)$
8. $O(n \log n)$

Ответ: _____ (2 балла)

Эталонный ответ: 3, 5

3. Методы сортировки, имеющие нотацию трудоемкости $O(n^2)$? Выберите номера ответов:

Варианты ответа:

1. Пирамидальная сортировка
2. Метод выбора
3. Метод разделения
4. Поразрядная сортировка
5. Обменная сортировка
6. Шейкер сортировка
7. Метод слияния
8. Метод вставки
9. Дерево выбора

Ответ: _____ (2 балла)

Эталонный ответ: 2, 5, 8

4. Методы сортировки, имеющие нотацию трудоемкости $O(n \log_2 n)$? Выберите номера ответов:

Варианты ответа:

1. Пирамидальная сортировка
2. Метод выбора
3. Метод разделения
4. Поразрядная сортировка
5. Обменная сортировка
6. Шейкер сортировка
7. Метод слияния
8. Метод вставки
9. Дерево выбора

Ответ: _____ (3 балла)

Эталонный ответ: 1, 3, 7, 9

5. Дан неупорядоченный массив: 5 23 6 2 9 78 8 12. Выберите перестановку элементов в массиве после первого шага построения пирамиды для сортировки по убыванию (номер ответа).

Варианты ответа:

- | | |
|----|--------------------|
| 1. | 5 23 6 2 9 78 8 12 |
| 2. | 2 5 6 9 12 78 8 23 |
| 3. | 2 6 5 12 9 8 23 78 |
| 4. | 2 5 6 9 12 78 8 23 |
| 5. | 2 6 5 12 9 8 23 78 |

Ответ: _____ (4 балла)

Эталонный ответ: 1

6. Дана пирамида, размещенная в массиве: 2 5 6 12 9 78 8 23. Выберите перестановку элементов в массиве после первого шага пирамидальной сортировки по убыванию (номер ответа).

Варианты ответа:

1. 78 5 6 12 9 2 8 23
2. 5 2 6 12 9 78 8 23
3. 23 5 6 12 9 78 8 2
4. 5 9 6 12 23 78 8 2
5. 5 23 6 12 9 78 8 2

Ответ: _____ (4 балла)

Эталонный ответ: 4

7. Дан неупорядоченный массив: 5 23 6 2 9 78 8 12. Выберите перестановку элементов в массиве после первого шага сортировки по возрастанию методом вставки для значений в массиве (номер ответа).

Варианты ответа:

1. 2 23 6 5 9 78 8 12
2. 2 23 5 6 9 78 8 12
3. 5 23 6 2 9 78 8 12
4. 2 5 23 6 8 9 78 12
5. 5 23 6 2 9 78 8 12

Ответ: _____ (4 балла)

Эталонный ответ: 3

8. Дан неупорядоченный массив: 5 23 6 2 9 78 8 12. Выберите перестановку элементов в массиве после первого шага сортировки по возрастанию методом обмена для значений в массиве (номер ответа).

Варианты ответа:

1. 2 23 6 5 9 78 8 12
2. 2 23 5 6 9 78 8 12
3. 5 23 6 2 9 78 8 12
4. 2 5 23 6 8 9 78 12
5. 5 23 6 2 9 78 8 12

Ответ: _____ (4 балла)

Эталонный ответ: 4

9. Дан неупорядоченный массив: 5 23 6 2 9 78 8 12. Выберите перестановку элементов в массиве после первого шага сортировки по возрастанию методом выбора для значений в массиве (номер ответа).

Варианты ответа:

1. 2 23 6 5 9 78 8 12
2. 2 23 5 6 9 78 8 12
3. 5 23 6 2 9 78 8 12
4. 2 5 23 6 8 9 78 12
5. 5 23 6 2 9 78 8 12

Ответ: _____ (4 балла)

Эталонный ответ: 1

10. Дан неупорядоченный массив: 5 23 6 2 9 78 8 12. Выберите перестановку элементов в массиве после первого шага сортировки по возрастанию методом Шелла для значений в массиве (номер ответа).

Варианты ответа:

1. 2 23 6 5 9 78 8 12
2. 2 23 5 6 9 78 8 12
3. 5 23 6 2 9 78 8 12
4. 2 5 23 6 8 9 78 12
5. 5 23 6 2 9 78 8 12

Ответ: _____ (4 балла)

Эталонный ответ: 5

11. Дан неупорядоченный массив: 5 23 6 2 9 78 8 12. Выберите перестановку элементов в массиве после первого разделения при сортировке по возрастанию методом разделения для значений в массиве (номер ответа).

Варианты ответа:

1. 2 23 6 5 9 78 8 12
2. 2 23 5 6 9 78 8 12
3. 5 23 6 2 9 78 8 12
4. 2 5 23 6 8 9 78 12
5. 5 23 6 2 9 78 8 12

Ответ: _____ (4 балла)

Эталонный ответ: 1

12. Дан неупорядоченный массив: 5 23 6 2 9 78 8 12. Выберите перестановку элементов в массиве после первого шага сортировки по возрастанию методом восходящего слияния для значений в массиве (номер ответа).

Варианты ответа:

1. 2 23 6 5 9 78 8 12
2. 5 23 2 6 9 78 8 12
3. 2 23 5 6 9 78 8 12
4. 5 23 6 2 9 78 8 12
5. 2 5 23 6 8 9 78 12

Ответ: _____ (4 балла)

Эталонный ответ: 2

13. Дан неупорядоченный массив: 3285 0023 1459 6346 8453 2000 1578 0341. Выберите перестановку элементов в массиве после первого разделения при десятичной MSD-сортировке по возрастанию для значений в массиве (номер ответа).

Варианты ответа:

1. 0314 0023 1459 1578 2000 3285 6346 8453
2. 0023 0341 1459 1578 2000 3285 6346 8453
3. 8453 6346 3285 2000 1578 1459 0341 0023
4. 1459 1578 6346 3285 8453 0023 0341 2000
5. 2000 0341 0023 8453 3285 6346 1578 1459

Ответ: _____ (4 балла)

Эталонный ответ: 2

14. Дан неупорядоченный массив: 3285 0023 1459 6346 8453 2000 1578 0341 0009 1245. Выберите перестановку элементов в массиве после первого разделения при десятичной LSD-сортировке по возрастанию для значений в массиве (номер ответа).

Варианты ответа:

1. 0314 0023 1459 1578 2000 3285 6346 8453
2. 0023 0341 1459 1578 2000 3285 6346 8453
3. 8453 6346 3285 2000 1578 1459 0341 0023
4. 1459 1578 6346 3285 8453 0023 0341 2000
5. 2000 0341 0023 8453 3285 6346 1578 1459

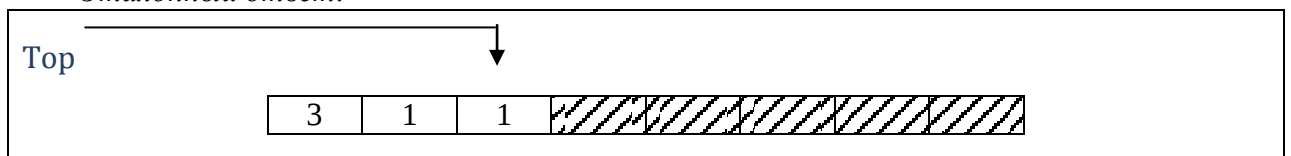
Ответ: _____ (3 балла)

Эталонный ответ: 5

15. Приведите схему стека на базе массива после серии операций: вставка (3), вставка (1), вставка (23), удаление, вставка (15), удаление, вставка (1), вставка (4), удаление. Размер массива равен 8. Стек первоначально пуст.

Ответ: (3 балла)

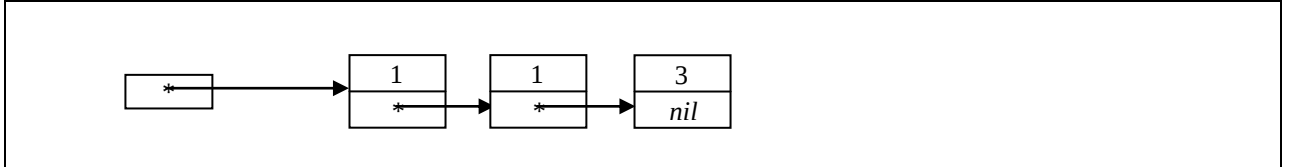
Эталонный ответ:



16. Приведите схему стека на базе односвязной структуры с адресными указателями после серии операций: вставка (3), вставка (1), вставка (23), удаление, вставка (15), удаление, вставка (1), вставка (4), удаление. Стек первоначально пуст.

Ответ: (3 балла)

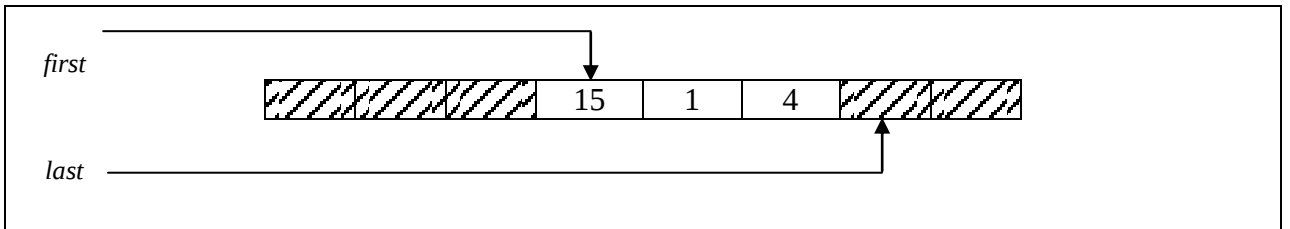
Эталонный ответ:



- 17.Приведите схему очереди на базе массива после серии операций: вставка (3), вставка (1), вставка (23), удаление, вставка (15), удаление, вставка (1), вставка (4), удаление. Размер массива равен 8, очередь первоначально пуста.

