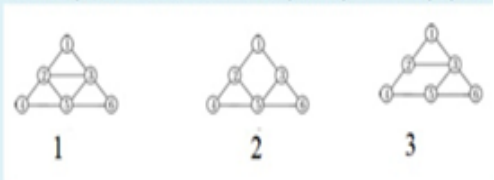
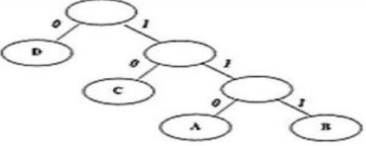


№	Задание	Правильный ответ	Вариант ответов
1.	В результате применения алгоритма RLE был получен сжатый текст 9A-4BCAB7C какой коэффициент сжатия этого текста обеспечил алгоритм RLE	2	0.7 0.5 2 1
2.	В результате применения алгоритма RLE был получен сжатый текст 9A-4BCAB7C какой коэффициент сжатия этого текста обеспечил алгоритм RLE	Факториал числа n	a. Степень числа n b. n-ое просто число c. факториал числа n d. n-ое число Фибоначчи
3.	Определите теоретическую вычислительную сложность (функцию роста времени) алгоритма: int count = 1; while(count<n) { count = count*2; /* Последовательность шагов программы с временной сложностью O(1) */ */ }	Логарифмическая	. линейная b. константная c. квадратичная d. логарифмическая
4.	Какой зависимостью описывается функция вычислительной сложности алгоритма поиска по бинарному дереву в худшем случае (несбалансированное бинарное дерево):	f(n)=Θ(n)	f(n)=Θ(n log(n)) f(n)=Θ(n ²) f(n)=Θ(n) * f(n)=Θ(log(n))
5.	Какой зависимостью описывается функция вычислительной сложности алгоритма линейного (последовательного) поиска в худшем случае	f(n)=Θ(n)	• f(n)=Θ(log(n)) • f(n)=Θ(n log(n)) • f(n)=Θ(n) * • f(n)=Θ(n ²)
6.	Какой зависимостью описывается функция вычислительной сложности алгоритма быстрой сортировки методом Шелла в наилучшем случае	• f(n)=Θ(n)	• f(n)=Θ(log(n)) • f(n)=Θ(n log(n)) • f(n)=Θ(n) * • f(n)=Θ(n ²)
7.	Какой вычислительной сложностью характеризуется алгоритм "грубой силы" (brute force attack) прямого поиска по образцу?	O(n*m)	a. O(n log m) b. O(n*m) c. O(n+m) d. O(n ²)
8.	Какой из приведенных кодов является префиксным (коды отдельных кодовых слов перечислены через запятую)?	0, 100, 101, 110, 111	a. 1, 12, 31 b. 1, 10, 100, 1000 c. 0, 10, 11, 100, 110 d. 0, 100, 101, 110, 111
9.	Зачем нужен заглавный элемент в двусвязном кольцевом списке	Чтобы указать на первый элемент	a. Для удаления пустых указателей б. Чтобы указать на первый элемент c. Чтобы иметь указатель

			на текущий элемент д. Для связи последнего и первого элемента
10.	Зачем нужен заголовок в двунаправленном циклическом списке?		
11.	При каком значении показателя сбалансированности b (разницы высот правого и левого поддеревьев) необходимо выполнить перестройку AVL-дерева	$ b = 2$	a. $ b = 2$ b. $ b = 1$ c. $[b] = 0$ d. $ b > 2$
12.	При каком значении баланс-фактора b (т.е. разницы высот правого и левого поддеревьев) необходимо выполнить перестройку AVL-дерева?		
13.	Какие из представленных связных неориентированных графов являются Эйлеровыми (содержат цикл Эйлера)? 	1	1 и 3 3 2 1
14.	При создании бинарного дерева поиска ключи поступали в следующей последовательности: 25, 15, 18, 10, 11, 7, 8, 17, 16. Из дерева удалили узел с ключем 18. Узел с каким значением был выбран замещающим?	17	a. 15 b. 18 c. 16 d. 17
15.	Как определить, что функция реализует каскадную (множественную) рекурсию:	Вызов функции по любой из всех возможных ветвей алгоритма встречается более одного раза	a. Вызов функции по любой из всех возможных ветвей алгоритма встречается более одного раза b. Функция вызывает параллельно несколько внешних функций c. Функция вызывает несколько внешних функций по любому из всех возможных ветвей алгоритма d. Функция вызывает последовательно несколько внешних функций
16.	Дана следующая рекурсивная функция: <code>int fib(int n)</code>	7	7 4

