

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Косенок Сергей Михайлович
Должность: ректор
Дата подписания: 19.06.2024 06:50:04
Уникальный программный ключ:
e3a68f3eaa1e62674b54f4998099d3d6bfdcf836

Тестовое задание для диагностического тестирования по дисциплине:

Структуры и алгоритмы обработки данных

Код, направление подготовки	09.03.04 Программная инженерия
Направленность (профиль)	Программное обеспечение компьютерных систем
Форма обучения	очная
Кафедра-разработчик	автоматики и компьютерных систем
Выпускающая кафедра	автоматики и компьютерных систем

Диагностический тест по дисциплине «Структуры и алгоритмы обработки данных»

Проверяемые компетенции	Задание	Варианты ответов	Тип сложности
ПК-4.1 ПК-6.2 ПК-7.2 ПК-7.3	1) Рекуррентное выражение временной сложности алгоритма $T(N)$ определяется выражением $T(N) = T(N-1) + 1$, если $N > 1$ и $T(N) = 1$ в противном случае. Асимптотическая сложность алгоритма равна	1) N 2) N^2 3) N^N 4) N^3	легкий
ПК-4.1 ПК-6.2 ПК-7.2 ПК-7.3	2) Рекуррентное выражение временной сложности алгоритма $T(N)$ определяется выражением $T(N) = 2T(N/2) + N$, если $N > 1$, $T(1) = 0$. Асимптотическая сложность алгоритма равна	1) $\ln N$ 2) $N \ln N$ 3) $N^2 \ln N$ 4) N^2	легкий
ПК-4.1 ПК-6.2 ПК-7.2 ПК-7.3	3) Модификация сортировкой вставками сортировки слиянием позволяет	1) Получить естественную сортировку 2) Улучшить временные характеристики сортировки 3) Уменьшить требованиям по памяти 4) Уменьшить асимптотическую сложность от N^2 до $N \log(N)$	легкий
ПК-4.1 ПК-6.2 ПК-7.2 ПК-7.3	4) Алгоритм сортировки распределяющим подсчетом не используют для сортировки строк потому, что	1) Он не обладает необходимыми временными характеристиками 2) Требуется дополнительной памяти 3) Применим к целым числам 4) Имеет линейную асимптотическую сложность	легкий
ПК-4.1 ПК-6.2 ПК-7.2 ПК-7.3	5) Алгоритм последовательного поиска в худшем случае при неудачном поиске имеет асимптотическую сложность	1) $O(1)$ 2) $O(N)$ 3) $O(\log N)$ 4) Нет правильных вариантов ответов	легкий
ПК-4.1 ПК-6.2 ПК-7.2	6) Временная сложность некоторого алгоритма определяется выражением $f(N) = N^3/3 + (10N \cdot \ln N)^2$.	1) $N^3/3$ 2) N^3 3) $(10N \cdot \ln N)^2$	средний

ПК-7.3	Асимптотическая сложность $O(f(N))$ будет равна (выберите два подходящие варианта ответов)	4) $(N \cdot \ln N)^2$	
ПК-4.1 ПК-6.2 ПК-7.2 ПК-7.3	7) Какая структура данных обеспечивает эффективность добавление элемента, имеющую сложность $O(1)$ (выберите три подходящие варианта ответов)	1) связный список 2) стек 3) очередь 4) бинарное дерево поиска	средний
ПК-4.1 ПК-6.2 ПК-7.2 ПК-7.3	8) Пусть алгоритм представлен следующим псевдокодом: <pre>for (i = 0; i < N/2; i++) { for (j = 0; j < N/3; j++) { f(N, other); } }</pre> причем асимптотическая сложность $f(N, other)$ составляет $O(N)$. Асимптотическая сложность алгоритма равна (выберите два подходящие варианта ответов)	1) $O(N^2)$ 2) $O(N)$ 3) $O(N \log N)$ 4) $O(N^2 \log N)$ 5) $O(N^3)$ 6) $O(N^3/6)$	средний
ПК-4.1 ПК-6.2 ПК-7.2 ПК-7.3	9) Пусть алгоритм представлен следующим псевдокодом: <pre>for (i = N; i > 0; i /= 2) { for (j = 0; j < N/3; j++) { f(N, other); } }</pre> причем асимптотическая сложность $f(N, other)$ составляет $O(N)$. Асимптотическая сложность алгоритма равна (выберите два подходящие варианта ответов)	1) $O(N^2)$ 2) $O(N)$ 3) $O(N \log N)$ 4) $O(N^2 \log N)$ 5) $O(N^3)$ 6) $O(N^3/6)$	средний
ПК-4.1 ПК-6.2 ПК-7.2 ПК-7.3	10) Принцип организации абстрактного типа данных «стек» (выберите все подходящие варианты ответов)	1) FILO (First Input Last Output) 2) FIFO (First Input First Output) 3) LIFO (Last Input First Output) 4) Справедливы варианты 1 и 2	средний

