

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ



**УТВЕРЖДАЮ**

Проректор по учебной работе и  
образовательным инновациям

О.Г. Прохоренко

«05» июля 2023 г.

Регистрационный № УД – 12450/уч.

**АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ**

**Учебная программа учреждения высшего образования  
по учебной дисциплине для специальности:**

**1-31 03 08**

**Математика и информационные технологии  
(по направлениям)**

Направление специальности:

**1-31 03 08-02**

**Математика и информационные технологии  
(математическое и программное обеспечение  
мобильных устройств)**

2023 г.

Учебная программа составлена на основе образовательного стандарта высшего образования ОСВО 1-31 03 08-2021 типового учебного плана, регистрационный № G31-1-012/пр-тип. от 31.03.2021 и учебных планов: № G31-1-017/уч. от 25.05.2021, № G31-1-001/уч. ин. от 31.05.2021 №G31-1-003/уч.з. от 31.05.2021, №G31-1-221/уч. от 22.03.2022, №G31-1-235/уч.ин. от 27.05.2022, №G31-1-219/уч.з от 27.05.2022.

### **СОСТАВИТЕЛИ:**

С.В. Суздаль, доцент кафедры веб-технологий и компьютерного моделирования Белорусского государственного университета, кандидат физико-математических наук, доцент.

С.А. Вельченко, старший преподаватель кафедры веб-технологий и компьютерного моделирования Белорусского государственного университета.

### **РЕЦЕНЗЕНТЫ:**

А.А. Лобатый, профессор кафедры робототехнических систем ФИТР БНТУ, доктор технических наук, профессор.

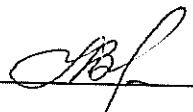
Г.А. Пискун, заместитель декана по научной работе, доцент кафедры проектирования информационно-компьютерных систем ФКП БГУИР, кандидат технических наук, доцент.

### **РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:**

Кафедрой веб-технологий и компьютерного моделирования Белорусского государственного университета  
(протокол № 12 от 30.05.2023 г.);

Научно-методическим советом БГУ  
(протокол № 9 от 29.06.2023 г.)

Заведующий кафедрой  
кандидат физ.-мат. наук, доцент

  
\_\_\_\_\_

М.В. Игнатенко

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дисциплина «Алгоритмы и структуры данных» знакомит студентов с такими фундаментальными понятиями как информация, размерность задачи и трудоемкость алгоритмов. В программу дисциплины включены разделы, позволяющие строить эффективные алгоритмы для разнообразных задач дискретной и комбинаторной оптимизации с использованием различных структур данных. Большое внимание уделяется таким способам решения задач, как организация перебора вариантов с отсечениями и построение приближенных алгоритмов. Даются начальные знания, необходимые для построения и анализа алгоритмов в условиях наличия неполной информации о входных данных.

### **Цели и задачи учебной дисциплины**

**Основной целью** учебной дисциплины «Алгоритмы и структуры данных» является ознакомление студентов с наиболее часто используемыми комбинаторными алгоритмами, с основными идеями, методами и алгоритмическими стратегиями, что позволит подготовить их к решению реальных задач, возникающих на практике.

**Развивающей целью** учебной дисциплины является дальнейшее формирование у студентов навыков алгоритмического мышления и умения применять его в конкретных задачах.

**Основными задачами**, решаемыми в рамках изучения дисциплины «Алгоритмы и структуры данных», являются изучение терминологии, основных утверждений и методов их доказательства, освоение приемов решения типовых задач, а также ознакомление со способами моделирования практических задач в терминах задач из рассматриваемых разделов дискретной математики.

### **Место учебной дисциплины**

Учебная дисциплина «Алгоритмы и структуры данных» относится к модулю «Дискретная математика» компонента учреждения образования.

**Связи** с другими учебными дисциплинами, включая учебные дисциплины компонента учреждения высшего образования, дисциплины специализации и др.

Основой для изучения дисциплины «Алгоритмы и структуры данных» являются дисциплины «Дискретная математика и математическая логика», и «Методы программирования», «Технологии программирования».

Методы, излагаемые в учебной дисциплине, используются при изучении дисциплины «Основы виртуальной и дополненной реальностей», а также при изучении ряда дисциплин специализации. Изучение дисциплины «Алгоритмы и структуры данных» позволяет дать студентам базу, необходимую для успешно-

го усвоения материала перечисленных выше учебных дисциплин, а также получить знания, необходимые им в дальнейшем для успешной работы

### **Требования к компетенциям**

Освоение учебной дисциплины «Алгоритмы и структуры данных» должно обеспечить формирование следующей **специализированной компетенции:**

СК – 5. Применять основные понятия, утверждения и методы для решения базовых задач дискретной математике.

В результате изучения дисциплины студент должен:

**знать:** принципы оценки комбинаторных алгоритмов; наиболее распространенные оценки алгоритмов; структуры данных, используемые при оценке алгоритмов; приемы исчерпывающего поиска; приемы декомпозиции; понятия полиномиальной разрешимой и NP-полной задач; списки наиболее распространенных NP-полных задач;

**уметь:** определять трудоемкость алгоритмов; применять структуры данных для построения алгоритмов; использовать поиск с возвратом для построения алгоритмов; использовать принцип «разделяй и властвуй» для декомпозиции задач; разрабатывать программные реализации основных алгоритмов и структур данных; применять основные алгоритмы и структуры данных для практических задач, возникающих при разработке программно-аппаратных систем обработки информации; разрабатывать эффективные алгоритмы поиска в графах;

**владеть:** методами создания и реализации структур данных; методами оценки трудоемкости алгоритмов; подходами к решению алгоритмических задач на основе известных алгоритмических стратегий.

### **Структура учебной дисциплины**

Дисциплина изучается в 5-6 семестре очной формы обучения и в 3, 4, 5 семестрах заочной формы обучения. Всего на изучение учебной дисциплины «Алгоритмы и структуры данных» отведено:

– для очной формы получения высшего образования –198 часов, в том числе 122 аудиторных часов, из них:

- в 5 семестре –108 часов, в том числе 72 аудиторных часов, из них: лекции – 36 часов, практические занятия – 32 часа, управляемая самостоятельная работа – 4 часа.

- в 6 семестре – 90 часов, в том числе 50 аудиторных часов, из них: лекции – 16 часов, практические занятия – 30 часов, управляемая самостоятельная работа – 4 часа.

Трудоемкость учебной дисциплины составляет 6 зачетных единиц.

Форма текущей аттестации – зачет в 5 семестре, зачет и экзамен в 6 семестре очной формы обучения;

– для заочной формы получения высшего образования – всего 198 часов, в том числе 28 аудиторных часов, из них:

- в 3 семестре – 80 часов, в том числе 14 аудиторных часов, из них: лекции – 8 часов, практические занятия – 6 часов;

- в 4 семестре – 82 часа, в том числе 14 аудиторных часов, из них: лекции – 8 часов, практические занятия – 6 часов;

- в 5 семестре – всего 36 часов.

Трудоемкость учебной дисциплины составляет 6 зачетных единиц.

Форма текущей аттестации – зачет в 3 - 4 семестре, экзамен в 5 семестре заочной формы обучения; а также предусмотрено написание контрольных работ в 4 и 5 семестре.

## СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

### Раздел 1. Структуры данных и сложность алгоритмов

**Тема 1.1** Предмет теории алгоритмов. Прикладное значение эффективности алгоритмов. Связь с дискретной математикой, математической кибернетикой, программированием.

**Тема 1.2** Структуры данных. Стеки, очереди, связанные списки, бинарные деревья.

**Тема 1.3** Понятие алгоритма. Теоретико-числовые задачи: «НОД», «факториал», «возведение в степень», «дискретный логарифм».

**Тема 1.4** Необходимость оценки трудоемкости алгоритмов. Принципы оценки трудоемкости комбинаторных алгоритмов. Асимптотики трудоемкости алгоритмов  $O$ ,  $\Omega$ ,  $\Theta$ .

**Тема 1.5** Алгоритмы поиска подстроки в строке: прямой, Рабина-Карпа, конечный автомат, Кнута-Морриса-Пратта.

**Тема 1.6** Алгоритмы сортировки. Быстрая сортировка. Сортировка слиянием. Линейные сортировки. Теорема о невозможности существования алгоритма сортировки в «худшем» и «в среднем» с трудоемкостью лучшей, чем  $O(n \log n)$ .

