

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
БУРЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ ДОРЖИ БАНАЗАРОВА
ИНСТИТУТ МАТЕМАТИКИ, ФИЗИКИ И КОМПЬЮТЕРНЫХ НАУК
КАФЕДРА ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ И ИНФОРМАТИКИ

Утверждена на заседании
Ученого совета ИМФКН
«__»_____ 202__ г.
Протокол № __

Рабочая программа дисциплины
Курс по программированию

Направление подготовки / специальность
09.04.02 Информационные системы и технологии
Профиль

Проектирование, разработка и эксплуатация информационных систем

Квалификация (степень) выпускника
Магистр

Форма обучения
Очная

Улан-Удэ
2025

Пояснительная записка

Цели освоения дисциплины

Цель изучения факультативной дисциплины "Курс по программированию" - изучение основных парадигм современного программирования; освоение методов и средств разработки алгоритмов и программ для решения прикладных задач, позволяющих выпускнику успешно работать в избранной сфере деятельности с применением современных компьютерных технологий.

Место дисциплины в структуре образовательной программы

Данная дисциплина входит в блок ФТД для направления подготовки 09.04.02 Информационные системы и технологии. Полученные умения и навыки дают возможность овладеть основами теоретических и практических знаний в области создания программных приложений.

Планируемые результаты обучения по дисциплине и индикаторы достижения компетенций.

ОПК-8. Способен осуществлять эффективное управление разработкой программных средств и проектов

ОПК-8.2. Осуществляет выбор средств разработки, оценивает сложность проектов, планирует ресурсы, контролирует сроки выполнения и оценивает качество полученного результата

В результате освоения дисциплины студент должен:

Знать:

основы современных парадигм, технологий программирования;
методы алгоритмизации и технологии программирования, пригодные для практического применения в области информационных систем и технологий

Уметь:

проводить разработку и анализ алгоритмов;
программировать алгоритм, используя средства языка высокого уровня

Владеть:

способностью формализовать прикладную задачу, выбирать для неё подходящие структуры данных и алгоритмы обработки;
навыками разработки программы, ее отладки и тестирования

Планируемые результаты освоения образовательной программы:

Объем дисциплины в зачетных единицах с указанием количества часов, выделенных на контактную работу обучающихся с преподавателем и на самостоятельную работу обучающихся

Общая трудоемкость дисциплины составляет 1 зачетные единицы, 36 часа.

№	Название разделов дисциплины	Лабораторная работа	Самостоятельная работа
Семестр 3		22	14
1	Анализ вычислительной сложности алгоритмов.	10	8
	Алгоритмы обработки данных		
2	Структуры данных	12	6

Тематическое планирование курса

Анализ вычислительной сложности алгоритмов. Алгоритмы обработки данных

Семестр 3

Анализ вычислительной сложности алгоритмов

Лабораторная работа. 2(0) ч. Основы анализа. Оценка размера входных данных. Порядок роста. Эффективность алгоритма в разных случаях. Анализ вычислительной сложности алгоритмов.

Лабораторная работа. 2(0) ч. Асимптотический анализ. Асимптотические отношения. Асимптотический анализ и пределы.

Лабораторная работа. 2(0) ч. Рекурсивные алгоритмы. Вычислительная сложность рекурсивных алгоритмов. Асимптотический анализ рекурсивных алгоритмов. Анализ вычислительной сложности рекурсивных алгоритмов.

Самостоятельная работа. 4(0) ч. Вычислительная сложность рекурсивных алгоритмов. Асимптотический анализ рекурсивных алгоритмов.

Алгоритмы обработки данных

Лабораторная работа. 2(0) ч. Общая задача сортировки. Классификация алгоритмов сортировки. Алгоритмы сортировки с квадратичной вычислительной сложностью.

Алгоритмы сортировки с линейно-логарифмической вычислительной сложностью.

Алгоритмы сортировки, неоснованные на сравнениях. Анализ вычислительной сложности алгоритмов сортировок в различных случаях

Лабораторная работа. 2(0) ч. Общая задача поиска. Последовательный поиск. Бинарный поиск. Поиск подстроки в строке. Реализация различных алгоритмов поиска.

Самостоятельная работа. 4(0) ч. Алгоритмы сортировки, неоснованные на сравнениях.

