

**ГОУ ВПО Российско-Армянский (Славянский)  
университет**

**Утверждено**  
**Директор Института математики и информатики**  
**Арамян Р.Г.**



**«21» мая 2025, протокол №9.1**

**УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ДИСЦИПЛИНЫ**

**Наименование дисциплины: Алгоритмы**

**Автор: Аслабян Айк Каренович, канд.физ.-мат.наук**

**Направление подготовки: 01.03.02 Прикладная математика и информатика**  
**Наименование образовательной программы: 01.03.02 Прикладная математика и информатика**

# 1. АННОТАЦИЯ

## 1.1. Краткое описание содержания данной дисциплины;

Дисциплина "Алгоритмы в программировании" изучает теоретические основы и практические методы разработки, анализа и применения алгоритмов в программировании. Курс охватывает фундаментальные алгоритмические концепции, структуры данных и методы решения вычислительных задач.

## 1.2. Трудоемкость в академических кредитах и часах, формы итогового контроля (экзамен/зачет);

72 часа, экзамен.

## 1.3. Взаимосвязь дисциплины с другими дисциплинами учебного плана специальности (направления)

Для удачного прохождения данного курса студенты должны знать структуры данных, дискретную математику уметь писать код на некотором языке программирования.

Иметь навыки аналитического мышления.

## 1.4. Результаты освоения программы дисциплины:

| Код компетенции | Наименование компетенции                                                                                                                 | Код индикатора достижения компетенций | Наименование индикатора достижений компетенций                                                                                                   |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| ПК-7            | способностью к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения | 1                                     | Знать методы и технологии разработки и применения системного и прикладного программного обеспечения                                              |
|                 |                                                                                                                                          | 2                                     | Разрабатывать и применять алгоритмические и программные решения в области системного и прикладного программного обеспечения                      |
|                 |                                                                                                                                          | 3                                     | Владеть способностью разрабатывать и применять алгоритмические и программные решения в области системного и прикладного программного обеспечения |
| ПК-8            | способностью приобретать и использовать организационно-управленческие навыки в профессиональной и социальной                             | 1                                     | Знать основные методы координации деятельности органов управления, организации надзора,                                                          |

|             |                                                                                                                                                                  |   |                                                                                                                                                                                 |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|             | деятельности                                                                                                                                                     |   | контроля и информационного обеспечения профессиональной деятельности                                                                                                            |
|             |                                                                                                                                                                  | 2 | Уметь применять нормативноправовые основы действующего законодательства в области решения профессиональных задач                                                                |
|             |                                                                                                                                                                  | 3 | Владеть навыками принятия управленческих решений в области профессиональной деятельности                                                                                        |
| <b>ПК-9</b> | способностью составлять и контролировать план выполняемой работы, планировать необходимые для выполнения работы ресурсы, оценивать результаты собственной работы | 1 | Знать план выполняемой работы в ходе практической деятельности для получения профессиональных умений и опыта профессиональной деятельности                                      |
|             |                                                                                                                                                                  | 2 | Уметь составлять и контролировать план выполняемой работы, планировать необходимые для выполнения работы ресурсы, оценивать результаты собственной работы                       |
|             |                                                                                                                                                                  | 3 | Регулярная оценка и анализ результатов выполненной работы с целью выявления достижений, а также выявления возможных областей для улучшения и развития профессиональных навыков. |
| <b>УК-1</b> | Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач                                   | 1 | Знает, как осуществлять поиск, критический анализ и синтез информации для решения поставленных профессиональных задач                                                           |
|             |                                                                                                                                                                  | 2 | Умеет применять системный подход на основе поиска, критического анализа и синтеза информации для решения задач профессиональной области                                         |

|              |                                                                                                                                                                    |   |                                                                                                                                                                        |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              |                                                                                                                                                                    | 3 | Владеет навыками поиска, синтеза и критического анализа информации в своей профессиональной области; владеет системным подходом для решения поставленных задач         |
| <b>ОПК-2</b> | Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач | 1 | Знает основные задачи и области применения математических методов, основные принципы математического моделирования, методы построения и анализа математических моделей |
|              |                                                                                                                                                                    | 2 | Умеет выбирать математические методы, адаптировать и использовать их для разработки и реализации алгоритмов решения прикладных задач                                   |
|              |                                                                                                                                                                    | 3 | Осуществляет выбор и адаптацию математических методов и систем программирования для разработки и реализации алгоритмов решения прикладных задач                        |

