

**Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
"Национальный исследовательский университет
"Высшая школа экономики"**

Факультет компьютерных наук
Департамент программной инженерии

**Рабочая программа дисциплины
Алгоритмы и структуры данных. Часть 1**

для образовательной программы «Программная инженерия»
направления подготовки 09.03.04 «Программная инженерия»
уровень - бакалавр

Разработчики программы

Ульянов М.В., д.т.н., профессор, muljanov@mail.ru

Ахметсафина Р.З., к.т.н., доцент, rakhmetsafina@hse.ru

Одобрена на заседании департамента программной инженерии «__» _____ 2015 г.
Руководитель департамента Авдошин С.М. _____

Рекомендована Академическим советом образовательной программы
«__» _____ 2015 г., № протокола _____

Утверждена «__» _____ 2015 г.
Академический руководитель образовательной программы
Шилов В.В. _____

Москва, 2015

*Настоящая программа не может быть использована другими подразделениями университета
и другими вузами без разрешения подразделения-разработчика программы.*



1 Область применения и нормативные ссылки

Настоящая программа учебной дисциплины «Алгоритмы и структуры данных. Часть 1» устанавливает минимальные требования к знаниям и умениям студента и определяет содержание и виды учебных занятий и отчетности.

Программа предназначена для преподавателей, ведущих данную дисциплину, учебных ассистентов и студентов образовательной программы «Программная инженерия» направления подготовки 09.03.04 «Программная инженерия», изучающих дисциплину «Алгоритмы и структуры данных. Часть 1».

Программа разработана в соответствии с:

- образовательным стандартом Федерального государственного автономного образовательного учреждения высшего профессионального образования «Национальный исследовательский университет «Высшая школа экономики» по направлению 09.03.04 «Программная инженерия»;
- образовательной программой «Программная инженерия» направления подготовки 09.03.04 «Программная инженерия»;
- рабочим учебным планом университета по направлению 09.03.04 «Программная инженерия», утвержденным в 2015 г.

2 Цели освоения дисциплины

Цели освоения дисциплины «Алгоритмы и структуры данных. Часть 1»:

- формирование у студентов профессиональных компетенций, связанных с использованием теоретических знаний в области теории алгоритмов и теории сложности вычислений;
- получение практических навыков в области разработки ресурсно-эффективных алгоритмов на основе теоретического анализа;
- развитие умений, основанных на полученных теоретических знаниях, позволяющих на творческом и репродуктивном уровне применять и создавать эффективные алгоритмы для решения задач обработки информации;
- получение студентам навыков самостоятельной исследовательской работы, предполагающей изучение специфических методов анализа алгоритмов, инструментов и средств, необходимых для решения актуальной, в аспекте программной инженерии, задачи выбора рациональных алгоритмов, в зависимости от особенностей применения разрабатываемых программ.

3 Компетенции обучающегося, формируемые в результате освоения дисциплины

В результате освоения дисциплины студент должен:

- Знать:
 - методы анализа алгоритмов в итерационной реализации;
 - методы анализа алгоритмов в рекурсивной реализации;
 - метод декомпозиции и метод динамического программирования как методы разработки эффективных алгоритмов.
- Уметь:



- оценивать компьютерные алгоритмы с использованием комплексных критериев качества, в том числе оценивать ресурсную эффективность алгоритмов;
 - планировать эксперимент, проводить экспериментальное исследование алгоритмов.
- Иметь навыки (приобрести опыт) и владеть:
 - инструментами измерения времени в программных реализациях алгоритмов;
 - методами и средствами оценки трудоемкости алгоритмов в их итерационной и рекурсивной реализации;
 - методами разработки эффективных алгоритмов на основе их сравнительного анализа.

В результате изучения дисциплины студент осваивает следующие компетенции:

1. Научно-исследовательская деятельность:

- способность к формализации в своей предметной области с учетом ограничений используемых методов исследования (ПК-2);
- готовность к использованию методов и инструментальных средств исследования объектов профессиональной деятельности (ПК-3);
- готовность обосновать принимаемые проектные решения, осуществлять постановку и выполнение экспериментов по проверке их корректности и эффективности (ПК-4);

2. Проектная деятельность:

- способность оценивать временную и емкостную сложность программного обеспечения (ПК-13);

3. Технологическая деятельность:

- навыки использования различных технологий разработки программного обеспечения (ПК-16);

4. Производственная деятельность:

- умение применять основные методы и инструменты разработки программного обеспечения (ПК-17).