Ответ: (3 балла)

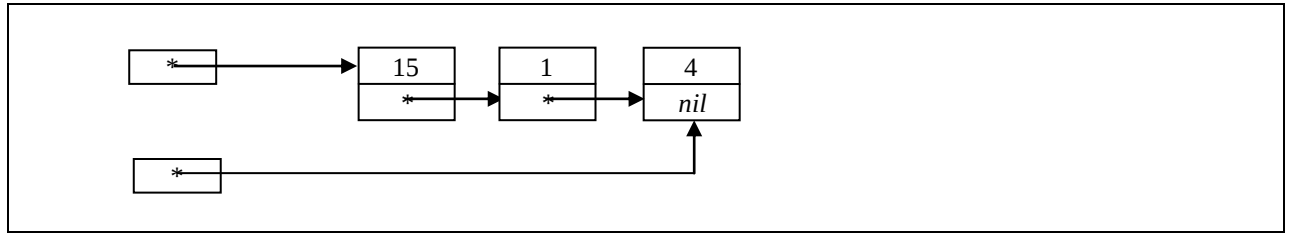
Эталонный ответ:



- 18.Приведите схему очереди на базе односвязной структуры с адресными указателями после серии операций: вставка (3), вставка (1), вставка (23), удаление, вставка (15), удаление, вставка (1), вставка (4), удаление. Очередь первоначально пуста.

Ответ: (3 балла)

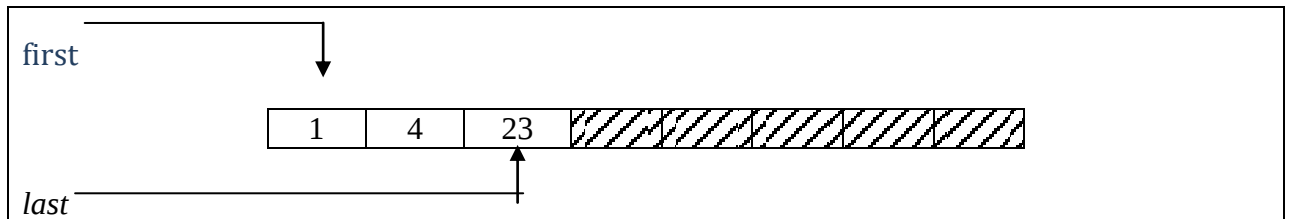
Эталонный ответ:



19.Приведите схему упорядоченного списка на базе массива после серии операций: вставка (3), вставка (1), вставка (23), вставка (15), удаление (1), удаление (3), вставка (1), вставка (4), удаление (15). Список первоначально пуст.

Ответ: (3 балла)

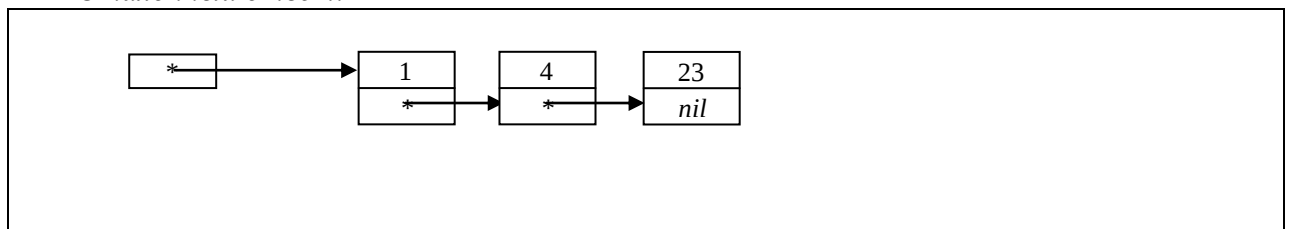
Эталонный ответ:



20.Приведите схему упорядоченного односвязного списка с адресными указателями после серии операций: вставка (3), вставка (1), вставка (23), вставка (15), удаление (1), удаление (3), вставка (1), вставка (4), удаление (15).Список первоначально пуст.

Ответ: (3 балла)

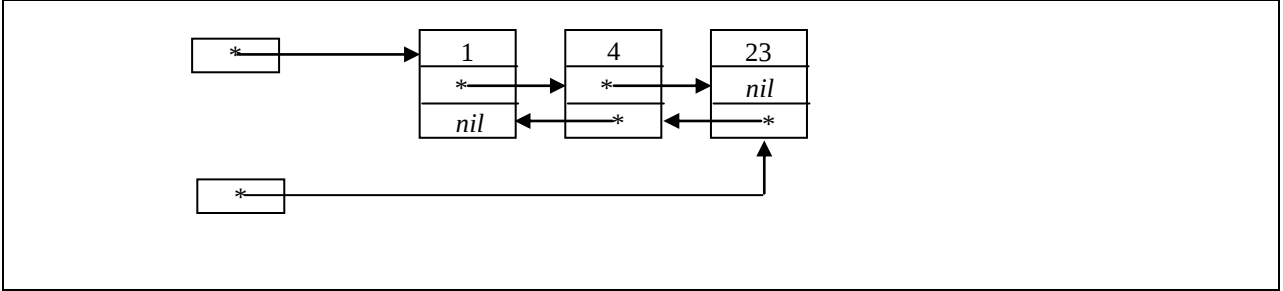
Эталонный ответ:



21.Приведите схему упорядоченного двусвязного списка с адресными указателями после серии операций: вставка (3), вставка (1), вставка (23), вставка (15), удаление (1), удаление (3), вставка (1), вставка (4), удаление (15).Список первоначально пуст.

Ответ: (3 балла)

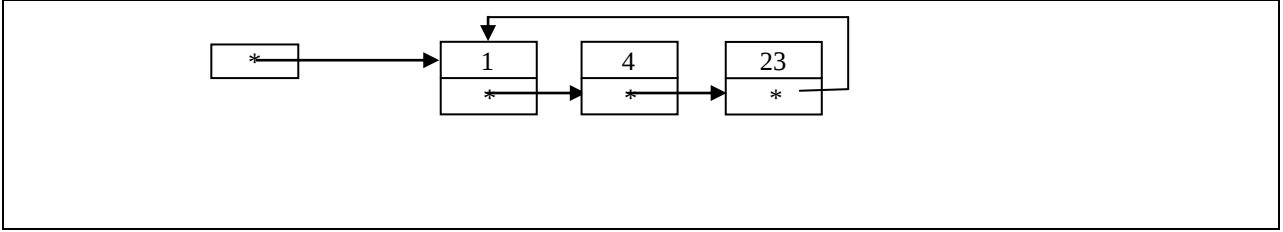
Эталонный ответ:



22. Приведите схему упорядоченного циклического односвязного списка с адресными указателями после серии операций: вставка (3), вставка (1), вставка (23), вставка (15), удаление (1), удаление (3), вставка (1), вставка (4), удаление (15).Список первоначально пуст.

Ответ: (3 балла)

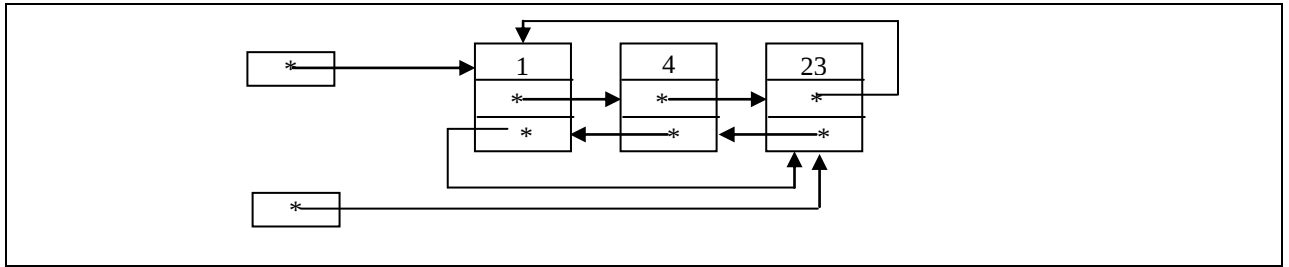
Эталонный ответ:



23.Приведите схему упорядоченного циклического двусвязного списка с адресными указателями после серии операций: вставка (3), вставка (1), вставка (23), вставка (15), удаление (1), удаление (3), вставка (1), вставка (4), удаление (15).Список первоначально пуст.