	<pre>{ if (n < 2) { return 1; } else { return fib(n - 2) + fib(n - 1); } }</pre> <p>Какова глубина рекурсии при вызове fib(5)</p>		<p>5 3</p>
17.	<p>Для оценки порядка роста функций, описывающих вычислительную сложность алгоритмов, используются асимптотические обозначения (символики) или нотации. Что обозначает запись $f(n)=\Theta(g(n))$:</p>	<p>Множество всех функций, порядок роста которых ограничен сверху и снизу функцией $g(n)$ с точностью до постоянных множителей c_1 и c_2 соответственно</p>	<p>a. Множество всех функций, порядок роста которых при достаточно больших n не меньше (больше или равен) некоторой константы c, умноженной на значение функции $g(n)$ b. Множество всех функций, порядок роста которых не ограничен сверху и снизу функцией $g(n)$ с точностью до постоянных множителей c_1 и c_2 соответственно c. Множество всех функций, порядок роста которых ограничен сверху и снизу функцией $g(n)$ с точностью до постоянных множителей c_1 и c_2 соответственно d. Множество всех функций, порядок роста которых при достаточно больших n не превышает (меньше или равен) некоторую константу c, умноженную на значение функции $g(n)$</p>
18.	<p>Имеется следующее оптимальное кодовое дерево (ОКД) Выберите исходную строку, которая была закодирована в битовый код 111 0 0 10 10 0 110 0 10 0 по алгоритму Хаффмана с использованием приведенного ОКД:</p> 	<p>BDDCCDADCD</p>	<p>a. ABCCDDDDDD b. BDDCCDADCD c. DDDDDCCCBA d. ABDCDCDCDD</p>
19.	<p>Структура хранения данных, в которой элементы явно хранят связи с соседними элементами структуры:</p>	<p>списочная</p>	<p>Массив векторная списочная реляционная</p>

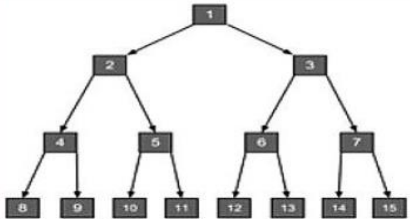
<p>20.</p>	<p>К графу на рисунке 1 применен алгоритм обхода в глубину. Какое из остовных деревьев соответствует обходу в глубину с вершины 3</p>	<p>создается на этапе выполнения программы и размер может быть изменен на этапе выполнения</p>	<p>a. создается на этапе выполнения программы и размер может быть изменен на этапе выполнения</p> <p>b. создается на этапе компиляции кода и размер может быть изменен во время выполнения программы</p> <p>c. создается на этапе компиляции кода и размер не может быть изменен во время выполнения программы</p> <p>d. создается на этапе выполнения программы и размер не может быть изменен на этапе выполнения</p>
<p>21.</p>	<p>Свойство динамической структуры данных (переменной)</p>		
<p>22.</p>	<p>Сколько указателей используется в структуре узла стека ? (Введите только число)</p>	<p>1</p>	
<p>23.</p>	<p>Какое из определений соответствует структуре данных граф ?</p>	<p>Нелинейная, многосвязная, динамическая структура</p>	<p>a. Иерархическая, динамическая структура данных</p> <p>b. Линейная, динамическая структура данных</p> <p>c. Нелинейная, многосвязная динамическая структура данных</p> <p>d. Нелинейная, статическая, упорядоченная структура данных</p>
<p>24.</p>	<p>Какое из перечисленных авл деревьев требует балансировки</p>	<p>3</p>	<p>1</p> <p>4</p> <p>3</p> <p>2</p>
<p>25.</p>	<p>Как называется алгоритм, который выполнит сортировку исходного массива (3,1,5,2,4) после следующей последовательностью проходов</p>	<p>Простого обмена</p>	<p>Простого выбора</p> <p>Шелла</p> <p>Простой вставки</p> <p>Простого обмена</p>