4 Место дисциплины в структуре образовательной программы

Настоящая дисциплина относится к базовой части дисциплин профессионального цикла.

В соответствии с рабочим учебным планом по направлению «Программная инженерия» дисциплина «Алгоритмы и структуры данных. Часть 1» читается студентам второго курса бакалавриата в 1-ом и 2-ом модулях.

Изучение данной дисциплины базируется на знаниях, полученных студентами при освоении учебных дисциплин «Дискретная математика», «Программирование», общих знаниях математики и основ программирования в части базовых алгоритмических конструкций.

Дисциплина является основой для последующего изучения дисциплин: «Проектирование и архитектура программных систем», «Обеспечение качества и тестирование», «Компьютерная графика» и других.



5 Тематический план учебной дисциплины

№	Название темы	Всего часов по дисциплине	Аудиторные часы		Самостоятельная работа
			Лекции	Практические занятия	
1	Понятие алгоритма. Формализм Э.Л. Поста. Алгоритм как финитный 1-процесс.	6	2		4
2	Основная терминология и обозначения в анализе ресурсной эффективности алгоритмов.	5	1		4
3	Классификация алгоритмов по трудоёмкости.	5	1		4
4	Анализ NPR алгоритмов методом классов входных данных.	6	2		4
5	Метод вероятностного анализа для получения трудоёмкости в среднем.	6	2		4
6	Сравнительный анализ алгоритмов по трудоёмкости и решение задачи рационального выбора.	32	2	10	20
7	Сложность алгоритмов. Теоретическая нижняя граница сложности задачи.	4	2		2
8	Введение в теорию сложности вычислений. Основные сложностные классы задач.	4	2		2
9	Временная эффективность и особенности перехода к временным оценкам.	12	2	4	6
10	Рекурсивные алгоритмы. Особенности реализации и анализа.	8	2	2	4
11	Разработка алгоритмов методом декомпозиции и особенности его применения	8	2	2	4
12	Анализ рекурсивных алгоритмов методом подсчёта вершин дерева рекурсии.	8	2	2	4
13	Рекурсивный алгоритм возведения в степень и его анализ.	6	2		4



14	Задачи умножения длинных целых чисел и умножения матриц.	18	2	6	10
15	Динамическое программирование. Задача оптимальной упаковки (задача о рюкзаке).	8	2	2	4
16	Рекурсивный алгоритм метода динамического программирования для задачи упаковки.	8	2	2	4
Итого:		144	30	30	84

6 Формы контроля знаний студентов

Тип контроля	Форма контроля	модули		Параметры
		1	2	
Текущий (неделя)	Контрольная работа		7-я неделя	Решение задания на компьютере
	Домашнее задание		2-я – 7-я недели	Реализация и исследование алгоритмов решения задачи
Промежуточный	Экзамен		*	Устный экзамен 60 мин.

6.1 Критерии оценки знаний, навыков

Оценки по всем формам контроля выставляются по 10-ти балльной шкале.

Текущий контроль предусматривает контрольную работу и домашнее задание, выполняемые во втором модуле.

Контрольная работа предусматривает программную реализацию на компьютере предложенного алгоритма. Выполняется на практическом занятии. Оценка за контрольную работу выставляется по критериям, описанным в примерном задании на контрольную работу (см. п. 9.1.2). Передача контрольной работы с целью повышения оценки не допускается. Допускается выполнение контрольной работы в более поздний срок при пропуске по уважительной причине.

Домашнее задание включает разработку, кодирование, тестирование и отладку программ реализации одной задачи (по выбору), исследование и сравнительный анализ алгоритмов ее решения. По домашнему заданию оформляется отчет в электронном виде.

Домашнее задание размещается в LMS в разделе «Проекты». В установленный срок студент загружает в LMS архив, содержащий полностью оформленный отчет и программу решения контрольного домашнего задания. Оценка за домашнее задание выставляется с учетом полноты выполнения задания и оформления результатов.