**Тема 1.7** Алгоритмы поиска и выборки. Бинарный поиск. Интерполяционный поиск.

**Тема 1.8** Деревья бинарного поиска. Сбалансированные деревья. Операции над деревьями. Хеширование. Открытая и закрытая адресация. Первичные и вторичные хеш-функции.

**Тема 1.9** Детерминированные и недетерминированные машины Тьюринга. Клеточные автоматы. Игра «Жизнь». Программирование на игре «Жизнь».

**Тема 1.10** Понятие о классах  $P$ ,  $NP$  и  $co-NP$ .  $NP$ -полные задачи. Теорема Кука–Карпа–Левина. Сводимость по Куку и по Карпу. Доказательство  $NP$ -полноты для ряда комбинаторных задач. Списки наиболее известных  $NP$ -полных задач.

### Раздел 2. Алгоритмы на графах

**Тема 2.1** Структуры данных для представления графов: матрицы смежности, матрицы инцидентности, списки смежности, списки ребер. Алгоритмы поиска в ширину и глубину, их реализация

**Тема 2.2** Поиск минимального остовного дерева и кратчайшего пути в графе.

**Тема 2.3** Алгоритм нахождения эйлерова цикла.

**Тема 2.4** Паросочетания в двудольных графах, метод увеличивающей цепи.

### **Раздел 3. Основные алгоритмические стратегии**

**Тема 3.1** Эвристики и метаэвристики. Алгоритмы локального поиска, поиска с запретами.

**Тема 3.2** Генетические алгоритмы, алгоритмы имитации отжига.

**Тема 3.3.** Алгоритмы полного перебора, метод ветвей и границ.

**Тема 3.4** Динамическое программирование и метод «разделяй и властвуй». Понятие о методах динамического программирования и «разделяй и властвуй».

### **Раздел 4. Приближенные алгоритмы и аппроксимация с гарантированной точностью**

**Тема 4.1** Алгоритмы с гарантированной оценкой точности. Жадные алгоритмы для покрытия множеств. Приближенные алгоритмы для вершинного покрытия.

**Тема 4.2** Жадный алгоритм для задачи о рюкзаке. Алгоритм Кристофидеса.

**Тема 4.3** Аппроксимация с заданной точностью. Решение задачи о рюкзаке с помощью динамического программирования.

**Тема 4.4** Полностью полиномиальная приближенная схема для задачи о рюкзаке.

### **Раздел 5. Вероятностный анализ детерминированных алгоритмов**

**Тема 5.1** Вероятностный анализ задачи об упаковке выполнимости КНФ.

**Тема 5.2** Точность алгоритма для почти всех входов. Полиномиальность в среднем.

### **Раздел 6. Вероятностные алгоритмы и их анализ**

**Тема 6.1** Вероятностная проверка тождеств. Вероятностные методы в перечислительных задачах.

**Тема 6.2** Вероятностное округление для задачи MAX-SAT. Максимальный разрез в графе.

**Тема 6.3** Методы дерандомизации. Метод условных вероятностей. Метод малых вероятностных пространств. Полиномиальная проверка простоты.

## УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

Очная форма получения высшего образования с применением дистанционных образовательных технологий (ДОТ)

| Номер раздела, темы | Название раздела, темы  | Количество аудиторных часов |                      |                     |                      |      | Количество часов УСР | Форма контроля знаний   |
|---------------------|---|-----------------------------|----------------------|---------------------|----------------------|------|----------------------|---|
|                     |   | Лекции                      | Практические занятия | Семинарские занятия | Лабораторные занятия | Иное |                      |   |
| 1                   | 2   | 3                           | 4                    | 5                   | 6                    | 7    | 8                    | 9   |
| 1                   | <b>Структуры данных и сложность алгоритмов</b>  | 20                          | 16                   |                     |                      |      | 3                    |   |
| 1.1                 | Предмет теории алгоритмов. Понятие алгоритма. Прикладное значение эффективности алгоритмов. Связь с дискретной математикой, математической кибернетикой, программированием.   | 2                           |                      |                     |                      |      |                      |   |
| 1.2                 | Структуры данных. Стеки, очереди, связанные списки, бинарные деревья.   | 2                           | 2                    |                     |                      |      |                      | Устный опрос  |
| 1.3                 | Понятие алгоритма. Теоретико-числовые задачи: «НОД», «факториал», «возведение в степень», «дискретный логарифм».  | 2                           | 2                    |                     |                      |      |                      | Отчет по аудиторным практическим упражнениям                    |
| 1.4                 | Необходимость оценки трудоемкости алгоритмов. Принципы оценки трудоемкости комбинаторных алгоритмов. Асимптотики трудоемкости алгоритмов $O$ , $\Omega$ , $\Theta$ .  | 2                           | 2                    |                     |                      |      |                      | Отчет по аудиторным практическим упражнениям                    |
| 1.5                 | Алгоритмы поиска подстроки в строке: прямой, Рабина-Карпа, конечный автомат, Кнута-Морриса-Пратта.  | 2                           | 2                    |                     |                      |      |                      | Отчет по аудиторным практическим упражнениям                    |
| 1.6                 | Алгоритмы сортировки. Быстрая сортировка. Сортировка слиянием. Линейные сортировки. Теорема о невозможности существования алгоритма сортировки в «худшем» и «в среднем» с трудоемкостью лучшей, чем $O(n \log n)$ . | 2                           |                      |                     |                      |      | 1                    | Отчет по аудиторным практическим упражнениям, учебная дискуссия |
| 1.7                 | Алгоритмы поиска и выборки. Бинарный поиск. Интерполяционный поиск.   | 2                           | 2                    |                     |                      |      |                      | Отчет по аудиторным   |



|          |  |          |          |  |  |  |          |   |
|----------|--|----------|----------|--|--|--|----------|---|
|          |  |          |          |  |  |  |          | практическим упражнениям                                    |
| 1.8      | Деревья бинарного поиска. Сбалансированные деревья. Операции над деревьями. Хеширование. Открытая и закрытая адресация. Первичные и вторичные хеш-функции.   | 2        | 2        |  |  |  | 1        | Отчет по аудиторным практическим упражнениям, анализ кейсов |
| 1.9      | Детерминированные и недетерминированные машины Тьюринга. Клеточные автоматы. Игра «Жизнь». Программирование на игре «Жизнь».   | 2        | 2        |  |  |  | 1        | Отчет по аудиторным практическим упражнениям                |
| 1.1<br>0 | Понятие о классах $P$ , $NP$ и $co-NP$ . $NP$ -полные задачи. Теорема Кука–Карпа–Левина. Сводимость по Куку и по Карпу. Доказательство $NP$ -полноты для ряда комбинаторных задач. Списки наиболее известных $NP$ -полных задач. | 2        | 2        |  |  |  |          | Контрольная работа по разделу 1                             |
| <b>2</b> | <b>Алгоритмы на графах</b>   | <b>8</b> | <b>8</b> |  |  |  | <b>3</b> |   |
| 2.1      | Структуры данных для представления графов: матрицы смежности, матрицы инцидентности, списки смежности, списки ребер. Алгоритмы поиска в ширину и глубину, их реализация  | 2        | 2        |  |  |  | 1        | Отчет по аудиторным практическим упражнениям                |
| 2.2      | Поиск минимального остовного дерева и кратчайшего пути в графе.  | 2        | 2        |  |  |  | 1        | Отчет по аудиторным практическим упражнениям                |
| 2.3      | Алгоритм нахождения эйлерова цикла.  | 2        | 2        |  |  |  | 1        | Отчет по аудиторным практическим упражнениям                |
| 2.4      | Паросочетания в двудольных графах, метод увеличивающей цепи.   | 2        | 2        |  |  |  |          | Коллоквиум  |
| <b>3</b> | <b>Основные алгоритмические стратегии</b>  | <b>8</b> | <b>8</b> |  |  |  |          |   |
| 3.1      | Эвристики и метаэвристики. Алгоритмы локального поиска, поиска с запретами.  | 2        | 2        |  |  |  |          | Отчет по аудиторным практическим упражнениям                |
| 3.2      | Генетические алгоритмы, алгоритмы имитации отжига.   | 2        | 2        |  |  |  |          | Отчет по аудиторным практическим упражнениям                |
| 3.3      | Алгоритмы полного перебора, метод  | 2        | 2        |  |  |  |          | Отчет по  |