Структуры данных

Семестр 3

Линейные структуры данных. Нелинейные структуры данных

Лабораторная работа. 2(0) ч. Стеки. Очереди. Деки. Понятие стека. Способы реализации стека. Алгоритмы, использующие стек. Префиксная и постфиксная нотации

арифметических выражений. Понятие очереди. Алгоритмы, использующие очередь.

Способы реализации очереди. Модель электронной очереди банка. Понятие дека.

Способы реализации дека. Алгоритмы, использующие дек. Реализация односвязных и двусвязных списков

Лабораторная работа. 4(0) ч. Реализация односвязных и двусвязных списков

Лабораторная работа. 2(0) ч. Деревья. Понятие нелинейной структуры данных. Дерево как частный случай графа. Бинарные деревья. Обходы бинарных деревьев. Бинарные деревья

поиска. Кучи. Структуры данных множество и очередь с приоритетом. АВЛ-деревья

Лабораторная работа. 2(0) ч. Графы. Понятие графа. Виды графов. Способы представления графов. Матрица смежности. Списки смежности. Виды обходов графа.

Задачи на графах. Поиск минимального остовного дерева. Поиск кратчайших путей на графе.

Лабораторная работа. 2(0) ч. Хэш-таблицы. Структуры хранения данных. Понятие хэш-таблицы. Способы разрешения коллизий. Открытая адресация. Метод цепочек. Хэш-

функции: числовые и строковые. Алгоритмы кодирования данных.

Самостоятельная работа. 6(0) ч. Поиск кратчайших путей на графе.

БРС

Семестр	Контрольные точки	Баллы
3	Текущий контроль в разделе «Структуры данных »	
	Тест	30
	Контрольная работа	30
3	Зачет	
	Устный ответ на вопрос экзаменационного билета	15
	Решение комплектов задач	25
Итого за семестр 3:		100

Учебно-методическое и информационное обеспечение учебного процесса

Образовательные технологии (в том числе на занятиях, проводимых в интерактивных формах).

Теоретическая часть курса, общие вопросы курса по программированию излагаются в лекционном курсе. Отдельные вопросы могут выноситься на самостоятельное изучение. Для приобретения навыков работы на ПК предназначены лабораторные занятия. При изучении дисциплины используются интерактивные формы занятий (лекция-дискуссия, защита рефератов).

- [Методич рек СРС.doc](#)

Режим доступа: https://my.bsu.ru/content/file/2/21/213/69325_metodich_rek_srs.doc

Учебно-методические материалы, в том числе методические указания для обучающихся по освоению дисциплины

Теоретическая часть курса, общие вопросы методики и технологий применения компьютерных средств излагаются преподавателем в лекционном курсе. Отдельные вопросы могут выноситься на самостоятельное изучение. Студент должен иметь в виду, что на лекциях преподаватель определяет такие вопросы и рекомендует необходимую для их изучения литературу, источники и др. ресурсы. Для успешного освоения курса необходимо внимательно фиксировать основные положения лекции, своевременно их усваивать, при необходимости самостоятельно прорабатывать, используя основную и дополнительную литературу.

Для приобретения навыков общения с ПК в процессе освоения инструментальных систем и отладки программ предназначены лабораторные занятия. Лабораторные занятия проводятся в специальных классах, оборудованных средствами вычислительной техники. На первом лабораторном занятии студенты получают инструктаж по технике безопасности при работе в классе и знакомятся с особенностями работы на конкретной вычислительной машине. Последующие лабораторные работы заключаются в освоении инструментальных систем и отладке программ решения типовых задач. Индивидуальные задания и методические указания к выполнению каждой последующей лабораторной работы студент получает, как правило, на предыдущем занятии. Подготовка к

выполнению лабораторных работ осуществляется в часы самостоятельной работы. Студенты, не подготовившиеся к занятиям, к работе на компьютере не допускаются. По каждой выполненной лабораторной работе студент оформляет отчет по установленной форме.

Самостоятельные занятия под контролем преподавателя предназначены для самостоятельного изучения студентами тех разделов курса, по которым не предусмотрено чтение лекций, либо проводятся лекции обзорного характера. По усмотрению преподавателя в часы индивидуальных занятий студентам может поручаться выполнение других заданий. Занятия проводятся с академической группой или с половиной группы в часы, установленные расписанием занятий. На занятиях студент должен иметь конспект лекций, учебную и справочную литературу, отдельную тетрадь для записей. Весь теоретический материал, изученный в процессе индивидуальных занятий, должен быть законспектирован.

