Государственное образовательное учреждение
«Приднестровский государственный университет им. Т.Г. Шевченко»

Физико-технический институт

Физико-математический факультет

Кафедра высшей и прикладной математики и информатики

УТВЕРЖДАЮ
Заведующий кафедрой высшей и
прикладной математики и информатики
_____ Коровай А.В.
(подпись) *(Ф.И.О.)*
протокол № 1 «14» сентября 2023 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине
Б1.О.26 «Алгоритмы и структуры данных»

Направление

01.03.04 «Прикладная математика»

Профиль

Математические и компьютерные методы для современных цифровых технологий

Квалификация

Бакалавр

ГОД НАБОРА 2022

Разработчик: ст. преподаватель
_____ Калинкова Е.В.
(подпись) *(Ф.И.О.)*
«14» сентября 2023 г.

г. Тирасполь 2023 г.

Паспорт фонда оценочных средств по учебной дисциплине

«Алгоритмы и структуры данных»

1. В результате изучения дисциплины «Алгоритмы и структуры данных» у обучающихся должны быть сформированы следующие компетенции:

2.

Категория (группа) компетенций	Код и наименование	Код и наименование индикатора достижения компетенции
Универсальные компетенции и индикаторы их достижения		
Учебным планом не предусмотрены		
Общепрофессиональные компетенции и индикаторы их достижения		
Теоретические и практические основы профессиональной деятельности	ОПК-4 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения.	ИД-1$_{ОПК-4}$ Знает методы алгоритмизации, языки и технологии программирования, пригодные для практического применения в области информационных технологий.
		ИД-2$_{ОПК-4}$ Умеет применять методы алгоритмизации, языки и технологии программирования для решения задач профессиональной деятельности.
		ИД-3$_{ОПК-4}$ Владеет навыками программирования, отладки и тестирования программных средств.
Обязательные профессиональные компетенции и индикаторы их достижения		
	ПК-1 Способен демонстрировать базовые знания математических и естественных наук, применять соответствующие процессу математические модели и проверять их адекватность, проводить анализ результатов моделирования, принимать решение на основе полученных результатов.	ИД-1$_{ПК-1}$ Знает основы фундаментальной и прикладной математики для формализации исследуемых процессов и явлений.
		ИД-2$_{ПК-1}$ Умеет самостоятельно разрабатывать математические модели на основе содержательного и физического описания процессов и объектов.
		ИД-3$_{ПК-1}$ Владеет способностью применить соответствующую процессу математическую модель и проверить ее адекватность, провести анализ результатов моделирования, принять решение на основе полученных результатов.
	ПК-2 Способен самостоятельно изучать новые разделы фундаментальных наук.	ИД-1$_{ПК-2}$ Знает основы фундаментальных наук и их задачи.
		ИД-2$_{ПК-2}$ Умеет использовать изученные разделы фундаментальных наук для решения конкретных научно-практических задач.
		ИД-3$_{ПК-2}$ Владеет навыками применения математических и естественных наук для решения конкретных научно-практических задач.
	ПК-4 Способен создавать и исследовать новые математические модели в естественных науках, промышленности и бизнесе с учетом возможностей современных информационных технологий и программирования.	ИД-1$_{ПК-4}$ Знает математические методы, основы программирования и современные компьютерные технологии.
		ИД-2$_{ПК-4}$ Умеет использовать математический аппарат, основы программирования и современные компьютерные технологии для решения практических задач
		ИД-3$_{ПК-4}$ Владеет навыками использования математического аппарата, основ программирования и современных компьютерных технологий для решения практических задач профессиональной деятельности.

