

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Косенок Сергей Михайлович
Должность: ректор
Дата подписания: 06.06.2024 07:50:55
Уникальный программный ключ:
e3a68f3eaa1e62674b54f4998099d3d6bfdcf836

Тестовое задание для диагностического тестирования по дисциплине:

Алгоритмы и структуры данных

Квалификация выпускника	бакалавр
Направление подготовки	01.03.02 «Прикладная математика и информатика»
Направленность (профиль)	«Технологии программирования и анализ данных»
Форма обучения	Очная
Кафедра-разработчик	Прикладной математики
Выпускающая кафедра	Прикладной математики

Диагностический тест по дисциплине «Алгоритмы и структуры данных» за третий семестр

Проверяемые компетенции	Задание	Варианты ответов	Тип сложности
ПК-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1 ОПК-5.2 ОПК-5.3	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	низкий
ПК-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1 ОПК-5.2 ОПК-5.3	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	низкий
ПК-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1 ОПК-5.2 ОПК-5.3	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 $N \log(N)$	низкий
ПК-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1	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	низкий

ОПК-5.2 ОПК-5.3			
ПК-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1 ОПК-5.2 ОПК-5.3	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	низкий
ПК-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1 ОПК-5.2 ОПК-5.3	6) Временная сложность некоторого алгоритма определяется выражением $f(N)=N^3/3 + (10N \cdot \ln N)^2$. Асимптотическая сложность $O(f(N))$ будет равна (выберите два подходящие варианта ответов)	1) $N^3/3$ 2) N^3 3) $(10N \cdot \ln N)^2$ 4) $(N \cdot \ln N)^2$	средний
ПК-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1 ОПК-5.2 ОПК-5.3	7) Какая структура данных обеспечивает эффективность добавление в начало, имеющую сложность $O(1)$ (выберите три подходящие варианта ответов)	1) связный список 2) стек 3) очередь 4) дерево	средний
ПК-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1 ОПК-5.2 ОПК-5.3	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 $O(N)$.	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

	Асимптотическая сложность алгоритма равна (выберите два подходящие варианта ответов)		
ПК-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1 ОПК-5.2 ОПК-5.3	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-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1 ОПК-5.2 ОПК-5.3	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-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1 ОПК-5.2 ОПК-5.3	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-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1	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

ОПК-5.2 ОПК-5.3			
ПК-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1 ОПК-5.2 ОПК-5.3	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-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1 ОПК-5.2 ОПК-5.3	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-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1 ОПК-5.2 ОПК-5.3	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-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1 ОПК-5.2 ОПК-5.3	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

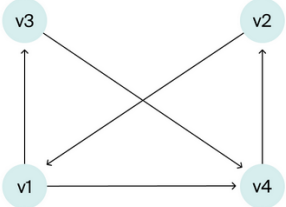
ПК-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1 ОПК-5.2 ОПК-5.3	17) Расположите алгоритмы в порядке повышения производительности (оцениваемой по асимптотической сложности в среднем по количеству операций сравнения ключей) в среднем	1) Сортировка Шелла 2) Сортировка Хоара 3) Сортировка выбором 4) Сортировка вставками	высокий
ПК-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1 ОПК-5.2 ОПК-5.3	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-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1 ОПК-5.2 ОПК-5.3	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	—	высокий
ПК-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1 ОПК-5.2 ОПК-5.3	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

Диагностический тест по дисциплине «Алгоритмы и структуры данных» за четвертый семестр