Оценочные средства

По данной дисциплине разработаны оценочные средства, критерии их оценивания, а также методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

- [Методич рек СРС.doc](#)
- [ФОС Курс по программированию Маги.docx](#)

Список литературы

Перечень основной и дополнительной литературы, необходимой для освоения дисциплины.

Основная

1. [Структурное программирование](#): Учебное пособие для вузов/Якимов С. П.. — Москва: Юрайт, 2022. — 342 с.
Режим доступа: <https://urait.ru/bcode/484252>
2. [Программирование](#): Учебник и практикум для вузов/Зыков С. В.. — Москва: Юрайт, 2022. — 320 с.
Режим доступа: <https://urait.ru/bcode/489754>

Дополнительная

1. [Программирование. Объектно-ориентированный подход](#): Учебник и практикум для вузов/Зыков С. В.. — Москва: Юрайт, 2021. — 155 с.
Режим доступа: <https://urait.ru/bcode/470281>

Перечень ресурсов информационно-коммуникационной сети «Интернет», необходимых для освоения дисциплины

<http://www.intuit.ru/departament/informatics/intinfo/> - Введение в информатику. Курс Интернет-университета информационных технологий

<http://www.junior.ru/wwwexam/> - Информатика и информационные технологии - web-конспект и тесты

<http://www.intuit.ru/departament/se/pinform/> - Практическая информатика. Курс Интернет-университета информационных технологий

<http://www.ict.edu.ru> - Информационно-коммуникационные технологии в образовании:

федеральный образовательный портал

<http://www.computer-museum.ru> - Виртуальный компьютерный музей

<http://www.problems.ru/inf> - Задачи по информатике

<http://iit.metodist.ru> Информатика - и информационные технологии: сайт лаборатории информатики МИОО

<http://www.intuit.ru> - Интернет-университет информационных технологий (ИНТУИТ.ру)

<http://www.edu-it.ru> - ИТ-образование в России: сайт открытого е-консорциума

<http://edu.of.ru> - Конструктор образовательных сайтов (проект Российского общеобразовательного портала)

<http://labinfo.ioso.ru> - Лаборатория обучения информатике Института содержания и методов обучения РАО

<http://test.specialist.ru> - Онлайн-тестирование и сертификация по информационным технологиям

<http://www.firststeps.ru> - Первые шаги: уроки программирования

<http://algolist.manual.ru> - Проект AlgoList: алгоритмы и, методы

<http://alglib.sources.ru> - Проект Alglib.ru: библиотека алгоритмов

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем (при необходимости)

CodeBlocks

Office Standard 2007

Windows 7 Корпоративная

Личный кабинет преподавателя или студента БГУ <http://my.bsu.ru>

Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Аудитория 4204

Корпус:4

Назначение аудитории:

учебная аудитория для проведения занятий лекционного типа, занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации

Число посадочных мест:20

Площадь (кв. м):48.5

Оборудование:

персональный компьютер - 12 шт.

доска аудиторная - 1 шт.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФГБОУ ВО «БУРЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ ДОРЖИ БАНЗАРОВА»

Фонд оценочных средств по дисциплине
Курс по программированию

Направление подготовки/ специальность
09.04.02 Информационные системы и технологии

Профиль подготовки /специализация
Проектирование, разработка и эксплуатация информационных систем

Квалификация (степень) выпускника
магистр

Форма обучения
очная

Паспорт фонда оценочных средств

ОПК-8.2 Осуществляет выбор средств разработки, оценивает сложность проектов, планирует ресурсы, контролирует сроки выполнения и оценивает качество полученного результата.