## 2. УЧЕБНАЯ ПРОГРАММА

### 2.1. Цели и задачи дисциплины

### 2.2. Трудоемкость дисциплины и виды учебной работы (в академических часах и зачетных единицах)

| Виды учебной работы                                                     | Всего, в<br>акад.<br>часах | Распределение по семестрам |          |          |           |          |           |
|-------------------------------------------------------------------------|----------------------------|----------------------------|----------|----------|-----------|----------|-----------|
|                                                                         |                            | IV<br>сем                  | —<br>сем | —<br>сем | —<br>сем. | —<br>сем | —<br>сем. |
| 1                                                                       | 2                          | 3                          | 4        | 5        | 6         | 7        | 8         |
| <b>1. Общая трудоемкость изучения дисциплины по семестрам, в т. ч.:</b> | <b>216</b>                 | <b>216</b>                 |          |          |           |          |           |
| 1.1. Аудиторные занятия, в т. ч.:                                       | <b>64</b>                  | <b>64</b>                  |          |          |           |          |           |
| 1.1.1. Лекции                                                           | <b>32</b>                  | <b>32</b>                  |          |          |           |          |           |
| 1.1.2. Практические занятия, в т. ч.                                    | <b>32</b>                  | <b>32</b>                  |          |          |           |          |           |
| 1.1.3. Самостоятельная работа                                           | <b>125</b>                 | <b>125</b>                 |          |          |           |          |           |
| 1.1.4. Другое                                                           | <b>27</b>                  | <b>27</b>                  |          |          |           |          |           |

|                   |  |             |  |  |  |  |  |
|-------------------|--|-------------|--|--|--|--|--|
| Итоговый контроль |  | Экзам<br>ен |  |  |  |  |  |
|-------------------|--|-------------|--|--|--|--|--|

## 2.3. Содержание дисциплины

### 2.3.1. Тематический план и трудоемкость аудиторных занятий (модули, разделы дисциплины и виды занятий) по рабочему учебному плану

| Разделы и темы дисциплины                                                                                                    | Всего<br>(ак. часов)    | Лекции<br>(ак.<br>часов) | Практ.<br>Занятия<br>(ак.<br>часов) | Семинары<br>(ак. часов) | Лабор.<br>(ак.<br>часов) |
|------------------------------------------------------------------------------------------------------------------------------|-------------------------|--------------------------|-------------------------------------|-------------------------|--------------------------|
| <b>1</b>                                                                                                                     | <b>2=3+4+5+6<br/>+7</b> | <b>3</b>                 | <b>4</b>                            | <b>5</b>                | <b>6</b>                 |
| Тема 1. Корректность алгоритмов, анализ алгоритмов (worst-case, average-case, best-case), асимптотический анализ алгоритмов. | <b>4</b>                | <b>2</b>                 | <b>2</b>                            |                         |                          |
| Тема 2. Алгоритмы сортировки (bubble, insertion, radix, bucket), нижняя оценка для сортировки сравнениями.                   | <b>8</b>                | <b>4</b>                 | <b>4</b>                            |                         |                          |
| Тема 3. Парадигма разделяй и властвуй, сортировка слиянием, бинарный поиск, умножения матриц, алгоритм Штрассена).           | <b>4</b>                | <b>2</b>                 | <b>2</b>                            |                         |                          |
| Тема 4. Динамическое программирование, числа Фибоначи, поиск наибольшей общей подпоследовательности.                         | <b>4</b>                | <b>2</b>                 | <b>2</b>                            |                         |                          |
| Тема 5. Самая длинная возрастающая подпоследовательность, задача о рюкзаке.                                                  | <b>4</b>                | <b>2</b>                 | <b>2</b>                            |                         |                          |
| Тема 6. Представления графов. Обход графов в ширину (BFS) и глубину (DFS), оценка сложности.                                 | <b>8</b>                | <b>4</b>                 | <b>4</b>                            |                         |                          |
| Тема 7. Топологическая сортировка, поиск сильно связанных компонентов, оценка сложности.                                     | <b>8</b>                | <b>4</b>                 | <b>4</b>                            |                         |                          |

|                                                                               |           |           |           |  |  |
|-------------------------------------------------------------------------------|-----------|-----------|-----------|--|--|
| Тема 8. Деревья (свойства, расчет диаметра дерева, поиск центра).             | 6         | 3         | 3         |  |  |
| Тема 9. Минимальное остовное дерево (алгоритм Краскала и/или алгоритм Прима). | 6         | 3         | 3         |  |  |
| Тема 10. Жадные алгоритмы.                                                    | 6         | 3         | 3         |  |  |
| Тема 11. NP полнота.                                                          | 6         | 3         | 3         |  |  |
| <b>ИТОГО</b>                                                                  | <b>64</b> | <b>32</b> | <b>32</b> |  |  |

