Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Институт прикладной математики и компьютерных наук

УТВЕРЖДЕНО:
Директор
А. В. Замятин

Оценочные материалы по дисциплине

Алгоритмы и структуры данных

по направлению подготовки

02.03.02 Фундаментальная информатика и информационные технологии

Направленность (профиль) подготовки:
Искусственный интеллект и разработка программных продуктов

Форма обучения
Очная

Квалификация
Бакалавр

Год приема
2025

СОГЛАСОВАНО:
Руководитель ОП
А.В. Замятин

Председатель УМК
С.П. Сущенко

Томск – 2025

1. Компетенции и индикаторы их достижения, проверяемые данными оценочными материалами

Целью освоения дисциплины является формирование следующих компетенций:

ОПК-3. Способен к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям.

ОПК-6. Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности.

Результатами освоения дисциплины являются следующие индикаторы достижения компетенций:

ИОПК-3.1. Использует методы построения и анализа алгоритмов при проектировании и разработке программных систем.

ИОПК-3.3. Разрабатывает алгоритмы и программы при решении задач профессиональной деятельности.

ИОПК-6.1. Обладает необходимыми знаниями в области информационных технологий, в том числе понимает принципы их работы.

ИОПК-6.2. Применяет знания, полученные в области информационных технологий, при решении задач профессиональной деятельности.

ИОПК-6.3. Использует современные информационные технологии на всех этапах разработки программных систем.

2. Оценочные материалы текущего контроля и критерии оценивания

Элементы текущего контроля:

– лабораторные работы;

– контрольная работа (первая часть экзамена в форме коллоквиума) в середине семестра.

При сдаче каждой лабораторной работы и ответах на вопросы на коллоквиуме проверяются знания и умения по индикаторам всех компетенций дисциплины.

Каждая лабораторная работа заключается в разработке программы по заданной теме. Примерные темы лабораторных работ:

- эффективные алгоритмы сортировки
- сортировка объектов
- хеш-таблицы
- задачи дискретной оптимизации
- задачи на графах.

Критерии оценивания лабораторных работ

Работы оцениваются по 5 основным параметрам:

1. Полнота и правильность решения (сделано все, что и как нужно, тесты проходят)
2. Способность внести простые изменения в программу
3. Знание области применения и основной идеи используемых алгоритмов
4. Знание назначения и принципов организации используемых структур данных
5. Ориентация в программе, основных переменных/атрибутах, функциях/методах.

Работы оцениваются в баллах – 10 или 15 баллов за каждую (максимально 60 баллов за 5 работ).

Для получения итоговой оценки по курсу нужно сдать не менее четырех заданий.

3. Оценочные материалы итогового контроля (промежуточной аттестации) и критерии оценивания

Экзамен в четвертом семестре проводится в 2 этапа – коллоквиумы в середине и в конце семестра.

Темы первого коллоквиума: алгоритмы сортировки и поиска, хеш-таблицы, информационные деревья.

Темы второго коллоквиума: задачи дискретной оптимизации, задачи на графах, алгоритмы поиска подстроки.

На коллоквиумах студент получает по 5 вопросов и отвечает устно. За каждый ответ можно получить от 0 до 4 баллов (максимально 20 баллов за коллоквиум).
Для получения итоговой оценки по курсу нужно набрать не менее пяти баллов на каждом коллоквиуме.

При ответах на вопросы на экзамене проверяются знания и умения по индикаторам всех компетенций дисциплины.

При формировании итоговой оценки суммируются баллы за лабораторные задания и баллы за коллоквиумы. Итоговая оценка:

80 – 100 баллов - отлично

60 – 79 баллов – хорошо

40 – 59 баллов – удовлетворительно

меньше 40 баллов – неудовлетворительно.

4. Оценочные материалы для проверки остаточных знаний (сформированности компетенций)

Список вопросов для коллоквиумов (около 100 вопросов) можно использовать и при проверке остаточных знаний.

Некоторые вопросы требуют простого ответа (например, «Что такое глубина рекурсии и как избежать бесконечной рекурсии»).

Вопросы по алгоритмам и структурам данных более сложные, при ответе на них необходимо объяснить:

- что дано и что требуется получить
- идею и особенности реализации алгоритма или структуры (кратко)
- трудоемкость (если она оценивалась).

Список вопросов для оценки остаточных знаний

1. Что такое трудоемкость алгоритма, в чем она измеряется, что такое элементарный шаг
2. Что такое асимптотическая трудоемкость, почему она используется на практике
3. Какие типы трудоемкостей вы знаете. Приведите пример алгоритма
4. Когда важно оценивать трудоемкость в среднем. Приведите пример алгоритма
5. Какой алгоритм можно считать эффективным
6. Что такое рекурсивный алгоритм, как он может быть реализован. Приведите примеры рекурсивных алгоритмов
7. Что такое глубина рекурсии, как избежать бесконечной рекурсии
8. Затраты памяти, которые нужно учитывать при разработке рекурсивной функции
9. Идея и трудоемкость дихотомического (бинарного) поиска