Ответ: (3 балла)

Эталонный ответ:

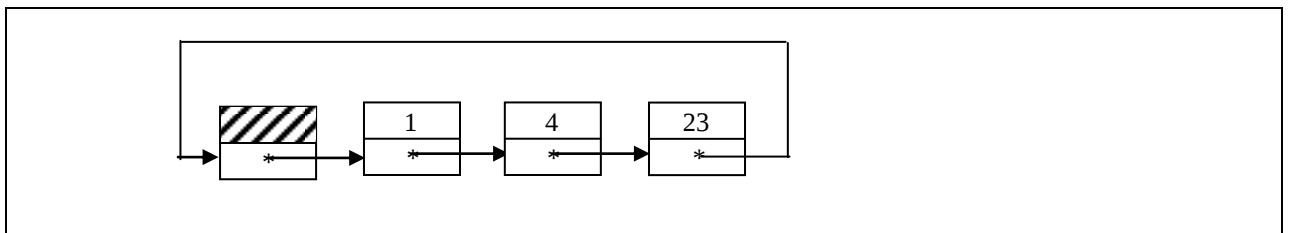


24.Приведите схему упорядоченного односвязного списка с адресными указателями, с барьерным элементом после серии операций: вставка (3), вставка (1), вставка (23), вставка (15), удаление (1), удаление (3), вставка (1), вставка (4), удаление (15).Список первоначально пуст.

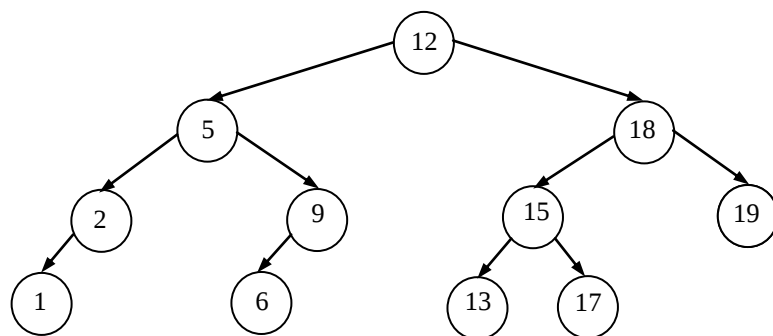
Ответ: (3 балла)



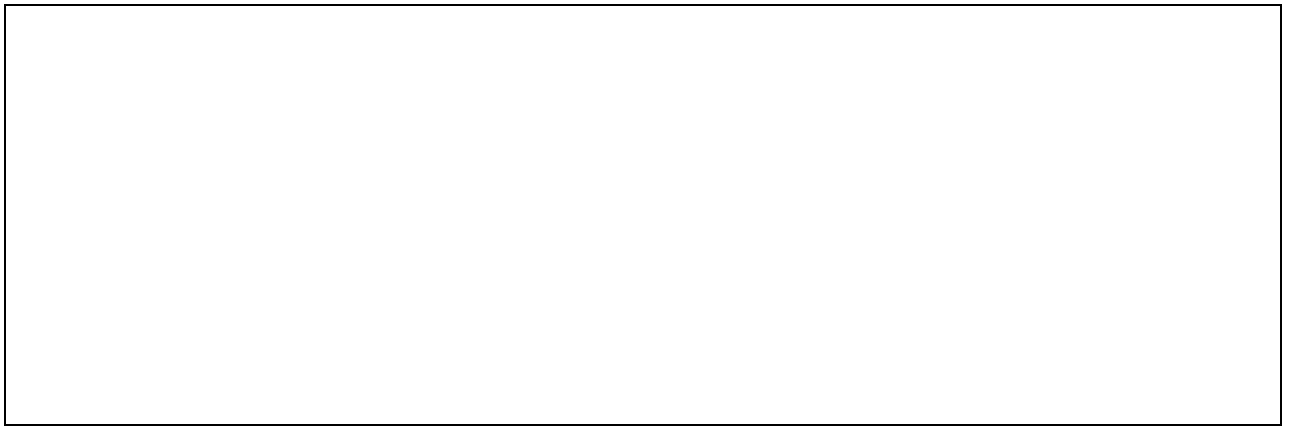
Эталонный ответ:



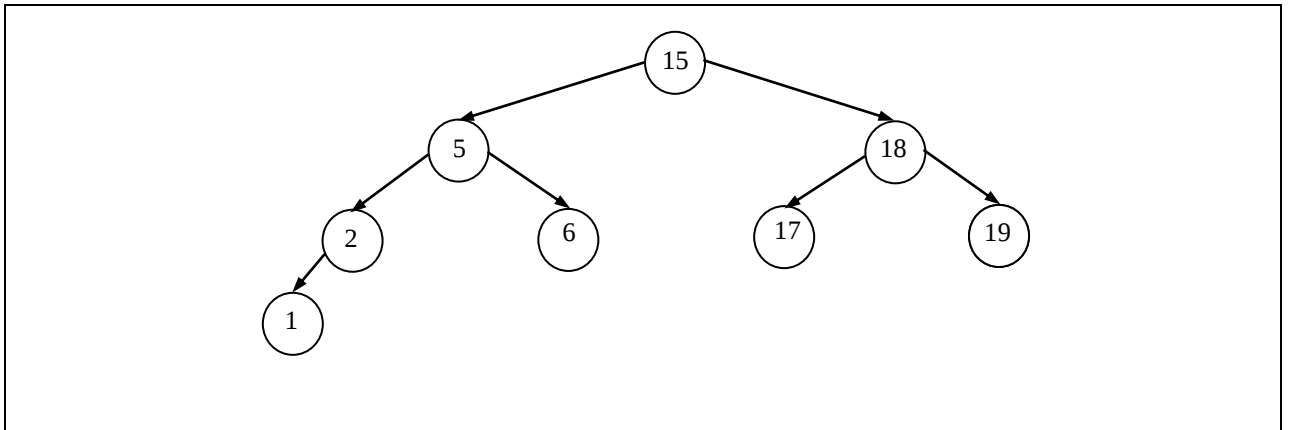
25.Приведите заданную структуру **BST**-дерева после серии операций: удаление (9), удаление (13), удаление (12).



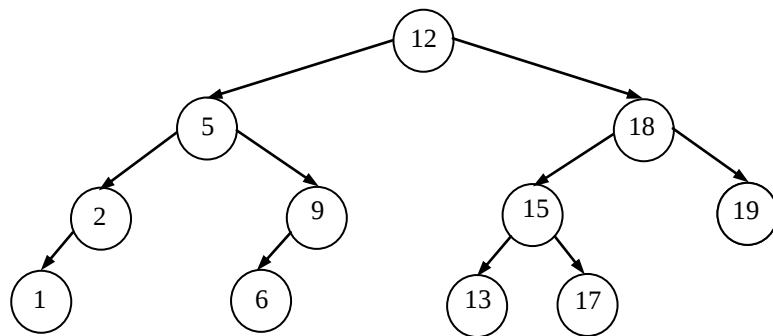
Ответ: (3 балла)



Эталонный ответ:



26.Приведите ключ элемента с порядковым номером 6.

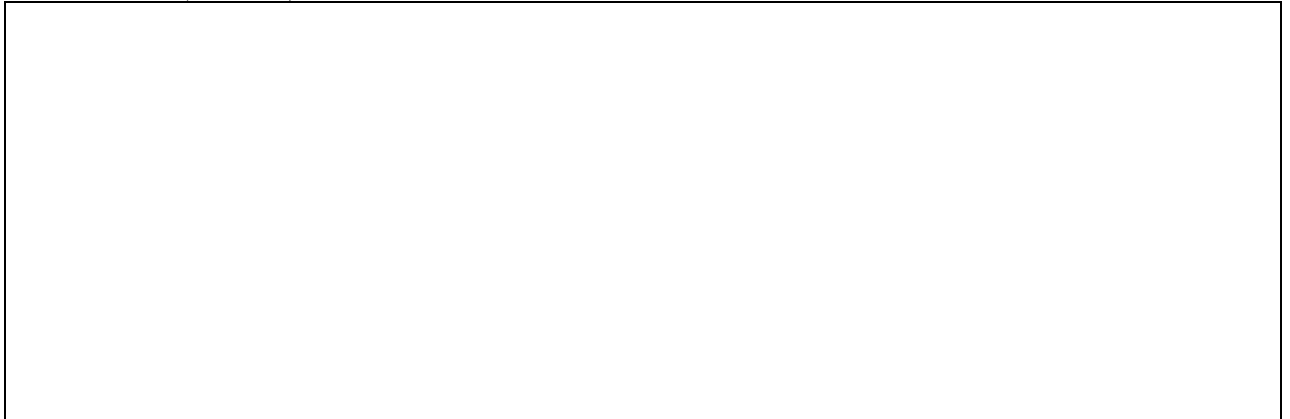


Ответ: _____ (3 балла)

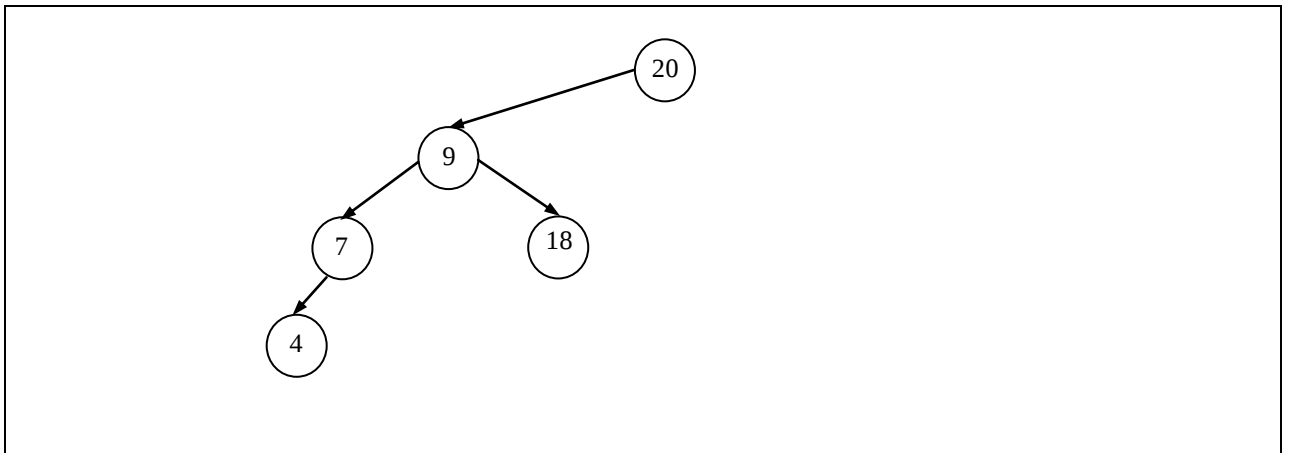
Эталонный ответ: 13

27.Приведите изображение изначально пустого **BST**-дерева после вставки в корень дерева последовательности ключей 4, 18, 7, 9, 10.