	(1,3,5,2,4), (1,3,5,2,4), (1,3,2,5,4), (1,3,2,4,5), (1,3,2,4,5), (1,2,3,4,5), (1,2,3,4,5), (1,2,3,4,5)		
26.	Укажите правильную аналогию между асимптотическим сравнением двух функций f и g для обозначения $f(n)=\Omega(g(n))$ и сравнением двух действительных чисел a и b :	$a \geq b$	a. $a = b$ b. $a \leq b$ c. $a > b$ d. $a \geq b$
27.	какой зависимостью описывается функция вычислительной сложности алгоритма поиска ключа в бинарном дереве поиска в наихудшем случае?	• $f(n)=\Theta(n)$	• $f(n)=\Theta(n \log(n))$ • $f(n)=\Theta(1)$ • $f(n)=\Theta(n)$ • $f(n)=\Theta(\log(n))$
28.	Из числа приведенных сортировок выберите наименее эффективную на больших массивах:	Пузырьковая	Пузырьковая Вставками Выбором Пирамидальная
29.	Дисциплина обслуживания (порядок выполнения операций) в очереди	FIFO	LIFO LOFI OFIL FIFO
30.	Какая вычислительная сложность характерна для алгоритмов со стратегией "разделяй и властвуй" (divide and conquer approach) ?	$O(n \log n)$	$O(n)$ $O(n \log n)$ $O(n^2)$ $O(1)$
31.	Что будет результатом применения алгоритма RLE ("групповое кодирование") к сжатию текста AAAAAAAAAAAAAAAAAAAAABBBBBBAAAB BBBBBVC	19A5B3A7B1C	a. 19A5B3A7B1C b. ABC(19,5,3,7,1) c. (22)A(12)B(1)C d. ABABC(19,5,3,7,1)
32.	Метод сортировки считается устойчивым, если	относительное расположение элементов с равными ключами не изменяется	a. относительное расположение элементов с равными ключами не изменяется б. начальная отсортированность массива не важна с. время доступа к значению константное д. относительное расположение элементов с равными ключами всегда изменяется
33.	Прямой доступ к элементу данных в памяти делает возможным:	адрес элемента в ОЗУ	О а. адрес элемента в ОЗУ О Б. любая списочная структура Э с. идентификатор элемента в программе О д. порядковый номер элемента в структуре

34.	Какой зависимостью описывается функция вычислительной сложности алгоритма сортировки методом простого обмена (с условием Айверсона) в наилучшем случае	$f(n)=O(n)$	<p>a. $f(n) = O(n^2)$</p> <p>b. $f(n) = O(n \log(n))$</p> <p>c. $f(n) = O(n)$</p> <p>d. $f(n) = O(\log(n))$</p>
35.	В результате применения алгоритма RLE был получен сжатый текст 9A-4BCAB7C. Какой текст был сжат этим алгоритмом?	AAAAAAAAABCVBCCCCC C	<p>a. AAAAAAAAAABCVBCCCCC</p> <p>b. AAAAAAAAAABCVBCCCCC</p> <p>c. (9)A-BCAB(7)C</p> <p>d. AAAAAAAAAABCVBVCABV CACCCCCC</p>
36.	Что помещается в стек рекурсивных вызовов при вызове рекурсивной функции?	Адрес точки возврата, значения всех переменных функции текущего вызова	<p>a. Имена локальных переменных и параметров функции</p> <p>b. Описание всех переменных функции при текущем вызове</p> <p>c. Значения всех переменных функции при текущем вызове для возврата в точку вызова</p> <p>d. Адрес точки возврата, значения всех переменных функции текущего вызова</p>
37.	Каков максимальный размер кодового слова, полученного алгоритмом Шеннона-Фано, для текста, частота появления символов в котором определяется таблицей: а-11, и-8, пробел - 6, л-4, п-3, с-3, м-2, ы-2, н-1, ш-1, ч-1, к-1, у-1, т-1, ь-1?	6	<p>7</p> <p>4</p> <p>6</p> <p>5</p>
38.	Какой формулой определяется максимальное количество узлов в бинарном дереве высотой k (у корня высота 0)?	$2^{(k+1)} - 1$	<p>a. $2^k + 1$</p> <p>b. $2^{(k-1)} + 1$</p> <p>c. $2^k - 1$</p> <p>d. $2^{(k+1)} - 1$</p>
39.	Какие основные операции следует учитывать при оценке временной (вычислительной) сложности алгоритмов сортировки:	Операции сравнения и перемещения данных	<p>a. Все имеющиеся операции в алгоритме</p> <p>b. Операции сравнения и перемещения данных</p> <p>c. Операции сравнения данных</p> <p>d. Операции перемещения данных</p>
40.	Что обеспечивает метод пробирования "двойное хеширование" при вставке в хеш-таблицу, разрешающей коллизии по методу открытого адреса?	Вычисление смещения, зависящее от значения ключа	<p>a. Вычисление смещения, не зависящее от значения ключа</p> <p>b. Вычисление смещения, зависящее от значения ключа</p> <p>c. Длину первичного кластера</p>