|          |  |          |           |  |  |  |          |   |
|----------|--|----------|-----------|--|--|--|----------|---|
|          | ветвей и границ.   |          |           |  |  |  |          | аудиторным<br>практическим<br>упражнениям             |
| 3.4      | Динамическое программирование и метод «разделяй и властвуй». Понятие о методах динамического программирования и «разделяй и властвуй». | 2        | 2         |  |  |  |          | Контрольная<br>работа по раз-<br>делу 3               |
| <b>4</b> | <b>Приближенные алгоритмы и аппроксимация с гарантированной точностью</b>  | <b>8</b> | <b>14</b> |  |  |  | <b>1</b> |   |
| 4.1      | Алгоритмы с гарантированной оценкой точности. Жадные алгоритмы для покрытия множеств. Приближенные алгоритмы для вершинного покрытия.  | 2        | 4         |  |  |  |          | Отчет по<br>аудиторным<br>практическим<br>упражнениям |
| 4.2      | Жадный алгоритм для задачи о рюкзаке. Алгоритм Кристофидеса.   | 2        | 2         |  |  |  |          | Отчет по<br>аудиторным<br>практическим<br>упражнениям |
| 4.3      | Аппроксимация с заданной точностью. Решение задачи о рюкзаке с помощью динамического программирования.                                 | 2        | 4         |  |  |  | 1        | Отчет по<br>аудиторным<br>практическим<br>упражнениям |
| 4.4      | Полностью полиномиальная приближенная схема для задачи о рюкзаке   | 2        | 4         |  |  |  |          | Контрольная<br>работа по раз-<br>делу 4               |
| <b>5</b> | <b>Вероятностный анализ детерминированных алгоритмов</b>   | <b>2</b> | <b>4</b>  |  |  |  | <b>1</b> |   |
| 5.1      | Вероятностный анализ задачи об упаковке выполнимости КНФ.  | 1        | 2         |  |  |  | 1        | Отчет по<br>аудиторным<br>практическим<br>упражнениям |
| 5.2      | Точность алгоритма для почти всех входов. Полиномиальность в среднем.  | 1        | 2         |  |  |  |          | Отчет по<br>аудиторным<br>практическим<br>упражнениям |
| <b>6</b> | <b>Вероятностные алгоритмы и их анализ</b>   | <b>6</b> | <b>12</b> |  |  |  |          |   |
| 6.1      | Вероятностная проверка тождеств. Вероятностные методы в перечислительных задачах.  | 2        | 4         |  |  |  |          | Отчет по<br>аудиторным<br>практическим<br>упражнениям |
| 6.2      | Вероятностное округление для задачи MAX-SAT. Максимальный разрез в графе.  | 2        | 4         |  |  |  |          | Отчет по<br>аудиторным<br>практическим                |

|     |  |           |           |  |  |  |          |  |
|-----|--|-----------|-----------|--|--|--|----------|--|
|     |  |           |           |  |  |  |          | упражнениям                                  |
| 6.3 | Методы дерандомизации. Метод условных вероятностей. Метод малых вероятностных пространств. Полиномиальная проверка простоты. | 2         | 4         |  |  |  |          | Отчет по аудиторным практическим упражнениям |
|     | <b>ИТОГО</b>   | <b>52</b> | <b>62</b> |  |  |  | <b>8</b> |  |

## УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

Заочная и форма получения высшего образования

| Номер раздела, темы | Название раздела, темы  | Количество аудиторных часов |                      |                     |                      |      | Форма контроля знаний   |
|---------------------|---|-----------------------------|----------------------|---------------------|----------------------|------|---|
|                     |   | Лекции                      | Практические занятия | Семинарские занятия | Лабораторные занятия | Иное |   |
| 1                   | 2   | 3                           | 4                    | 5                   | 6                    | 7    | 8   |
| <b>1</b>            | <b>Структуры данных и сложность алгоритмов</b>  | <b>7</b>                    | <b>4</b>             |                     |                      |      |   |
| 1.1                 | Предмет теории алгоритмов. Прикладное значение эффективности алгоритмов. Связь с дискретной математикой, математической кибернетикой, программированием.  | 1                           |                      |                     |                      |      |   |
| 1.2                 | Структуры данных. Стеки, очереди, связанные списки, бинарные деревья.   |                             |                      |                     |                      |      | Устный опрос  |
| 1.3                 | Понятие алгоритма. Теоретико-числовые задачи: «НОД», «факториал», «возведение в степень», «дискретный логарифм».  | 1                           |                      |                     |                      |      |   |
| 1.4                 | Необходимость оценки трудоемкости алгоритмов. Принципы оценки трудоемкости комбинаторных алгоритмов. Асимптотики трудоемкости алгоритмов $O$ , $\Omega$ , $\Theta$ .  | 1                           | 1                    |                     |                      |      | Отчет по аудиторным практическим упражнениям, учебная дискуссия |
| 1.5                 | Алгоритмы поиска подстроки в строке: прямой, Рабина-Карпа, конечный автомат, Кнута-Морриса-Пратта.  |                             |                      |                     |                      |      |   |
| 1.6                 | Алгоритмы сортировки. Быстрая сортировка. Сортировка слиянием. Линейные сортировки. Теорема о невозможности существования алгоритма сортировки в «худшем» и «в среднем» с трудоемкостью лучшей, чем $O(n \log n)$ . | 1                           | 0,5                  |                     |                      |      | Отчет по аудиторным практическим упражнениям, анализ кейсов     |

|          |   |          |          |  |  |  |  |
|----------|---|----------|----------|--|--|--|--|
| 1.7      | Алгоритмы поиска и выборки. Бинарный поиск. Интерполяционный поиск  |          |          |  |  |  | Отчет по аудиторным практическим упражнениям (инд работам) |
| 1.8      | Деревья бинарного поиска. Сбалансированные деревья и реализуемые с их помощью структуры. Операции над деревьями. Хеширование. Открытая и закрытая адресация. Первичные и вторичные хеш-функции                            | 1        | 1        |  |  |  |  |
| 1.9      | Детерминированные и недетерминированные машины Тьюринга. Клеточные автоматы. Игра «Жизнь». Программирование на игре «Жизнь».  | 1        | 0,5      |  |  |  | Отчет по аудиторным практическим упражнениям (инд работам) |
| 1.10     | Понятие о классах $P$ , $NP$ и $co-NP$ . $NP$ -полные задачи. Теорема Кука–Карпа–Левина. Сводимость по Куку и по Карпу. Доказательство $NP$ -полноты для ряда комбинаторных. Списки наиболее известных $NP$ -полных задач | 1        | 1        |  |  |  | Контрольная работа по разделу 1 с устной защитой           |
| <b>2</b> | <b>Алгоритмы на графах</b>  | <b>2</b> | <b>2</b> |  |  |  |  |
| 2.1      | Структуры данных для представления графов: матрицы смежности, матрицы инцидентности, списки смежности, списки ребер. Алгоритмы поиска в ширину и глубину, их реализация   | 1        | 1        |  |  |  | Устный опрос   |
| 2.2      | Поиск минимального остовного дерева и кратчайшего пути в графе.   |          |          |  |  |  |  |
| 2.3      | Алгоритм нахождения эйлерова цикла.   | 1        | 1        |  |  |  | Отчет по аудиторным практическим упражнениям (инд работам) |
| 2.4      | Паросочетания в двудольных графах, метод увеличивающей цепи   |          |          |  |  |  |  |
| <b>3</b> | <b>Основные алгоритмические стратегии</b>   | <b>2</b> | <b>2</b> |  |  |  |  |
| 3.1      | Эвристики и метаэвристики. Алгоритмы локального поиска, поиска с запретами.   | 0,5      | 0,5      |  |  |  | Отчет по аудиторным практическим упражнениям (инд работам) |

