

Тестовое задание для диагностического тестирования по дисциплине

Технологии программирования, 6 семестр

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

№	Проверяемая компетенция	Задание	Варианты ответов	Тип сложности вопроса
1	ОПК-1.1, ОПК-2.1, ПК-3.1, ПК-5.1, ПК-11.1	Какие существуют метрики, отражающие эффективность алгоритма?	1. процессорное время, память 2. адаптивность, простота реализации 3. надежность, масштабируемость	Низкий
2	ОПК-1.1, ОПК-2.1, ПК-3.1, ПК-5.1, ПК-11.1	Динамические структуры данных – это структуры данных, ... под которые ... и ... по мере	—	Низкий
3	ОПК-1.1, ОПК-2.1, ПК-3.1, ПК-5.1, ПК-11.1	При рассмотрении времени работы $T(M)$ и памяти $M(N)$ что нас интересует?	1. приближенный вид функций. Используется O -символика 2. точный вид функций. Используется O -символика 3. приближенный до константы вид функций. Используется O -символика 4. точный вид функций $T(N)$ и $M(N)$	Низкий

4	ОПК-1.1, ОПК-2.1, ПК-3.1, ПК-5.1, ПК-11.1	Какая оценка снизу справедлива для сортировок?	1. $O(\log N)$ 2. $O(N \cdot \log N)$ 3. $O(N^2)$ 4. $O(N)$	Низкий
5	ОПК-1.1, ОПК-2.1, ПК-3.1, ПК-5.1, ПК-11.1	При размере входных данных N , как рассчитывается время работы алгоритма?	1. в сравнении с N 2. как функция от параметра N 3. не зависимо от N 4. как $O(N)$	Низкий
6	ОПК-2.2, ПК-3.2, ПК-5.2, ПК-11.2	Соотнесите алгоритмы сортировки с их временной сложностью	1. Пузырьком $O(N+K)$ 2. Быстрая $O(N \cdot \log(N))$ 3. Подсчётом $O(N^2)$	Средний
7	ОПК-2.2, ПК-3.2, ПК-5.2, ПК-11.2	Для алгоритма сортировки слиянием merge-sort при каком количестве элементов в последовательности рекурсивное деление должно прерываться, в стандартном виде?	1. 2 2. 1 3. 3 4. 4	Средний
8	ОПК-2.2, ПК-3.2, ПК-5.2, ПК-11.2	Какое максимальное число потомков может быть у узла бинарного дерева?	—	Средний

9	ОПК-2.2, ПК-3.2, ПК-5.2, ПК-11.2	Какое дерево называется разбалансированным?	<ol style="list-style-type: none"> 1. если существуют вершины-потомки, ключи которых больше ключей родителей, если в остальных вершинах это свойство не нарушено 2. размеры левых и правых поддеревьев в нем сильно различаются 3. если значения ключей в левом поддереве намного меньше значений ключей в правом поддереве 4. если в нем нарушен порядок неубывания ключей 	Средний
10	ОПК-2.2, ПК-3.2, ПК-5.2, ПК-11.2	Где будет находиться наиболее часто встречающийся символ в дереве кодирования Хаффмана?	<ol style="list-style-type: none"> 1. на нижнем уровне дерева 2. в самой крайней левой вершине 3. в самой крайней правой вершине 4. на верхнем уровне дерева 5. может находиться в любом месте 	Средний
11	ОПК-2.2, ПК-3.2, ПК-5.2, ПК-11.2	Бинарное дерево — это ... структура данных, в которой каждый ... содержит ... и ... на левого и правого	—	Средний
12	ОПК-2.2, ПК-3.2, ПК-5.2, ПК-11.2	Какие две операции должен выполнять стек?	<ol style="list-style-type: none"> 1. enqueue, dequeue 2. push, pop 3. set, get 4. insert, delete 	Средний