	$f(n)=O(g(n))$ и сравнением двух действительных чисел a и b :		$a \geq b$ $a < b$
50.	Как называется алгоритм для нахождения кратчайших путей от одной из вершин графа до всех остальных	Алгоритм Дейкстры	Алгоритм Крускала Алгоритм Флойда-Уоршала Алгоритм Дейкстры Алгоритм Йены
51.	Как называется алгоритм для нахождения кратчайшего пути между двумя вершинами графа		
52.	Система байтового кодирования символов ASCII кодирует символы входной последовательности кодами:	фиксированной длины в один байт	хеш-кодами переменной длины до одного байта фиксированной длины в один байт переменной длины свыше одного байта
53.	Структура хранения, в которой элементы данных физически размещаются в последовательных ячейках памяти строго один за другим:	векторная или вектор	сетевая списочная иерархическая векторная
54.	Структура хранения данных, в которой элементы (ячейки) физически размещаются в последовательных ячейках памяти один за другим		
55.	Время выполнения рекурсивной функции $f(n)$ в наихудшем случае определяется рекуррентным соотношением $T(n) = \begin{cases} \theta(1) & \text{при } n = 1 \\ 2T\left(\frac{n}{2}\right) + \theta(n) & \text{при } n > 1 \end{cases}$	$O(n \log 2n)$	$O(\log 2n)$ $O(n)$ $O(2n)$ $O(n \log 2n)$
56.	Основное требование, предъявляемое к массиву для возможности выполнения двоичного поиска:	Упорядоченность массива	Статический массив Массив небольшого размера Массив целочисленных значений Упорядоченность массива
57.	Для структуры данных очередь справедливо:	Удаление элемента возможно с одной из сторон списка	Доступ возможен к произвольному элементу Удаление элемента возможно с одной из сторон списка Удаление элемента возможно из начала и из конца списка Вставка нового элемента возможна в начало и в конец списка

58.	Укажите свойство, характеризующее структуру данных Очередь:		
59.	Чему равна сумма степеней вершин графа?	Удвоенному числу его ребер	Сумме веса ребер Удвоенному весу ребер Удвоенному числу его ребер Числу его ребер
60.	Массив длиной n сортируется пузырьковым алгоритмом по неубыванию. За сколько проходов по массиву наименьшее значение массива окажется "вверху"?	Один	n n-1 n^2 Один
61.	Какой зависимостью описывается функция вычислительной сложности алгоритма поиска в тексте по образцу Бойера-Мура в лучшем случае?	$f(n, m) = \Theta(\log(n/m))$ На самом деле $\Omega(n/m)$	<input type="radio"/> a. $f(n, m) = \Theta(\log(n*m))$ <input type="radio"/> b. $f(n) = \Theta(n)$ <input type="radio"/> c. $f(n, m) = \Theta(\log(n+m))$ <input checked="" type="radio"/> d. $f(n, m) = \Theta(\log(n/m))$
62.	Какой зависимостью описывается функция вычислительной сложности алгоритма поиска в тексте по образцу методом Бойера-Мура в лучшем случае:		
63.	Имеется указатель q на узел в середине линейного односвязного списка со следующей структурой узла: <pre>struct Tnode { Tdata data; Tnode* next; }</pre> Требуется вставить новый узел (узел содержит данные), ссылку на который хранит указатель qq, в позицию, в которой находится узел q. Какую последовательность операторов необходимо выполнить, чтобы корректно выполнялась данная операция вставки?	qq->next=q->next; q->next=qq; swap(qq->data, q->data);	<pre>a. (*qq)=(*q); b. qq->next=q->next; q->next=qq; swap(qq->data, q->data); c. qq=q; d. q->next=qq; qq->next=q; swap(qq->data, q->data);</pre>
64.	Есть ли ошибка в коде функции f, которая осуществляет получение ссылки на последний узел линейного односвязного списка со структурой узла? <pre>struct Tnode(Tdata data; Tnode* next;); Tnode *f(Tnode *L) { Tnode *q=L; while (q) { q=q->next; } return q; }</pre>	Ошибка в условии while(q) - должно быть while (q->next)	Ошибка в записи оператора q=q->next Ошибка в записи оператора Tnode *q=L - должно быть Tnode q=L Ошибки нет Ошибка в условии while(q) - должно быть while (q->next)

