

Документ подписан в электронном виде  
 Информация о владельце:  
 ФИО: Косенок Сергей Михайлович  
 Должность: ректор  
 Дата подписания: 19.06.2024 07:24:06  
 Уникальный программный ключ:  
 e3a68f3eaa1e62674b54f4998099d3d6bfdcf836

## Тестовое задание для диагностического тестирования по дисциплине:

Технологии программирования, 3 семестр

Код, направление подготовки	09.03.02 Информационные системы технологии
Направленность (профиль)	Безопасность информационных систем и технологий
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	Какие существуют метрики, отражающие эффективность алгоритма?	1. процессорное время, память 2. адаптивность, простота реализации	Низкий

ОПК-3.3 ОПК-4.1 ОПК-4.2 ОПК-4.3 ОПК-6.1 ОПК-6.2 ОПК-6.3		3. надежность, масштабируемость	
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?	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	Какая оценка снизу справедлива для сортировок?	1. O(N <sup>2</sup> ) 2. O(N) 3. O(log N) 4. O(N*log N)	Низкий

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	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 $\leq$ $O(N+K)$ 2. Быстрая $\leq$ $O(N*\log(N))$ 3. Подсчётом $\leq$ $O(N^2)$	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	Для алгоритма сортировки слиянием mergesort при каком количестве	1. 2 2. 1 3. 4 4. 3	Средний

ОПК-3.2 ОПК-3.3 ОПК-4.1 ОПК-4.2 ОПК-4.3 ОПК-6.1 ОПК-6.2 ОПК-6.3	элементов в последовательности 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"> <li>1. <math>O(N^2)</math></li> <li>2. <math>O(N^3)</math></li> <li>3. <math>O(N)</math></li> <li>4. алгоритм не использует дополнительную память</li> </ol>	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	Что означает устойчивость алгоритма сортировки?	<ol style="list-style-type: none"> <li>1. сортировка происходит на любых данных</li> <li>2. время работы алгоритма относительно стабильно при различной величине входных данных</li> <li>3. процент ошибок при сортировке меньше</li> <li>4. если при работе алгоритма относительный порядок пар с равными ключами не меняется</li> </ol>	Средний
ОПК-1.1 ОПК-1.2	Какие высказывания	1. в конце структуры нулевой указатель ,	Средний

<p>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>относятся к структуре данных связный список?</p>	<p>указатель на первый элемент хранится отдельно</p> <ol style="list-style-type: none"> <li>2. эта структура используется для реализации стека</li> <li>3. в каждом узле содержатся указатель на следующий узел и данные</li> <li>4. время доступа к элементу константное</li> </ol>	
<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>Какое дерево называется разбалансированным?</p>	<ol style="list-style-type: none"> <li>1. размеры левых и правых поддеревьев в нем сильно различаются</li> <li>2. если существуют вершины-потомки, ключи которых больше ключей родителей, если в остальных вершинах это свойство не нарушено</li> <li>3. если значения ключей в левом поддереве намного меньше значений ключей в правом поддереве</li> <li>4. если в нем нарушен порядок неубывания ключей</li> </ol>	<p>Средний</p>
<p>ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-3.1</p>	<p>Бинарное дерево — это _____ структура данных, в которой каждый _____ содержит</p>		<p>Средний</p>

ОПК-3.2 ОПК-3.3 ОПК-4.1 ОПК-4.2 ОПК-4.3 ОПК-6.1 ОПК-6.2 ОПК-6.3	_____ и _____ на левого и правого _____.		
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	Где будет находиться наиболее часто встречающийся символ в дереве кодирования Хаффмана?	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	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	Какие действия включает в себя операция вставки (Insert(x)) в двоичном дереве поиска?	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 $key(v) < key(w)$	Высокий
ОПК-1.1	Выберите	1. теорема	Высокий

<p>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>компоненты L- системы</p>	<p>2. набор ограничений 3. правила 4. алфавит 5. аксиома 6. множество целых чисел, называемое ключами</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>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>ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-3.1 ОПК-3.2 ОПК-3.3 ОПК-4.1</p>	<p>Что можно сделать для алгоритма Quick-sort, чтобы дерево рекурсии было всегда сбалансированным ?</p>	<p>1. увеличить количество рекурсивных вызовов для функции 2. уменьшить число рекурсий в</p>	<p>Высокий</p>

ОПК-4.2 ОПК-4.3 ОПК-6.1 ОПК-6.2 ОПК-6.3		рекурсивной функции 3. заменить рекурсию на цикл 4. выбирать правильный опорный элемент (pivot)	
---	--	--	--