13	ОПК-2.2, ПК-3.2, ПК-5.2, ПК-11.2	Сколько дополнительной памяти требуется для работы алгоритма quick-sort?	<ol style="list-style-type: none"> 1. алгоритм не использует дополнительную память 2. $O(N^2)$ 3. $O(N^3)$ 4. $O(N)$ 	Средний
14	ОПК-2.2, ПК-3.2, ПК-5.2, ПК-11.2	Что означает устойчивость алгоритма сортировки?	<ol style="list-style-type: none"> 1. процент ошибок при сортировке меньше 2. сортировка происходит на любых данных 3. если при работе алгоритма относительный порядок пар с равными ключами не меняется 4. время работы алгоритма относительно стабильно при различной величине входных данных 	Средний
15	ОПК-2.2, ПК-3.2, ПК-5.2, ПК-11.2	Какие высказывания относятся к структуре данных связный список?	<ol style="list-style-type: none"> 1. эта структура используется для реализации стека 2. в конце структуры нулевой указатель, указатель на первый элемент хранится отдельно 3. время доступа к элементу константное 4. в каждом узле содержатся указатель на следующий узел и данные 	Средний
16	ОПК-1.3, ОПК-2.3, ПК-5.3, ПК-3.3, ПК-11.3	Какие действия включает в себя операция вставки (Insert(x)) в двоичном дереве поиска?	<ol style="list-style-type: none"> 1. вершину w объявим левым сыном v, если $key(v) > key(w)$ 2. поиск ключа x в дереве 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

17	ОПК-1.3, ОПК-2.3, ПК-5.3, ПК-3.3, ПК-11.3	Типичный порядок полей триплета для LZ77:	1. length 2. next 3. offset	Высокий
18	ОПК-1.3, ОПК-2.3, ПК-5.3, ПК-3.3, ПК-11.3	Выберите компоненты L-системы	1. алфавит 2. множество целых чисел, называемое ключами 3. правила 4. аксиома 5. набор ограничений 6. теорема	Высокий
19	ОПК-1.3, ОПК-2.3, ПК-5.3, ПК-3.3, ПК-11.3	Основные проблемы, которые необходимо решать при реализации алгоритма RLE:	1. корректная работа со скользящим окном 2. хранение таблицы символов 3. способность алгоритма отличать закодированные данные от исходных 4. сохранение закодированных данных на диск	Высокий
20	ОПК-1.3, ОПК-2.3, ПК-5.3, ПК-3.3, ПК-11.3	Что можно сделать для алгоритма Quick-sort, чтобы дерево рекурсии было всегда сбалансированным?	1. уменьшить число рекурсий в рекурсивной функции 2. увеличить количество рекурсивных вызовов для функции 3. заменить рекурсию на цикл 4. выбирать правильный опорный элемент (pivot)	Высокий

№	ПРАВИЛЬНЫЕ ОТВЕТЫ
1	процессорное время, память
2	память; выделяется; освобождается; необходимости
3	приближенный до константы вид функций. Используется O-символика
4	$O(N \cdot \log N)$

5	как функция от параметра N
6	Пузырьком $O(N^2)$; Быстрая $O(N \cdot \log(N))$; Подсчётом $O(N+K)$
7	1
8	2 шт.
9	размеры левых и правых поддеревьев в нем сильно различаются
10	на верхнем уровне дерева
11	иерархическая; узел ; значение ; ссылки ; потомка
12	push, pop
13	алгоритм не использует дополнительную память
14	если при работе алгоритма относительный порядок пар с равными ключами не меняется
15	в каждом узле содержатся указатель на следующий узел и данные; эта структура используется для реализации стека; в конце структуры нулевой указатель , указатель на первый элемент хранится отдельно; время доступа к элементу константное
16	если поиск завершился удачей, создадим новую вершину w с ключем x
17	offset; length; next
18	алфавит; аксиома; правила
19	сохранение закодированных данных на диск; способность алгоритма отличать закодированные данные от исходных
20	уменьшить число рекурсий в рекурсивной функции; выбирать правильный опорный элемент (pivot)