	ПК-6 Способен разрабатывать алгоритмы, проектировать и разрабатывать компоненты программного обеспечения на основе современных парадигм, технологий и языков программирования.	ИД-1$_{ПК-6}$ Знает основные языки программирования и основы работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий.
		ИД-2$_{ПК-6}$ Умеет применять языки программирования, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ.
		ИД-3$_{ПК-6}$ Владеет навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач.
Рекомендуемые профессиональные компетенции и индикаторы их достижения		
Учебным планом не предусмотрены		

3. Программа оценивания контролируемой компетенции:

Текущая аттестация	Контролируемые модули, разделы (темы) дисциплины и их наименование	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
1	Раздел 1. Формализация понятия алгоритма. Анализ сложности алгоритмов.	ОПК-4, ПК-1, ПК-2, ПК-4, ПК-6	Комплект тестовых заданий № 1
2	Раздел 2. Алгоритмы поиска и сортировки.	ОПК-4, ПК-1, ПК-2, ПК-4, ПК-6	Комплект тестовых заданий № 2
3	Раздел 3. Линейные структуры данных. Нелинейные структуры данных.	ОПК-4, ПК-1, ПК-2, ПК-4, ПК-6	Комплект тестовых заданий № 3
4	Раздел 4. Построение эффективных алгоритмов.	ОПК-4, ПК-1, ПК-2, ПК-4, ПК-6	Комплект тестовых заданий № 4
5	Раздел 5. Алгоритмы на графах.	ОПК-4, ПК-1, ПК-2, ПК-4, ПК-6	Комплект тестовых заданий № 5
Промежуточная аттестация		Код контролируемой компетенции (или ее части)	Наименование оценочного средства
1		ОПК-4, ПК-1, ПК-2, ПК-4, ПК-6	Список вопросов к экзамену

Комплект тестовых заданий № 1

по дисциплине «Алгоритмы и структуры данных»

1. Алгоритм – это
   1) понятное и точное описание конечной последовательности команд, приводящей от исходных данных к искомому результату
   2) пошаговое описание процесса решения какой-либо задачи
   3) последовательность действий, применяемая к некоторым исходным данным

2. Свойство алгоритма записываться в виде упорядоченной совокупности отделенных друг от друга предписаний (директив) называется …
   1) понятность
   2) определенность
   3) дискретность
   4) массовость

3. Свойство алгоритма записываться в виде только тех команд, которые находятся в системе команд исполнителя, называется …
   1) понятность
   2) определенность
   3) дискретность
   4) результативность

4. Свойство алгоритма записываться только директивами однозначно и одинаково интерпретируемыми разными исполнителями, называется …
   1) массовость
   2) понятность
   3) результативность
   4) определенность

5. Свойство алгоритма, определяющее, что при точном исполнении всех предписаний процесс должен прекратиться за конечное число шагов с определенным ответом на поставленную задачу называется …
   1) понятность
   2) конечность
   3) детерминированность
   4) дискретность

6. Свойство алгоритма обеспечивать решение не одной задачи, а целого класса задач данного типа называется …
   1) понятность
   2) определенность
   3) массовость
   4) дискретность

7. Команда машины Поста имеет структуру nKm, где:
   1) n – действие, выполняемое головкой; K – номер следующей команды, подлежащей выполнению; m – порядковый номер команды
   2) n – порядковый номер команды; K – действие, выполняемое головкой; m – номер следующей команды, подлежащей выполнению
   3) n – порядковый номер команды; K – номер следующей команды, подлежащей выполнению; m – действие, выполняемое головкой
   4) n – порядковый номер команды; K – действие, выполняемое головкой; m – номер клетки, с которой данную команду надо произвести

8. В машине Поста останов будет результативным:
   1) при выполнении недопустимой команды
   2) если машина не останавливается никогда
   3) если результат выполнения программы такой, какой и ожидался
   4) по команде "Стоп"

9. Сколько существует команд у машины Поста?
   1) 2
   2) 4
   3) 6
   4) 8

10. В алгоритмах Маркова дана система подстановок в алфавите Л={a,b,c}: abc - c; ba - cb; ca - ab. Преобразуйте с помощью этой системы слово bacaabc
   1) cbc

2) ccbcbbc
3) cbacba
4) cbabc

11. Какова функция сложности программы, если каждый элемент входных данных требуется обработать лишь линейное число раз?
1) O(1)
2) O(N)
3) $O(N^2)$
4) O(Log(N))

12. Алгоритмы константной сложности характеризуются следующей функцией сложности:
1) O(1)
2) O(N)
3) $O(N^2)$
4) O(Log(N))