### 2.3.2. Краткое содержание разделов дисциплины в виде тематического плана

#### **Тема 1. Корректность алгоритмов, анализ алгоритмов (worst-case, average-case, best-case), асимптотический анализ алгоритмов.**

Корректность алгоритмов определяется их способностью решать поставленную задачу для всех допустимых входных данных и доказывается через инварианты циклов, предусловия и постусловия. Анализ алгоритмов включает оценку производительности в трех сценариях: worst-case (наихудший случай) показывает максимальное время выполнения, average-case (средний случай) учитывает вероятностное распределение входных данных, а best-case (наилучший случай) демонстрирует минимальное время работы. Асимптотический анализ использует нотации  $O$ ,  $\Theta$  и  $\Omega$  для описания поведения алгоритмов при росте размера входных данных, позволяя сравнивать эффективность различных подходов независимо от конкретной реализации и аппаратной платформы, что является основой для выбора оптимальных алгоритмических решений..

## **Тема 2. Алгоритмы сортировки (bubble, insertion, radix, bucket), нижняя оценка для сортировки сравнениями.**

Алгоритмы сортировки делятся на основанные на сравнениях и не основанные на сравнениях, каждый с различными характеристиками сложности. Сортировка пузырьком (bubble sort) и сортировка вставками (insertion sort) имеют сложность  $O(n^2)$  в худшем случае, но insertion sort эффективна для небольших или частично отсортированных массивов с лучшим случаем  $O(n)$ . Поразрядная сортировка (radix sort) и блочная сортировка (bucket sort) не используют сравнения элементов: radix sort работает за  $O(d \cdot n)$  где  $d$  — количество разрядов, а bucket sort достигает  $O(n)$  в среднем случае при равномерном распределении данных, но требует  $O(n^2)$  в худшем случае. Теоретическая нижняя граница для любого алгоритма сортировки, основанного на сравнениях, составляет  $\Omega(n \log n)$ , что доказывается через анализ дерева решений: для сортировки  $n$  элементов существует  $n!$  различных перестановок, и каждое сравнение может дать максимум два исхода, поэтому минимальная высота дерева решений равна  $\log_2(n!) \approx n \log n$ , что делает алгоритмы типа merge sort и heap sort асимптотически оптимальными среди сортировок сравнениями.

## **Тема 3. Парадигма разделяй и властвуй, сортировка слиянием, бинарный поиск, умножения матриц, алгоритм Штрассена)**

Парадигма "разделяй и властвуй" рекурсивно разбивает задачу на подзадачи, решает их независимо и объединяет результаты. Сортировка слиянием делит массив пополам до одноэлементных частей, затем сливает их в отсортированном порядке за  $O(n \log n)$ . Бинарный поиск отбрасывает половину элементов на каждом шаге, сравнивая искомое значение со средним элементом, что дает  $O(\log n)$ . Стандартное умножение матриц требует  $O(n^3)$  операций, но алгоритм Штрассена использует семь рекурсивных умножений вместо восьми, снижая сложность до  $O(n^{2.81})$ . Эффективность таких алгоритмов анализируется через рекуррентные соотношения вида  $T(n) = aT(n/b) + f(n)$ .

## **Тема 4. Динамическое программирование, числа Фибоначчи, поиск наибольшей общей подпоследовательности.**

Динамическое программирование — это метод решения задач путем разбиения их на перекрывающиеся подзадачи с сохранением результатов для избежания повторных вычислений. Классический пример — числа Фибоначчи, где наивная рекурсия  $F(n) = F(n-1) + F(n-2)$  имеет экспоненциальную сложность  $O(2^n)$  из-за многократного пересчета одних

значений, но с мемоизацией или итеративным подходом сложность снижается до  $O(n)$ . Задача поиска наибольшей общей подпоследовательности (LCS) для двух строк длиной  $m$  и  $n$  решается через двумерную таблицу, где  $dp[i][j]$  представляет длину LCS для первых  $i$  символов первой строки и первых  $j$  символов второй строки, с рекуррентным соотношением: если символы совпадают, то  $dp[i][j] = dp[i-1][j-1] + 1$ , иначе  $dp[i][j] = \max(dp[i-1][j], dp[i][j-1])$ . Алгоритм LCS работает за  $O(m \times n)$  времени и пространства, демонстрируя типичный подход динамического программирования: оптимальное решение строится из оптимальных решений подзадач с использованием принципа оптимальности Беллмана.