ПК-4.1 ПК-6.2 ПК-7.2 ПК-7.3	11) Алгоритм сортировка вставками имеет в худшем и лучшем случаях асимптотическую сложность соответственно	1) $O(N^2)$ и $O(N^2/2)$ 2) $O(N^2/2)$ и $O(N^2/4)$ 3) $O(N^2/2)$ и $O(N)$ 4) $O(N^2)$ и $O(\ln N)$	средний
ПК-4.1 ПК-6.2 ПК-7.2 ПК-7.3	12) Какие из следующих алгоритмов имеют асимптотическую сложность $N \log(N)$ в среднем (выберите два подходящие варианта ответов)	1) Пирамидальная сортировка 2) Сортировка Хоара 3) Сортировка вставками 4) Сортировка выбором	средний
ПК-4.1 ПК-6.2 ПК-7.2 ПК-7.3	13) В пустое бинарное дерево поиска последовательно добавляются ключи 3, 2, 5, 4. Чему равна разность сумм ключей между левым и правым поддеревьями.	1) 5 2) 6 3) -6 4) -7	средний
ПК-4.1 ПК-6.2 ПК-7.2 ПК-7.3	14) Количество возможных вариантов построения бинарного дерева поиска (его структуры), состоящего из четырех узлов, равно	1) 12 2) 10 3) 14 4) 18	средний
ПК-4.1 ПК-6.2 ПК-7.2 ПК-7.3	15) Предложите наиболее оптимальный способ реализации абстрактного типа данных «Множество» (известно, что ключами будут целые числа типа unsigned char)	1) упорядоченный список 2) упорядоченный массив 3) бинарное дерево поиска 4) сбалансированное дерево поиска 5) хеш-таблица 6) битовый массив	средний
ПК-4.1 ПК-6.2 ПК-7.2 ПК-7.3	16) Расположите алгоритмы в порядке повышения производительности (оцениваемой по временной асимптотической сложности в среднем) в среднем	1) Бинарный поиск 2) Последовательный поиск 3) Сортировка вставками 4) Сортировка Шелла 5) Пирамидальная сортировка	высокий
ПК-4.1 ПК-6.2 ПК-7.2	17) Расположите алгоритмы в порядке повышения производительности (оцениваемой по асимптотической сложности	1) Сортировка Шелла 2) Сортировка Хоара 3) Сортировка выбором	высокий

ПК-7.3	в среднем по количеству операций сравнения ключей) в среднем	4) Сортировка вставками	
ПК-4.1 ПК-6.2 ПК-7.2 ПК-7.3	18) Расположите алгоритмы в порядке повышения производительности (оцениваемой по асимптотической сложности в среднем по количеству операций сравнения ключей) в среднем	1) Последовательный поиск 2) Интерполяционный поиск 3) Поиск прыжками 4) Бинарный поиск	высокий
ПК-4.1 ПК-6.2 ПК-7.2 ПК-7.3	19) Пусть есть бинарное дерево, у которого каждый не листовой узел имеет ровно два потомка. Если у такого дерева 11 листьев, то общее количество узлов равно	Вводимый ответ	высокий
ПК-4.1 ПК-6.2 ПК-7.2 ПК-7.3	20) Предложите наиболее два наиболее оптимальных способа реализации абстрактного типа данных «Множество» (известно, что ключами будут строки)	1) упорядоченный список 2) упорядоченный массив 3) бинарное дерево поиска 4) сбалансированное дерево поиска 5) хеш-таблица 6) битовый массив	высокий