Кроме текущего контроля учитывается

- работа студентов на практических занятиях, а именно: реализация алгоритмов, решение тестов по некоторым темам,
- выполнение домашней работы (заданий к семинарам). Задания к семинарам размещаются в LMS, сдаются студентами в указанный срок в виде проектов.



В случае несвоевременной сдачи домашних работ и домашнего задания оценка снижается на один балл за каждый день задержки. При задержке по уважительной причине баллы не снимаются.

Промежуточный контроль: экзамен в конце 2-го модуля. Проводится в устной форме. Экзамен состоит из двух частей:

- теоретической, проводится в форме устной беседы по тематике дисциплины (30 мин.);
- практической, связанной с обсуждением результатов домашнего задания (30 мин.).

Итоговый контроль: экзамен в конце 4-го модуля (Часть 2 программы).

6.2 Порядок формирования оценок по дисциплине

По всем видам работ выставляется 10-балльная оценка.

Оценивается работа студентов на практических занятиях $O_{\text{аудиторная}}$, в которую входят:

- результаты тестирования по текущей теме,
- разработка программ, реализующих алгоритмы.

Оценки за работу на практических занятиях выставляются в рабочую ведомость.

Оценивается самостоятельная работа студентов $O_{\text{сам. работа}}$: правильность и полнота выполнения домашних работ по темам практических занятий. Оценки за самостоятельную работу студента преподаватель выставляет в рабочую ведомость.

Накопленная оценка за текущий контроль учитывает результаты студента по текущему контролю следующим образом:

$$O_{\text{накопленная}} = 0,5 * O_{\text{текущий}} + 0,2 * O_{\text{ауд}} + 0,3 * O_{\text{сам. работа}},$$

где оценка $O_{\text{текущий}}$ рассчитывается как взвешенная сумма оценок за домашнее задание и контрольную работу:

$$O_{\text{текущий}} = 0,5 * O_{\text{дз}} + 0,5 * O_{\text{кр}}.$$

Способ округления — арифметический.

Оценка промежуточного контроля во втором модуле в форме экзамена определяется соотношением:

$$O_{\text{промежуточный}} = 0,5 * O_{\text{накопленная}} + 0,5 * O_{\text{экс}}.$$

В случае получения неудовлетворительной оценки и повторной сдачи экзамена формула оценивания не меняется.



Перевод в пятибалльную оценку осуществляется в соответствии со следующей таблицей.
Таблица соответствия оценок по десятибалльной и пятибалльной системам

По десятибалльной шкале	По пятибалльной шкале
1 – неудовлетворительно 2 – очень плохо 3 – плохо	неудовлетворительно – 2
4 – удовлетворительно 5 – весьма удовлетворительно	удовлетворительно – 3
6 – хорошо 7 – очень хорошо	хорошо – 4
8 – почти отлично 9 – отлично 10 – блестяще	отлично – 5

7 Содержание дисциплины

7.1 Содержание лекций

Тема 1. Понятие алгоритма. Формализм Э.Л. Поста. Алгоритм как финитный 1-процесс

История теории алгоритмов. Понятие алгоритма. Формализация понятия алгоритма. Формализм Э.Л. Поста. Пространство символов и примитивные операции в машине Поста, алгоритм как финитный 1-процесс, доставляющий решение общей проблемы, гипотеза Поста.

Литература по теме 1:

1. Post E.L. Finite combinatory process — formulation 1 // J. Symbolic Logic (1936) 1, pp. 103–105.
2. Успенский В.А. Машина Поста. — М.: Наука, 1979. — 96 с. [стр. 1–96].
3. Ульянов М.В. Ресурсно-эффективные компьютерные алгоритмы: разработка и анализ. — М.: ФИЗМАТЛИТ, 2008. — 304 с. [стр. 18–23].
4. Фалевич Б.Я. Теория алгоритмов: Учебное пособие. — М.: Машиностроение, 2004. — 160 с. [стр. 15–22].

Тема 2. Основная терминология и обозначения в анализе ресурсной эффективности алгоритмов

Вход как эквивалент конкретной проблемы по Посту. Длина входа как функция от мощности множества входных данных. Оценки качества алгоритма. Понятие трудоёмкости. Трудоёмкость в лучшем, худшем и среднем случаях. Ёмкостная эффективность.