65.	В чем смысл анализа алгоритма	Предсказать требуемые для его выполнения вычислительные ресурсы	Определить требования к интерфейсу Предсказать требуемые для его выполнения вычислительные ресурсы Предсказать форматы структур данных и внешние связи между ними Обеспечить безопасность данных
66.	В чем заключается анализ алгоритма		
67.	Что означает утверждение, что алгоритм X асимптотически более эффективен, чем алгоритм Y ?	X будет лучшим выбором для любых входных данных, начиная с некоторого граничного значения	X будет лучшим выбором для любых входных данных, начиная с некоторого граничного значения X будет лучшим выбором для любых входных данных, кроме очень больших X будет лучшим выбором для любых входных данных X будет лучшим выбором для любых входных данных до некоторого граничного значения
68.	В какой последовательности располагаются вершины дерева при прямом обходе (preorder) ? 	а. 1, 2, 4, 8, 9, 5, 10, 11, 3, 6, 12, 13, 7, 14, 15	а. 1, 2, 4, 8, 9, 5, 10, 11, 3, 6, 12, 13, 7, 14, 15 б. 8, 4, 9, 2, 10, 5, 11, 1, 12, 6, 13, 3, 14, 7, 15 в. 8, 9, 4, 10, 11, 5, 2, 12, 13, 6, 14, 15, 7, 3, 1 г. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
69.	Какой алгоритм сортировки из перечисленных ниже будет самым эффективным на уже отсортированном массиве ? Пирамидальная Слиянием Вставками Быстрая сортировка	Вставками	Пирамидальная Слиянием Вставками Быстрая сортировка
70.	Величина сдвига при поиске образца в тексте определяется наибольшим значением из трех, вычисленных, соответственно, по трем правилам (эвристикам) в алгоритме	Бойера-Мура с турбосдвигом	Кнута-Морриса-Прата Бойера-Мура с турбосдвигом Бойера-Мура Бойера-Мура-Хорспула

71.	Идея алгоритма интерполяционного поиска основана на:	Выборе новой области поиска по расстоянию между ключом и текущим значением элемента	Сравнении каждого элемента с искомым Использование хеш-таблицы Учете знака разности между ключом и текущим значением элемента Выборе новой области поиска по расстоянию между ключом и текущим значением элемента
72.	Укажите свойство, характеризующее структуру данных Дек	Обладает двумя первыми элементами	Удаление элемента возможно из заданной позиции Вставка нового элемента возможна в заданную позицию Линейный список с заданным количеством узлом Обладает двумя первыми элементами
73.	Каков главный недостаток хеш-таблицы	Фиксированный размер, при неизвестном окончательном числе элементов	В отказе на повторное хеширование Фиксированное число элементов, располагающихся в таблице В трудоемкой функции хеширования Фиксированный размер, при неизвестном окончательном числе элементов
74.	Бинарное дерево - это	Дерево, у которого каждый узел может содержать до двух дочерних узлов	Дерево, у которого каждый узел может содержать до двух дочерних узлов дерево, элементы которого являются двоичными числами дерево, у которого каждый узел должен содержать два дочерних узла дерево, у которого каждый узел содержит до двух различных значений
75.	Алгоритм Флойда-Уоршала - это	Алгоритм для нахождения кратчайших путей между всеми вершинами взвешенного ориентированного графа	Алгоритм для нахождения кратчайших путей от одной из вершин графа до всех остальных Алгоритм поиска заданного пользователем количества