Ответ: (3 балла)

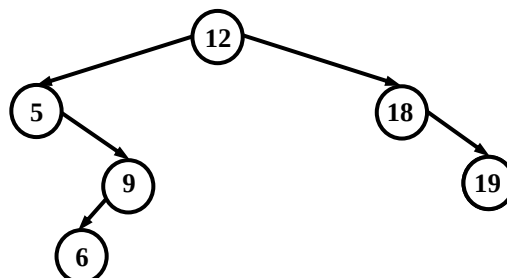


Эталонный ответ:



28.Приведите изображение рандомизированного дерева, полученного в результате работы операции вставки ключа 7:
Значение RAND_MAX = 32767.

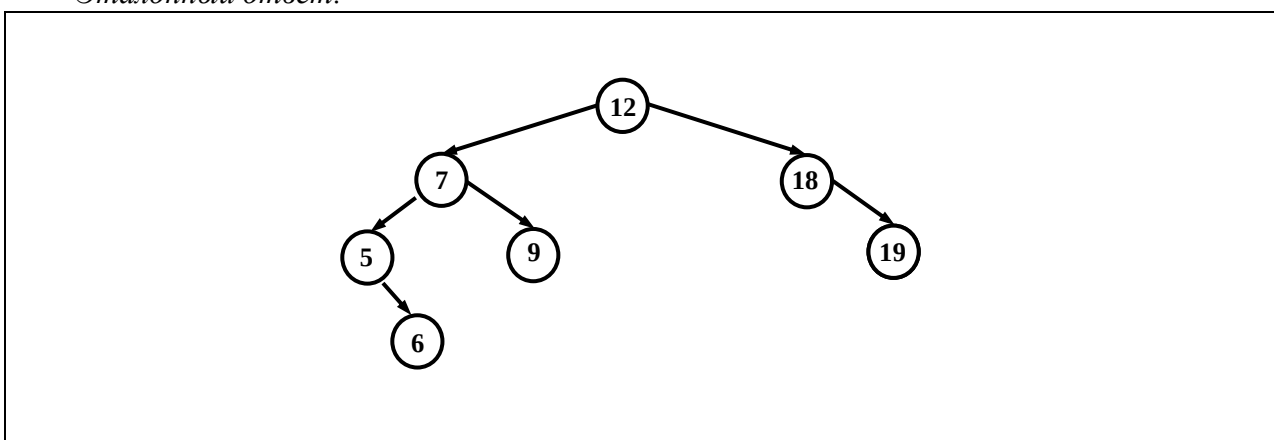
В процессе работы алгоритма генератор случайных чисел Rand() вычисляет



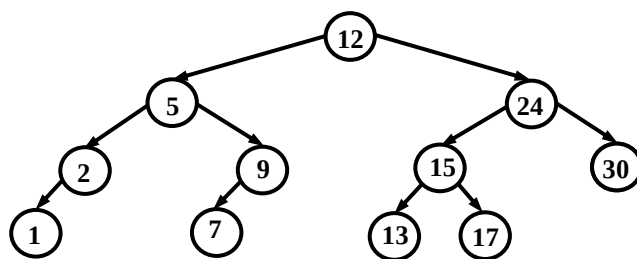
следующую последовательность значений: 6782, 653, 5187, 154, 23567,

Ответ: (5 баллов)

Эталонный ответ:

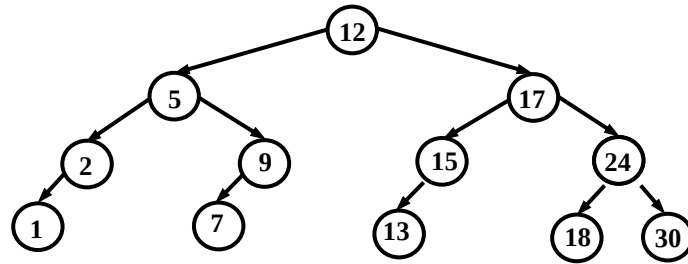


29. Приведите изображение структуры AVL – дерева после выполнения операции вставки ключа 18:



Ответ: (5 баллов)

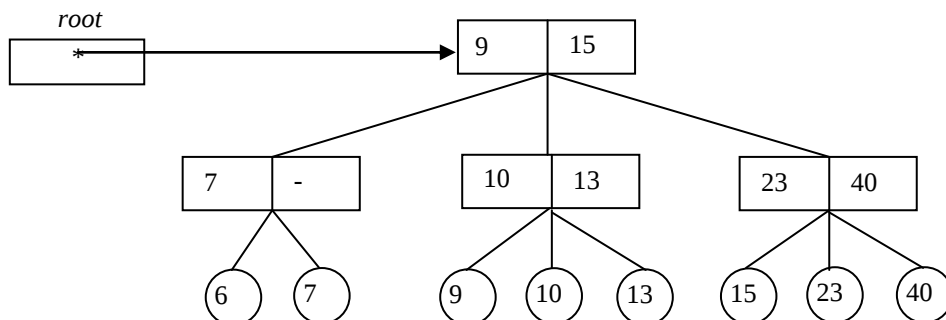
Эталонный ответ:



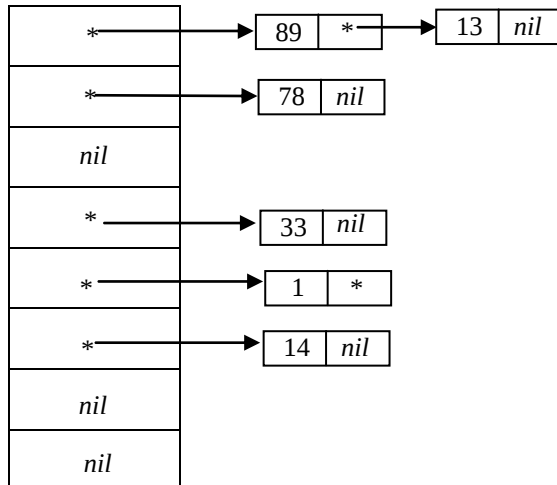
30. Приведите структуру 2 - 3-дерева после серии операций: вставка (10), вставка (6), вставка (23), вставка (15), вставка (4), вставка (9), вставка (7), вставка (13), вставка (40). Дерево первоначально пусто.

Ответ: (5 баллов)

Эталонный ответ:



Эталонный ответ:



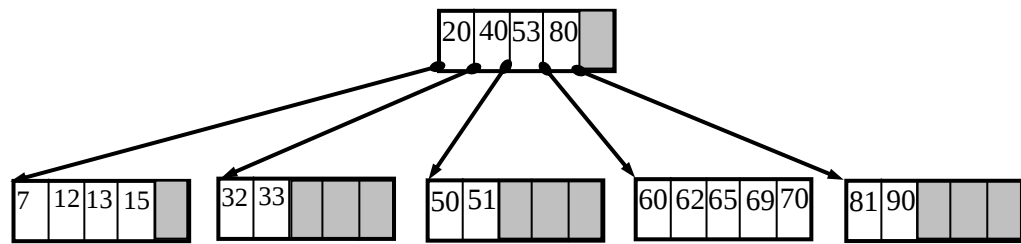
33.Приведите вид первоначально пустой хеш-таблицы с открытой адресацией после вставки ключей 1, 89, 45, 78, 2, 13. Размер хеш-таблицы равен 7, используется модульное хеширование. Метод разрешения коллизий – линейный.

