**Программа учебной дисциплины «Алгоритмы и структуры данных. Часть 1»**

Утверждена

Академическим советом ООП

Протокол № от «\_\_»\_\_\_\_\_20\_\_ г.

|  |  |
| --- | --- |
| Автор | Ульянов М. В., д.т.н., профессор, Ахметсафина Р. З., к.т.н., доцент,  Мицюк А. А., преподаватель |
| Число кредитов | 8 |
| Контактная работа (час.) | 60 |
| Самостоятельная работа (час.) | 84 |
| Курс | 2 |
| Формат изучения дисциплины | Без использования он-лайн курса |

1. **ЦЕЛЬ, РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ И ПРЕРЕКВИЗИТЫ**

Целями освоения дисциплины «Алгоритмы и структуры данных. Часть 1» являются

* 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.

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.

**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.

Изучение дисциплины «Алгоритмы и структуры данных. Часть 1» базируется на знаниях, полученных студентами при освоении учебных дисциплин «Дискретная математика», «Программирование», общих знаниях математики и основ программирования в части базовых алгоритмических конструкций

Основные положения дисциплины могут быть использованы в дальнейшем при изучении следующих дисциплин:

* Проектирование архитектуры программных систем,
* Обеспечение качества и тестирование,
* Компьютерная графика

и других.

1. **СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ**

**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.

**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.

**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.

**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.

**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.

**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.

**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.

**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.

**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 трудоемкости. Особенности измерения времени выполнения программной реализации алгоритма.

**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.

**Тема 11. Метод декомпозиции и особенности его применения**

Собственно метод декомпозиции. Рекурсивные алгоритмы как реализация метода декомпозиции. Методы анализа рекурсивных алгоритмов. Теорема Бентли-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.

**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.

**Тема 13. Рекурсивный алгоритм возведения в степень**

Метод повторного возведения в квадрат. Функция 1(n). Рекурсивный и итерационный алгоритм быстрого возведения в степень и их анализ с использованием функции 1(n).

**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.

**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.

**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.

**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. Динамическое программирование. Постановка задачи одномерной упаковки. Сравнение алгоритмов решения задачи одномерной оптимальной по стоимости упаковки – рекурсивного, табличного, итерационного;
5. Указатели и списки;
6. Реализация некоторых структур данных и задач, использующих структуры;.
7. Алгоритмы порождения комбинаторных объектов;
8. Алгоритмы обработки строк;
9. Экспериментальное исследование алгоритмов решения задачи путем измерения времени или подсчета количества операций, анализ результатов экспериментов.

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

1. **ОЦЕНИВАНИЕ**

**Формы контроля знаний студентов**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Тип контроля | Форма контроля | модули | | Параметры |
| 1 | 2 |
| Текущий  (неделя) | Домашнее задание |  | 2-я – 5-я недели | Реализация и исследование алгоритмов решения задачи |
| Промежуточный | Экзамен |  | \* | Устный экзамен 60 мин. |

**Критерии оценки знаний, навыков**

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

**Текущий контроль** предусматривает домашнее задание, выполняемое во втором модуле.

***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 Oдз выставляется с учетом полноты выполнения задания и оформления результатов.

Кроме текущего контроля оценивается

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

Для студентов, не сдавших своевременно домашнюю работу или домашнее задание, могут создаваться в ЛМС проекты «для опоздавших». При этом оценка за работу делится пополам. При задержке по уважительной причине оценка не снижается.

**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).

**Порядок формирования оценок по дисциплине**

По всем видам работ выставляется 10-балльная оценка.

Оценивается работа студентов на практических занятиях Оаудиторная, в которую входят:

* результаты тестирования по текущей теме,
* разработка программ, реализующих алгоритмы,
* участие в разборе задач во время практических занятий.

Оценки за работу на практических занятиях выставляются в рабочую ведомость.

Оценивается самостоятельная работа студентов Осам.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:

Онакопленная= 0,5 \* Отекущий + 0,2 \* Оауд + 0,3 \* Осам.работа,

где оценка Отекущий - это оценка за домашнее задание Отекущий = Одз.

Способ округления — арифметический.

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

Опромежуточный = 0,5 \* Онакопленная + 0,5 \*·Оэкз.

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 |

1. **ПРИМЕРЫ ОЦЕНОЧНЫХ СРЕДСТВ**

# **Оценочные средства для текущего контроля студента**

Домашнее задание

Предполагается выбор студентами для последующей программной реализации и сравнительного анализа нескольких алгоритмов из перечисленных в темах 2–9 практических занятий. Должно быть выполнено планирование эксперимента, в т.ч. определение входных и выходных данных. Результаты сравнительного анализа должны быть оформлены в виде таблиц, графиков, текста.

**Оценочные средства для промежуточной аттестации**

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.

1. **РЕСУРСЫ**
   1. **Основная литература**
2. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ, 3-е издание. — Пер. с англ. — М.; СПб.; Киев: ООО «И.Д. Вильямс», 2005. — 1328 с. ) (или более поздние издания)
3. Дональд Э. Кнут. «Искусство программирования, том 3. Сортировка и поиск», 2-е издание, — Пер. с англ. — М.; СПб.; Киев: ООО «И.Д. Вильямс», 2012. - 832 стр. (или более поздние издания)
4. Дональд Э. Кнут. «Искусство программирования, том 4,А. Комбинаторные алгоритмы. Часть 1» — Пер. с англ. — М.; СПб.; Киев: ООО «И.Д. Вильямс», 2013. - 960 стр. (или более поздние издания)
5. Дональд Э. Кнут. «Искусство программирования, том 2. Получисленные алгоритмы», **3**-е издание, — Пер. с англ. — М.; СПб.; Киев: ООО «И.Д. Вильямс», 2012. - 832 стр. (или более поздние издания)
6. Дональд Э. Кнут. «Искусство программирования, том 1. Основные алгоритмы», **3**-е издание,— Пер. с англ. — М.; СПб.; Киев: ООО «И.Д. Вильямс», 2011. - 720 стр. (или более поздние издания)
   1. **Программное обеспечение**

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Наименование** | **Условия доступа** |
| 1. | Microsoft Windows 7 Professional RUS  Microsoft Windows 10 | *Из внутренней сети университета (договор)* |
| 2. | Microsoft Office Professional Plus 2010 | *Из внутренней сети университета (договор)* |
| 3 | Microsoft Visual Studio 2015 Community | *Свободно распространяемое ПО* |
| 4 | JetBrains CLion Community Edition | *Свободно распространяемое ПО* |
| 5 | GNU Compiler Collection (GCC) | *Свободно распространяемое ПО* |

* 1. **Материально-техническое обеспечение дисциплины**

Учебные аудитории для лекционных занятий по дисциплине, оснащенные мультимедийным проектором с дистанционным управлением.

Учебные аудитории для практический занятий и самостоятельной работы, оснащенные компьютерами, на которых установлены интегрированные среды разработки Microsoft Visual Studio версии 2015 и выше и/или CLion версии 2016.0 и выше, набор компиляторов GNU Compiler Collection (GCC) версии 6.1 и выше. Компьютеры подключены к сети Интернет для обеспечения доступа студентов к справочным материалам.