|          |  |          |            |  |  |  |
|----------|--|----------|------------|--|--|--|
|          |  |          |            |  |  | работам)   |
| 3.2      | Генетические алгоритмы, алгоритмы имитации отжига.   |          |            |  |  | Отчет по аудиторным практическим упражнениям (инд работам) |
| 3.3      | Алгоритмы полного перебора, метод ветвей и границ  | 1        | 1          |  |  |  |
| 3.4      | Динамическое программирование и метод «разделяй и властвуй». Понятие о методах динамического программирования и «разделяй и властвуй». | 0,5      | 0,5        |  |  | Контрольная работа по разделу 3 с устной защитой           |
| <b>4</b> | <b>Приближенные алгоритмы и аппроксимация с гарантированной точностью</b>  | <b>2</b> | <b>1,5</b> |  |  |  |
| 4.1      | Алгоритмы с гарантированной оценкой точности. Жадные алгоритмы для покрытия множеств. Приближенные алгоритмы для вершинного покрытия.  | 1        | 0,5        |  |  | Отчет по аудиторным практическим упражнениям (инд работам) |
| 4.2      | Жадный алгоритм для задачи о рюкзаке. Алгоритм Кристофидеса.   |          |            |  |  |  |
| 4.3      | Аппроксимация с заданной точностью. Решение задачи о рюкзаке с помощью динамического программирования.                                 | 1        | 1          |  |  | Контрольная работа по разделу 4 с устной защитой           |
| 4.4      | Полностью полиномиальная приближенная схема для задачи о рюкзаке   |          |            |  |  |  |
| <b>5</b> | <b>Вероятностный анализ детерминированных алгоритмов</b>   | <b>1</b> | <b>1</b>   |  |  |  |
| 5.1      | Вероятностный анализ задачи об упаковке выполнимости КНФ.  | 1        | 1          |  |  | Отчет по аудиторным практическим упражнениям (инд работам) |
| 5.2      | Точность алгоритма для почти всех входов. Полиномиальность в среднем   |          |            |  |  |  |
| <b>6</b> | <b>Вероятностные алгоритмы и их анализ</b>   | <b>2</b> | <b>1,5</b> |  |  |  |
| 6.1      | Вероятностная проверка тождеств. Вероятностные методы в перечислительных задачах.  | 1        | 0,5        |  |  | Отчет по аудиторным практическим упражнениям               |
| 6.2      | Вероятностное округление для задачи MAX-SAT. Максимальный разрез в   |          |            |  |  |  |

|     |  |           |           |  |  |  |                       |
|-----|--|-----------|-----------|--|--|--|-----------------------|
|     | графе.   |           |           |  |  |  | ниям (инд<br>работам) |
| 6.3 | Методы дерандомизации. Метод<br>условных вероятностей. Метод малых<br>вероятностных пространств. Полино-<br>миальная проверка простоты | 1         | 1         |  |  |  | Коллокви-<br>ум       |
|     | <b>ИТОГО</b>   | <b>16</b> | <b>12</b> |  |  |  |                       |

## ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

### Перечень основной литературы

1. Лафоре, Р. Структуры данных и алгоритмы Java / Роберт Лафоре ; [пер. с англ. Е. Матвеев]. - 2-е изд. - Санкт-Петербург ; Москва ; Минск : Питер, 2023. - 701 с.
2. Программирование. Сборник задач : учебное пособие / [О.Г.Архипов и др.]; под ред. М. М. Марана. - Изд. 2-е, стер. - Санкт-Петербург ; Москва ; Краснодар : Лань, 2022. - 137 с. - URL: <https://reader.lanbook.com/book/223418>.
3. Алгоритмы: построение и анализ : [пер. с англ.] / Т. Кормен [и др.]. - 3-е изд. - Москва ; Санкт-Петербург : Вильямс, 2018. - 1323 с
4. Луридадас, П. Алгоритмы для начинающих : теория и практика для разработчика / Панос Луридадас ; [пер. с англ. Е. М. Егоровой]. - Москва : Бомбора, 2020.
5. Апанасевич, С. А. Структуры и алгоритмы обработки данных. Линейные структуры : учебное пособие / Апанасевич С. А. - Санкт-Петербург : Лань, 2022. - 136 с. - URL: <https://e.lanbook.com/book/206261>.
6. Павлов, Л. А. Структуры и алгоритмы обработки данных : учебник / Л. А. Павлов, Н. В. Первова. - Изд. 2-е, испр. и доп. - Санкт-Петербург ; Москва ; Краснодар : Лань, 2020. - 254 с. - URL: <https://e.lanbook.com/book/156929>.
7. Андрианова, А. А. Алгоритмизация и программирование. Практикум : учебное пособие / А. А. Андрианова, Л. Н. Исмагилова, Т. М. Мухтарова. - Санкт-Петербург ; Москва ; Краснодар : Лань, 2019. - 236 с. - URL: <https://e.lanbook.com/book/206258>.
8. Рафгарден, Т. Совершенный алгоритм. Основы = Algorithms Illuminated / Тим Рафгарден ; [пер. с англ. А. Логунов]. - Санкт-Петербург [и др.] : Питер, 2019. - 253 с. - URL: <https://ibooks.ru/reading.php?short=1&productid=365286>.

### Перечень дополнительной литературы

1. Вирт, Н. Алгоритмы и структуры данных / Н. Вирт. – Санкт-Петербург: Невский диалект, 2001. – 352 с.
2. Скиена, С. Алгоритмы. Руководство по разработке / С. Скиена. – Санкт-Петербург: БХВ, 2016. – 720 с.
3. Емеличев, В.А. Лекции по теории графов / В.А. Емеличев, О.И. Мельников, В.И. Сарванов, Р.И. Тышкевич. – М.: Наука, 1990. – 384 с.
4. Кнут, Д. Искусство программирования. Т. 2. Получисленные алгоритмы / Д. Кнут. – М.: Вильямс, 2007. – 832 с.
5. Ахо, А. Построение и анализ вычислительных алгоритмов / А. Ахо, Дж. Хопкрофт, Дж. Ульман. – М.: Издательский дом «Вильямс», 2001. – 384 с.



6. Гэри, М. Вычислительные машины и труднорешаемые задачи / М. Гэри, Д. Джонсон. – М.: Мир, 1982. – 416 с.
7. Кнут, Д. Искусство программирования. Т. 1. Основные алгоритмы / Д. Кнут. – М.: Вильямс, 2006. – 720 с.
8. Кузюрин, Н.Н. Эффективные алгоритмы и сложность вычислений / Н.Н. Кузюрин, С.А. Фомин. – М.: МФТИ, 2015. – 363 с.
9. Шень, А.Х. Программирование: теоремы и задачи / А.Х. Шень. – М.: МЦНМО, 2004. – 296 с.

## Перечень рекомендуемых средств диагностики и методика формирования итоговой отметки

Объектом диагностики компетенций студентов являются знания, умения, полученные ими в результате изучения учебной дисциплины. Выявление учебных достижений студентов осуществляется с помощью мероприятий текущего контроля и промежуточной аттестации.

Для диагностики компетенций могут использоваться следующие средства текущего контроля: проведение контрольных работ, устные опросы, отчет по аудиторным практическим упражнениям, коллоквиум, дискуссия.

Диагностика компетенций проводится в устной (собеседование) и устно-письменной (отчеты по аудиторным и домашним практическим упражнениям с устной защитой) формах.

Оценка за ответы на лекциях и практических занятиях определяется полнотой ответа, обоснованием аргументов, приведением практических примеров и т.д.

Формой текущей аттестации по дисциплине «Алгоритмы и структуры данных» предусмотрен **зачет** в 5 -6 семестре и **экзамен** в 6 семестре очной формы обучения; **зачет** в 3-4 семестре и **экзамен** в 5 семестре заочной формы обучения.

При формировании итоговой отметки используется рейтинговая система оценки знаний студента, дающая возможность проследить и оценить динамику процесса достижения целей обучения. Рейтинговая система предусматривает использование весовых коэффициентов для текущего контроля знаний и текущей аттестации студентов по дисциплине.

Весовые коэффициенты, определяющие вклад текущего контроля знаний в итоговую отметку:

формирование отметки за текущую успеваемость:

- опрос на практических занятиях – 30%;
- отчет по аудиторным практическим упражнениям, отчет по индивидуальным работам – 30%;
- контрольная работа – 40%;

Итоговая отметка по дисциплине рассчитывается на основе отметки текущей успеваемости (рейтинговая система оценки знаний) и экзаменационной отметки с учетом их весовых коэффициентов. Вес отметки по текущей успеваемости составляет 40 %, экзаменационной отметки – 60 %.

## Примерный перечень заданий для управляемой самостоятельной работы студентов

### Раздел 1. Структуры данных и сложность алгоритмов

#### Тема 1.6 Алгоритмы сортировки. (1 ч)

##### *Задание 1. Quicksort*

Реализуйте сортировку массива заданной размерности с помощью алгоритма быстрой сортировки. В качестве опорного элемента выберите:

- а) последний элемент массива,
- б) первый элемент массива,
- в) средний элемент массива,
- г) случайный элемент массива.

##### *Задание 2. MergeSort*

Реализуйте сортировку слиянием для массива заданной размерности. Форма контроля – отчет по аудиторным практическим упражнениям.

