

Документ подписан простой электронной подписью  
Информация о владельце:

ФИО: Косенок Сергей Михайлович

Должность: ректор

Дата подписания: 11.06.2026 09:32:23

Уникальный программный ключ:

e3a68f3eaa1e62674b54f4998099d3d6bfdcf88f6

## Гестовое задание для диагностического тестирования по дисциплине

Технологии программирования, 5 семестр

Код, направление подготовки	09.03.01 Информатика и вычислительная техника
Направленность (профиль)	Искусственный интеллект и экспертные системы
Форма обучения	Очная
Кафедра разработчик	АСОИУ
Выпускающая кафедра	АСОИУ

№	Проверяемая компетенция	Задание	Варианты ответов	Тип сложности вопроса	
1	ПК-7.1 ПК-7.3 ПК-8.2 ПК-13.1 ПК-13.3	ПК-7.2 ПК-8.1 ПК-8.3 ПК-13.2	Какие существуют метрики, отражающие эффективность алгоритма?	1. процессорное время, память 2. адаптивность, простота реализации 3. надежность, масштабируемость	Низкий
2	ПК-7.1 ПК-7.3 ПК-8.2 ПК-13.1 ПК-13.3	ПК-7.2 ПК-8.1 ПК-8.3 ПК-13.2	Динамические структуры данных – это структуры данных, ... под которые ... и ... по мере ... .	—	Низкий
3	ПК-7.1 ПК-7.3 ПК-8.2 ПК-13.1 ПК-13.3	ПК-7.2 ПК-8.1 ПК-8.3 ПК-13.2	При рассмотрении времени работы $T(M)$ и памяти $M(N)$ что нас интересует?	1. точный вид функций. Используется $O$ -символика 2. приближенный вид функций. Используется $o$ -символика 3. точный вид функций $T(N)$ и $M(N)$ 4. приближенный до константы вид функций. Используется $O$ -символика	Низкий
4	ПК-7.1 ПК-7.3 ПК-8.2 ПК-13.1 ПК-13.3	ПК-7.2 ПК-8.1 ПК-8.3 ПК-13.2	Какая оценка снизу справедлива для сортировок?	1. $O(N)$ 2. $O(N^2)$ 3. $O(\log N)$ 4. $O(N \cdot \log N)$	Низкий
5	ПК-7.1 ПК-7.3 ПК-8.2 ПК-13.1 ПК-13.3	ПК-7.2 ПК-8.1 ПК-8.3 ПК-13.2	При размере входных данных $N$ , как рассчитывается время работы алгоритма?	1. в сравнении с $N$ 2. как функция от параметра $N$ 3. как $O(N)$ 4. не зависимо от $N$	Низкий

6	ПК-7.1 ПК-7.3 ПК-8.2 ПК-13.1 ПК-13.3	ПК-7.2 ПК-8.1 ПК-8.3 ПК-13.2	Соотнесите алгоритмы сортировки с их временной сложностью	1. Пузырьком $\leftrightarrow O(N^2)$ 2. Быстрая $\leftrightarrow O(N+K)$ 3. Подсчётом $\leftrightarrow O(N*\log(N))$	Средний
7	ПК-7.1 ПК-7.3 ПК-8.2 ПК-13.1 ПК-13.3	ПК-7.2 ПК-8.1 ПК-8.3 ПК-13.2	Для алгоритма сортировки слиянием merge-sort при каком количестве элементов в последовательности рекурсивное деление должно прерываться, в стандартном виде?	1. 2 2. 3 3. 1 4. 4	Средний
8	ПК-7.1 ПК-7.3 ПК-8.2 ПК-13.1 ПК-13.3	ПК-7.2 ПК-8.1 ПК-8.3 ПК-13.2	Какое максимальное число потомков может быть у узла бинарного дерева?	—	Средний
9	ПК-7.1 ПК-7.3 ПК-8.2 ПК-13.1 ПК-13.3	ПК-7.2 ПК-8.1 ПК-8.3 ПК-13.2	Какое дерево называется разбалансированным?	1. если существуют вершины-потомки, ключи которых больше ключей родителей, если в остальных вершинах это свойство не нарушено 2. если в нем нарушен порядок неубывания ключей 3. если значения ключей в левом поддереве намного меньше значений ключей в правом поддереве 4. размеры левых и правых поддеревьев в нем сильно различаются	Средний