Литература по теме 2:

1. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ, 2-е издание. — Пер. с англ. — М.; СПб.; Киев: Издательский дом «Вильямс», 2011. — 1296 с. [стр. 22–25].
2. Ульянов М.В. Классификация и методы сравнительного анализа вычислительных алгоритмов. Научное издание. — М.: Издательство физико-математической литературы, 2004. — 212 с. [стр. 49–52].



3. Ульянов М.В. Теоретико-множественный подход к определению функций трудоемкости алгоритма // Информация и безопасность. 2004. № 2. С. 53–58.
4. Ульянов М.В. Ресурсно-эффективные компьютерные алгоритмы: разработка и анализ. — М.: ФИЗМАТЛИТ, 2008. — 304 с. [стр. 60–64].

Тема 3. Классификация алгоритмов по трудоёмкости

Влияние длины входа и значений на трудоёмкость. Классификация алгоритмов по характеристическим особенностям множества исходных данных, определяющим вид функции трудоемкости. Классы N, PR, NPR. Примеры алгоритмов в рамках классификации. Дополнительная детализация классов. Особенности анализа алгоритмов по классам.

Литература по теме 3:

1. Ульянов М.В. Классификация алгоритмов в целях практического анализа // Информационные технологии. 2003. № 11. С. 29–36.
2. Ульянов М.В. Ресурсно-эффективные компьютерные алгоритмы: разработка и анализ. — М.: ФИЗМАТЛИТ, 2008. — 304 с. [стр. 95–102].

Тема 4. Анализ NPR алгоритмов методом классов входных данных

Метод классов входных данных. Особенности его применения. Задача сортировки трёх чисел по месту. Классы входных данных, порождаемые этой задачей. Алгоритмы и их анализ методом классов входных данных.

Литература по теме 4:

1. Макконелл Дж. Основы современных алгоритмов. — 2-е дополненное издание. — М.: Техносфера, 2006. — 368 с. [стр. 19–26].
2. Кнут Д. Искусство программирования. Том 1. 3-е изд. Пер. с англ. : Уч. пос. — М.: Изд. дом «Вильямс», 2011. [стр. 198–217].
3. Ульянов М.В. Ресурсно-эффективные компьютерные алгоритмы: разработка и анализ. — М.: ФИЗМАТЛИТ, 2008. — 304 с. [стр. 199–204].

Тема 5. Метод вероятностного анализа для получения трудоёмкости в среднем

Собственно метод вероятностного анализа. Алгоритм сортировки вставками и анализ его трудоёмкости в среднем. Замечания об экспериментальном анализе алгоритмов по трудоёмкости. Выборочное среднее и математическое ожидание. Определение необходимого числа экспериментов.

Литература по теме 5:

1. Петрушин В.Н, Ульянов М.В. Информационная чувствительность компьютерных алгоритмов. — М.: ФИЗМАТЛИТ, 2010. — 224 с. ISBN: 978-5-9221-1264-2. [стр. 76–85].
2. Кнут Д. Искусство программирования. Тома 1, 2, 3. 3-е изд. Пер. с англ. : Уч. пос. — М.: Изд. дом «Вильямс», 2011. [стр. 127–131].
3. Ульянов М.В. Ресурсно-эффективные компьютерные алгоритмы: разработка и анализ. — М.: ФИЗМАТЛИТ, 2008. — 304 с. [стр. 171–1743].



Тема 6. Сравнительный анализ алгоритмов по трудоёмкости и решение задачи рационального выбора

Методика сравнительного анализа алгоритмов и определение границ применимости по характеристикам исходных данных. Простейшие примеры сравнительного анализа для задачи сортировки — алгоритм сортировки вставками и сортировки методом индексов. Рациональный выбор в параметрическом пространстве.

Литература по теме 6:

1. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ, 2-е издание. — Пер. с англ. — М.; СПб.; Киев: Издательский дом «Вильямс», 2011. — 1296 с. [стр. 18–32].
2. Скиена С. Алгоритмы. Руководство по разработке. — 2-е изд.: Пер. с англ. — СПб.: БХВ-Петербург, 2011. — 720 с.: ил. [стр. 59–64].
3. Ульянов М.В. Ресурсно-эффективные компьютерные алгоритмы: разработка и анализ. — М.: ФИЗМАТЛИТ, 2008. — 304 с. [стр. 119–123, 225–230].