#### Тема 1.8 Дерево бинарного поиска. Хеширование. (1 ч)

**Задание 1.** Реализуйте структуру данных «дерево бинарного поиска». При этом обязательно должны присутствовать как базовые функции: добавить элемент, удалить элемент, найти элемент, - так и дополнительные: ротировать два узла, вставить в корень, обойти дерево слева направо, просмотреть все дерево.

##### *Задание 2. Хеш-таблица: разрешение коллизий методом цепочек*

Реализуйте хеш-таблицу размерности 128, в которой коллизии разрешаются методом цепочек. Первичное хеширование осуществите при помощи функций:

- а)  $h(i) = i \bmod 128$ ,
- б)  $h(i) = \lfloor 128 \{(\sqrt{5}-1)i/2\} \rfloor$ .

При этом обязательно должны присутствовать три базовые функции: добавить элемент, удалить элемент, найти элемент, - а также функция просмотра всей таблицы.

##### *Задание 3. Хеш-таблица с открытой адресацией*

Реализуйте хеш-таблицу с открытой адресацией размерности 128, в которой осуществляется:

- а) линейное пробирование,
- б) квадратичное пробирование.

При этом обязательно должны присутствовать три базовые функции: добавить элемент, удалить элемент, найти элемент, - а также функция просмотра всей таблицы.

Форма контроля – отчет по аудиторным практическим упражнениям, анализ кейсов.

## **Тема 1.9. Детерминированные и недетерминированные машины Тьюринга. Программирование на машине Тьюринга (1 ч)**

### **Примерный перечень заданий:**

- а) Напишите программу для машины Тьюринга, реализующую следующее преобразование.  
Вход: массив из 0 и 1.  
Выход: ответ «да» (1), если число 0 и число 1 отличаются не более, чем на 3; «нет» (0) - в противном случае.
- б) Напишите программу для машины Тьюринга, реализующую следующее преобразование.  
Вход: массив из 0 и 1.  
Выход: ответ «да» (1), если в массиве есть последовательность из трех подряд идущих 0; «нет» (0) - в противном случае.
- в) Напишите программу для машины Тьюринга, реализующую следующее преобразование.  
Вход: массив из 0 и 1.  
Выход: массив, записанный в обратном порядке.
- г) Напишите программу для машины Тьюринга, реализующую следующее преобразование.  
Вход: массив из 0 и 1.  
Выход: ответ «да» (1), если число 0 и число 1 отличаются на два; «нет» (0) - в противном случае.
- д) Напишите программу для машины Тьюринга, реализующую следующее преобразование.  
Вход: массив из 0 и 1.  
Выход: сумма элементов массива по модулю 2.
- е) Напишите программу для машины Тьюринга, реализующую следующее преобразование.  
Вход: массив из символов  $a, b, c$ .  
Выход: массив, из которого удалены все вхождения символа  $c$ .
- ж) Напишите программу для машины Тьюринга, реализующую следующее преобразование.  
Вход: массив из 0, 1, 2, 3.  
Выход: «да» (1), если если четных элементов больше, чем нечетных; «нет» (0) - в противном случае.
- з) Напишите программу для машины Тьюринга, реализующую следующее преобразование.  
Вход: массив из 0 и 1.

Выход: ответ «да» (1), если сумма элементов массива делится на три; «нет» (0) - в противном случае.

и) Напишите программу для машины Тьюринга, реализующую следующее преобразование.

Вход: натуральное число  $x$  в унарной системе счисления.

Выход: число  $x + 5$ .

**Форма контроля** – отчет по аудиторным практическим упражнениям.

## Раздел 2. Алгоритмы на графах.

**Тема 2.1. Структуры данных для представления графов: матрицы смежности, матрицы инцидентности, списки смежности, списки ребер. Алгоритмы поиска в ширину и глубину, их реализация. (1 ч)**

### Примерный перечень заданий

#### *Задание 1. Граф: поиск в ширину и в глубину*

Реализуйте неориентированный граф, заданный при помощи списка ребер и количества вершин, в виде:

- а) списка смежности вершин,
- б) матрицы смежности,
- в) матрицы инцидентности.

Начиная с некоторой заданной вершины этого графа осуществите поиск в ширину и в глубину.

**Форма контроля** - отчет по аудиторным практическим упражнениям

**Тема 2.2. Поиск минимального остовного дерева и кратчайшего пути в графе. (1 ч)**

### Примерный перечень заданий

#### *Построение минимального остовного дерева*

**Задание 1.** Для связного взвешенного  $n$ -вершинного графа, заданного матрицей инцидентности, реализуйте алгоритмы Краскала и Прима.

#### *Задание 2. Алгоритм Дейкстры*

Для связного взвешенного графа с неотрицательными весами ребер, заданного матрицей смежности, и некоторой его вершины реализуйте алгоритм Дейкстры.

#### *Задание 3. Алгоритм Флойда*

Для связного взвешенного графа с неотрицательными весами ребер, заданного матрицей смежности, реализуйте алгоритм Флойда.

#### *Задание 4. Алгоритм Куна*

Для связного двудольного графа с заданными долями  $X$  и  $Y$  найдите

наибольшее паросочетание по алгоритму Куна.

**Задание 5. Алгоритм Форда-Фалкерсона**

Для транспортной сети  $G = G(V, E)$ ,  $V = \{v_0, v_1, \dots, v_n\}$ , с истоком  $v_0$ , стоком  $v_n$ , заданными пропускными способностями дуг  $c_1, c_2, \dots, c_{|E|}$  и заданным допустимым потоком  $f_1, f_2, \dots, f_{|E|}$  построить поток максимальной мощности. В качестве выходных данных выведите величину потока максимальной мощности и разрез, который определяет этот поток.

**Форма контроля** – отчет по аудиторным практическим упражнениям.

**Тема 2.3 Алгоритм нахождения эйлерова цикла. (1 ч)**

**Задание (примерный вариант)**

Для связного  $n$ -вершинного графа, заданного матрицей инцидентности, определите, эйлеров ли он. Если да, то выведите его эйлеров цикл.

**Форма контроля** – отчет по аудиторным практическим упражнениям.

**Раздел 4. Приближенные алгоритмы и аппроксимация с гарантированной точностью.**

**Тема 4.3. Аппроксимация с заданной точностью. Решение задачи о рюкзаке с помощью динамического программирования. (1 ч)**

**Примерный перечень заданий**

**Задание 1. Раскраски**

Дан  $n$ -граф  $G$  с множеством вершин  $VG = \{v_1, v_2, \dots, v_n\}$ . Нужно найти его хроматическое число  $\chi(G)$ .

**Форма контроля** – отчет по аудиторным практическим упражнениям.

**Задание 2. Задачи динамического программирования.**

**Проход по матрице штрафов**

Задана матрица натуральных чисел  $A = [a_{ij}]_{n \times m}$ . За каждый проход через клетку матрицы с индексами  $(i, j)$  взимается штраф  $a_{ij}$ . Необходимо с минимальным суммарным штрафом пройти из первой клетки первой строки в клетку  $(n, m)$ . Движение по матрице осуществляется следующим образом: из текущей клетки с координатами  $(i, j)$  можно перейти в любую из трех соседних клеток (если они существуют):  $(i+1, j), (i+1, j-1), (i+1, j+1)$ . Найдите величину минимального суммарного штрафа.

**Проход по матрице платежей**

Задана матрица натуральных чисел  $A = [a_{ij}]_{n \times m}$ . За каждый проход через клетку матрицы с индексами  $(i, j)$  выплачивается вознаграждение  $a_{ij}$ . Необходимо найти максимальную сумму, которую можно получить, если двигаться из

клетки  $(1,1)$  в клетку  $(n,m)$ . Движение по матрице осуществляется следующим образом: из текущей клетки с координатами  $(i, j)$  можно перейти в любую из двух соседних клеток (если они существуют):  $(i+1, j), (i, j+1)$ .

#### *Файловая система*

В файловой системе персонального компьютера файлы организованы в каталоги. В компьютере полное имя файла является строкой, состоящей из имен каталогов и имени файла, разделенных символом "\", причем "\" не может быть первым, последним символом, а также идти два раза подряд. Имя файла (каталога) может быть произвольной длины, но длина полного имени файла не может быть длиннее  $n$  символов. В качестве символов, допустимых к употреблению в именах файлов (каталогов), могут использоваться символы из алфавита, состоящего из  $k$  букв (символ "\" не входит в их число). Необходимо для данных  $k$  ( $1 < k < 13$ ) и  $n$  ( $1 < n < 30$ ) определить максимальное число файлов, которое можно записать на данный компьютер.