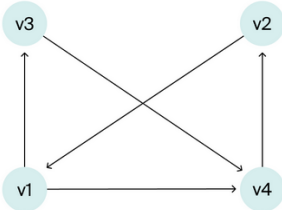
Проверяемые компетенции	Задание	Варианты ответов	Тип сложности
ПК-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1 ОПК-5.2 ОПК-5.3	1) Чтобы алгоритм бинарного поиска работал правильно, нужно, чтобы массив (список) был:	1) Отсортированным 2) В куче 3) Несортированным 4) Выходящим из стека	низкий
ПК-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1 ОПК-5.2 ОПК-5.3	2) Определите максимальное количество узлов в двоичном дереве с высотой k , где корень — нулевая высота (0)	1) $2^k - 1$ 2) $2^{k-1} + 1$ 3) $2^{k+1} - 1$ 4) $2^k + 1$	низкий
ПК-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1 ОПК-5.2 ОПК-5.3	3) Что означает следующая фраза: «алгоритм X асимптотически более эффективен, чем Y »?	1) X будет лучшим выбором для всех входов 2) X будет лучшим выбором для всех входов, за исключением, возможно, небольших входов 3) X будет лучшим выбором для всех входов, кроме больших входов 4) Y будет лучшим выбором для небольших входов	низкий
ПК-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1	4) Алгоритм обхода графа отличается от алгоритма обхода вершин дерева тем, что...	1) Деревья не соединяются 2) Графы могут иметь циклы 3) У деревьев есть корни 4) Все утверждения выше ошибочны: дерево — подмножество графа	низкий

ОПК-5.2 ОПК-5.3			
ПК-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1 ОПК-5.2 ОПК-5.3	5) Какой алгоритм из нижеперечисленных будет самым производительным, если дан уже отсортированный массив?	1) Сортировка слиянием 2) Сортировка вставками 3) Быстрая сортировка 4) Пирамидальная сортировка	низкий
ПК-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1 ОПК-5.2 ОПК-5.3	6) Какой алгоритм реализует следующая функция на языке C++? <pre>int Trial (int a, int b, int c) { if ((a >= b) && (c < b)) return b; else if (a>=b) return Trial(a, c, b); else return Trial(b, a, c); }</pre>	1) Находит максимальное значение a, b и c 2) Находит минимальное значение a, b и c 3) Находит среднее число a, b и c 4) Ничего из вышеперечисленного	средний
ПК-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1 ОПК-5.2 ОПК-5.3	7) Какая структура данных обеспечивает эффективность добавление в начало, имеющую сложность O(1) (выберите три подходящие варианта ответов)	1) связный список 2) стек 3) очередь 4) дерево	средний
ПК-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1 ОПК-5.2	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>	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

ОПК-5.3	причем асимптотическая сложность $f(N, other)$ составляет $O(N)$. Асимптотическая сложность алгоритма равна (выберите два подходящие варианта ответов)		
ПК-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1 ОПК-5.2 ОПК-5.3	9) Алгоритм Дейкстры основан на:	1) Парадигме «разделяй и властвуй» 2) Динамическом программировании 3) Жадном подходе 4) Поиске с возвратом	средний
ПК-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1 ОПК-5.2 ОПК-5.3	10) Какие алгоритмы не основан на жадном подходе?	1) Алгоритм нахождения кратчайшего пути Дейкстры 2) Алгоритм Прима 3) Алгоритм Крускала 4) Алгоритм Хаффмана 5) Алгоритм сортировки слиянием 6) Алгоритм быстрой сортировки 7) Алгоритм нахождения кратчайшего пути Беллмана-Форда	средний
ПК-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1 ОПК-5.2 ОПК-5.3	11) Рассмотрите подпрограмму ниже и определите её сложность: <pre>void function(int n) { int i, j, count=0; for (i=n/2; i <= n; i++) for (j = 1; j <= n; j = j*2) count++; }</pre>	1) $O(\log N)$ 2) $O(N^2)$ 3) $O(N^2 \log N)$ 4) $O(N \log N)$	средний
ПК-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3	12) Какие недостатки присуще методу цепочек для разрешения коллизий хеш-функции?	1) Недостатков нет. 2) При добавлении элемента нужно выделять дополнительную память, при удалении — освобождать.	средний