13. Какова сложность этого алгоритма:
```
int s = 0;
for(int i = 0; i < N; i++)
    s = s + i;
```
1) O(1)
2) O(N)
3) O(2*N)
4) O(N*Log(N))
5) $O(N^2)$

14. Какова сложность этого алгоритма:
```
int count = 0;
for (int i = 0; i < n; i++)
   for (int j = i; j > 0; j--)
      count++;
```
1) O(1)
2) O(N)
3) O(N*Log(N))
4) $O(N^2)$
5) $O(N^2/2)$

15. Какова сложность этого алгоритма:
```
int count = 0;
for (int i = n; i > 0; i /= 2)
   for (int j = 0; j < i; j++)
      count++;
```
1) O(1)
2) O(N)
3) O(N*Log(N))
4) $O(N^2)$
5) O(N*log(N*Log(N)))

16. Время работы алгоритма $T=n^2+25n+3$. Выберите все верные утверждения.
1. $T=O(n^2)$
2. $T=O(n)$
3. $T=\Omega(n^2)$
4. $T=\Omega(n)$

Критерии оценки:

За каждое правильно выполненное задание назначается 1 балл. В случае задания с выбором нескольких правильных ответов 1 балл дается за полностью выполненное задание, 0 баллов – за хотя бы один неверный ответ. Затем первичные баллы пропорционально переводятся в тестовые баллы согласно технологической карте дисциплины.

Комплект тестовых заданий № 2

по дисциплине «Алгоритмы и структуры данных»

1. Предположим, вы проводите последовательный поиск в списке [15, 8, 2, 19, 18, 0, 8, 14, 18, 14]. Сколько сравнений вам понадобится, чтобы найти ключ 18?
2. Предположим, вы проводите последовательный поиск в упорядоченном списке [3, 5, 6, 8, 11, 12, 14, 15, 17, 18]. Сколько сравнений вам потребуется, чтобы найти ключ 13?
3. Какова временная сложность бинарного поиска в отсортированном массиве из *n* элементов?
   1) O(n)
   2) O(log n)
   3) O(n log n)
   4) $O(n^2)$
4. Какой алгоритм поиска может быть эффективно применен для поиска в отсортированном массиве, если известно, что данные распределены равномерно?
   1) линейный поиск
   2) линейный поиск с барьером
   3) бинарный поиск
   4) интерполяционный поиск
5. Какой алгоритм сортировки обычно считается самым эффективным для случая общего набора данных?
   1) сортировка пузырьком
   2) сортировка выбором
   3) быстрая сортировка
   4) сортировка вставками
6. Какой алгоритм сортировки обычно считается самым неэффективным для случая общего набора данных?
   1) сортировка пузырьком
   2) сортировка слиянием
   3) сортировка выбором
   4) сортировка вставками
7. Что такое устойчивая сортировка?
   1) сортировка, которая не требует дополнительной памяти
   2) сортировка, которая сохраняет относительный порядок равных элементов
   3) сортировка, которая выполняется за линейное время
   4) сортировка, которая всегда возвращает отсортированный массив
8. Какой/какие из следующих алгоритмов сортировки имеют сложность $O(n^2)$ в среднем случае?
   1) пузырьковая сортировка
   2) сортировка вставками
   3) сортировка выбором
   4) быстрая сортировка
   5) сортировка слиянием
9. Какой/какие из следующих алгоритмов сортировки имеют сложность O(*n*) в лучшем случае?
   1) пузырьковая сортировка
   2) сортировка вставками
   3) сортировка выбором
   4) быстрая сортировка
   5) сортировка слиянием
10. Какова асимптотическая сложность времени (в худшем случае) быстрой сортировки?
    1) O(n log n)
    2) $O(n^2)$
    3) O(n)
    4) $O(2^n)$
11. Какова асимптотическая сложность времени (в худшем случае) алгоритма сортировки слиянием?
    1) O(n log n)
    2) $O(n^2)$
    3) O(n)
    4) $O(2^n)$
12. Как называется этот вид сортировки?