			<p>путей между двумя вершинами во взвешенном графе</p> <p>Алгоритм для нахождения кратчайших путей между всеми вершинами взвешенного ориентированного графа</p> <p>Алгоритм поиска минимального остовного дерева во взвешенном неориентированном связном графе</p>
76.	Какие поля должны содержать записи хеш-таблицы при использовании метода цепочек (цепного хеширования)	Указатель на начало списка элементов с одним и тем же значением хеш-функции	<p>Указатель на начало списка элементов с одним и тем же значением хеш-функции</p> <p>Адрес следующего элемента списка</p> <p>Ключ элемента</p> <p>Значение хеш-функции</p>
77.	Какой зависимостью описывается функция вычислительной сложности алгоритма быстрой сортировки методом Хоара (quicksort) в наихудшем случае	$O(n^2)$	<p>$O(n \log n)$</p> <p>$O(\log n)$</p> <p>$O(n^2)$</p> <p>$O(n^3)$</p>
78.	Как называется алгоритм, который выполнит сортировку исходного массива (3,1,5,2,4) следующей последовательностью проходов (3,1,4,2,5), (3,1,2,4,5), (2,1,3,4,5), (1,2,3,4,5)	Простого выбора	<p>Шелла</p> <p>Простого обмена</p> <p>Простой вставки</p> <p>Простого выбора</p>
79.	Какой зависимостью описывается функция вычислительной сложности алгоритма поиска в глубину на списке смежных вершин, если V - количество вершин в графе, а E - количество ребер	$O(V^2)$ - В худшем случае Но вообще $O(V + E)$ https://www.simplilearn.com/tutorials/data-structure-tutorial/dfs-algorithm	<p>$O(\log(E))$</p> <p>$O(V^2)$</p> <p>$O(V)$ (ТОЧНО НЕ ВЕРНЫЙ)</p> <p>$O(E)$</p>
80.	Какие отношения между элементами неявно поддерживает структура данных массив	Линейные	<p>Нелинейные</p> <p>Наследование</p> <p>Линейные</p> <p>Иерархические</p>
81.	Для структуры данных дек справедливо:	Вставка нового элемента возможна в начало или в конец списка	<p>Удаление элемента возможно из произвольной позиции</p> <p>Вставка нового элемента возможна в начало или в конец списка</p> <p>Вставка нового элемента</p>

			возможна только в начало списка Линейный список с произвольным количеством вершин
82.	Какая структура данных относится к категории линейных списков?	Дек	Бинарное дерево Множество Дек Массив
83.	Какой зависимостью описывается функция вычислительной сложности алгоритма сортировки методом простой вставки в наилучшем случае:	O(n)	O(n) O(n ²) O(logn) O(nlogn)
84.	Что будет результатом применения алгоритма RLE - "Групповое кодирование" к сжатию текста AAAAAAAAAAAAAAAAAAAAABCABCACCCCC C?	18A-6BCABCA6C	<input type="radio"/> a. A(18)BCA(2)C(6) <input type="radio"/> b. 18A-2BCA6C <input type="radio"/> c. A18ABCABC6C <input checked="" type="radio"/> d. 18A-6BCABCA6C
85.	Дан рекурсивный алгоритм: int F(int n) { if (n > 2) return F(n-1)+F(n-2)+F(n-3) else return n; } Найдите значение F(5). (В поле ответа - только число)	11	
86.	Выберите код, являющийся результатом сжатия по методу скользящего окна текста "abacabacabadaca". Примечание: Кавычки ограничивают текст и не кодируются.	(0,0,a)(0,0,b)(2,1,c)(4,7,d)(2,1,c)(2,1,пусто) (Первый вариант ответа)	<input checked="" type="radio"/> a. (0,0,a)(0,0,b)(2,1,c)(4,7,d)(2,1,c)(2,1,пусто) <input type="radio"/> b. (0,0,a)(0,0,b)(2,1,c)(3,7,d)(2,1,c)(2,1,пусто) <input type="radio"/> c. (0,0,a)(0,0,b)(2,1,c)(4,4,d)(2,1,c)(2,1,пусто) <input type="radio"/> d. (0,0,a)(0,0,b)(2,1,c)(4,7,d)(2,1,c)(4,1,пусто)
87.	Имеется бинарное дерево поиска, содержащее целые числа от 1 до 7. Каким будет результат вывода при обратном обходе дерева (postorder)	7,6,5,4,3,2,1	7,6,5,4,3,2,1 4.1.2.3.5.6.7 4,1,5,2,6,3,7 1,2,3,4,5,6,7
88.	Имеется бинарное дерево поиска, содержащее целые числа от 1 до 7. Каким будет результат вывода при обходе дерева снизу вверх (обратный/postorder обход)		
89.	В теории вычислимости важную роль играет функция Аккермана A(m,n), определенная следующим образом:	7	