№	Контролируемые разделы, темы, модули	Формируемые компетенции	Этапы формирования	Оценочные средства	
				Вид	Количество
1.	Анализ вычислительной сложности алгоритмов. Алгоритмы обработки данных	ОПК-8.2	3 семестр	Тест	1
2.	Структуры данных	ОПК-8.2	3 семестр	Контрольная работа	1
Всего:				2	2

Описание показателей и критериев оценивания уровня приобретенных компетенций на различных этапах их формирования

Результаты обучения	Показатели оценивания	Критерии сформированности	Шкала оценивания
Знает: базовые теоретические понятия, лежащие в основе процесса разработки алгоритмов и структур данных; основные классы алгоритмов (поиска, кодирования (сжатия) данных, быстрого поиска, сортировки), используемых в них структурах данных и общих схемах решения задач на их основе;	Пороговый уровень (как обязательный для всех студентов)	Знает: разновидности структур данных, используемых для обработки сложной информации; Умеет: проводить структурирование информационного пространства заданной предметной области; Владеет: знаниями об анализе сложности алгоритмов и программ	60-69 баллов
	Базовый уровень	<i>помимо удовлетворения критериям порогового уровня:</i> Знает: основные алгоритмы обработки структур данных: пополнение, удаление, модификация; Умеет: на основе анализа разрабатываемой задачи (программы) выбирать структуры данных, обеспечивающие реализацию задачи (программы); Владеет: методологией проектирования программ	70-84 баллов
	Высокий уровень	<i>помимо удовлетворения критериям порогового и базового уровня:</i> Знает: основные алгоритмы обработки	85-100 баллов

Умеет: выбирать алгоритмы и структуры данных, обеспечивающие реализацию задачи Владеет: знаниями об анализе сложности алгоритмов и программ; методологией проектирования программ со сложной организацией данных		структур данных: поиск, сортировка (упорядочение); Умеет: на основе анализа разрабатываемой задачи (программы) выбирать наиболее рациональные и экономичные структуры данных, обеспечивающие эффективную реализацию задачи (программы); Владеет: методологией проектирования программ со сложной организацией данных, начиная с разработки модели предметной области и кончая описанием алгоритмов и структур данных средствами языка программирования	
---	--	--	--

Балльно-рейтинговая система
по дисциплине «Курс по программированию»

Общая максимальная сумма баллов, которую студент может набрать по дисциплине в течение семестра – 100 баллов: 60 баллов текущий контроль и рубежный контроль + 40 баллов зачет/экзамен (итоговый контроль);

– общая максимальная сумма баллов, которую студент может набрать в течение семестра за выполнение всех видов работ во время аудиторных и внеаудиторных занятий, активность и посещаемость, должна быть равна 60 баллам;

– минимальная сумма баллов, при которой студент допускается к зачету/экзамену (итоговому контролю), равна 36 баллам (60% от 60 баллов);

– минимальная сумма баллов, при которой студент получает положительную итоговую оценку по дисциплине равна 60 баллам (60% от 100 баллов).

Связь между четырехбалльной и столбальной системами
оценки качества обучения студентов

Оценка	Буквенный эквивалент оценки	Рейтинговые баллы
Отлично	A+	95-100
	A	90-94
	A-	85-89
Хорошо	B+	80-84
	B	75-79
	B-	70-74
Удовлетворительно	C+	67-69
	C	64-66
	C-	60-63
Неудовлетворительно	D	40-59
–	F	<40
Зачтено	S	60-100
Не зачтено	U	<60

Оценочные средства и критерии их оценки

Примерные вопросы к зачету:

1. Оценка вычислительной сложности алгоритмов.

2. Основные классы вычислительной сложности.
3. Оценка вычислительной сложности алгоритмов.
4. Скорость роста функций. Сравнение скорости роста функций.
5. Оценка вычислительной сложности рекурсивных алгоритмов. Основная теорема.
6. Общая задача сортировки. Виды алгоритмов сортировки. Сортировка «пузырьком», быстрая сортировка, поразрядная сортировка для чисел
7. Общая задача сортировки. Виды алгоритмов сортировки. Сортировка выбором, сортировка слиянием, сортировка подсчетом
8. Общая задача сортировки. Виды алгоритмов сортировки. Сортировка вставками, пирамидальная сортировка, поразрядная сортировка для строк
9. Общая задача поиска. Виды алгоритмов поиска. Линейный поиск. Бинарный поиск.
10. Линейные структуры данных. Односвязные списки
11. Линейные структуры данных. Двусвязные списки
12. Линейные структуры данных. Стек. Способы реализации стека
13. Линейные структуры данных. Стек. Построение обратной польской нотации
14. Линейные структуры данных. Очередь. Способы реализации очереди
15. Линейные структуры данных. Очередь. Обход графа в ширину
16. Линейные структуры данных. Дек. Способы реализации дека
17. Линейные структуры данных. Дек. Построение выпуклого многоугольника по множеству точек
18. Нелинейные структуры данных. Графы. Граф как модель. Способы реализации графа. Насыщенность. Обходы графов.
19. Нелинейные структуры данных. Графы. Задачи на графах (поиск кратчайших путей, построение минимального остовного дерева)
20. Нелинейные структуры данных. Деревья. Виды деревьев. Бинарные деревья. Способы реализации произвольных и бинарных деревьев. Обходы бинарных деревьев
21. Нелинейные структуры данных. Бинарные деревья поиска. Реализация бинарных деревьев поиска.
22. Нелинейные структуры данных. Множества. Операции над множествами. Реализация множества.
23. Нелинейные структуры данных. Очередь с приоритетом. Реализация очереди с приоритетом.
24. Нелинейные структуры данных. Хеш-таблицы. Понятие хеш-функции. Способы разрешения коллизий.
25. Нелинейные структуры данных. Ассоциативные массивы.
26. Библиотека STL. Линейные структуры данных
27. Библиотека STL. Нелинейные структуры данных