### **Тема 5. Самая длинная возрастающая подпоследовательность, задача о рюкзаке.**

Задача о самой длинной возрастающей подпоследовательности (LIS) решается динамическим программированием за  $O(n^2)$ , где  $dp[i]$  хранит длину LIS, заканчивающейся в позиции  $i$ , с рекуррентным соотношением  $dp[i] = \max(dp[j] + 1)$  для всех  $j < i$  где  $arr[j] < arr[i]$ , но существует оптимизированное решение за  $O(n \log n)$  с использованием бинарного поиска. Задача о рюкзаке (knapsack problem) имеет две основные версии: в задаче 0/1 каждый предмет можно взять только один раз, решение строится через двумерную таблицу  $dp[i][w]$ , где  $dp[i][w]$  — максимальная стоимость для первых  $i$  предметов при весе рюкзака  $w$ , с рекуррентным соотношением  $dp[i][w] = \max(dp[i-1][w], dp[i-1][w - \text{weight}[i]] + \text{value}[i])$ . Неограниченная задача о рюкзаке позволяет брать предметы многократно, что упрощает рекуррентное соотношение до  $dp[w] = \max(dp[w], dp[w - \text{weight}[i]] + \text{value}[i])$  для каждого предмета  $i$ . Обе задачи имеют временную сложность  $O(n \times W)$  где  $n$  — количество предметов,  $W$  — вместимость рюкзака, и демонстрируют классический подход динамического программирования для задач комбинаторной оптимизации.

### **Тема 6. Представления графов. Обход графов в ширину (BFS) и глубину (DFS), оценка сложности.**

Графы представляются двумя основными способами: матрица смежности размером  $V \times V$  занимает  $O(V^2)$  памяти и позволяет проверить наличие ребра за  $O(1)$ , но неэффективна для разреженных графов, в то время как список смежности требует  $O(V+E)$  памяти и  $O(\text{degree}(v))$  времени для проверки ребра, где  $\text{degree}(v)$  — степень вершины. Обход в глубину (DFS) использует стек (явно или через рекурсию) для посещения вершин, углубляясь в граф до тупика перед возвратом, что позволяет обнаруживать циклы, выполнять топологическую сортировку и находить компоненты связности. Обход в ширину (BFS) применяет очередь для

посещения всех соседей текущей вершины перед переходом к следующему уровню, что гарантирует нахождение кратчайшего пути в невзвешенных графах и позволяет определять связность графа. Временная сложность обоих алгоритмов составляет  $O(V+E)$  для списка смежности и  $O(V^2)$  для матрицы смежности, где  $V$  — количество вершин,  $E$  — количество ребер, а пространственная сложность равна  $O(V)$  для хранения информации о посещенных вершинах и структур данных обхода.

## **Тема 7. Топологическая сортировка, поиск сильно связанных компонентов, оценка сложности.**

Топологическая сортировка применяется к направленным ациклическим графам (DAG) для упорядочивания вершин таким образом, чтобы для каждого ребра  $(u,v)$  вершина  $u$  предшествовала вершине  $v$  в итоговой последовательности. Алгоритм Кана использует очередь и массив входящих степеней: на каждом шаге удаляется вершина с нулевой входящей степенью, а степени ее соседей уменьшаются на единицу, что дает сложность  $O(V+E)$ . Альтернативный подход через DFS выполняет обход в глубину и добавляет вершины в стек при завершении обработки всех их потомков, после чего элементы стека извлекаются в обратном порядке. Поиск сильно связанных компонентов (SCC) решается алгоритмом Косарайо за  $O(V+E)$ : выполняется DFS на исходном графе с записью времени завершения обработки вершин, затем DFS на транспонированном графе в порядке убывания времени завершения, где каждый обход дает одну сильно связанную компоненту. Алгоритм Тарьяна находит SCC за один проход DFS с использованием стека и двух массивов (время обнаружения и наименьшее достижимое время), также работая за  $O(V+E)$  времени и  $O(V)$  памяти.

## **Тема 8. Деревья (свойства, расчет диаметра дерева, поиск центра).**

Дерево — это связный ациклический граф с  $V$  вершинами и  $V-1$  ребрами, где между любыми двумя вершинами существует единственный простой путь. Основные свойства деревьев включают отсутствие циклов, связность, минимальность (удаление любого ребра делает граф несвязным) и максимальность (добавление любого ребра создает цикл). Диаметр дерева — это максимальное расстояние между любыми двумя вершинами, которое находится за  $O(V)$  с помощью двух BFS: первый BFS из произвольной вершины находит наиболее удаленную от нее вершину  $u$ , второй BFS от  $u$  находит наиболее удаленную вершину  $v$ , и расстояние между  $u$  и  $v$  является диаметром. Центр дерева — это вершина (или две

вершины), минимизирующая максимальное расстояние до всех остальных вершин, и находится путем итеративного удаления листьев: на каждом шаге удаляются все листья и их ребра, процесс продолжается до тех пор, пока не останется одна или две вершины, которые и являются центром. Все алгоритмы для деревьев имеют линейную сложность  $O(V)$  благодаря их простой структуре без циклов.