Ответ: (4 балла)

Эталонный ответ:

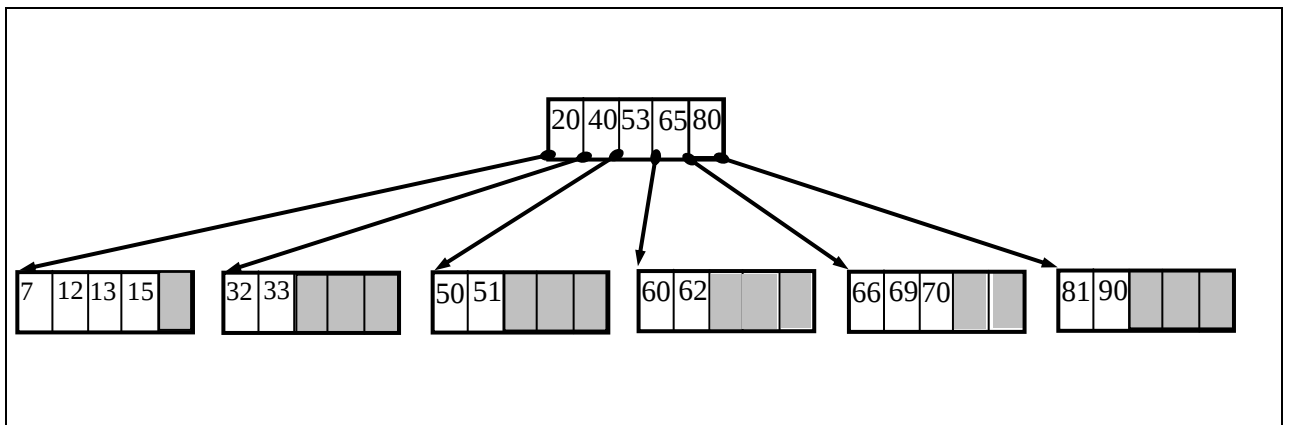
1
78
45
2
89
13

34. Приведите вид структуры В-дерева после вставки элемента с ключом 66. Минимальная степень В-дерева $t = 3$. 66.

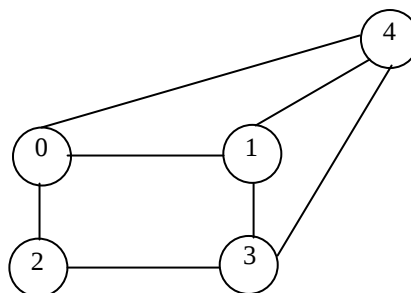


Ответ: (4 балла)

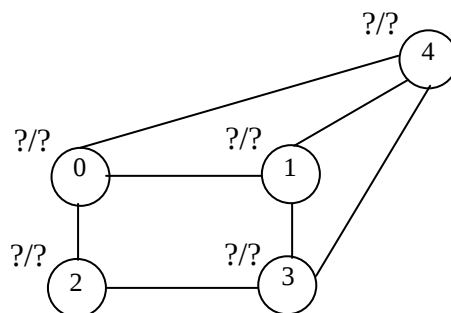
Эталонный ответ:



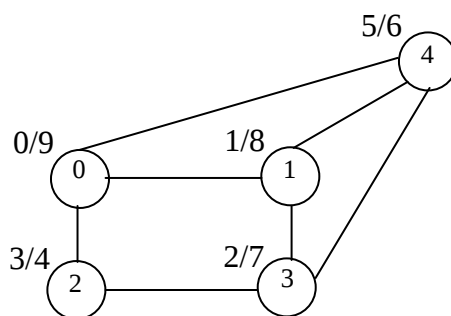
35. Дан неориентированный граф с упорядоченными списками смежности. Проставьте метки прямой и обратной нумерации при обходе графа методом поиска в глубину от вершины 0.



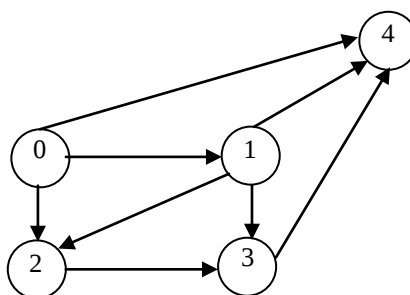
Ответ: (3 балла)



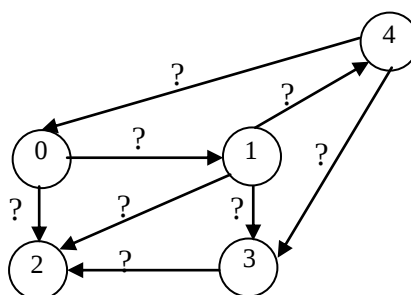
Эталонный ответ:



36. Дан ориентированный граф (орграф) с упорядоченными списками смежности. Приведите классификацию ребер на основе полного обхода в глубину от вершины 0.

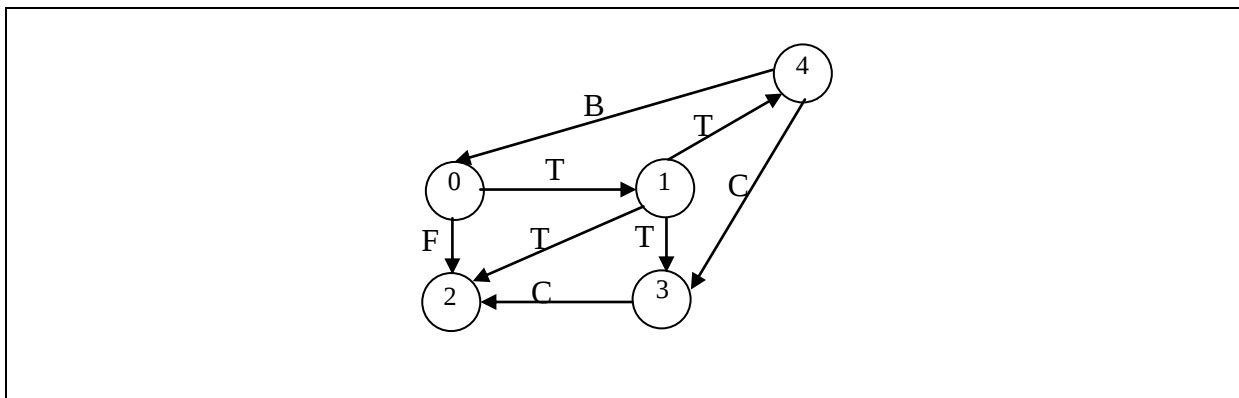


Ответ: (3 балла)

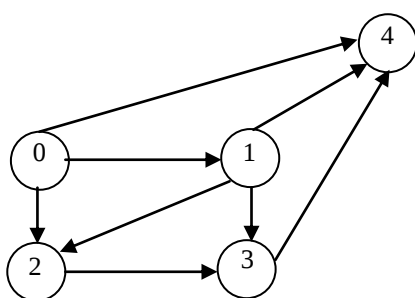


Обозначения ребер:
 Т – древесное,
 В – обратное,
 F – прямое,
 С - поперечное

Эталонный ответ:



37. Дан ориентированный граф (орграф). Составьте матрицу транзитивного замыкания



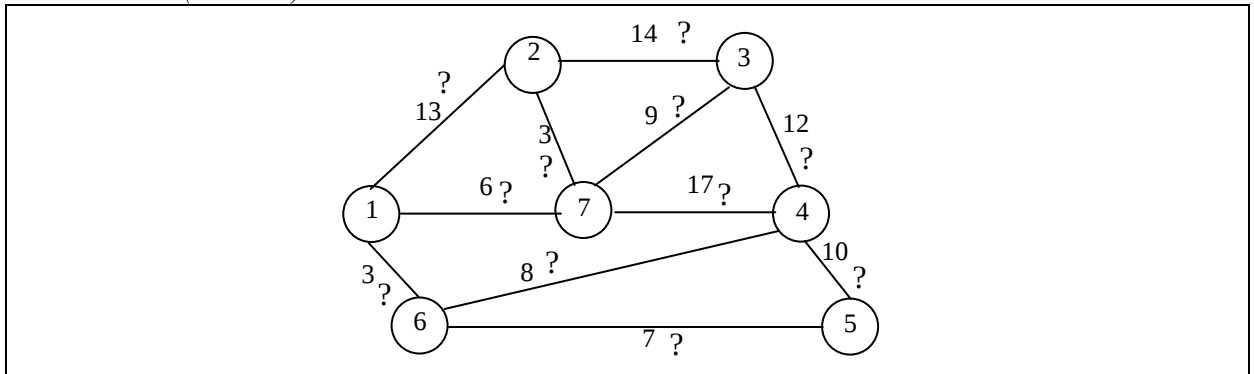
Ответ: (3 балла)

Эталонный ответ:

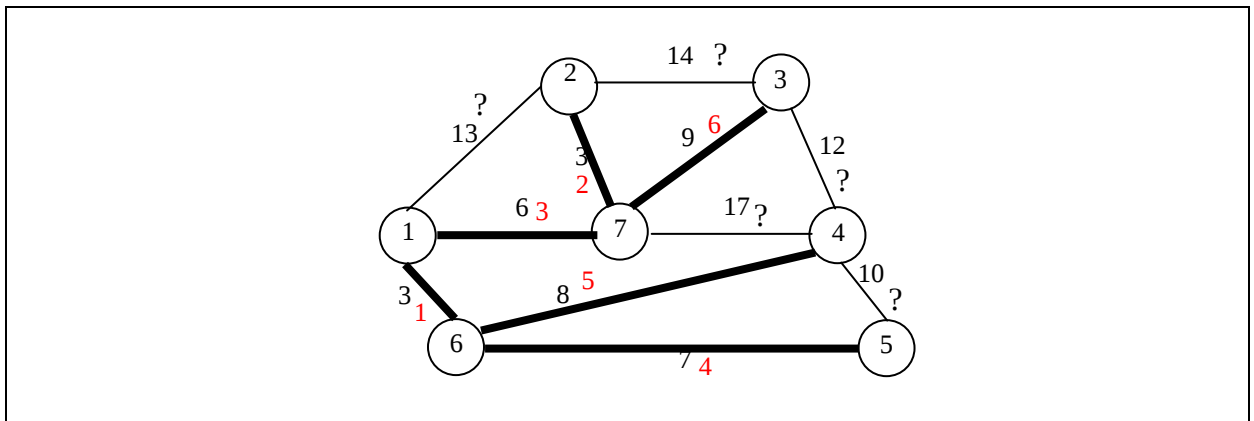
1	1	1	1	1
0	1	1	1	1
0	0	1	1	1
0	0	0	1	1
0	0	0	0	1

38. Дан неориентированный взвешенный граф. Постройте минимальное остовное дерево методом Крускала. Ребра остовного дерева выделите другим цветом на чертеже графа. Укажите, в каком порядке алгоритм Крускала выбирает ребра остова.