#### *Радиоактивные отходы*

При переработке радиоактивных материалов образуются отходы двух видов: опасные и неопасные. Для их хранения используются одинаковые контейнеры.

После помещения отходов в контейнеры, последние укладываются одной вертикальной стопкой. Стопка считается опасной, если в ней подряд идет не менее трех опасных контейнеров, в противном случае стопка считается безопасной. Для заданного количества контейнеров  $1 \leq n \leq 13$  определите число вариантов формирования безопасной стопки.

#### *Системы счисления*

Вычислите количество  $n$ -значных чисел в системе счисления с основанием  $k$  ( $2 \leq k \leq 10, k+n \leq 18$ ) таких, что их запись не содержит двух подряд идущих нулей.

#### *Локомотивы*

На железнодорожной колее размещены  $n$  локомотивов с номерами от 1 до  $n$ ,  $1 \leq k \leq 16$ . Они начинают двигаться в одну сторону, причем локомотив номер  $i$  движется со скоростью  $i$  м/с. Если локомотив, движущийся с большей скоростью, нагоняет более медленный локомотив, дальше они движутся один за другим со скоростью впереди идущего локомотива. Определите, сколько начальных расстановок дадут в результате ровно  $k$   $1 \leq k \leq n$  групп движущихся локомотивов?

#### *Блок из единиц*

В таблице размера  $n$  на  $m$ , состоящей из 0 и 1, найдите квадратный блок максимального размера, состоящий только из 1.

#### *Обмен валют*

Матрица  $A = [a_{ij}]_{n \times m}$  заполнена неотрицательными вещественными числами. Число  $a_{ij}$  определяет курс обмена валюты  $i$  на валюту  $j$ . Так, например,

если  $a_{ij} = 4$ , то это значит, что за единицу валюты  $i$  дают 4 единицы валюты  $j$ . Если  $a_{ij} = 0$ , то полагаем, что курс обмена валюты  $i$  на валюту  $j$  прямо не установлен. Необходимо определить, можно ли, имея некоторую сумму денег в одной из валют, получить большую сумму денег в той же валюте, совершив несколько обменов.

#### *Лента*

Задана лента шириной в одну клетку и длиной в  $n$  клеток. На каждой клетке написано некоторое целое число. Играют два игрока, которые ходят поочередно. За один ход игрок отрезает от ленты одну из крайних клеток и забирает ее. Игра останавливается, когда лента заканчивается. Выигрыш игрока равен сумме чисел находящихся у него клеток ленты. Необходимо определить величину выигрыша, которую может гарантировать себе игрок, начинающий игру первым.

#### *Игра в числа*

Задается некоторое натуральное число  $n < 998$ . Двое играющих называют по очереди числа, меньшие 1000, по следующим правилам. Начиная с числа  $n$ , каждое новое число должно увеличивать одну из цифр предыдущего числа (возможно, незначащий нуль) на 1, 2 или 3. Проигравшим считается тот, кто называет число 999. Для заданного  $n$  необходимо определить, может ли выиграть игрок, делающий первый ход, при наилучших последующих ходах противника. В случае возможности выигрыша первым игроком, требуется напечатать все его возможные выигрышные первые ходы.

#### *Удали до палиндрома*

Вводится непустая строка, которая имеет длину не более 100 символов и состоит только из прописных латинских букв. Палиндромом называется такая строка символов, которая читается слева направо и справа налево одинаково. Необходимо удалить из строки минимальное количество символов так, чтобы получился палиндром. Этот палиндром нужно вывести на печать.

#### *Добавь до палиндрома*

Вводится непустая строка, которая имеет длину не более 100 символов и состоит только из прописных латинских букв. Палиндромом называется такая строка символов, которая читается слева направо и справа налево одинаково. Необходимо вставить в строку минимальное количество символов так, чтобы получился палиндром. Этот палиндром нужно вывести на печать.

#### *Последовательная упаковка дисков*

Имеется  $n$  дисков одинаковой толщины с радиусами  $r_1, r_2, \dots, r_n$ . Эти диски упаковываются в коробку таким образом, что каждый из них стоит ребром на дне коробки, и все диски находятся в одной плоскости. Для последовательности дисков необходимо найти длину коробки, в которую все диски могут быть упакованы, при условии, что диски можно упаковывать только в порядке их следования.



### *Произвольная упаковка дисков*

Имеется  $n$  дисков одинаковой толщины с радиусами  $r_1, r_2, \dots, r_n$ . Эти диски упаковываются в коробку таким образом, что каждый из них стоит ребром на дне коробки, и все диски находятся в одной плоскости. Для последовательности дисков необходимо определить минимальную длину коробки, если порядок упаковки дисков в коробку может быть произвольным (указать при этом порядок, в котором упаковываются диски).

### *Триангуляция многоугольника*

Выпуклый  $n$ -угольник ( $n \geq 3$ ) задается координатами своих вершин в порядке обхода по контуру. Необходимо разбить  $n$ -угольник на треугольники  $n-3$  диагоналями, не пересекающимися, кроме как в вершинах многоугольника, таким образом, чтобы сумма их длин была минимальной.

### *Триангуляция многоугольника с минимаксной диагональю*

Выпуклый  $n$ -угольник ( $n \geq 3$ ) задается координатами своих вершин в порядке обхода по контуру. Необходимо разбить  $n$ -угольник на треугольники  $n-3$  диагоналями, не пересекающимися, кроме как в вершинах многоугольника, таким образом, чтобы максимальная из диагоналей имела наименьшую длину.

### *Невозможная покупка*

Покупатель имеет  $n$  купюр достоинством  $a_1, a_2, \dots, a_n$  и продавец имеет  $m$  купюр достоинством  $b_1, b_2, \dots, b_m$ . Необходимо найти максимальную стоимость товара, которую покупатель не может купить, потому что нет возможности точно рассчитаться за этот товар с продавцом, хотя денег на покупку у него достаточно.

### *Точная сдача*

Покупатель имеет  $n$  купюр достоинством  $a_1, a_2, \dots, a_n$  и продавец имеет  $m$  купюр достоинством  $b_1, b_2, \dots, b_m$ . Необходимо определить, может ли покупатель приобрести вещь стоимости  $s$  так, чтобы у продавца нашлась точная сдача (если она необходима).

### *Железная дорога*

Железная дорога с односторонним движением имеет  $n$  станций. Известны цены билетов от  $i$ -ой станции до  $j$ -ой. Найдите минимальную стоимость проезда от начала до конца с учетом возможной экономии за счет пересадок.

### *Возрастающая подпоследовательность без разрывов*

Необходимо из заданной числовой последовательности  $a$ , состоящей из  $n$  элементов, вычеркнуть минимальное количество элементов так, чтобы оставшиеся элементы образовали строго возрастающую подпоследовательность элементов. Построенный алгоритм должен иметь трудоемкость  $O(n \log n)$ .

### *Возрастающая подпоследовательность с разрывом*

Необходимо из заданной числовой последовательности  $a$ , состоящей из  $n$  элементов, вычеркнуть минимальное количество элементов так, чтобы в оставшейся подпоследовательности каждый последующий элемент был строго больше предыдущего, кроме не более одной пары соседних элементов (одного «разрыва»). Построенный алгоритм должен иметь трудоемкость  $O(n \log n)$ .

#### *Выбор подпоследовательности*

Из элементов заданной последовательности  $a$  целых чисел длины  $n$  необходимо выбрать элементы таким образом, чтобы они образовали подпоследовательность наибольшей длины, в которой каждый последующий элемент делился бы нацело на предыдущий. Порядок следования элементов можно менять.

#### *Бинарная операция*

Задано конечное непустое множество с бинарной операцией  $\square$  (вообще говоря, некоммутативной и неассоциативной). Имеется  $n$  элементов  $a_1, a_2, \dots, a_n$  этого множества и еще один элемент  $x$ . Проверьте, можно ли так расставить скобки в выражении  $a_1 \square a_2 \square \dots \square a_n$ , чтобы в результате получился  $x$ . Построенный алгоритм должен иметь трудоемкость  $O(n^3)$ .

#### *Задача о рюкзаке*

Имеется  $n$  положительных чисел  $a_1, a_2, \dots, a_n$  - веса различных предметов, и число  $m$  — максимально допустимый вес рюкзака. Выясните, можно ли получить  $m$ , складывая некоторые из чисел  $a_1, a_2, \dots, a_n$ . Построенный алгоритм должен иметь трудоемкость  $O(nm)$ .