## **Тема 9. Минимальное остовное дерево (алгоритм Краскала и/или алгоритм Прима).**

Минимальное остовное дерево (MST) для взвешенного связного графа — это поддерево, которое соединяет все вершины с минимальной суммой весов ребер, содержащее  $V-1$  ребро для  $V$  вершин. Алгоритм Краскала использует жадный подход: сортирует все ребра по возрастанию веса и последовательно добавляет их в MST, если они не создают цикл, что проверяется через структуру данных Union-Find (система непересекающихся множеств). Временная сложность алгоритма Краскала составляет  $O(E \log E)$  из-за сортировки ребер, где  $E$  — количество ребер, а операции Union-Find выполняются практически за константное время с использованием эвристик сжатия путей и объединения по рангу. Алгоритм Прима строит MST инкрементально, начиная с произвольной вершины и на каждом шаге добавляя минимальное ребро, соединяющее уже включенные вершины с еще не включенными, что реализуется через приоритетную очередь с временной сложностью  $O(E \log V)$  при использовании двоичной кучи или  $O(E + V \log V)$  с фибоначчией кучей. Оба алгоритма гарантированно находят оптимальное решение благодаря свойствам MST: теореме о разрезе (минимальное ребро, пересекающее любой разрез, принадлежит некоторому MST) и теореме о цикле (максимальное ребро в любом цикле не принадлежит MST).

## **Тема 10. Жадные алгоритмы.**

Жадные алгоритмы принимают локально оптимальные решения на каждом шаге, надеясь получить глобально оптимальное решение, что работает только для задач с определенными математическими свойствами. Классические примеры включают алгоритмы для минимальных остовных деревьев (Краскал и Прим), где выбор минимального ребра на каждом шаге гарантирует оптимальность благодаря матроидным свойствам графа. Задача о планировании занятий решается жадным выбором заданий с наименьшим временем окончания, алгоритм Хаффмана строит оптимальное префиксное кодирование путем последовательного объединения символов с наименьшими частотами, а задача о сдаче решается выбором монет наибольшего номинала. Корректность жадных алгоритмов

доказывается через свойство жадного выбора (локально оптимальное решение входит в глобально оптимальное) и оптимальную подструктуру (оптимальное решение содержит оптимальные решения подзадач).

### **Тема 11. NP полнота**

NP-полнота — это класс вычислительных задач, которые находятся в пересечении двух множеств: NP (задачи, для которых решение можно проверить за полиномиальное время) и NP-трудные (задачи, к которым любая NP-задача сводится за полиномиальное время). Задача является NP-полной, если она принадлежит NP и любая другая NP-задача может быть полиномиально сведена к ней, что делает NP-полные задачи "самыми трудными" в классе NP. Классические примеры включают задачу выполнимости булевых формул (SAT), задачу о клике, задачу коммивояжера, задачу о рюкзаке 0/1 и задачу о раскраске графа в  $k$  цветов. Центральный открытый вопрос теории сложности — равенство  $P = NP$  — спрашивает, можно ли все NP-задачи решить за полиномиальное время; большинство ученых считает, что  $P \neq NP$ , что означает отсутствие эффективных алгоритмов для NP-полных задач. Практические подходы к NP-полным задачам включают аппроксимационные алгоритмы (дающие приближенные решения с гарантированной точностью), эвристические методы, экспоненциальные алгоритмы для небольших входных данных и сведение к частным случаям с полиномиальными алгоритмами.

**Литература:** Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ = Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to Algorithms. Second edition / пер. с англ. канд. техн. наук И. В. Красикова, Н. А. Ореховой, В. Н. Романова под ред. канд. техн. наук И. В. Красикова. — 2-е изд. — М.: Издательский дом «Вильямс», 2011. — 1290 с., ил. — 1000 экз. — ISBN 978-5-8459-0857-5 (рус.). — ISBN 0-07-013151-1 (англ.).