Ответ: (3 балла)

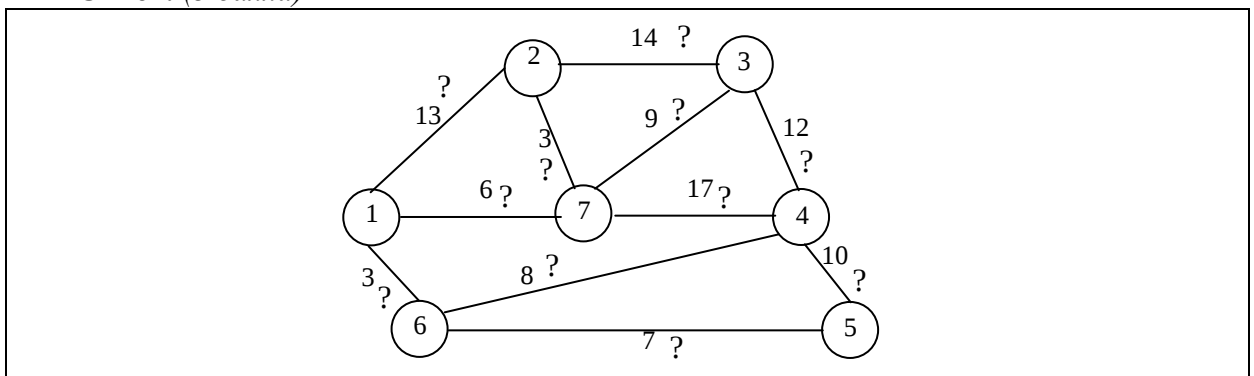


Эталонный ответ:

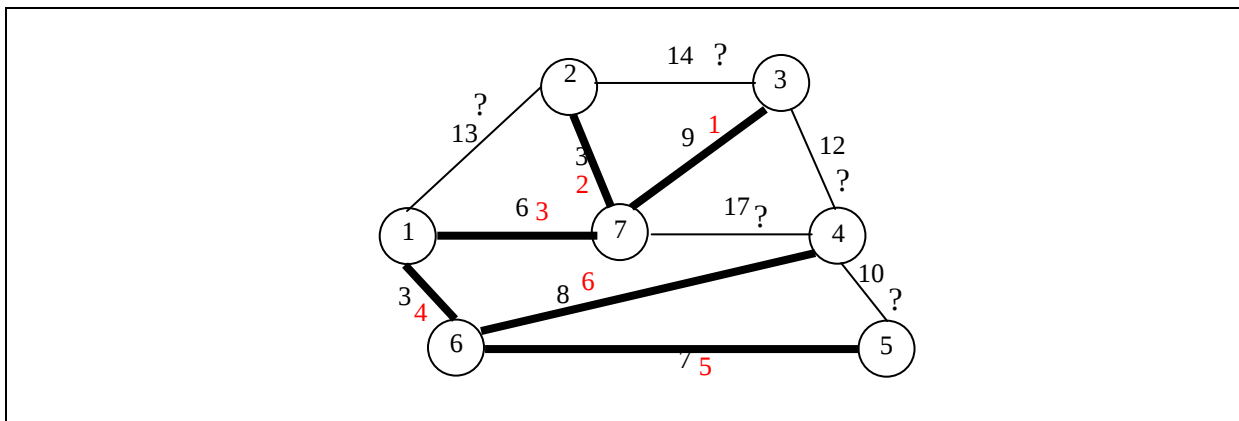


39. Дан неориентированный взвешенный граф. Постройте минимальное остовное дерево методом Прима с корневой вершиной 3. Ребра остовного дерева выделите другим цветом на чертеже графа. Укажите, в каком порядке алгоритм Прима выбирает ребра остова.

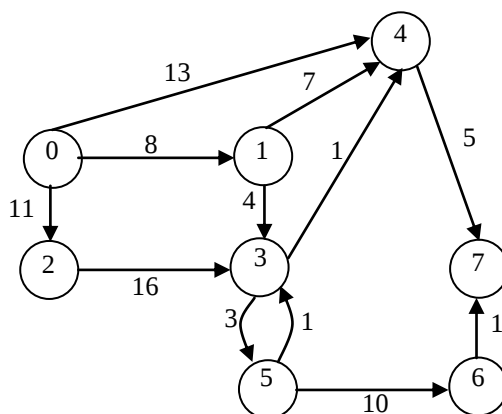
Ответ: (3 балла)



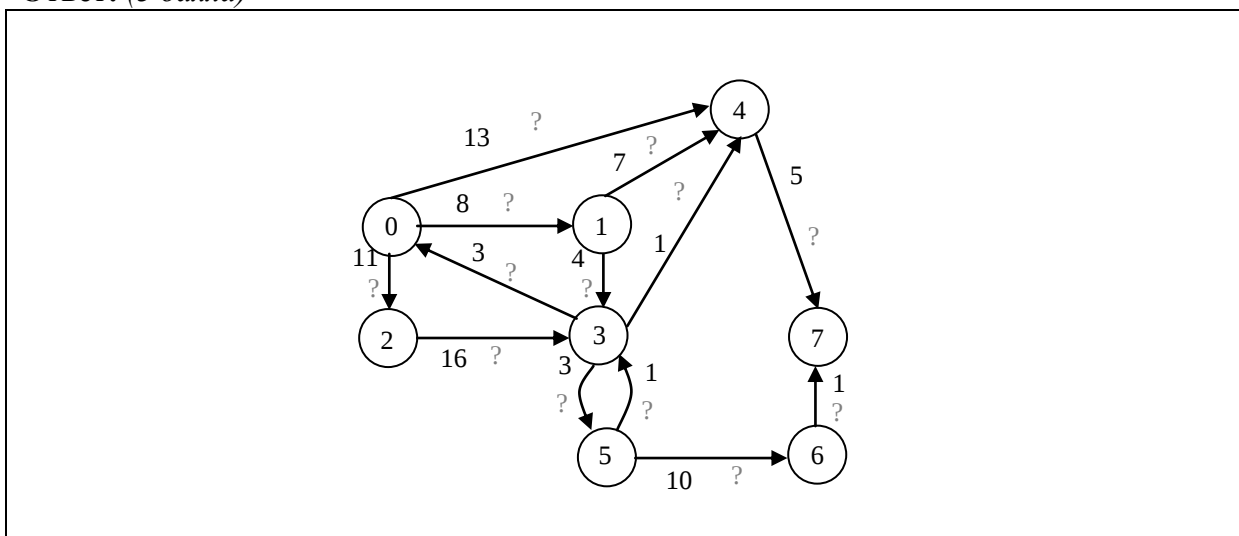
Эталонный ответ:



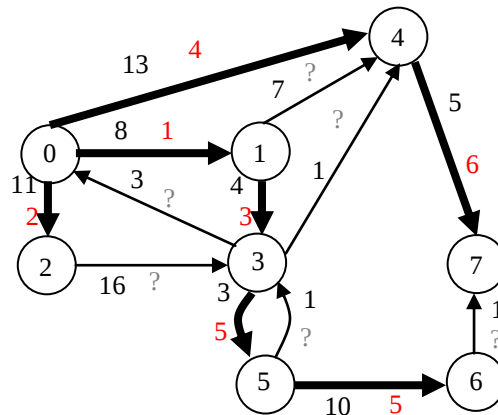
40. Дан ориентированный взвешенный граф (орграф) с упорядоченными списками смежности. Постройте все кратчайшие пути от вершины 0 до остальных вершин. Дуги, вошедшие в кратчайшие пути, выделите другим цветом на чертеже графа. Укажите порядок включения ребер в кратчайший маршрут



Ответ: (3 балла)



Эталонный ответ:



41. Составьте таблицу соответствия между номерами задач и номерами соответствующих методов поиска решения задач.