Фиксируется какой-либо ключ (базовый), относительно которого все элементы с большим весом перебрасываются вправо, а с меньшим — влево. При этом весь список элементов делится относительно базового ключа на две части. Для каждой части процесс повторяется.

1) Поразрядная
2) Шелла
3) Шейкерная
4) Хоара

13. Какие сортировки являются устойчивыми?
    1) пузырьковая сортировка
    2) сортировка вставками
    3) быстрая сортировка
    4) сортировка слиянием

14. В чем заключается основной принцип сортировки подсчетом?
    1) элементы сортируются путем последовательного сравнения друг с другом
    2) элементы сортируются на основе их относительного положения в массиве
    3) частота встречаемости каждого элемента учитывается для его правильного расположения
    4) элементы сортируются путем выбора случайного элемента в качестве опорного

15. Какова временная сложность поразрядной сортировки для массива из *n* элементов?
    1) O(n)
    2) O(n log n)
    3) $O(n^2)$
    4) O(kn), где k – количество разрядов чисел в массиве

16. Какой тип данных наиболее эффективно сортируется с использованием поразрядной сортировки?
    1) строки
    2) целые числа
    3) вещественные числа
    4) булевы значения

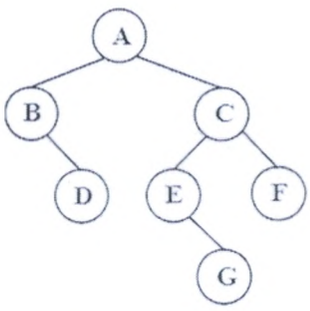
Критерии оценки:

За каждое правильно выполненное задание назначается 1 балл. В случае задания с выбором нескольких правильных ответов 1 балл дается за полностью выполненное задание, 0 баллов – за хотя бы один неверный ответ. Затем первичные баллы пропорционально переводятся в тестовые баллы согласно технологической карте дисциплины.

Комплект тестовых заданий № 3
по дисциплине «Алгоритмы и структуры данных»

1. Что такое бинарное дерево поиска?
   1) структура данных, в которой каждый узел имеет максимум два потомка
   2) структура данных, в которой каждый узел может иметь произвольное количество потомков
   3) структура данных, в которой каждый узел может иметь только одного потомка
   4) структура данных, в которой каждый узел может иметь три потомка

2. Укажите вид обхода дерева, представленного на рисунке, если порядок просмотра вершин следующий: B D A E G C F



   1) прямой
   2) обратный
   3) симметричный
   4) этот путь не является обходом

3. В двоичное дерево поиска были добавлены следующие элементы: 8, 3, 6, 10, 1, 7, 9, 14, 4, 13, 5. Сколько листьев у данного дерева?

4. Какова сложность операции поиска элемента в бинарном дереве поиска в среднем случае?
   1) O(1)
   2) O(log n)
   3) O(n)
   4) O(n log n)

5. Какова сложность операции поиска элемента в бинарном дереве поиска в худшем случае?
   1) O(1)
   2) O(log n)
   3) O(n)
   4) O(n log n)

6. Какова сложность операции поиска элемента в сбалансированном бинарном дереве поиска в худшем случае?
   1) O(1)
   2) O(log n)
   3) O(n)
   4) O(n log n)

7. Что такое хэш-таблица?
   1) структура данных, в которой элементы располагаются в виде дерева
   2) структура данных, основанная на очереди
   3) структура данных, использующая хэш-функцию для быстрого поиска элементов
   4) структура данных, использующая массив для хранения элементов

8. Какова сложность операции поиска элемента в хэш-таблице в среднем случае (в среднем по времени)?
   1) O(1)
   2) O(log n)
   3) O(n)
   4) O(n log n)

9. Как называется процесс преобразования входного значения в хэш-код?
   1) хэширование
   2) шифрование
   3) кодирование
   4) сжатие

10. Какая структура данных обычно используется для разрешения коллизий в хэш-таблице?
   1) двоичное дерево поиска