### **2.3.3. Краткое содержание семинарских/практических занятий/лабораторного практикума**

Семинарские и практические занятия включают программную реализацию и анализ основных алгоритмических парадигм для закрепления теоретических знаний. Студенты изучают алгоритмы сортировки (bubble sort, insertion sort, merge sort, quick sort) с измерением их производительности на различных наборах данных, реализуют бинарный поиск и анализируют его логарифмическую сложность. Практические задания охватывают

динамическое программирование через классические задачи (числа Фибоначчи, LCS, LIS, задача о рюкзаке), алгоритмы на графах включая BFS, DFS, топологическую сортировку и поиск минимальных остовных деревьев методами Краскала и Прима. Лабораторные работы включают реализацию жадных алгоритмов (планирование занятий, алгоритм Хаффмана), изучение алгоритмов "разделяй и властвуй" с анализом рекуррентных соотношений, а также знакомство с NP-полными задачами через аппроксимационные алгоритмы. Все задания сопровождаются экспериментальным анализом временной сложности, сравнением теоретических оценок с практическими измерениями и обсуждением применимости различных алгоритмических подходов к реальным задачам, что развивает навыки выбора оптимальных алгоритмов для конкретных сценариев.

#### **2.3.4. Материально-техническое обеспечение дисциплины**

«Материально-техническое обеспечение дисциплины» включает компьютерные классы с современными персональными компьютерами, оснащенными процессорами не менее 2 ГГц, оперативной памятью от 4 ГБ и достаточным дисковым пространством для установки IDE и компиляторов. Программное обеспечение включает среды разработки (Visual Studio или аналогичные), компиляторы C++ (GCC, Clang, MSVC), отладчики и профилировщики для анализа производительности алгоритмов. Аудитории оборудованы проекторами и интерактивными досками для демонстрации алгоритмов и структур данных. Лабораторные занятия требуют доступа к сети Интернет для работы с онлайн-ресурсами. Дополнительно обеспечивается доступ к электронным учебникам, базам данных с алгоритмическими задачами и специализированным программам для визуализации структур данных, что способствует лучшему пониманию материала и развитию практических навыков программирования.

#### **2.4. Модульная структура дисциплины с распределением весов по формам контролей**

| Формы контролей                                                                                  | Вес формы (форм) текущего контроля в результирующей оценке текущего контроля (по модулям) |     | Вес формы промежуточного контроля в итоговой оценке промежуточного контроля |    | Вес итоговой оценки промежуточного контроля в результирующей оценке промежуточных контролей |    | Вес итоговой оценки промежуточного контроля в результирующей оценке промежуточных контролей (семестровой оценке) |  | Веса результирующей оценки промежуточных контролей и оценки итогового контроля в результирующей оценке итогового контроля |  |
|--------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|-----|-----------------------------------------------------------------------------|----|---------------------------------------------------------------------------------------------|----|------------------------------------------------------------------------------------------------------------------|--|---------------------------------------------------------------------------------------------------------------------------|--|
|                                                                                                  | М1 <sup>1</sup>                                                                           | М2  | М1                                                                          | М2 | М1                                                                                          | М2 |                                                                                                                  |  |                                                                                                                           |  |
| Вид учебной работы/контроля                                                                      |                                                                                           |     |                                                                             |    |                                                                                             |    |                                                                                                                  |  |                                                                                                                           |  |
| Контрольная работа (при наличии)                                                                 | 0.5                                                                                       | 0.5 |                                                                             |    |                                                                                             |    |                                                                                                                  |  |                                                                                                                           |  |
| Устный опрос (при наличии)                                                                       |                                                                                           |     |                                                                             |    |                                                                                             |    |                                                                                                                  |  |                                                                                                                           |  |
| Тест (при наличии)                                                                               |                                                                                           |     |                                                                             |    |                                                                                             |    |                                                                                                                  |  |                                                                                                                           |  |
| Лабораторные работы (при наличии)                                                                |                                                                                           |     |                                                                             |    |                                                                                             |    |                                                                                                                  |  |                                                                                                                           |  |
| Письменные домашние задания (при наличии)                                                        |                                                                                           |     |                                                                             |    |                                                                                             |    |                                                                                                                  |  |                                                                                                                           |  |
| Реферат (при наличии)                                                                            |                                                                                           |     |                                                                             |    |                                                                                             |    |                                                                                                                  |  |                                                                                                                           |  |
| Эссе (при наличии)                                                                               |                                                                                           |     |                                                                             |    |                                                                                             |    |                                                                                                                  |  |                                                                                                                           |  |
| Проект (при наличии)                                                                             |                                                                                           |     |                                                                             |    |                                                                                             |    |                                                                                                                  |  |                                                                                                                           |  |
| Другие формы (при наличии)                                                                       |                                                                                           |     |                                                                             |    |                                                                                             |    |                                                                                                                  |  |                                                                                                                           |  |
| Веса результирующих оценок текущих контролей в итоговых оценках промежуточных контролей          |                                                                                           |     |                                                                             |    | 0.3                                                                                         | 4  |                                                                                                                  |  |                                                                                                                           |  |
| Веса оценок промежуточных контролей в итоговых оценках промежуточных контролей                   |                                                                                           |     |                                                                             |    | 0.7                                                                                         |    |                                                                                                                  |  |                                                                                                                           |  |
| Вес итоговой оценки 1-го промежуточного контроля в результирующей оценке промежуточных контролей |                                                                                           |     |                                                                             |    |                                                                                             |    | 0.5                                                                                                              |  |                                                                                                                           |  |
| Вес итоговой оценки 2-го промежуточного контроля в результирующей оценке                         |                                                                                           |     |                                                                             |    |                                                                                             |    | 0.5                                                                                                              |  |                                                                                                                           |  |