Тема 7. Сложность алгоритмов. Теоретическая нижняя граница сложности задачи

Асимптотический анализ функций и основные обозначения. Сложность как асимптотическая оценка трудоёмкости в худшем случае. Теоретическая нижняя граница сложности задачи. Доказательство теоретической нижней границы для задачи сортировки сравнениями.

Литература по теме 7:

1. Макконелл Дж. Основы современных алгоритмов. — 2-е дополненное издание. — М.: Техносфера, 2006. — 368 с. [стр. 32–36].
2. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ, 2-е издание. — Пер. с англ. — М.; СПб.; Киев: Издательский дом «Вильямс», 2011. — 1296 с. [стр. 36–49].
3. Ульянов М.В. Ресурсно-эффективные компьютерные алгоритмы: разработка и анализ. — М.: ФИЗМАТЛИТ, 2008. — 304 с. [стр. 84–89].

Тема 8. Введение в теорию сложности вычислений. Основные сложностные классы задач

Исторический очерк теории сложности вычислений. Сложностные классы задач. Определение и взаимосвязь классов P и NP. Определение и примеры NP-полных задач.

Литература по теме 8:

1. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. — М.: Мир, 1982. — 416 с. [Введение].
2. Хопкрофт Дж., Мотовани Р., Ульман Дж. Введение в теорию автоматов, языков и вычислений, 2-е изд.: Пер. с англ. — М.: Издательский дом «Вильямс», 2002. — 528 с. [стр. 19–26]. [стр. 457–472].
3. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ, 2-е издание. — Пер. с англ. — М.; СПб.; Киев: Издательский дом «Вильямс», 2011. — 1296 с. [стр. 837–880].



Тема 9. Временная эффективность и особенности перехода к временным оценкам

Переход к пооперационному анализу трудоёмкости алгоритмов. Построение временных оценок. Методы перехода к временным оценкам на основе функции трудоёмкости. Особенности измерения времени выполнения программной реализации алгоритма.

Литература по теме 9:

1. Ульянов М.В. Метод прогнозирования временных оценок программных реализаций алгоритмов на основе функции трудоёмкости // Информационные технологии. 2004. № 5. С. 54–62.
2. Ульянов М.В. Ресурсно-эффективные компьютерные алгоритмы: разработка и анализ. — М.: ФИЗМАТЛИТ, 2008. — 304 с. [стр. 72–82].
3. Макконелл Дж. Основы современных алгоритмов. — 2-е дополненное издание. — М.: Техносфера, 2006. — 368 с. [стр. 346–349].

Тема 10. Рекурсивные алгоритмы — Особенности реализации и анализа

Рекурсивные функции и рекурсивные алгоритмы. Рекурсивная реализация алгоритмов. Анализ механизма рекурсивного вызова. Дерево рекурсивных вызовов. Простейшие примеры рекурсивных алгоритмов.

Литература по теме 10:

1. Головешкин В.А., Ульянов М.В. Теория рекурсии для программистов. — М.: ФИЗМАТЛИТ, 2006. — 296 с. [стр. 19–26]. [стр. 51–67].
2. Ульянов М.В. Ресурсно-эффективные компьютерные алгоритмы: разработка и анализ. — М.: ФИЗМАТЛИТ, 2008. — 304 с. [стр. 178–191].

Тема 11. Метод декомпозиции и особенности его применения

Собственно метод декомпозиции. Рекурсивные алгоритмы как реализация метода декомпозиции. Методы анализа рекурсивных алгоритмов. Теорема Бенгли-Хакен-Сакса и её применение для получения сложностных оценок алгоритмов, разработанных методом декомпозиции. Анализ рекурсивных алгоритмов методом прямого подсчета вершин рекурсивного дерева. Пример анализа — алгоритм сортировки слиянием.