2) связанный список
3) очередь
4) стек

11. Какой метод разрешения коллизий позволяет уменьшить вероятность возникновения коллизий?
1) метод цепочек
2) линейное пробирование
3) двойное хеширование

12. Какова сложность операции вставки элемента в хэш-таблицу, если не учитывать возможные коллизии?
1) O(1)
2) O(log n)
3) O(n)
4) O(n log n)

13. Какое утверждение верно относительно размера хэш-таблицы и ее производительности?
1) больший размер таблицы всегда ведет к лучшей производительности
2) меньший размер таблицы всегда ведет к лучшей производительности
3) размер таблицы не влияет на производительность
4) оптимальный размер таблицы зависит от количества элементов и хэш-функции

14. Какая операция является основной для хэш-таблицы?
1) вставка элемента
2) поиск элемента
3) удаление элемента
4) обновление элемента

15. Какой из нижеперечисленных методов разрешения коллизий характеризуется наибольшим количеством затрат памяти?
1) линейное пробирование
2) квадратичное пробирование
3) метод цепочек
4) метод деления

Критерии оценки:

За каждое правильно выполненное задание назначается 1 балл. В случае задания с выбором нескольких правильных ответов 1 балл дается за полностью выполненное задание, 0 баллов – за хотя бы один неверный ответ. Затем первичные баллы пропорционально переводятся в тестовые баллы согласно технологической карте дисциплины.

Комплект тестовых заданий № 4

по дисциплине «Алгоритмы и структуры данных»

1. Что представляют собой жадные алгоритмы в области компьютерных наук?
   1) алгоритмы, которые всегда дают оптимальное решение для любой задачи;
   2) алгоритмы, которые выбирают локально оптимальное решение на каждом шаге с надеждой на глобальную оптимальность
   3) алгоритмы, которые используются только для сортировки данных
   4) алгоритмы, которые не могут быть применены к оптимизационным задачам

2. В чем заключается основное преимущество жадных алгоритмов?
   1) они всегда дают оптимальное решение для любой задачи
   2) они гарантированно работают для всех типов задач
   3) они могут решать задачи любой сложности
   4) они легко реализуемы и обычно работают достаточно быстро

3. Какие основные факторы делают жадные алгоритмы неприменимыми для некоторых задач?
   1) сложность алгоритма
   2) отсутствие глобальной оптимальности
   3) недостаточность ресурсов
   4) невозможность реализации решения задачи в виде жадного подхода

4. Что представляет собой метод динамического программирования в области компьютерных наук?
   1) метод, который использует случайный выбор для поиска оптимального решения
   2) метод, который разбивает задачу на более мелкие подзадачи и решает их последовательно
   3) метод, который всегда даёт точное оптимальное решение
   4) метод, который использует только жадные алгоритмы для решения задач

5. В чем состоит основное отличие между динамическим программированием и жадными алгоритмами?
   1) динамическое программирование всегда дает оптимальное решение, в то время как жадные алгоритмы выбирают локально оптимальное решение на каждом шаге
   2) динамическое программирование не использует рекурсию, в отличие от жадных алгоритмов
   3) динамическое программирование использует только жадные подходы к решению задач
   4) динамическое программирование не может быть применено к оптимизационным задачам

6. Какой основной недостаток динамического программирования?
   1) высокая вычислительная сложность
   2) ограничение на размер входных данных
   3) невозможность применения к задачам оптимизации
   4) необходимость хранения промежуточных результатов

7. Используя метод динамического программирования, решите следующую задачу: «На вершине лесенки, содержащей N ступенек, находится мячик, который начинает прыгать по ним вниз, к основанию. Мячик может прыгнуть на следующую ступеньку, на ступеньку через одну или через 2. (То есть, если мячик лежит на 8-ой ступеньке, то он может переместиться на 5-ую, 6-ую или 7-ую.) Определить число всевозможных "маршрутов" мячика с вершины на землю». Решить задачу при $N = 10$.

