

Документ подписан и проверен владельцем подлинно
 Информация о владельце:
 ФИО: Косенок Сергей Михайлович
 Должность: ректор
 Дата подписания: 19.06.2024 07:40:57
 Уникальный программный ключ:
 e3a68f55ea1e62674b54f4998099d3d6b1dctf856

Тестовое задание для диагностического тестирования по дисциплине:

Технологии программирования, 3 семестр

Код, направление подготовки	09.03.02 Информационные системы технологии
Направленность (профиль)	Информационные системы и технологии
Форма обучения	Очная
Кафедра разработчик	Информатики и вычислительной техники
Выпускающая кафедра	Информатики и вычислительной техники

Проверяемая компетенция	Задание	Варианты ответов	Тип сложности вопроса
ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-3.1 ОПК-3.2 ОПК-3.3 ОПК-4.1 ОПК-4.2 ОПК-4.3 ОПК-6.1 ОПК-6.2 ОПК-6.3	При рассмотрении времени работы $T(M)$ и памяти $M(N)$ что нас интересует?	1. точный вид функций $T(N)$ и $M(N)$ 2. приближенный до константы вид функций. Используется Осимволика 3. приближенный вид функций. Используется осимволика 4. точный вид функций. Используется Осимволика	Низкий
ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-3.1 ОПК-3.2 ОПК-3.3 ОПК-4.1	Какие существуют метрики, отражающие эффективность алгоритма?	1. процессорное время, память 2. адаптивность, простота реализации 3. надежность, масштабируемость	Низкий

ОПК-4.2 ОПК-4.3 ОПК-6.1 ОПК-6.2 ОПК-6.3			
ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-3.1 ОПК-3.2 ОПК-3.3 ОПК-4.1 ОПК-4.2 ОПК-4.3 ОПК-6.1 ОПК-6.2 ОПК-6.3	Динамические структуры данных – это структуры данных, _____ под которые _____ и _____ по мере _____		Низкий
ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-3.1 ОПК-3.2 ОПК-3.3 ОПК-4.1 ОПК-4.2 ОПК-4.3 ОПК-6.1 ОПК-6.2 ОПК-6.3	При размере входных данных N, как рассчитывается время работы алгоритма?	<ol style="list-style-type: none"> 1. как функция от параметра N 2. не зависимо от N 3. в сравнении с N 4. как O(N) 	Низкий
ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-3.1 ОПК-3.2 ОПК-3.3 ОПК-4.1 ОПК-4.2 ОПК-4.3 ОПК-6.1 ОПК-6.2 ОПК-6.3	Какая оценка снизу справедлива для сортировок?	<ol style="list-style-type: none"> 1. $O(N^2)$ 2. $O(N)$ 3. $O(\log N)$ 4. $O(N \cdot \log N)$ 	Низкий

ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-3.1 ОПК-3.2 ОПК-3.3 ОПК-4.1 ОПК-4.2 ОПК-4.3 ОПК-6.1 ОПК-6.2 ОПК-6.3	Какие две операции должен выполнять стек?	1. enqueue, dequeue 2. set, get 3. insert, delete 4. push, pop	Средний
ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-3.1 ОПК-3.2 ОПК-3.3 ОПК-4.1 ОПК-4.2 ОПК-4.3 ОПК-6.1 ОПК-6.2 ОПК-6.3	Соотнесите алгоритмы сортировки с их временной сложностью	1. Пузырьком $\Leftrightarrow O(N+K)$ 2. Быстрая $\Leftrightarrow O(N*\log(N))$ 3. Подсчётом $\Leftrightarrow O(N^2)$	Средний
ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-3.1 ОПК-3.2 ОПК-3.3 ОПК-4.1 ОПК-4.2 ОПК-4.3 ОПК-6.1 ОПК-6.2 ОПК-6.3	Какое максимальное число потомков может быть у узла бинарного дерева?	Правильные ответы: 1. 2	Средний
ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-3.1 ОПК-3.2 ОПК-3.3	Для алгоритма сортировки слиянием mergesort при каком количестве элементов в последовательности	1. 2 2. 1 3. 4 4. 3	Средний

ОПК-4.1 ОПК-4.2 ОПК-4.3 ОПК-6.1 ОПК-6.2 ОПК-6.3	и рекурсивное деление должно прерываться, в стандартном виде?		
ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-3.1 ОПК-3.2 ОПК-3.3 ОПК-4.1 ОПК-4.2 ОПК-4.3 ОПК-6.1 ОПК-6.2 ОПК-6.3	Сколько дополнительной памяти требуется для работы алгоритма quicksort?	<ol style="list-style-type: none"> 1. $O(N^2)$ 2. $O(N^3)$ 3. $O(N)$ 4. алгоритм не использует дополнительную память 	Средний