10. Когда простой исчерпывающий поиск в массиве использовать выгоднее, чем бинарный
11. Косвенная упорядоченность в массиве, как ее реализовать
12. Как можно упорядочить объекты, которые не являются элементами одного массива
13. Минимальная гарантированная трудоемкость в наихудшем для задач поиска и сортировки со сравнениями

14. Идея и трудоемкость рекурсивной сортировки слиянием (с пояснением)
15. Идея и трудоемкость рекуррентной сортировки слиянием (с пояснением)
16. Идея и трудоемкость сортировки Шелла
17. Как выбирается шаг в цепочках в алгоритме Шелла
18. Как сортируются цепочки в алгоритме Шелла
19. Что такое бинарная куча (пирамида)
20. Для чего в пирамидальной сортировке строится пирамида
21. Что такое просеивание элемента в пирамиде. Трудоемкость просеивания
22. Как строится пирамида. Трудоемкость построения
23. Идея и трудоемкость пирамидальной сортировки
24. Идея быстрой сортировки
25. Трудоемкость быстрой сортировки (все варианты, когда они реализуются)
26. Основной недостаток алгоритма быстрой сортировки с двумя рекурсивными вызовами
27. Как работает алгоритм быстрой сортировки с одним рекурсивным вызовом
28. Идея и трудоемкость алгоритма поиска k-го минимального элемента в массиве
29. Что такое цифровая сортировка, когда ее можно использовать
30. Как реализовать цифровую сортировку целых чисел

31. Идея хеширования
32. Что такое хеш-функция и хеш-таблица
33. Что такое коллизии при хешировании, причины их появления
34. Идея метода цепочек для борьбы с коллизиями
35. Как происходит добавление и удаление элементов при использовании метода цепочек
36. Как при использовании метода цепочек выбрать размер хеш-таблицы q, если известно, что в таблице будут размещаться до n объектов
37. Идея метода открытой адресации для борьбы с коллизиями
38. Какие методы вычисления шага в методе открытой адресации вы знаете
39. Как работает метод линейных проб при добавлении и поиске элементов
40. Что такое двойное хеширование в методе открытой адресации
41. Как работает метод двойного хеширования при добавлении и поиске элементов
42. Как работает метод квадратичных проб при добавлении и поиске элементов
43. Как удалить элемент из хеш-таблицы, организованной по методу открытой адресации
44. Трудоемкость поиска в хеш-таблице - наилучший и наихудший вариант
45. Когда и почему выгодно/невыгодно использовать хеш-таблицы

46. Что такое случайное бинарное дерево, правило его формирования
47. Трудоемкость поиска на случайном бинарном дереве – все варианты
48. Как производится добавление элемента к бинарному дереву

49. Как производится удаление элемента из бинарного дерева, если удаляемая вершина имеет два непустых поддерева
50. Что такое АВЛ-дерево. Худший вариант с точки зрения трудоемкости поиска
51. Как производится добавление элемента к АВЛ-дереву
52. Как производится удаление элемента из АВЛ-дерева
53. Что такое красно-черное дерево, правила его формирования
54. Преимущества сбалансированного бинарного дерева по сравнению с идеальным и случайным
55. Когда и почему выгодно/невыгодно использовать бинарные деревья
56. Что такое B-дерево. Правила формирования B-дерева порядка k
57. Структура вершины B-дерева порядка k
58. Как происходит поиск элемента в B-дереве
59. Наихудший с точки зрения трудоемкости поиска и затрат памяти случай B-дерева порядка k
60. Когда и почему выгодно использовать B-деревья.

61. Поиск в глубину на графе
62. Поиск в ширину на графе
63. Алгоритм Прима
64. Алгоритм Краскала
65. Поиск эйлерова цикла/пути
66. Поиск гамильтонова цикла/пути

67. Алгоритм Дейкстры
68. Алгоритм Флойда-Уоршала
69. Выделение двусвязных компонент графа
70. Выделение сильно связных компонент графа
71. Задача о максимальном потоке в сети
72. Топологическая сортировка графа

73. Бэктрекинг в сочетании с методом ветвей и границ
74. Поиск минимальной раскраски графа
75. Алгоритмы раскраски, основанные на степенях вершин
76. Алгоритмы раскраски, основанные на склеивании вершин
77. Транзитивная ориентация графов
78. Раскраска транзитивно-ориентированных графов

79. Алгоритм Литтла поиска оптимального маршрута коммивояжера
80. Алгоритм ближайшего города
81. Маршрут коммивояжера на основе минимального остова
82. Алгоритм имитации отжига, применение для решения задачи коммивояжера
83. Муравьиный алгоритм, применение для решения задачи коммивояжера
84. Генетический алгоритм, применение для решения задачи коммивояжера

85. Алгоритм Бойера-Мура-Хопгуда
86. Поиск подстроки с помощью конечного автомата

87. Алгоритм Кнута-Морриса-Пратта
88. Алгоритм Рабина-Карпа
89. Алгоритм Ахо-Корасик, построение бора
90. Алгоритм Ахо-Корасик, выделение подстрок в тексте

Информация о разработчиках

Фукс Александр Львович, канд. техн. наук, доцент кафедры теоретических основ информатики ТГУ.