#### *Преобразование строк*

На вход подаются две символьные строки  $S_1$  и  $S_2$ . Необходимо преобразовать строку  $S_1$  в строку  $S_2$  с минимальным суммарным штрафом, который определяется следующим образом:

- штраф за удаление символа из строки  $S_1$  равен  $a$  баллов;
- штраф за вставку символа в строку  $S_1$  -  $b$  баллов;
- штраф за выполнение замены символа в строке  $S_1$  на любой другой символ -  $c$  баллов.

**Форма контроля** - отчет по аудиторным практическим упражнениям

## **Раздел 5. Вероятностный анализ детерминированных алгоритмов**

### **Тема 5.1 Вероятностный анализ задачи об упаковке выполнимости КНФ. (1 ч)**

*Одномерная упаковка.*

#### **Примерный перечень заданий**

**Задание 1** Дано множество  $A = \{a_1, a_2, \dots, a_n\}$  предметов и их веса (длины)  $l(a_i) \in (0, 1)$ ,  $i \in N_n = \{1, 2, \dots, n\}$ . Нужно найти разбиение множества  $A$  на минимальное число подмножеств (контейнеров)  $A_1, A_2, \dots, A_m$  такое, что для каждого

контейнера  $A_j$  выполняется ограничение по вместимости  $\sum_{a_i \in A_j} l(a_i) \leq 1$ .

Форма контроля – отчет по аудиторным практическим упражнениям.

**Задание 2** Дано множество  $A = \{a_1, a_2, \dots, a_n\}$  прямоугольных предметов длины  $l(a_i) \in (0, 1)$  и ширины  $w(a_i) \in (0, 1)$ ,  $i \in N_n = \{1, 2, \dots, n\}$ . Поворачивать предметы нельзя. Стороны длины  $l(a_i)$  параллельны оси абсцисс, а стороны длины  $w(a_i)$  параллельны оси ординат. Необходимо найти упаковку.

- а) в положительном квадранте с минимальной площадью окаймляющего прямоугольника;
- б) в полубесконечную полосу ширины 1 с минимальной длиной занимаемой полосы;
- в) в минимальное число контейнеров размера  $1 \times 1$ .

**Форма контроля** – отчет по аудиторным практическим упражнениям.

### Рекомендуемые темы контрольных работ

#### 1. Раздел 1. Структуры данных и сложность алгоритмов.

Оценка трудоемкости алгоритмов; доказательство  $NP$ -полноты и  $NP$ -трудности алгоритмических задач; машины Тьюринга; бинарный и интерполяционный поиск; деревья бинарного поиска; хеширование с открытой и закрытой адресацией; алгоритмы поиска в ширину и глубину; поиск минимального остовного дерева и кратчайшего пути в графе; алгоритм нахождения эйлера цикла; паросочетания в двудольных графах, метод увеличивающей цепи; динамическое программирование.

#### 2. Раздел 3. Основные алгоритмические стратегии.

Эвристики и метаэвристики. Алгоритмы локального поиска, поиска с запретами. Генетические алгоритмы, алгоритмы имитации отжига. Алгоритмы полного перебора, метод ветвей и границ. Динамическое программирование и метод «разделяй и властвуй». Понятие о методах динамического программирования и «разделяй и властвуй».

#### 3. Раздел 4. Приближенные алгоритмы и аппроксимация с гарантированной точностью.

Жадный алгоритм для задачи о рюкзаке; полностью полиномиальная приближенная схема; жадные алгоритмы для покрытия множеств. приближенные алгоритмы для вершинного покрытия; вероятностный анализ задачи об упаковке выполнимости КНФ; вероятностная проверка тождеств; вероятностное округ-

ление для задачи MAX-SAT; нахождение максимального разреза в графе; метод условных вероятностей; полиномиальная проверка простоты.

### Примерная тематика практических занятий

**Занятие № 1.** Структуры данных. Стеки, очереди, связные списки, бинарные деревья.

**Занятие № 2.** Теоретико-числовые задачи: «НОД», «факториал», «возведение в степень», «дискретный логарифм».

**Занятие № 3.** Необходимость оценки трудоемкости алгоритмов. Принципы оценки трудоемкости комбинаторных алгоритмов. Асимптотики трудоемкости алгоритмов  $O$ ,  $\Omega$ ,  $\Theta$ .

**Занятие № 4.** Алгоритмы поиска подстроки в строке: прямой, Рабина-Карпа, конечный автомат, Кнута-Морриса-Пратта.

**Занятие № 5.** Алгоритмы сортировки. Быстрая сортировка. Сортировка слиянием. Линейные сортировки. Теорема о невозможности существования алгоритма сортировки в «худшем» и «в среднем» с трудоемкостью лучшей, чем  $O(n \log n)$ .

**Занятие № 6.** Алгоритмы поиска и выборки. Бинарный поиск. Интерполяционный поиск.

**Занятие № 7.** Деревья бинарного поиска. Сбалансированные деревья. Операции над деревьями. Хеширование. Открытая и закрытая адресация. Первичные и вторичные хеш-функции.

**Занятие № 8.** Детерминированные и недетерминированные машины Тьюринга. Клеточные автоматы. Игра «Жизнь». Программирование на игре «Жизнь».

**Занятие № 9.** Понятие о классах  $P$ ,  $NP$  и  $co-NP$   $NP$ -полные задачи. Теорема Кука–Карпа–Левина. Сводимость по Куку и по Карпу. Доказательство  $NP$ -полноты для ряда комбинаторных задач. Списки наиболее известных  $NP$ -полных задач.

**Занятие № 10.** Структуры данных для представления графов: матрицы смежности, матрицы инцидентности, списки смежности, списки ребер. Алгоритмы поиска в ширину и глубину, их реализация

**Занятие № 11.** Поиск минимального остовного дерева и кратчайшего пути в графе

**Занятие № 12.** Алгоритм нахождения эйлерова цикла.

**Занятие № 13.** Паросочетания в двудольных графах, метод увеличивающей цепи.

**Занятие № 14.** Генетические алгоритмы, алгоритмы имитации отжига.

**Занятие № 15.** Алгоритмы полного перебора, метод ветвей и границ.

**Занятие № 16.** Динамическое программирование и метод «разделяй и властвуй». Понятие о методах динамического программирования и «разделяй и властвуй».

**Занятие № 17.** Алгоритмы с гарантированной оценкой точности. Жадные алгоритмы для покрытия множеств. Приближенные алгоритмы для вершинного покрытия.

**Занятие № 18.** Алгоритмы с гарантированной оценкой точности. Жадные алгоритмы для покрытия множеств. Приближенные алгоритмы для вершинного покрытия.

**Занятие № 19.** Жадный алгоритм для задачи о рюкзаке. Алгоритм Кристофидеса.

**Занятие № 20.** Аппроксимация с заданной точностью. Решение задачи о рюкзаке с помощью динамического программирования.

**Занятие № 21.** Аппроксимация с заданной точностью. Решение задачи о рюкзаке с помощью динамического программирования.

**Занятие № 22.** Полностью полиномиальная приближенная схема для задачи о рюкзаке

**Занятие № 23.** Полностью полиномиальная приближенная схема для задачи о рюкзаке

**Занятие №24.** Вероятностный анализ задачи об упаковке выполнимости КНФ.

**Занятие № 25.** Точность алгоритма для почти всех входов. Полиномиальность в среднем.

**Занятие № 26.** Вероятностная проверка тождеств. Вероятностные методы в перечислительных задачах.

**Занятие № 27.** Вероятностная проверка тождеств. Вероятностные методы в перечислительных задачах.

**Занятие № 28.** Вероятностное округление для задачи MAX-SAT. Максимальный разрез в графе.

**Занятие № 29.** Вероятностное округление для задачи MAX-SAT. Максимальный разрез в графе.

**Занятие № 30.** Методы дерандомизации. Метод условных вероятностей. Метод малых вероятностных пространств. Полиномиальная проверка простоты.

**Занятие № 31.** Методы дерандомизации. Метод условных вероятностей. Метод малых вероятностных пространств. Полиномиальная проверка простоты.

## Описание инновационных подходов и методов к преподаванию учебной дисциплины

Следующие подходы могут быть использованы при организации образовательного процесса:

– ***эвристический подход***

- демонстрация многообразия решений большинства профессиональных задач и жизненных проблем;

- индивидуализация обучения через возможность самостоятельно ставить цели, осуществлять рефлексию собственной образовательной деятельности.