Критерии оценки теоретической части:

- оценка «отлично» (14-15 баллов) выставляется студенту, если он

- Четко знает принципы и базовые концепции технологий программирования, умеет применять основные алгоритмы для решения типовых учебных задач, овладел умением создавать программы на языке высокого уровня, знает основные алгоритмические конструкции, используемые при построении алгоритмов.

- Дает четкий и правильный ответ, выявляющий понимание учебного материала и характеризующий прочные знания, излагает материал в логической последовательности с использованием принятой терминологии;

- Ошибок не делает, но допускает оговорки по невнимательности при работе с программными продуктами, которые легко исправляет по требованию преподавателя;

- Ответ логичен, последователен, технически грамотен.

оценка «хорошо» (11-13 баллов) выставляется студенту, если он

- Овладел программным материалом, ориентируется в базовых концепциях технологий программирования, умеет применять основные алгоритмы с небольшим затруднением, но знает алгоритмические конструкции;
- Дает правильный ответ в определенной логической последовательности;
оценка «удовлетворительно» (9-10 баллов) выставляется студенту, если он
- Основной программный материал знает нетвердо, но большинство изученных понятий и обозначений усвоил;
- Ответ дает неполный, построенный несвязно, но выявивший общее понимание вопросов;
оценка «неудовлетворительно» (0-8 баллов) выставляется студенту, если он
- Обнаруживает незнание или непонимание большей или наиболее важной части учебного материала;
- Ответы строит несвязно, допускает существенные ошибки, которые не может исправить даже с помощью преподавателя.

Примерные задачи к экзамену:

1. Перед вами данные об изменении курса доллара за n дней. Обозначим за $r(i)$ курс доллара к рублю в i -ый день. Вы хотите понять, в какой из этих n дней стоило купить доллары, а в какой — продать, чтобы максимизировать прибыль в рублях (в предположении, что в первый день у вас были только рубли). Разработайте и реализуйте эффективный алгоритм, решающий эту задачу. Если прибыль получить невозможно, алгоритм должен сообщить об этом. Национальный исследовательский университет «Высшая школа экономики» Программа дисциплины «Алгоритмы и структуры данных» для направления 01.03.02 «Прикладная математика и информатика» подготовки бакалавра

2. Перед вами ряд из n карт, на каждой из которых написано некоторое число. Все числа различны, а карты лежат числом вниз. Требуется найти карту, на которой написано число, меньшее, чем числа на всех соседних картах. Разрешается перевернуть $O(\log n)$ карт. Как решить задачу?

3. Назовем два пути в графе непересекающимися по ребрам, если они не содержат одинаковых ребер (но могут содержать одинаковые вершины). Требуется выяснить, есть ли в заданном ориентированном графе k путей, непересекающихся по ребрам, из вершины s в вершину t . Разработайте и реализуйте эффективный алгоритм решения этой задачи.