Литература по теме 11:

1. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ, 2-е издание. — Пер. с англ. — М.; СПб.; Киев: Издательский дом «Вильямс», 2011. — 1296 с. [стр. 18–29].
2. Головешкин В.А., Ульянов М.В. Теория рекурсии для программистов. — М.: ФИЗМАТЛИТ, 2006. — 296 с. [стр. 71–74].
3. Ульянов М.В. Ресурсно-эффективные компьютерные алгоритмы: разработка и анализ. — М.: ФИЗМАТЛИТ, 2008. — 304 с. [стр. 131–134].
4. Макконелл Дж. Основы современных алгоритмов. — 2-е дополненное издание. — М.: Техносфера, 2006. — 368 с. [стр. 133–135].



Тема 12. Анализ рекурсивных алгоритмов методом подсчёта вершин дерева рекурсии

Анализ рекурсивных алгоритмов методом прямого подсчета вершин рекурсивного дерева. Пример анализа — алгоритм сортировки слиянием. Отличие детального анализа от асимптотического — граница применимости алгоритма сортировки вставками.

Литература по теме 12:

1. Головешкин В.А., Пономарёв А.В., Ульянов М.В., Аналитическое решение класса рекуррентных соотношений с аддитивной функцией степенного вида в целях анализа трудоёмкости рекурсивных алгоритмов // Автоматизация и современные технологии. 2011. №3. С. 25–29.
2. Головешкин В.А., Ульянов М.В. Теория рекурсии для программистов. — М.: ФИЗМАТЛИТ, 2006. — 296 с. [стр. 181–185].
3. Ульянов М.В. Ресурсно-эффективные компьютерные алгоритмы: разработка и анализ. — М.: ФИЗМАТЛИТ, 2008. — 304 с. [стр. 178–181].

Тема 13. Рекурсивный алгоритм возведения в степень

Метод повторного возведения в квадрат. Функция $\beta_1(n)$. Рекурсивный и итерационный алгоритм быстрого возведения в степень и их анализ с использованием функции $\beta_1(n)$.

Литература по теме 13:

1. Головешкин В.А., Ульянов М.В. Теория рекурсии для программистов. — М.: ФИЗМАТЛИТ, 2006. — 296 с. [стр. 204–209].
2. Ульянов М.В. Ресурсно-эффективные компьютерные алгоритмы: разработка и анализ. — М.: ФИЗМАТЛИТ, 2008. — 304 с. [стр. 204–209].

Тема 14. Задачи умножения длинных целых чисел и умножения матриц

Задача умножения длинных целых чисел. Гипотеза Колмогорова. Алгоритм умножения в столбик. Алгоритм Карацубы. Классический алгоритм умножения матриц. Алгоритм Штрассена.

Литература по теме 14:

1. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ, 2-е издание. — Пер. с англ. — М.; СПб.; Киев: Издательский дом «Вильямс», 2011. — 1296 с. [стр. 679–684, 721–728].
2. Скиена С. Алгоритмы. Руководство по разработке. — 2-е изд.: Пер. с англ. — СПб.: БХВ-Петербург, 2011. — 720 с.: ил. [стр. 422-425, 443-448].
3. Головешкин В.А., Ульянов М.В. Теория рекурсии для программистов. — М.: ФИЗМАТЛИТ, 2006. — 296 с. [стр. 209–219].

Тема 15. Динамическое программирование. Задача оптимальной упаковки (задача о рюкзаке)

Основы метода динамического программирования. Пошаговая оптимизация. Классическая задача упаковки и основное уравнение Беллмана для точного решения задачи упаковки — табличная реализация функционального уравнения.



Литература по теме 15:

1. Беллман Р., Дрейфус Р. Прикладные задачи динамического программирования: Пер. с англ. — М.: Наука, 1965, — 457 с. [стр. 24–57].
2. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ, 2-е издание. — Пер. с англ. — М.; СПб.; Киев: Издательский дом «Вильямс», 2011. — 1296 с. [стр. 288–313].
3. Головешкин В.А., Ульянов М.В. Дополнение к книге Р. Хаггарти «Дискретная математика для программистов». — М.: Техносфера, 2005. С. 305–394. [стр. 372–381].
4. Скиена С. Алгоритмы. Руководство по разработке. — 2-е изд.: Пер. с англ. — СПб.: БХВ-Петербург, 2011. — 720 с.: ил. [стр. 448–451].