	<p>Вычислите значение $A(2,2)$. (Введите только число)</p> $A(m, n) = \begin{cases} n + 1 & m = 0 \\ A(m - 1, 1) & m > 0, n = 0 \\ A(m - 1, A(m, n - 1)) & m > 0, n > 0 \end{cases}$		
90.	Какая функция реализует линейную рекурсию:	Содержит один вызов самой себя	<p>Содержит несколько вызовов самой себя</p> <p>Содержит один вызов самой себя</p> <p>Содержит линейную функцию</p> <p>Количество вызовов самой себя определяется линейной функцией</p>
91.	Как называется алгоритм поиска минимального остовного дерева во взвешенном неориентированном связном графе	Алгоритм Крускала	<p>Дейкстры</p> <p>Крускала</p> <p>Белмана-Форда</p> <p>Флойда-Уоршала</p>
92.	<p>Для выполнения какой операции в линейном двусвязном списке необходимо выполнить два "подготовительных" оператора:</p> <p>$q \rightarrow prev \rightarrow next = q \rightarrow next$;</p> <p>$q \rightarrow next \rightarrow prev = q \rightarrow prev$;</p> <p>($q$ - указатель на крайний узел списка)</p>	Удаление узла по указателю q	<p>Удаление узла по указателю q</p> <p>Вставка нового узла по указателю q</p> <p>Вставка нового узла после элемента, ссылку на который хранит указатель q</p> <p>Обмен значениями двух узлов по указателю q</p>
93.	Какое из условий проверяется при определении сбалансированности красно-черного дерева?	Любой путь от корня дерева к листу содержит одно и то же количество черных узлов	<p>Высота левого и правого поддеревьев равны</p> <p>Любой путь от корня дерева к листу содержит одно и то же количество черных узлов</p> <p>Количество красных или черных узлов в левом и правом поддеревьях равны</p> <p>Любой путь от корня дерева к листу содержит одно и то же число кратных узлов</p>
94.	<p>Что делает следующая функция?</p> <pre>int trinity (int a, int b, int c){ if ((a >= b) && (c < b)) return b; else if (a >= b) return trinity (a, c, b); else return trinity (b, a, c) }</pre> <p>Вычисляет максимальное значение Вычисляет среднее значение Ничего из перечисленного Вычисляет минимальное значение</p>	ничего из вышеперечисленного	<p>Вычисляет максимальное значение</p> <p>*Вычисляет среднее значение</p> <p>Ничего из перечисленного</p> <p>Вычисляет минимальное значение</p>
95.	Какие из приведенных строк кода <code>int fib(int n){</code>	return fib(n - 1) + fib(n - 2)	<p>return (n-1) + (n-2)</p> <p>fib (n - 1) + fib (n - 2)</p>

	<pre>if (n < 2) return 1; else {...} }</pre> <p>необходимо записать после else в рекурсивной функции вычисления n-ного числа Фибаначи</p>		<pre>return fib ((n-1) + (n-2)) return fib(n - 1) + fib(n - 2)</pre>
96.	имеется бинарное дерево поиска, содержащее целые числа. Просмотр дерева дает следующий результат: 2, 4, 6, 8, 10, 12, 14. Какой способ обхода дерева при этом используется?	Симметричный	В ширину Обратный Прямой Симметричный
97.	как называется алгоритм, который выполнит сортировку исходного массива (3,1,5,2,4) следующей последовательностью проходов (1,3,5,2,4), (1,3,5,2,4), (1,2,3,5,4), (1,2,3,4,5)	Простой вставки	Простого выбора Простого обмена Простой вставки Пирамидальная
98.	В AVL-дерево, содержащее ключи 8,10,11,14, был вставлен ключ 12. Потребовалась ли перестройка дерева и если да, то каким способом?	да, двойной LR	нет да, двойной LR да, одинарный правый да, двойной RL
99.	Росту первичного кластера в хеш-таблице с открытым адресом способствует:	Смещение и размер таблицы имеют общие множители	Ключи равномерно распределены по таблице Размер таблицы определен как простое число и смещение простое число Для большого количества ключей хеш-функция сформировала один и тот же индекс Смещение и размер таблицы имеют общие множители
100.	Отличительная особенность алгоритма Бойера-Мура:	Сравнение символов производится начиная с конца образца;	Сравнение символов производится начиная с конца текста После каждого неудачного сравнения производится сдвиг образца вправо на одну позицию Производится посимвольное сравнение образца с текстом при равенстве хешей Сравнение символов производится начиная с конца образца;
101.	какой зависимостью описывается функция вычислительной сложности	$\Theta(n \log n)$	$\Theta(n \log n)$ $\Theta(\log n)$