4. Дед Мороз готовит подарки к Новому году. У Деда Мороза есть заводной автомобиль, книжка–раскраска, кукла Таня, кулек конфет и еще $n - 4$ различных подарка. Каждый из них имеется в неограниченном количестве, так что проблем вроде бы быть не должно. Однако Деду Морозу кажется, что кулек конфет и заводной автомобиль — неравноценные подарки, поэтому к кульку конфет он, скорее всего, добавит книжку–раскраску или что-то еще. Дед Мороз уже составил k подарочных наборов, каждый из которых содержит не более одного подарка каждого типа (т.е. в одном наборе — не более одной куклы Тани). Принципы Деда Мороза не позволяют ему дарить одному ребенку явно больше, чем другому, поэтому каждый из любых двух наборов содержит какой-то подарок, которого нет в другом наборе. Дед Мороз пытается составить еще один — $(k + 1)$ -ый — подарочный набор, согласующийся с его принципами, но у него не получается. Предложите алгоритм, определяющий, существует ли такой $(k + 1)$ -ый набор (для заданных k наборов и n типов подарков), приведите оценку времени работы предложенного алгоритма и реализуйте его.

5. Пусть $X[1..n]$ и $Y[1..n]$ — два отсортированных массива, каждый из которых содержит по n элементов. Разработайте и реализуйте

алгоритм, в котором поиск медианы всех $2n$ элементов, содержащихся в массивах X и Y , выполнялся бы за время $O(\log n)$.

6. Даны два массива по n элементов в каждом: i -ый элемент первого массива — ставка i -го игрока на тотализаторе, i -ый элемент второго массива — выигрыш i -го игрока. Все ставки различны, все выигрыши положительны. Обозначим за W сумму всех выигрышей. Требуется найти ставки каждого игрока x , для которого верно следующее: суммарный выигрыш игроков, поставивших меньше игрока x , не превышает $W/2$, и суммарный выигрыш игроков, поставивших больше, не превышает $W/2$. Разработайте и запрограммируйте эффективный алгоритм решения задачи и приведите оценку его времени работы.

Критерии оценки практической части:

оценка «отлично» (23-25 баллов) выставляется студенту, если

- задача решена полностью;
- в изображении графа, в теоретических выкладках решения нет пробелов и ошибок;
- в тексте программы нет синтаксических ошибок (возможны одна-две различные неточности, опiski, не являющиеся следствием незнания или непонимания учебного материала).

оценка «хорошо» (18-22 баллов) выставляется студенту, если

- задача решена полностью, но обоснования шагов решения недостаточны (если умение обосновывать рассуждения не являлось специальным объектом проверки);
- допущена одна ошибка или два-три недочета в графическом изображении графа или тексте программы.

оценка «удовлетворительно» (15-17 баллов) выставляется студенту, если

- допущены более одной ошибки или двух-трех недочетов в изображении графа или программе, но студент владеет обязательными умениями по проверяемой теме;
- студент не совсем понимает суть задачи и предлагает алгоритм, раскрывающий часть хода ее решения, но при помощи преподавателя исправляет указанные недоработки

оценка «неудовлетворительно» (0-14 баллов) выставляется студенту, если

- допущены существенные ошибки, показавшие, что студент не владеет обязательными знаниями по данной теме в полной мере.

Примерные темы контрольных работ

Вариант 1

1. Докажите или опровергните: если $f(n) = O(g(n))$ и $g(n) = O(h(n))$, то $f(n) = O(h(n))$.
2. Вершинным покрытием неориентированного графа называется подмножество его вершин, включающее хотя бы одну из двух вершин каждого ребра. Независимым множеством неориентированного графа называется подмножество его вершин, в котором никакие две вершины не связаны ребром. Допустим, у вас есть алгоритм A , вычисляющий вершинное покрытие минимального размера за время $O(f(n))$, где $f(n) = \Omega(n)$. Предложите алгоритм B , вычисляющий независимое множество максимального размера за время $O(f(n))$.
3. Решите рекуррентное соотношение: $T(n) = 8T(n/2) + n^3$.

Вариант 2

1. Докажите или опровергните: если $f_1(n) = O(g(n))$ и $f_2(n) = O(g(n))$, то $f_1(n) + f_2(n) = O(g(n))$.
2. Разработайте и реализуйте алгоритм, сортирующий массив из n чисел, каждое из которых равно -1 , 0 или 1 , за время $O(n)$ в худшем случае.

3. В некоторой программе осуществляется n последовательных вызовов операции f. Известно, что эта последовательность вызовов занимает время $\Theta(n \log n)$ в худшем случае. Каким может быть максимальное время выполнения (в терминах Θ) одной операции f из этой последовательности?