<sup>1</sup> Учебный Модуль

|                                                                                              |              |              |              |              |              |              |              |              |
|----------------------------------------------------------------------------------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| промежуточных контролей                                                                      |              |              |              |              |              |              |              |              |
| Вес результирующей оценки промежуточных контролей в результирующей оценке итогового контроля |              |              |              |              |              |              |              | 0.4          |
| <b>Вес итогового контроля (Экзамен/зачет)</b> в результирующей оценке итогового контроля     |              |              |              |              |              |              |              | 0.6          |
|                                                                                              | $\Sigma = 1$ | $\Sigma = 1$ | $\Sigma = 1$ | $\Sigma = 1$ | $\Sigma = 1$ | $\Sigma = 1$ | $\Sigma = 1$ | $\Sigma = 1$ |

### 3. Теоретический блок

#### 3.1. Материалы по теоретической части курса

##### 3.1.1. Учебник(и);

Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ = Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to Algorithms. Second edition / пер. с англ. канд. техн. наук И. В. Красикова, Н. А. Ореховой, В. Н. Романова под ред. канд. техн. наук И. В. Красикова. — 2-е изд. — М.: Издательский дом «Вильямс», 2011. — 1290 с., ил. — 1000 экз. — ISBN 978-5-8459-0857-5 (рус.). — ISBN 0-07-013151-1 (англ.).

##### 3.1.2. Учебное(ые) пособие(я);

##### 3.1.3. Курс лекций;

##### 3.1.4. Краткие конспекты лекций;

##### 3.1.5. Электронные материалы (электронные учебники, учебные пособия, курсы и краткие конспекты лекций, презентации РРТ и т.п.);

##### 3.1.6. Глоссарий/терминологический словарь;

##### 3.1.7. др. варианты материалов, необходимых для освоения учебной программы дисциплины.

### 4. Фонды оценочных средств

#### 4.1. Планы практических и семинарских занятий

1. Корректность и анализ алгоритмов: worst-, average-, best-case; асимптотики.

2. Классические алгоритмы сортировки: bubble, insertion, radix, bucket.
3. Нижняя оценка сложности сортировки сравнениями.
4. Парадигма «разделяй и властвуй»: сортировка слиянием, бинарный поиск, умножение матриц.
5. Алгоритм Штрассена.
6. Динамическое программирование: числа Фибоначчи, LCS.
7. ДП: самая длинная возрастающая подпоследовательность (LIS), задача о рюкзаке.
8. Представления графов: списки, матрицы, хэш-таблицы.
9. Алгоритмы BFS и DFS, оценка сложности.
10. Топологическая сортировка и компоненты сильной связности.
11. Деревья: свойства, нахождение центра, вычисление диаметра.
12. Алгоритмы построения MST: Краскал, Прим.
13. Жадные алгоритмы: разбор и реализация.
14. Введение в классы P и NP, понятие NP-полноты.

#### **4.2. Планы лабораторных работ и практикумов**

1. Сравнение временной сложности сортировок на реальных данных.
2. Реализация алгоритмов сортировки с логированием шагов.
3. Реализация алгоритма Штрассена.
4. Построение LCS, LIS и рюкзак — табличные и рекурсивные реализации.
5. Визуализация работы BFS и DFS.
6. Построение графа по списку рёбер и выполнение топологической сортировки.
7. Поиск компоненты сильной связности с помощью алгоритма Косарайю.
8. Вычисление диаметра дерева и его центра.
9. Реализация алгоритма Краскала и Прима.
10. Примеры и задачи на жадные алгоритмы.
11. Анализ задач на NP-полноту и редукции (на примере SAT, рюкзака, раскраски графа).