ОПК-5.1 ОПК-5.2 ОПК-5.3		3) Цепочки данных в памяти компьютера расположены не последовательно, а в произвольных местах.	
ПК-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1 ОПК-5.2 ОПК-5.3	13) Какие недостатки присуще методу открытой адресации для разрешения коллизий хеш-функции?	1) Недостатков нет, при использовании такого метода можно эффективно проводить операции поиска, вставки и удаления. 2) При увеличении количества ячеек, помеченных ярлыком «deleted», скорость выполнения операций будет замедляться. 3) Данные в памяти компьютера расположены не последовательно, а в произвольных местах.	средний
ПК-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1 ОПК-5.2 ОПК-5.3	14) Дан граф:  В каком порядке будут посещены вершины алгоритмом поиска в глубину, если начать из v1?	1) v1, v3, v4, v2 2) v1, v4, v2, v3 3) v1, v2, v3, v4 4) v1, v4, v3, v2	средний
ПК-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1 ОПК-5.2 ОПК-5.3	15) Какой алгоритм обхода в глубину эффективней — тот, что хранит список посещённых вершин, или тот, что хранит массив цветов?	1) Они одинаково эффективны. 2) Алгоритм, хранящий список посещённых вершин, эффективнее по времени. 3) Алгоритм, хранящий массив цветов, эффективнее по времени.	средний
ПК-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3	16) Дан граф:	1) v1, v2, v3, v4 2) v1, v3, v4, v2 3) v1, v4, v3, v2 4) v1, v3, v4, v2, v1	высокий

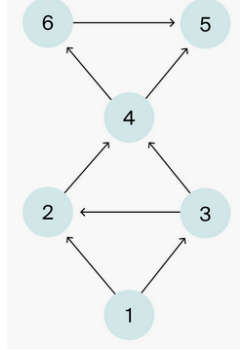
ОПК-5.1
ОПК-5.2
ОПК-5.3



Каким будет порядок обхода для алгоритма поиска в ширину, если начать с вершины v1?

ПК-2.3
ПК-4.3
ОПК-2.1
ОПК-2.2
ОПК-2.3
ОПК-5.1
ОПК-5.2
ОПК-5.3

17) Каким может быть порядок обхода BFS для этого графа при стартовой вершине 1?

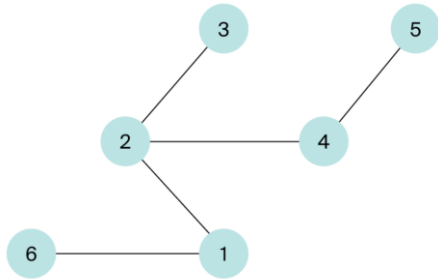


- 1) [1, 2, 3, 4, 6, 5]
- 2) [1, 2, 3, 4, 5, 6]
- 3) [1, 3, 2, 6, 5, 4]
- 4) [1, 3, 2, 4, 5, 6]
- 5) [1, 3, 2, 4, 6, 5]
- 6) [1, 3, 2, 5, 4, 6]

высокий

ПК-2.3
ПК-4.3
ОПК-2.1
ОПК-2.2
ОПК-2.3
ОПК-5.1
ОПК-5.2
ОПК-5.3

18) Каким может быть порядок обхода BFS для этого графа при стартовой вершине 1?



- 1) [1, 2, 3, 4, 5, 6]
- 2) [1, 2, 3, 4, 6, 5]
- 3) [1, 2, 6, 3, 4, 5]
- 4) [1, 2, 6, 4, 3, 5]
- 5) [1, 6, 2, 4, 3, 5]
- 6) [1, 6, 2, 4, 5, 3]

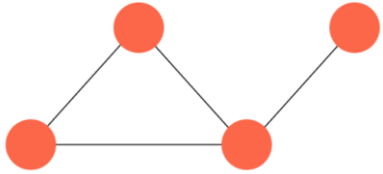
высокий

ПК-2.3
ПК-4.3

19) Сколько разных остовных деревьев у этого графа?

—

высокий

<p>ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1 ОПК-5.2 ОПК-5.3</p>			
<p>ПК-2.3 ПК-4.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-5.1 ОПК-5.2 ОПК-5.3</p>	<p>20) Если поиск и извлечение минимального по весу ребра происходит из массива, то какая у алгоритма Прима сложность?</p>	<p>1) $O(V ^2 \log E)$ 2) $O(V \cdot E)$ 3) $O(V + E)$ 4) $O(E + V \log V)$</p>	<p>высокий</p>