8. Используя метод динамического программирования, решите следующую задачу: «В левом верхнем углу прямоугольной таблицы размером $N \times M$ находится черепашка. В каждой клетке таблицы записано некоторое число. Черепашка может перемещаться вправо или вниз, при этом маршрут черепашки заканчивается в правом нижнем углу таблицы. Подсчитаем сумму чисел, записанных в клетках, через которую проползла черепашка (включая начальную и конечную клетку). Найдите наибольшее возможное значение этой суммы».

3	5	9	7	1	2
6	4	1	5	6	3
8	7	2	4	2	5
4	3	6	7	3	2
5	1	4	5	8	1

Критерии оценки:

За каждое правильно выполненное задание назначается 1 балл. В случае задания с выбором нескольких правильных ответов 1 балл дается за полностью выполненное задание, 0 баллов – за хотя бы один неверный ответ. Затем первичные баллы пропорционально переводятся в тестовые баллы согласно технологической карте дисциплины.

Комплект тестовых заданий № 5
по дисциплине «Алгоритмы и структуры данных»

1. Как называется граф, в котором нет циклов и все вершины связаны?
   1) дерево
   2) циклограф
   3) полный граф
   4) связный граф

2. Как называется вершина, к которой не ведут рёбра?
   1) конечная вершина
   2) начальная вершина
   3) изолированная вершина
   4) подвешенная вершина

3. Какое утверждение о неориентированных графах верно?
   1) вершины не имеют порядка
   2) рёбра не имеют направления
   3) рёбра имеют направление от одной вершины к другой
   4) вершины имеют порядок, в котором они были добавлены в граф

4. Как называется граф, в котором каждая пара вершин соединена ровно одним ребром?
   1) полный граф
   2) регулярный граф
   3) связный граф
   4) дерево

5. Что такое цикл в графе?
   1) путь, который проходит через каждую вершину графа ровно один раз
   2) путь, начальная и конечная вершины которого совпадают
   3) соединение двух вершин ребром
   4) граф, в котором есть хотя бы одна вершина

6. Конечный связный граф является деревом. У этого графа 8 вершин. Сколько ребер у него?
   1) 4
   2) 7
   3) 8
   4) 9
   5) 16

7. Какие из следующих алгоритмов используется для поиска кратчайшего пути во взвешенном графе?
   1) BFS
   2) DFS
   3) алгоритм Дейкстры
   4) алгоритм Флойда-Уоршелла

8. Какой из следующих алгоритмов не гарантирует нахождения кратчайшего пути в графе?
   1) BFS
   2) DFS
   3) алгоритм Дейкстры
   4) алгоритм Флойда-Уоршелла

9. Какой алгоритм используется для поиска пути с наименьшей стоимостью между всеми парами вершин во взвешенном графе?
   1) алгоритм Прима
   2) алгоритм Краскала
   3) алгоритм Дейкстры
   4) алгоритм Флойда-Уоршелла
   5) алгоритм Форда-Беллмана

10. Какой из следующих алгоритмов более эффективен для поиска кратчайшего пути в невзвешенном графе?
   1) BFS
   2) DFS
   3) алгоритм Дейкстры
   4) алгоритм Флойда-Уоршелла

11. Какие алгоритмы позволяют определить связность в графе?
   1) BFS

2) DFS
3) алгоритм Форда-Беллмана
4) алгоритм Дейкстры
5) алгоритм Флойда-Уоршелла

12. Какой алгоритм более эффективен для нахождения кратчайшего пути в графе с отрицательными весами ребер?

1) алгоритм Дейкстры
2) алгоритм Флойда-Уоршелла
3) алгоритм Форда-Беллмана
4) алгоритм Прима

Критерии оценки:

За каждое правильно выполненное задание назначается 1 балл. В случае задания с выбором нескольких правильных ответов 1 балл дается за полностью выполненное задание, 0 баллов – за хотя бы один неверный ответ. Затем первичные баллы пропорционально переводятся в тестовые баллы согласно технологической карте дисциплины.