Критерии оценки:

оценка «отлично» (28-30 баллов) выставляется студенту, если

- работа выполнена полностью;
- в изображении графа, в теоретических выкладках решения нет пробелов и ошибок;
- в тексте программы нет синтаксических ошибок (возможны одна-две различные неточности, опiski, не являющиеся следствием незнания или непонимания учебного материала).

оценка «хорошо» (23-27 баллов) выставляется студенту, если

- работа выполнена полностью, но обоснования шагов решения недостаточны (если умение обосновывать рассуждения не являлось специальным объектом проверки);
- допущена одна ошибка или два-три недочета в изображении графа или тексте программы.

оценка «удовлетворительно» (15-22 балла) выставляется студенту, если

- допущены более одной ошибки или двух-трех недочетов в изображении графа или программе, но студент владеет обязательными умениями по проверяемой теме.

оценка «неудовлетворительно» (0-14 баллов) выставляется студенту, если

- допущены существенные ошибки, показавшие, что студент не владеет обязательными знаниями по данной теме в полной мере.

Примерные тестовые задания

Тест

Вопрос 1

Какие операции характерны при использовании очереди

- добавление элемента в конец очереди
- удаление элемента из начала очереди
- добавление элемента в любое место очереди
- удаление любого элемента из очереди

Вопрос 2

Что характерно для статической реализации структур данных

- использование массивов как основы реализации
- распределение памяти под элементы структуры во время компиляции программы
- использование адресных переменных (указателей) для связывания элементов структуры
- возможность выделения памяти для элементов структуры во время выполнения программы

Вопрос 3

Какие объявления необходимы для статической реализации кольцевой очереди

- массив для хранения элементов очереди
- переменная-индекс для указания ячейки массива с первым элементом очереди
- переменная-индекс для указания ячейки массива с последним элементом очереди
- переменная-указатель размещения в памяти первого элемента очереди

Вопрос 4

В чем состоит отличие элемента двунаправленного списка от элемента обычного списка

- наличие двух связующих полей для адресации левого и правого соседа
- каждый элемент двунаправленного списка имеет указатели на начало и конец списка

- каждый элемент двунаправленного списка кроме основного указателя на следующий элемент имеет указатель на отдельно хранящуюся информационную часть
- никаких отличий между элементами нет

Вопрос 5

Какие утверждения относительно динамической реализации списков являются правильными

- каждому элементу списка во время выполнения программы выделяется своя область памяти
- каждый элемент списка имеет специальное поле с адресом следующего элемента
- логический порядок следования элементов в списке может не совпадать с физическим размещением элементов в памяти
- максимальное число элементов в списке должно быть известно заранее

Вопрос 6

Какие правила обхода дерева являются основными

- обход в прямом порядке
- обход в обратном порядке
- симметричный обход
- круговой обход

Вопрос 7

Какие высказывания относятся к рекурсивному определению дерева

- дерево может быть пустым
- дерево — это вершина и связанное с ней конечное число поддеревьев
- дерево всегда содержит хотя бы одну корневую вершину
- дерево — это вершина и связанное с ней единственное поддерево

Вопрос 8

В чем суть правила симметричного обхода дерева

- сначала обрабатывается корень поддерева, потом — левое поддерево, потом — правое поддерево
- сначала обрабатывается левое поддерево, потом — корень поддерева, потом — правое поддерево
- сначала обрабатывается правое поддерево, потом — левое поддерево, потом — корень поддерева
- сначала обрабатывается корень поддерева, потом — правое поддерево, потом — левое поддерево

Вопрос 9

Как выполняется удаление вершины из дерева поиска, если эта вершина имеет двух потомков

- удаление такой вершины невозможно
- определяется вершина-заменитель, которая подставляется на место удаляемой
- соответствующий указатель в родительской вершине устанавливается в адрес левого потомка удаляемой вершины
- соответствующий указатель в родительской вершине устанавливается в адрес правого потомка удаляемой вершины

Вопрос 10

Какое дерево называется идеально сбалансированным

- если для каждой вершины число вершин в левом и правом поддеревьях отличается не более чем на 1
- если для корневой вершины число вершин в левом и правом поддеревьях отличается не более чем на 1

- если для каждой вершины число вершин в левом и правом поддеревьях совпадают
- если для каждой вершины высота левого и правого поддерева одинаковы

Вопрос 11

Что происходит при удалении вершины с терминальной страницы Б-дерева

- удаление элемента из страничного массива
- уменьшение счетчика числа вершин
- проверка допустимого числа вершин
- поиск вершины-заменителя