Задача

1. Обход графа в глубину
2. Поиск кратчайших путей (алгоритм Дейкстры)
3. Построение минимального остова во взвешенном графе (алгоритм Крускала)
4. Поиск минимального гамильтонова цикла в графе
5. Определение всех кратчайших путей во взвешенном графе (алгоритм Флойда)

Метод решения

1. Жадный выбор
2. Исчерпывающий поиск с возвратом
3. Поиск с возвратом
4. Метод динамического программирования

Ответ: (4 балла)

Номер задачи	Номер метода
1	
2	
3	
4	
5	

Эталонный ответ:

Номер задачи	Номер метода
1	3
2	1
3	1
4	2
5	4

**Паспорт
расчетно-графического задания (работы)**

по дисциплине «Алгоритмы и структуры данных», 5 семестр

1. Методика оценки

Задание на РГЗ направлено на достижение знаний, умений и компетенций:

- ОПК3: Готовность применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов.
- ПК.22 Способность создавать программные интерфейсы.
- з3. Знать принципы и методологии структурного и объектно-ориентированного подходов к разработке программного обеспечения.

При выполнении расчетно-графического задания (работы) студенты должны спроектировать и реализовать библиотеку классов для решения различных задач на графах. Задание к РГЗ содержит три основных пункта:

1. Реализация универсального класса для программирования графов различной природы (неориентированных, ориентированных, взвешенных) и ассоциированных с ним вспомогательных классов
2. Реализация класса для решения заданной вариантом задачи 1 на ориентированном или неориентированном графе.
3. Реализация класса для решения заданной вариантом задачи 2 на ориентированном или неориентированном взвешенном графе.

1.1 Основные этапы выполнения и структурные части РГЗ:

1.1.1 Реализация класса «Универсальный граф» и ассоциированных вспомогательных классов

Реализация и отладка класса «Граф» и вспомогательных классов выполняется с использованием технологии объектно-ориентированного программирования. Граф реализуется в виде класса «Простой граф» и ассоциированных классов «Дескриптор вершины графа», «Дескриптор ребра графа», «Итератор вершин графа», «Итератор ребер графа», «Итератор исходящих ребер вершины».

- Объект «Простой граф» реализуется в виде шаблонного класса.

Параметрами шаблона являются типы «Дескриптор вершины графа» и «Дескриптор ребра графа».

- Объект «Дескриптор вершины графа» реализуется в виде шаблонного класса. Параметрами шаблона являются тип, задающий имя вершины, и тип данных, связанных с вершиной.
- Объект «Дескриптор ребра графа» реализуется в виде шаблонного класса. Параметрами шаблона являются тип «Дескриптор вершины графа», тип, задающий вес ребра, и тип данных, связанных с ребром.
- Ассоциированные классы («Итератор вершин графа», «Итератор ребер графа», «Итератор исходящих ребер вершины») реализовать, как вложенные в класс «Простой граф».
- Для реализации различных форм представления графа используются вспомогательные классы «L – граф» и «M – граф». Рекомендуется объединить эти классы в иерархию с использованием наследования и полиморфизма.
- Объект «Простой граф» содержит вектор дескрипторов вершин, вставленных в граф.
- Объект «Простой граф» содержит структуру графа в виде списков смежности (L-граф) или матрицы смежности (M-граф). Элементы списков смежности или матрицы смежности содержат дескрипторы ребер, вставленных в граф.
- Для согласования систем идентификации вершин в прикладной программе (имена вершин) и в объекте «Простой граф» (адреса дескрипторов вершин) рекомендуется использовать объект «Отображение имени на адрес дескриптора». Объект реализуется в виде шаблонного класса. Параметрами шаблона являются тип имени вершины графа и указатель на «Дескриптор вершины графа». Такой объект обеспечивает прямой доступ по имени вершины к адресу дескриптора вершины. Объект - отображение можно реализовать на базе ассоциативной структуры (сбалансированное дерево, хеш-таблица), объекта библиотеки STL - `map` или `ash_map`, где ключом является имя вершины и т.п.

1.1.2 Реализация классов для решения задачи 1 и/или задачи2

Для реализации задач 1 - 2, заданных в варианте задания, разрабатываются объекты – клиенты для объекта «Простой граф». Объект задачи связан с объектом графа, для которого решается задача, с помощью указателя на объект «Простой граф».

- Объект задачи для решения задачи использует интерфейс объекта «Простой

граф» и интерфейс ассоциированных с графом объектов («Дескриптор вершины графа», «Дескриптор ребра графа», «Итератор вершин графа», «Итератор ребер графа», «Итератор исходящих ребер вершины»).

- Объект задачи содержит все структуры, необходимые для решения задачи, сохранения результатов решения задачи.
- Решение задачи происходит в момент создания объекта задачи, в момент вызова методов Set (g) (связывание объекта задачи с другим графом g), вызова метода Restart () (повторное решение задачи для графа).
- Метод Result () формирует результат решения задачи для клиентской программы в удобной форме, в соответствии с целью решаемой задачи.

1.2 Критерии оценки

1. Работа считается не выполненной, если не выполнены все 3 пункта задания на РГЗ или выполнен первый пункт задания с существенными отступлениями от требований, сформулированных в разделе 1.1.1. Оценка составляет 0 баллов.

2. Работа считается выполненной на пороговом уровне, если выполнен пункт 1 задания с соблюдением требований к реализации, сформулированных в разделе 1.1.1. Оценка составляет 10 баллов.

3. Работа считается выполненной на базовом уровне, если выполнены пункт 1 и один из пунктов 2 или 3 задания. При реализации пунктов должны быть соблюдены требования, задания с соблюдением требований к реализации, сформулированные в разделах 1.1.1 и 1.1.2. Оценка составляет 20 баллов.

4. Работа считается выполненной на продвинутом уровне, если выполнены все три пункта задания к РГЗ. При этом соблюдены все требования к реализации, сформулированные в разделах 1.1.1 и 1.1.2. Оценка составляет 30 баллов.

1.3 Шкала оценки

В общей оценке по дисциплине баллы за РГЗ(Р) учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

Для аттестации студентов используется балльно-рейтинговая система. Рейтинг студента по дисциплине определяется как сумма баллов за работу в семестре (текущий рейтинг) и баллов, полученных в результате промежуточной аттестации (дифференцированный зачет).

Максимальная сумма баллов за 5 семестр составляет 100 (текущий рейтинг - 80

баллов, дифференцированный зачет - 20 баллов). Минимальная сумма баллов за 5 семестр - 50 баллов.

В таблице 1 приведено максимальное количество баллов, которое может набрать студент по разным видам учебной деятельности в течение 5 семестра и диапазоны баллов, соответствующие минимальному и максимальному количествам баллов. Максимальная сумма баллов за 5 семестр составляет 100 (текущий рейтинг – 80 баллов, зачет – 20).

Таблица 1 Сроки и диапазоны баллов для всех видов учебной работы

№п/п	Вид учебной работы (учебной деятельности)	Максимальное количество баллов	Диапазоны баллов	Срок представления и защиты (неделя семестра)
Пятый семестр:				
1	Лабораторная работа № 1	6	4-8	7
2	Лабораторная работа № 2	8	6 - 12	11
3	Лабораторная работа № 3	12	7 - 14	15
4	Лабораторная работа № 4	14	8- 16	17
5	Расчетно-графическое задание	30	10 - 30	17
Итого по текущему рейтингу:		80	40 - 80	
6	Дифференцированный зачет	20	10-20	
Итого за пятый семестр:		100	50-72 (удовл.) 73-87 (хор.) 88-100 (отл.)	

В результате текущей и промежуточной аттестации итоговые баллы переводятся в оценку системы ECTS и пятибальную шкалу согласно таблице 2:

Таблица 2 Сроки и диапазоны баллов для всех видов учебной работы

90 - 100	A+ (99 - 100)	отлично
	A (93 - 98)	
	A- (90 - 92)	
80 - 89	B+(87 - 89)	хорошо
	B (83 - 86)	
	B-(80 - 82)	
70 - 79	C+(77- 79)	удовлетворительно
	C (73 -76)	
	C-(70 - 72)	
60 - 69	D+(67 - 69)	не удовлетворительно
	D (63 -66)	
	D-(60 - 62)	
50 - 59	E (50 -59)	
25 - 49	FX	не удовлетворительно
0 - 24	F	

Правила текущей аттестации РГЗ:

В течение 5-го семестра обучения (до 17 учебной недели включительно) необходимо представить и защитить расчетно-графическое задание (РГЗ) согласно варианту.

Максимальное количество баллов - 30 выставляется в случае, если успешно выполнены пункты 1- 3 задания к РГЗ при условии своевременной (до конца 17 учебной недели включительно) и успешной защиты.

Количество баллов 20 выставляется в случае, если успешно выполнены пункты 1, и один из пунктов 2-3 к РГЗ при условии своевременной и успешной защиты.

Минимальное количество баллов 10 выставляется в случае, если успешно выполнен пункт 1 задания к РГЗ при условии своевременной и успешной защиты.

В случае пересдачи или представления и защиты РГЗ с опозданием от учебного графика происходит потеря баллов (опоздание на 1 неделю - потеря 1 балла).

2. Примерный перечень тем РГЗ(Р)

2.1 Пункт 1 задания

Реализация универсального класса для программирования графов различной природы (неориентированных, ориентированных, взвешенных) и ассоциированных с ним вспомогательных классов.

2.1.1 Разработать АТД «Простой граф».