<p>ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-3.1 ОПК-3.2 ОПК-3.3 ОПК-4.1 ОПК-4.2 ОПК-4.3 ОПК-6.1 ОПК-6.2 ОПК-6.3</p>	<p>Что означает устойчивость алгоритма сортировки?</p>	<ol style="list-style-type: none"> 1. сортировка происходит на любых данных 2. время работы алгоритма относительно стабильно при различной величине входных данных 3. процент ошибок при сортировке меньше 4. если при работе алгоритма относительный порядок пар с равными ключами не меняется 	<p>Средний</p>	<p>5</p>
<p>ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-3.1 ОПК-3.2 ОПК-3.3 ОПК-4.1 ОПК-4.2 ОПК-4.3 ОПК-6.1 ОПК-6.2 ОПК-6.3</p>	<p>Какие высказывания относятся к структуре данных связный список?</p>	<ol style="list-style-type: none"> 1. в конце структуры нулевой указатель , указатель на первый элемент хранится отдельно 2. эта структура используется для реализации стека 3. в каждом узле содержатся указатель на следующий узел и данные 4. время доступа к элементу 	<p>Средний</p>	<p>5</p>

		константное		
ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-3.1 ОПК-3.2 ОПК-3.3 ОПК-4.1 ОПК-4.2 ОПК-4.3 ОПК-6.1 ОПК-6.2 ОПК-6.3	Какое дерево называется разбалансированным?	1. размеры левых и правых поддеревьев в нем сильно различаются 2. если существуют вершины-потомки, ключи которых больше ключей родителей, если в остальных вершинах это свойство не нарушено 3. если значения ключей в левом поддереве намного меньше значений ключей в правом поддереве 4. если в нем нарушен порядок неубывания ключей	Средний	5

<p>ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-3.1 ОПК-3.2 ОПК-3.3 ОПК-4.1 ОПК-4.2 ОПК-4.3 ОПК-6.1 ОПК-6.2 ОПК-6.3</p>	<p>Бинарное дерево — это структура данных, в которой каждый _____ содержит _____ и _____ на левого и правого _____.</p>		<p>Средний</p>	<p>5</p>
<p>ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-3.1 ОПК-3.2 ОПК-3.3 ОПК-4.1 ОПК-4.2 ОПК-4.3 ОПК-6.1 ОПК-6.2 ОПК-6.3</p>	<p>Где будет находиться наиболее часто встречающийся символ в дереве кодирования Хаффмана?</p>	<p>1. на нижнем уровне дерева 2. на верхнем уровне дерева 3. в самой крайней правой вершине 4. в самой крайней левой вершине 5. может находиться в любом месте</p>	<p>Средний</p>	<p>5</p>

<p>ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-3.1 ОПК-3.2 ОПК-3.3 ОПК-4.1 ОПК-4.2 ОПК-4.3 ОПК-6.1 ОПК-6.2 ОПК-6.3</p>	<p>Какие действия включает в себя операция вставки (Insert(x)) в двоичном дереве поиска?</p>	<ol style="list-style-type: none"> 1. поиск ключа x в дереве 2. вершину w объявим левым сыном v, если $key(v) > key(w)$ 3. если поиск завершился неудачей, создадим новую вершину w с ключем x 4. если поиск завершился удачей, создадим новую вершину w с ключем x 5. вершину w объявим правым сыном v, если $key(v) < key(w)$ 	<p>Высокий</p>	<p>8</p>
<p>ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-3.1 ОПК-3.2 ОПК-3.3 ОПК-4.1 ОПК-4.2 ОПК-4.3 ОПК-6.1 ОПК-6.2 ОПК-6.3</p>	<p>Выберите компоненты L-системы</p>	<ol style="list-style-type: none"> 1. теорема 2. набор ограничений 3. правила 4. алфавит 5. аксиома 6. множество целых чисел, называемое ключами 	<p>Высокий</p>	<p>8</p>

<p>ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-3.1 ОПК-3.2 ОПК-3.3 ОПК-4.1 ОПК-4.2 ОПК-4.3 ОПК-6.1 ОПК-6.2 ОПК-6.3</p>	<p>Типичный порядок полей триплета для LZ77:</p>	<p>1. next 2. offset 3. length</p>	<p>Высокий</p>	<p>8</p>
<p>ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-3.1 ОПК-3.2 ОПК-3.3 ОПК-4.1 ОПК-4.2 ОПК-4.3 ОПК-6.1 ОПК-6.2 ОПК-6.3</p>	<p>Основные проблемы, которые необходимо решать при реализации алгоритма RLE:</p>	<p>1. способность алгоритма отличать закодированные данные от исходных 2. корректная работа со скользящим окном 3. хранение таблицы символов 4. сохранение закодированных данных на диск</p>	<p>Высокий</p>	<p>8</p>

<p>ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-3.1 ОПК-3.2 ОПК-3.3 ОПК-4.1 ОПК-4.2 ОПК-4.3 ОПК-6.1 ОПК-6.2 ОПК-6.3</p>	<p>Что можно сделать для алгоритма Quick-sort, чтобы дерево рекурсии было всегда сбалансированным ?</p>	<ol style="list-style-type: none"> 1. увеличить количество рекурсивных вызовов для функции 2. уменьшить число рекурсий в рекурсивной функции 3. заменить рекурсию на цикл 4. выбирать правильный опорный элемент (pivot) 	<p>Высокий</p>	<p>8</p>
--	---	--	----------------	----------