Вопрос 12

Какие утверждения справедливы относительно процедуры поиска заданного ключа в Б-дереве

- поиск начинается с корневой страницы
- поиск заканчивается как неудачный при попытке перехода к несуществующей дочерней странице
- поиск реализуется перебором всех страниц дерева
- поиск всегда заканчивается на терминальной странице

Вопрос 13

Какие утверждения справедливы относительно страничного массива Б-дерева порядка m

- массив имеет размерность $2m$
- элементами массива являются записи-структуры
- массив может изменять свою размерность
- в массиве всегда заняты все ячейки

Вопрос 14

Какие ситуации могут возникать при добавлении новой вершины как потомка заданной вершины в недвоичном дереве

- отсутствие в массиве указателей у заданной вершины свободного места
- наличие у заданной вершины потомков
- отсутствие у заданной вершины потомков
- отсутствие заданной вершины в дереве

Вопрос 15

Какие утверждения справедливы относительно представления графа с помощью матрицы смежности

- в этом представлении используются двумерные массивы
- данное представление рекомендуется использовать для графов с фиксированным числом вершин
- данное представление имеет простую программную реализацию
- данное представление позволяет легко изменять набор вершин в графе

Вопрос 16

В каких случаях поиск в методе цепочек дает положительный результат

- если ячейка массива, определяемая значением хеш-функции, содержит искомый ключ
- если в процессе просмотра ячеек массива по определенному правилу будет обнаружена ячейка с искомым ключом
- если один из элементов списка, связанного с исходной ячейкой, содержит искомый ключ
- если хеш-функция для искомого ключа дает положительное значение

Вопрос 17

Какие вспомогательные списки будут созданы при размещении десяти ключей (15, 19, 77, 33, 12, 35, 54, 14, 20, 10) в семиэлементной хеш-таблице

- (33, 12, 54)
- (35, 14)
- (77, 35, 14, 20)

- (19, 35)

Вопрос 18

Какие утверждения справедливы относительно правила поиска свободной ячейки с квадратичным шагом

- такой поиск обеспечивает более равномерное распределение ключей по массиву
- такой поиск не обеспечивает обход всех ячеек массива
- такой поиск требует использования датчика случайных чисел
- такой поиск требует минимальных затрат памяти по сравнению с остальными

Вопрос 19

Укажите два наилучших алгоритма по критерию трудоемкости

- алгоритм с логарифмической скоростью роста
- алгоритм с линейной скоростью роста
- алгоритм с линейно-логарифмической скоростью роста
- алгоритм с квадратичной скоростью роста

Вопрос 20

Сколько раз выполняется внешний цикл в программах простой сортировки

- $n-1$
- n
- 1
- $n+1$

Вопрос 21

Какие достоинства имеет пирамидальная сортировка по сравнению с другими методами

- пирамидальная сортировка при любых входных данных сохраняет трудоемкость порядка $O(n \cdot \log n)$
- пирамидальная сортировка эффективна при поиске нескольких первых наименьших элементов
- пирамидальная сортировка требует существенно меньше памяти
- пирамидальная сортировка имеет наиболее простую программную реализацию

Вопрос 22

Какой шаг выполняется самым первым при начале обработки очередного подмассива

- выбор опорного элемента
- поиск в подмассиве минимального элемента
- поиск в подмассиве максимального элемента
- поиск медианного элемента

Вопрос 23

Что выполняется на основном этапе сортировки файлов естественным слиянием

- распределение объединенных серий по выходным вспомогательным файлам
- сортировка отдельных серий методами сортировки массивов
- сортировка серий с помощью Б-дерева
- попарное объединение серий из входных вспомогательных файлов

Вопрос 24

Какие условия необходимы для применения поразрядной сортировки

- наличие в процессоре специальных поразрядных команд
- возможность преобразования строковых ключей в целые числа
- все ключи — целые числа
- все ключи имеют заранее известную разрядность

Критерии оценки:

- оценка «отлично»(27-30 баллов) выставляется студенту, если количество правильных ответов >90%;
- оценка «хорошо» (23-26 баллов): 76%-90%;
- оценка «удовлетворительно» (15-22 балла): -50% - 75%;
- оценка «неудовлетворительно» (0-14 баллов): <50%.