Вопросы к экзамену

по дисциплине «Алгоритмы и структуры данных»

1. Понятие алгоритма и исполнителя алгоритма. Свойства алгоритма. Способы описания алгоритмов. Виды алгоритмов.
2. Теоретические подходы к понятию алгоритма.
3. Машина Поста.
4. Машина Тьюринга.
5. Нормальные алгоритмы Маркова.
6. Показатели эффективности алгоритмов. Подсчет числа операций алгоритма.
7. Асимптотические обозначения. Этапы асимптотического анализа.
8. Классы сложности задач P, NP, EXP. Проблема «P=NP?».
9. Алгоритмы поиска в массиве: линейный, линейный с барьером, бинарный. Оценка их сложности.
10. Алгоритмы поиска в массиве: прыжками, интерполяционный. Оценка их сложности.
11. Методы сортировки: классификация и анализ сложности.
12. Описание алгоритма пузырьковой сортировки. Оценка его сложности.
13. Описание алгоритма шейкерной сортировки. Оценка его сложности.
14. Описание алгоритма сортировки выбором. Оценка его сложности.
15. Описание алгоритма сортировки вставками. Оценка его сложности.
16. Описание алгоритма быстрой сортировки (Хоара). Оценка его сложности.
17. Описание алгоритма сортировки слиянием. Оценка его сложности.
18. Описание алгоритма сортировки подсчетом. Оценка его сложности.
19. Описание алгоритма блочной сортировки. Оценка его сложности.
20. Описание алгоритма поразрядной сортировки. Оценка его сложности.
21. Динамические массивы: особенности реализации.
22. Понятие связанного списка, виды списков. Сравнение списков с массивами.
23. Стек на основе массиве: особенности реализации.
24. Очередь на основе массиве: особенности реализации.
25. Понятие хеш-таблицы и варианты ее реализации. Борьба с коллизиями.
26. Понятие дерева, виды деревьев. Бинарные деревья поиска. Обход дерева. Основные операции.
27. AVL-деревья. Принцип организации, основные операции.
28. Алгоритмы балансировки AVL-дерева.
29. Красно-черные деревья. Принцип организации.
30. Жадные алгоритмы: определение, принцип работы. Примеры задач, решаемые жадными алгоритмами.
31. Динамическое программирование: определение, принцип работы. Таблицы динамического программирования.
32. Классические примеры задач, решаемых с помощью динамического программирования.
33. Основные понятия теории графов. Способы представления графов. Способы программной реализации графовых структур, оценка их сложности.
34. Описание и анализ алгоритма обхода графов в глубину (DFS).
35. Описание и анализ алгоритма обхода графов в ширину (BFS).
36. Алгоритм Дейкстры: особенности реализации, оценка сложности.
37. Алгоритм Беллмана-Форда: особенности реализации, оценка сложности.
38. Алгоритм Флойда-Уоршелла: особенности реализации, оценка сложности.
39. Минимальное остовное дерево. Алгоритм Прима.
40. Минимальное остовное дерево. Алгоритм Краскала.

Критерии оценки:

– оценка «отлично» выставляется, если обучающийся в полном объеме усвоил материал программы учебной дисциплины, раскрыл теоретическое содержание вопросов билета, не затрудняется в ответах на дополнительные вопросы экзаменатора, успешно выполнил практическое задание, продемонстрировав необходимые навыки и умения правильного применения теоретических знаний в практической деятельности;

– оценка «хорошо» выставляется, если обучающийся знает материал программы учебной дисциплины, правильно, по существу и последовательно излагает содержание вопросов билета, в целом правильно выполнил практическое задание, владеет основными умениями и навыками, при ответе не допустил существенных ошибок и неточностей;

– оценка «удовлетворительно» выставляется, если обучающийся усвоил только основные положения материала программы учебной дисциплины, содержание вопросов билета изложил поверхностно, без должного обоснования, допускает неточности и ошибки, недостаточно правильные формулировки,

нарушает последовательность в изложении материала, практическое задание выполнил не в полном объеме, испытывает затруднения при ответе на часть дополнительных вопросов;

– оценка «неудовлетворительно» выставляется, если обучающийся не знает основных положений материала программы учебной дисциплины, при ответе на вопросы билета допускает существенные ошибки, не выполнил практическое задание, не смог ответить на большинство дополнительных вопросов или отказался отвечать.