Интерфейс АТД «Простой граф» включает операции:

Конструктор () по умолчанию: создает пустой L - граф с нулевым числом вершин и ребер,

Конструктор (V, D, F) создает граф с V вершинами, без ребер, типа D (ориентированный / неориентированный), формы представления F (L- граф/М-граф),

Конструктор (V, E, D, F) создает граф с V вершинами, с E случайными ребрами, типа

D (ориентированный / неориентированный), формы представления F (L- граф/М-граф),

Конструктор (G) - конструктор копирования создает объект – копию графа G,

Деструктор () уничтожает внутренние структуры графа,

V () - возвращает число вершин в графе,

E () - возвращает число ребер в графе,

Directed () - возвращает тип графа (ориентированный / неориентированный)

Dense () - возвращает форму представления графа (L- граф / M- граф),

K () - возвращает коэффициент насыщенности графа,

ToListGraph () преобразует граф к L- графу,

ToMatrixGraph () преобразует граф к M- графу,

InsertV () добавляет вершину к графу и возвращает адрес дескриптора вновь созданной вершины,

DeleteV (v) - удаляет вершину из графа, заданную адресом дескриптора **v**,

InsertE(v1, v2) - добавляет ребро (**v1, v2**) к графу, соединяющую вершины, заданные адресами дескрипторов **v1** и **v2**, и возвращает адрес дескриптора вновь созданного ребра - **e**,

DeleteE (v1, v2) удаляет ребро, соединяющее вершины, заданные адресами дескрипторов **v1** и **v2**,

GetEdge (v1, v2) возвращает адрес дескриптора ребра соединяющего вершины, заданные дескрипторами **v1** и **v2**,

2.1.2 Разработать ассоциированные с графом типы:

АТД «Дескриптор вершины графа»

Дескриптор вершины содержит поля:

name – имя вершины,

data – данные, связанные с вершиной,

index – индекс вершины в структуре графа или -1,

Интерфейс АТД «Дескриптор вершины графа» включает операции:

Конструктор (): поле **name** не определено, поле **data** не определено,

Конструктор (name, data): **name** - имя вершины, **data** - данные, связанные с вершиной,

GetName () - возвращает имя вершины,

GetData () - возвращает данные, связанные с вершиной,

SetName (name) – задает имя вершины,

SetData (data) – записывает данные **data** в дескриптор вершины.

АТД «Дескриптор ребра графа»

Дескриптор ребра содержит поля:

v1 - дескриптор вершины, из которой исходит ребро,

v2 - дескриптор вершины, в которую входит ребро,

w - вес ребра,

data - данные, связанные с ребром,

Интерфейс АТД «Дескриптор ребра графа» включает операции:

Конструктор (v1, v2): *v1* - дескриптор вершины, из которой исходит ребро, *v2* - дескриптор вершины, в которую входит ребро,

Конструктор (v1, v2, w): *v1* - дескриптор вершины, из которой исходит ребро, *v2* - дескриптор вершины, в которую входит ребро, *w* - вес ребра,

Конструктор (v1, v2, w, data): *v1* - дескриптор вершины, из которой исходит ребро, *v2* - дескриптор вершины, в которую входит ребро, *w* - вес ребра, *data* - данные, связанные с ребром

v1() - возвращает дескриптор вершины, из которой исходит ребро,

v2() - возвращает дескриптор вершины, в которую входит ребро,

GetW () - возвращает вес ребра,

SetW (w) - изменение веса ребра,

GetData () - возвращает данные, связанные с ребром,

SetData (data) - изменение данных, связанных с ребром.

АТД «Итератор вершин графа»

Интерфейс АТД «Итератор вершин графа» включает операции:

Конструктор () - создает итератор вершин графа,

beg () - возвращает итератор, установленный на первую вершину графа,

end () - возвращает итератор, соответствующий окончанию переходов итератора,

operator ++ - переход к следующей вершине графа,

operator * - возвращает дескриптор вершины графа, на которую указывает итератор.

АТД «Итератор ребер графа»

Интерфейс АТД «Итератор ребер графа» включает операции:

Конструктор () - создает итератор ребер графа,

beg () - возвращает итератор, установленный на первое ребро графа,

end () - возвращает итератор, соответствующий окончанию переходов итератора,

operator ++ - переход к следующему ребру графа,

operator * - возвращает дескриптор ребра графа, на которое указывает итератор.

АТД «Итератор исходящих ребер вершины»

Интерфейс АТД «Итератор исходящих ребер вершины» включает операции:

Конструктор (v) - создает итератор исходящих ребер графа для вершины, заданной дескриптором *v*,

beg () - возвращает итератор, установленный на первое исходящее ребро вершины,
end () - возвращает итератор, соответствующий окончанию переходов итератора,
operator ++ - переход к следующему исходящему ребру,
operator * - возвращает дескриптор исходящего ребра вершины, на которое указывает итератор.

2.2 Пункт 2 задания

Спроектировать и реализовать шаблонный класс для АТД «Задача 1» в соответствии с вариантом и использовать для решения задачи на неориентированном или ориентированном графе.

Интерфейс АТД «Задача 1» включает операции:

Конструктор (*g*) - создает объект задачи 1, ассоциированный с графом *g*, и выполняет решение задачи для графа *g*,

Конструктор (*T*) - конструктор копирования создает копию объекта – задачи *T*,

Деструктор () - уничтожает внутренние структуры объекта задачи,

Set (*g*) – связывает объект задачи с графом *g* и выполняет решение задачи 1 для графа *g*,

Restart () – повторно выполняет решение задачи 1 для графа *g*,

Result () – возвращает результат решения задачи 1

Варианты задачи 1:

- 1) формирования двудольного неориентированного графа с раскраской в два цвета,
- 2) определение кратчайших по числу ребер путей между всеми парами вершин орграфа,
- 3) определение гамильтонова цикла в орграфе,
- 4) определение диаметра связного неориентированного графа,
- 5) определение последовательности вершин в результате прямой топологической сортировки ациклического орграфа DAG (сортировка - на основе очереди истоков),
- 6) построение остовного дерева орграфа в пределах заданной высоты за счёт добавления минимального числа рёбер,
- 7) определение последовательности вершин на основе топологической сортировки ациклического орграфа DAG на основе поиска в глубину,
- 8) поиск всех простых циклов, включающих заданную вершину орграфа,
- 9) формирование реберно-связного неориентированного графа,
- 10) поиск простого цикла заданной длины, включающего заданную вершину,

- 11) построение редуцированного графа сильносвязных компонент,
- 12) классификация всех рёбер относительно заданной вершины орграфа,
- 13) определение вершин, отстоящих на расстоянии d от заданной вершины (d – число рёбер),
- 14) формирование двусвязного неориентированного графа,
- 15) определение эйлера цикла в неориентированном графе,
- 16) формирование списка ребер неориентированного графа в порядке двухпроходного эйлера цикла.

2.3 Пункт 3 задания

Спроектировать и реализовать шаблонный класс для АТД «Задача 2» в соответствии с вариантом и использовать для решения задачи на взвешенном графе.

Интерфейс АТД «Задача 2» включает операции:

Конструктор (g) - создает объект задачи 3, ассоциированный с графом g , и выполняет решение задачи для графа g ,

Конструктор (T) - конструктор копирования создает копию объекта – задачи T ,

Деструктор () - уничтожает внутренние структуры объекта задачи,

Set (g) – связывает объект задачи с графом g и выполняет решение задачи 3 для графа g ,

Restart () – повторно выполняет решение задачи 3 для графа g ,

Result() – возвращает результат решения задачи 3

Варианты алгоритмов:

- 1) определение диаметра и пути, соответствующего диаметру, на основе алгоритма Флойда,
- 2) приведение отрицательной весовой функции орграфа к положительной методом Джонсона,
- 3) определение входящих в заданную вершину кратчайших путей на основе алгоритма Дейкстры,
- 4) определение центра взвешенного орграфа на основе алгоритма Беллмана – Форда.
- 5) разбиение вершин взвешенного неориентированного графа на кластеры, объединяющие вершины ребрами с длиной, большей d . Для разбиения использовать алгоритм Крускала,
- 6) определение периферии взвешенного орграфа на основе алгоритма Дейкстры,

- 7) определение списка вершин, которые лежат в пределах заданного расстояния d от заданной вершины взвешенного орграфа с отрицательной весовой функции на основе алгоритма Беллмана-Форда,
- 8) определение первого по величине (по количеству вершин) кластера в неориентированном взвешенном графе, вершины которого объединены ребрами с длиной, большей d . Для определения кластера использовать алгоритм Прима,
- 9) определение периферии взвешенного орграфа на основе алгоритма Флойда,
- 10) нахождения ребра (ребер), устранение которых вызывает максимальное возрастание кратчайшего пути из вершины u в вершину v во взвешенном орграфе на основе алгоритма Дейкстры,
- 11) определения кратчайших путей для всех пар вершин взвешенного орграфа на основе алгоритма Беллмана-Форда,
- 12) построение минимального остовного дерева относительно заданной взвешенного неориентированного графа методом Прима,
- 13) определение радиуса и соответствующего радиусу пути взвешенного орграфа на основе алгоритма Дейкстры,
- 14) определение центра взвешенного орграфа на основе алгоритма Флойда,
- 15) определение эксцентриситета заданной вершины взвешенного орграфа с отрицательной весовой функции на основе алгоритма Беллмана-Форда,
- 16) формирование остовного дерева взвешенного неориентированного графа с суммарным весом ребер, ближайшим к заданному весу d . Для формирования дерева использовать алгоритм Крускала,