10	ПК-7.1 ПК-7.3 ПК-8.2 ПК-13.1 ПК-13.3	ПК-7.2 ПК-8.1 ПК-8.3 ПК-13.2	Где будет находиться наиболее часто встречающийся символ в дереве кодирования Хаффмана?	<ol style="list-style-type: none"> <li>1. в самой крайней правой вершине</li> <li>2. может находиться в любом месте</li> <li>3. на верхнем уровне дерева</li> <li>4. на нижнем уровне дерева</li> <li>5. в самой крайней левой вершине</li> </ol>	Средний
11	ПК-7.1 ПК-7.3 ПК-8.2 ПК-13.1 ПК-13.3	ПК-7.2 ПК-8.1 ПК-8.3 ПК-13.2	Бинарное дерево — это ... структура данных, в которой каждый ... содержит ... и ... на левого и правого ... .	—	Средний
12	ПК-7.1 ПК-7.3 ПК-8.2 ПК-13.1 ПК-13.3	ПК-7.2 ПК-8.1 ПК-8.3 ПК-13.2	Какие две операции должен выполнять стек?	<ol style="list-style-type: none"> <li>1. set, get</li> <li>2. insert, delete</li> <li>3. push, pop</li> <li>4. enqueue, dequeue</li> </ol>	Средний
13	ПК-7.1 ПК-7.3 ПК-8.2 ПК-13.1 ПК-13.3	ПК-7.2 ПК-8.1 ПК-8.3 ПК-13.2	Сколько дополнительной памяти требуется для работы алгоритма quick-sort?	<ol style="list-style-type: none"> <li>1. алгоритм не использует дополнительную память</li> <li>2. <math>O(N^3)</math></li> <li>3. <math>O(N)</math></li> <li>4. <math>O(N^2)</math></li> </ol>	Средний
14	ПК-7.1 ПК-7.3 ПК-8.2 ПК-13.1 ПК-13.3	ПК-7.2 ПК-8.1 ПК-8.3 ПК-13.2	Что означает устойчивость алгоритма сортировки?	<ol style="list-style-type: none"> <li>1. если при работе алгоритма относительный порядок пар с равными ключами не меняется</li> <li>2. время работы алгоритма относительно стабильно при различной величине входных данных</li> <li>3. сортировка происходит на любых данных</li> <li>4. процент ошибок при сортировке меньше</li> </ol>	Средний

15	ПК-7.1 ПК-7.3 ПК-8.2 ПК-13.1 ПК-13.3	ПК-7.2 ПК-8.1 ПК-8.3 ПК-13.2	Какие высказывания относятся к структуре данных связный список?	<ol style="list-style-type: none"> <li>1. время доступа к элементу константное</li> <li>2. в каждом узле содержится указатель на следующий узел и данные</li> <li>3. эта структура используется для реализации стека</li> <li>4. в конце структуры нулевой указатель, указатель на первый элемент хранится отдельно</li> </ol>	Средний
16	ПК-7.1 ПК-7.3 ПК-8.2 ПК-13.1 ПК-13.3	ПК-7.2 ПК-8.1 ПК-8.3 ПК-13.2	Какие действия включает в себя операция вставки (Insert(x)) в двоичном дереве поиска?	<ol style="list-style-type: none"> <li>1. поиск ключа <math>x</math> в дереве</li> <li>2. вершину <math>w</math> объявим левым сыном <math>v</math>, если <math>key(v) &gt; key(w)</math></li> <li>3. вершину <math>w</math> объявим правым сыном <math>v</math>, если <math>key(v) &lt; key(w)</math></li> <li>4. если поиск завершился неудачей, создадим новую вершину <math>w</math> с ключем <math>x</math></li> <li>5. если поиск завершился удачей, создадим новую вершину <math>w</math> с ключем <math>x</math></li> </ol>	Высокий
17	ПК-7.1 ПК-7.3 ПК-8.2 ПК-13.1 ПК-13.3	ПК-7.2 ПК-8.1 ПК-8.3 ПК-13.2	Типичный порядок полей триплета для LZ77:	<ol style="list-style-type: none"> <li>1. offset</li> <li>2. length</li> <li>3. next</li> </ol>	Высокий
18	ПК-7.1 ПК-7.3 ПК-8.2 ПК-13.1 ПК-13.3	ПК-7.2 ПК-8.1 ПК-8.3 ПК-13.2	Выберите компоненты L-системы	<ol style="list-style-type: none"> <li>1. множество целых чисел, называемое ключами</li> <li>2. алфавит</li> <li>3. теорема</li> <li>4. правила</li> <li>5. аксиома</li> <li>6. набор ограничений</li> </ol>	Высокий

19	ПК-7.1 ПК-7.3 ПК-8.2 ПК-13.1 ПК-13.3	ПК-7.2 ПК-8.1 ПК-8.3 ПК-13.2	Основные проблемы, которые необходимо решать при реализации алгоритма RLE:	<ol style="list-style-type: none"> <li>1. сохранение закодированных данных на диск</li> <li>2. корректная работа со скользящим окном</li> <li>3. хранение таблицы символов</li> <li>4. способность алгоритма отличать закодированные данные от исходных</li> </ol>	Высокий
20	ПК-7.1 ПК-7.3 ПК-8.2 ПК-13.1 ПК-13.3	ПК-7.2 ПК-8.1 ПК-8.3 ПК-13.2	Что можно сделать для алгоритма Quick-sort, чтобы дерево рекурсии было всегда сбалансированным?	<ol style="list-style-type: none"> <li>1. уменьшить число рекурсий в рекурсивной функции</li> <li>2. увеличить количество рекурсивных вызовов для функции</li> <li>3. заменить рекурсию на цикл</li> <li>4. выбирать правильный опорный элемент (pivot)</li> </ol>	Высокий