Тема 16. Рекурсивный алгоритм метода динамического программирования для задачи упаковки

Рекурсивная реализация алгоритма Беллмана для задачи упаковки. Параметризация задачи упаковки по среднему объёму грузов. Структура рекурсивного дерева и подсчет количества вершин порождённого дерева рекурсии. Оценка сложности алгоритма. Сравнительный анализ табличного и рекурсивного алгоритмов. Варианты построения комбинированного алгоритма.

Литература по теме 16:

1. Наумова О.А., Ульянов М.В. Комбинированный и волновой алгоритмы решения задачи упаковки: принципы построения и особенности // Бизнес-Информатика. 2009. № 2(8) .С. 27–33.
2. Головешкин В.А., Ульянов М.В. Теория рекурсии для программистов. — М.: ФИЗМАТЛИТ, 2006. — 296 с. [стр. 249–262].
3. Скиена С. Алгоритмы. Руководство по разработке. — 2-е изд.: Пер. с англ. — СПб.: БХВ-Петербург, 2011. — 720 с.: ил. [стр. 293–332].

7.2 Содержание практических занятий

Примерный перечень тем практических занятий:

1. Итерационные алгоритмы сортировки — пузырька, пузырька с условием Айверсона, простой и бинарной вставки.
2. Рекурсивные алгоритмы сортировки – пирамидальная, Хоара, слиянием.
3. Псевдолинейные алгоритмы сортировки – подсчетом, цифровая
4. Экспериментальное исследование алгоритмов сортировки для разных входов путем измерения времени, анализ результатов экспериментов.
5. Указатели и списки.
6. Алгоритмы порождения комбинаторных объектов.
7. Динамическое программирование. Постановка задачи одномерной упаковки. Сравнение алгоритмов решения задачи одномерной оптимальной по стоимости упаковки – рекурсивного, табличного, итерационного.
8. Алгоритмы обработки строк.



На практических занятиях языком программной реализации алгоритмов является язык программирования C++, в связи с этим на занятиях изучаются основы языка C++ в объеме, необходимом для выполнения практических заданий.

8 Образовательные технологии

Работа на практических занятиях предполагает разработку программ, реализующих алгоритмы, их экспериментальное исследование и сравнение результатов с теоретическими оценками алгоритмов. Домашнее задание предполагает самостоятельный анализ ресурсной эффективности нескольких выбранных алгоритмов решения определённой задачи и сравнительный анализ этих алгоритмов с целью формулировки рекомендаций по их применению в зависимости от особенностей проблемной области применения.

9 Оценочные средства для текущего контроля и аттестации студента

9.1 Тематика заданий текущего контроля

9.1.1 Домашнее задание

Предполагается выбор студентами для последующей программной реализации и сравнительного анализа двух-трех алгоритмов из перечисленных в темах 7–12 практических занятий (раздел 7.2).

9.1.2. Контрольная работа

Примерное задание:

По заданному алгоритму требуется

- 1) Написать программу на языке C++, реализующую заданный алгоритм
- 2) Определить максимальную глубину рекурсии

Формат входных данных:

- данные читаются из текстового файла test.txt;
- В первой строке целое число — количество элементов массива (≤ 2000);
- Во второй строке — разделенные пробелами элементы одномерного целочисленного массива.

Формат выходных данных

На консоль должно выводиться одно целое число — максимальная глубина рекурсии.

Пример входных и выходных данных

Входные данные – файл test.txt	Выходные данные - вывод на консоль
8 5 3 2 6 4 1 3 7	5

Оценивание контрольной работы

- 1) Алгоритм реализован верно — 4 балла.
- 2) Верно выполняется тестовый пример, приведенный выше — 5 баллов.
- 3) Тестовый пример выполняется не на всех входных данных — (+ до 2 баллов) (6-7 баллов).
- 4) Тестовый пример выполняется на всех входных данных (+ до 2 баллов) (8-9 баллов).

Работа выполнена самостоятельно менее чем за 60 минут — +1 балл.

9.2 Вопросы для оценки качества освоения дисциплины

1. Понятие модели вычислений.
2. Машина Поста, алгоритм как финитный 1-процесс.
3. Требования к алгоритмам.