#### **4.3. Материалы по практической части курса**

##### **4.3.1. Учебно-методические пособия**

- Кормен Т. и др. *Алгоритмы: построение и анализ*
- Седжвик Р. *Алгоритмы на C++*
- Тим Рафгарден (Stanford CS161)

##### **4.3.2. Учебные справочники**

- [cp-algorithms.com](http://cp-algorithms.com)
- GeeksForGeeks
- VisuAlgo (визуализация алгоритмов)

#### **4.3.3. Задачники (практикумы)**

- Timus Online Judge
- LeetCode: раздел Dynamic Programming, Graphs
- AtCoder Educational DP Contest

#### **4.3.4. Наглядно-иллюстративные материалы**

- Анимации сортировок и обходов графов
- Видео объяснения алгоритмов (YouTube: William Fiset, Computerphile)
- Схемы динамических таблиц

#### **4.3.5. Другие виды материалов**

- GitHub с шаблонами реализаций и задач
- Онлайн-интерпретаторы для визуализации работы алгоритмов
- Коллекции задач по темам

#### **4.4. Вопросы и задания для самостоятельной работы студентов**

- Реализовать и сравнить сортировки: bubble, insertion, merge.
- Выполнить пошаговую симуляцию LCS и LIS.
- Построить дерево и найти его центр.
- Составить граф зависимостей и выполнить топологическую сортировку.
- Разработать реализацию рюкзака с memoization.

#### **4.6. Образцы контрольных и промежуточных заданий**

##### **Фрагмент контрольной:**

1. Приведите пример задачи, решаемой жадным методом, и обоснуйте корректность.
2. Реализуйте поиск центра дерева.
3. Выполните топологическую сортировку графа.

#### **4.7. Перечень экзаменационных вопросов**

1. Понятие корректности алгоритма и асимптотическая сложность
2. Разновидности сортировок и их эффективность
3. Парадигма «разделяй и властвуй» и её применение
4. Основы динамического программирования
5. LIS, LCS и задача рюкзака
6. Обходы графов и построение графов

7. Алгоритмы топологической сортировки и компонент связности
8. Свойства деревьев: диаметр, центр
9. Минимальное остовное дерево: Краскал и Прим
10. Жадные алгоритмы и доказательства оптимальности
11. Классы P и NP, NP-полные задачи

#### 4.8. Образцы экзаменационных билетов

##### Билет №4

1. Объясните понятие NP-полноты и приведите пример задачи.
2. Реализуйте алгоритм Краскала.
3. Найдите самую длинную возрастающую подпоследовательность заданной последовательности.

#### 4.9. Образцы экзаменационных практических заданий

- Реализация merge sort с логированием шагов
- Построение дерева и нахождение его центра и диаметра
- Построение остовного дерева с визуализацией выбора рёбер
- Задача рюкзака: реализация и оптимизация по памяти

#### 4.10. Банк тестовых заданий для самоконтроля

##### Примеры:

1. Что обозначает  $O(n \log n)$ ?
  - ☐ Линейная сложность
  - ☒ Лог-линейная сложность
  - ☐ Квадратичная сложность
2. Какой из алгоритмов работает быстрее в среднем случае?
  - ☐ bubble sort
  - ☒ merge sort
  - ☐ insertion sort
3. Какой из алгоритмов обеспечивает корректную топологическую сортировку?
  - ☐ BFS
  - ☒ DFS с постобходом
  - ☐ Prim

#### 4.11. Методики решения и ответы к тестам

- **Задача:** Найти компоненту сильной связности  
**Методика:** Выполнить два прохода DFS (алгоритм Косарайю)
- **Вопрос:** Что делает алгоритм Прима?  
**Ответ:** Находит MST, начиная с одной вершины, расширяя дерево ближайшим ребром
- **Задача:** Реализовать LCS  
**Ответ:** Использовать двумерную таблицу, где  $dp[i][j]$  — длина LCS первых  $i$  и  $j$  символов

### 5. Методический блок

#### 5.1. Методика преподавания

5.1.1. Методика преподавания дисциплины строится на принципе постепенного усложнения материала от базовых алгоритмов к сложным парадигмам, сочетая теоретические обоснования с практической реализацией. Студентам рекомендуется готовиться к семинарским занятиям через предварительное изучение псевдокода алгоритмов, выполнение трассировки на небольших примерах вручную и анализ временной сложности на бумаге. Самостоятельная работа организуется через решение задач на алгоритмических платформах (LeetCode, Codeforces), чтение классической литературы по алгоритмам (Кормен, Седжвик), создание собственных тестовых наборов и ведение журнала с анализом сложности решенных задач. Особое внимание уделяется развитию алгоритмического мышления через декомпозицию сложных задач на подзадачи, выбор подходящих парадигм (жадные алгоритмы, динамическое программирование, разделяй и властвуй) и обоснование корректности решений через инварианты и доказательства, что формирует фундаментальные навыки для решения практических вычислительных задач.