– ***практико-ориентированный подход***, который предполагает

- освоение содержания образования через решения практических задач;

- освоение содержания образования через решения практических задач;

- приобретение навыков эффективного выполнения разных видов профессиональной деятельности;

- приобретение навыков эффективного выполнения разных видов профессиональной деятельности;

- ориентация на генерирование идей, реализацию групповых студенческих проектов, развитие предпринимательской культуры;

- использование процедур, способов оценивания, фиксирующих сформированность профессиональных компетенций.

– ***метод анализа конкретных ситуаций (кейс-метод)***:

- приобретение студентом знаний и умений для решения практических задач;

- анализ ситуации, используя профессиональные знания, собственный опыт, дополнительную литературу и иные источники.

– **метод развития критического мышления**, то есть способность оценивать и анализировать ситуации с различных позиций, с целью прихода к обоснованному и твердому решению.

– **метод группового обучения** часто встречающихся для математических специальностей выделяют восемь форм группового обучения (обучение в командах, метод учебного турнира, метод командной поддержки индивидуального обучения, метод кооперативного взаимообучения, метод группового исследования, метод группового исследования, лабораторный метод, поисковый метод, приложение).

– **метод учебной дискуссии**, с одной стороны, предполагает наличие у учащихся умения ясно и точно формулировать свои мысли, строить систему аргументированных доказательств, с другой стороны, учит их мыслить, спорить, доказывать свою правоту. Преподаватель должен сам демонстрировать

перед учениками образец такого стиля аргументации, учить их точно излагать свои мысли и терпимо относиться к формулировкам оппонентов, уважительно вносить поправки в их аргументацию, ненавязчиво сохранять за собой право на последнее слово, не претендуя на обладание истиной в последней инстанции.

### **Методические рекомендации по организации самостоятельной работы обучающихся**

Самостоятельная работа студентов – это любая деятельность, связанная с воспитанием мышления будущего профессионала. В широком смысле под самостоятельной работой следует понимать совокупность всей самостоятельной деятельности студентов, как в учебной аудитории, так и вне её, в контакте с преподавателем или в его отсутствии.

Самостоятельная работа реализуется:

1. Непосредственно в процессе аудиторных занятий на лекциях и практических занятиях.
2. В контакте с преподавателем вне рамок расписания на консультациях по учебным вопросам, в ходе творческих контактов, при ликвидации задолженностей, при выполнении индивидуальных заданий и т.д.
3. В библиотеке, дома, в общежитии, на кафедре при выполнении студентами учебных и творческих задач.

При изучении дисциплины организация самостоятельной работы студентов должна представлять единство трех взаимосвязанных форм:

1. Внеаудиторная самостоятельная работа;
2. Аудиторная самостоятельная работа, которая осуществляется под непосредственным руководством преподавателя;
3. Творческая, в том числе научно-исследовательская работа.

Виды внеаудиторной самостоятельной работы студентов могут принимать разнообразные формы: изучение рекомендуемой литературы по дисциплине, подготовка и написание рефератов, докладов, очерков и других письменных работ на заданные темы.

Аудиторная самостоятельная работа может реализовываться при проведении практических занятий, а также во время чтения лекций.

При чтении лекционного курса непосредственно в аудитории иногда возникает необходимость проконтролировать усвоение материала основной массой студентов путем проведения экспресс-опросов по конкретным темам.

На практических занятиях нужно значительную часть времени отводить на самостоятельное решение задач. Типичное практическое занятие целесообразно строить следующим образом:

1. Вводное слово преподавателя (цели занятия, основные вопросы, которые должны быть рассмотрены).
2. Беглый опрос.
3. Решение 1 – 2 типовых задач (с вызовом студента к доске).
4. Самостоятельное решение задач.
5. Разбор типовых ошибок при решении (в конце текущего занятия или в начале следующего).

Результативность самостоятельной работы студентов во многом определяется наличием активных методов ее контроля. В рамках дисциплины существуют следующие виды контроля:

- входной контроль знаний и умений студентов в начале изучения дисциплины;
- текущий контроль, то есть регулярное отслеживание уровня усвоения материала на лекциях и практических занятиях;
- промежуточный контроль по окончании изучения раздела учебной дисциплины;
- самоконтроль, осуществляемый студентом в процессе изучения дисциплины при подготовке к контрольным мероприятиям;
- итоговый контроль по дисциплине в виде экзамена.

### **Примерный перечень вопросов к зачету**

1. Трудоемкость алгоритмов
2. Алгоритмы сортировки. QuickSort. Сложность задачи сортировки
3. Сортировка слиянием. Сложность задачи сортировки
4. Бинарные поисковые деревья
5. Хеширование
6. Графы. Структуры данных для представления графов
7. Алгоритм для нахождения эйлерова цикла
8. Поиск в ширину и его применение
9. Поиск в глубину и его применение
10. Задача об остовном дереве. Алгоритмы Прима и Краскала, их реализация, доказательство оптимальности
11. Кратчайший путь между парой вершин в графе. Алгоритм Дейкстры
12. Кратчайшие пути между всеми вершинами графа. Алгоритм Флойда-Уоршелла



13. Паросочетания в двудольных графах. Алгоритм построения наибольшего паросочетания в двудольном графе
14. Алгоритм Гейла-Шепли решения задачи об устойчивом паросочетании
15. Потоки и разрезы в сетях, алгоритм Флойда-Фолкерсона

### **Примерный перечень вопросов к экзамену**

1. Трудоемкость алгоритмов
2. Алгоритмы сортировки. QuickSort. Сложность задачи сортировки
3. Сортировка слиянием. Сложность задачи сортировки
4. Бинарные поисковые деревья
5. Хеширование
6. Графы. Структуры данных для представления графов
7. Алгоритм для нахождения эйлерова цикла
8. Поиск в ширину и его применение
9. Поиск в глубину и его применение
10. Задача об остовном дереве. Алгоритмы Прима и Краскала, их реализация, доказательство оптимальности
11. Кратчайший путь между парой вершин в графе. Алгоритм Дейкстры
12. Кратчайшие пути между всеми вершинами графа. Алгоритм Флойда-Уоршелла
13. Паросочетания в двудольных графах. Алгоритм построения наибольшего паросочетания в двудольном графе
14. Алгоритм Гейла-Шепли решения задачи об устойчивом паросочетании
15. Алгоритм ссылочного ранжирования (на примере простейшего алгоритма подсчета PageRank)
16. Сжатие информации. Алгоритм Хаффмана
17. Потоки и разрезы в сетях, алгоритм Флойда-Фолкерсона
18. Алгоритмические стратегии (принцип «Разделяй и властвуй», динамическое программирование, жадные алгоритмы)
19. Алгоритмы поиска подстроки. Алгоритм Кнута-Морриса-Пратта
20. Алгоритмы поиска подстроки. Алгоритм Бойера-Мура
21. Алгоритмы поиска подстроки. Алгоритм Рабина-Карпа
22. Машины Тьюринга
23. Классы P и NP. Полиномиальное сведение
24. NP-полные задачи. Шесть основных NP-полных задач (по Гери и Джонсону)
25. Алгоритмы с гарантированной оценкой точности. Задача упаковки

- 26.Метод локального поиска и поиска с запретами. Алгоритм 2-замены для задачи коммивояжера
- 27.Метод ветвей и границ. Задача коммивояжера
- 28.Алгоритм Кристофидеса решения метрической задачи коммивояжера
- 29.Задача упаковки
- 30.Задача о раскраске графа. Эвристические алгоритмы ее решения
- 31.Генетические алгоритмы

## ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ УВО

| Название учебной дисциплины, с которой требуется согласование | Название кафедры                             | Предложения об изменениях в содержании учебной программы учреждения высшего образования по учебной дисциплине | Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола) |
|---|--|---|---|
| «Основы виртуальной и дополненной реальности»                 | Веб-технологий и компьютерного моделирования | Нет   | Оставить содержание учебной дисциплины без изменения<br>(протокол № 12 от 30.05.2023 г.)          |

## ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ УВО

на \_\_\_\_\_ год

| №<br>п/п | Дополнения и изменения | Основание |
|----------|------------------------|-----------|
|          |                        |           |

Учебная программа пересмотрена и одобрена заседании веб-технологий и компьютерного моделирования Белорусского государственного университета (протокол. № \_\_\_\_\_ от \_\_\_\_\_ г.)

Заведующий кафедрой,  
кандидат физ.-мат. наук, доцент

\_\_\_\_\_

М.В. Игнатенко

УТВЕРЖДАЮ  
Декан факультета,  
доктор физ.-мат. наук, профессор

\_\_\_\_\_

С.М. Босяков