4. Основные свойства алгоритмов.
5. Понятие функции трудоёмкости в модели вычислений.
6. Понятие вычислительной сложности алгоритма.
7. Комплексные оценки качества алгоритмов и их компоненты.
8. Классификации алгоритмов.
9. Теоретический нижний предел сложности задачи.
10. Трудоёмкость в худшем, среднем и лучшем случае как функция длины входа.
11. Сравнительный анализ алгоритмов по трудоёмкости.
12. Методика прогнозирования временных оценок по функции трудоёмкости.
13. Особенности анализа рекурсивных алгоритмов.
14. Метод подсчета вершин порожденного дерева рекурсии.
15. Способы повышения временной эффективности рекурсивных алгоритмов.
16. Этапы разработки алгоритмов методом декомпозиции.
17. Основная теорема о рекуррентных соотношениях.
18. Оценка вычислительной сложности рекурсивных алгоритмов.
19. Метод динамического программирования — основная идея.
20. Уравнение Беллмана для задачи одномерной упаковки.
21. Основные этапы табличного алгоритма решения задачи упаковки.
22. Структура дерева рекурсии, порожденного алгоритмом решения задачи упаковки.
23. Оценка сложности рекурсивного алгоритма решения задачи упаковки.
24. Понятие о комбинированных алгоритмах.
25. Принцип построения комбинированного алгоритма сортировки.
26. Идея волнового комбинированного алгоритма упаковки.
27. Комплексное решение задачи рационального выбора алгоритма.
28. Основные сложностные классы задач.
29. Теорема Кука и полиномиальная сводимость задач.
30. NP-полные задачи и особенности решающих их алгоритмов.

10 Учебно-методическое и информационное обеспечение дисциплины

10.1 Базовый учебник — отсутствует

10.2 Основная литература:

1. Ульянов М.В. Ресурсно-эффективные компьютерные алгоритмы: разработка и анализ. — М.: ФИЗМАТЛИТ, 2008. — 304 с.



2. Головешкин В.А., Ульянов М.В. Теория рекурсии для программистов. — М.: ФИЗМАТЛИТ, 2006. — 296 с.
3. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ, 2-е издание. — Пер. с англ. — М.; СПб.; Киев: Издательский дом «Вильямс», 2011. — 1296 с.
4. Петрушин В.Н, Ульянов М.В. Информационная чувствительность компьютерных алгоритмов. — М.: ФИЗМАТЛИТ, 2010. — 224 с. ISBN: 978-5-9221-1264-2.
5. Джосаттис Николаи М. Стандартная библиотека C++: Справочное руководство, 2-е изд. : Пер. с англ. – М.: ООО «И.Д. Вильямс», 2014 – 1136 с.
6. Кнут Д. Искусство программирования. Тома 1, 2, 3. 3-е изд. Пер. с англ. : Уч. пос. — М.: Изд. дом «Вильямс», 2011.
7. Ахо, А. В. Хопкрофт Дж, Ульман Дж. Структуры данных и алгоритмы. Вильямс, 2010. - 391 с.

10.3 Дополнительная литература и источники

1. Скиена С. Алгоритмы. Руководство по разработке. — 2-е изд.: Пер. с англ. — СПб.: БХВ-Петербург, 2011. — 720 с.: ил.
2. Макконелл Дж. Основы современных алгоритмов. — 2-е дополненное издание. — М.: Техносфера, 2006. — 368 с.

10.4 Справочники, словари, энциклопедии

1. <http://www.acm.org/calgo>
2. <http://www-cs-faculty.stanford.edu>
3. <http://combinatorica.com>

10.5 Программные средства

Практические занятия проводятся в компьютерном классе с выходом в Интернет и доступом к ресурсам электронной библиотеки НИУ ВШЭ. Каждый студент должен иметь рабочее место.

Необходимое программное обеспечение:

1. Microsoft Office Professional 2010;
2. Visual Studio 2013.
3. CLion 1.0

10.6 Дистанционная поддержка дисциплины

Дистанционная поддержка дисциплины обеспечивается использованием LMS. В разделе дисциплины «Алгоритмы и структуры данных. Часть 1» размещаются материалы лекций и практических занятий, тесты для самоподготовки, проекты, оценки текущего и итогового контроля.

11 Материально-техническое обеспечение дисциплины

Проектор для лекций и семинаров, классы для семинаров с компьютерами, на которых установлена инструментальная среда Visual Studio 2013 и/или CLion.