	алгоритма турнирной сортировки в наихудшем случае:		$\Theta(n)$ $\Theta(n^2)$
102.	Какой зависимостью описывается функция вычислительной сложности алгоритма турнирной сортировки в среднем и наихудшем случаях		
103.	Отличительная особенность алгоритма Бойера-Мура-Хорспула:	После каждого неудачного сравнения производится сдвиг образца вправо в соответствии с эвристикой стоп-символов ("плохих" символов)	<p>После каждого неудачного сравнения производится сдвиг образца вправо на одну позицию</p> <p>После каждого неудачного сравнения производится сдвиг образца вправо в соответствии с эвристикой стоп-символов ("плохих" символов)</p> <p>После каждого неудачного сравнения производится сдвиг образца вправо на количество позиций, равное количеству предшествующих удачных сравнений</p> <p>После каждого неудачного сравнения производится сдвиг образца вправо в соответствии с таблицей префиксов (префикс-функцией)</p>
104.	Известно, что при построении хеш-таблиц возможно появление коллизий. Коллизия это такая ситуация, когда:	для разных ключей хеш-функция может принимать одно и тоже значение $h(k_i) = h(k_j)$	<p>Способ нахождения промежуточных значений определяется величиной по имеющемуся дискретному набору значений</p> <p>Определение области поиска производится с помощью учета знака разности между ключом и текущим значением элемента</p> <p>Для одинаковых ключей хеш-функция может принимать разные значения $h(k_i) \neq h(k_j)$</p> <p>для разных ключей хеш-функция может принимать одно и тоже значение $h(k_i) = h(k_j)$</p>

105.	какой код будет получен из входной строки символов 1111223444 при использовании алгоритма группового кодирования RLE (Run Length Encoding)	1 4 2 2 3 1 4 3 - ЭТО ПРАВИЛЬНЫЙ ПО СДО 4 1 2 2 1 3 3 4 - ЭТО ПРАВИЛЬНЫЙ НА САМОМ ДЕЛЕ	1 4 2 2 3 1 4 3 (4)1 (2)2 (1)3 (3)4 4 1 2 2 1 3 3 4 0 0 0 0 01 01 10 11 11 11
106.	Какой зависимостью описывается функция вычислительной сложности алгоритма быстрой сортировки методом Хоара (quicksort) в среднем и наилучшем случаях ?	$\Theta(n \log n)$	$\Theta(n \log n)$ $\Theta(\log n)$ $\Theta(n^2)$ $\Theta(n)$
107.	Показатели эффективности алгоритмов (или программ) включают:	Количество выполняемых операций и объем требуемой памяти	Количество уровней подзадач при декомпозиции основной задачи Корректность и достоверность программы Количество выполняемых операций и объем требуемой памяти Размер программы (количество операторов)
108.	Укажите асимптотическую сложность операции удаления элемента, на который имеется указатель (например, ptr), из однонаправленного списка размером n	$O(1)$	$O(1)$ $O(n^2)$ $O(n)$ $O(\log n)$
109.	какая проблема может возникнуть в хеш-таблице с открытым адресом, после удаления ключа из первичного кластера ключей, хешированных с одним индексом?	Невозможность найти ключ в первичном кластере	Невозможность найти ключ в первичном кластере Ключи поменяют индексы, с которыми были хешированы при вставке в таблицу Невозможно вставить ключ в первичный кластер Неоднозначность результата поиска ключа
110.	Высота (глубина) бинарного дерева поиска из N узлов в наилучшем случае определяется по формуле:	$h = \log_2(N+1)-1$	$h = \log_2(N+1)$ $h = \log_2 N$ $h = \log_2(N+1)-1$ $h = N$
111.	Определите теоретическую вычислительную сложность (функцию роста времени) алгоритма: for (int i = n/2; i < n; i++) for (int j = 1; j < n; j = j*2) cout << "hello"	линейно-логарифмическая	a. линейно-логарифмическая b. логарифмическая c. квадратичная d. линейная
112.	Определите теоретическую вычислительную сложность (функцию роста времени) алгоритма: int count = 1;	Логарифмическая	

	<pre>while (count < n) { count = count * 2; /* Последовательность шагов программы с временной сложностью O(1) */ }</pre>		
113.	<p>Линейными называются структуры данных, в которых:</p>	<p>связи между элементами не зависят от выполнения какого-либо условия</p>	<p>а. связи между элементами зависят от выполнения какого-либо условия б. связи между элементами не зависят от выполнения какого-либо условия в. связи между элементами не зависят от линейной упорядоченности элементов д. связи между элементами зависят от упорядоченности значений элементов</p>
114.	<p>В каком типе деревьев лист не должен содержать значение?</p>	<p>красно-черное дерево</p>	<p>В-дерево бинарное дерево поиска AVL-дерево красно-черное дерево</p>
115.	<p>Какой зависимостью описывается функция вычислительной сложности алгоритма Флойда-Уоршала в графе, построенном на матрице смежности, если V — количество вершин в графе, а E — количество ребер:</p>	<p>$f(n) = O(V^3)$</p>	<p>$f(n) = O(E)$ $f(n) = O(V^3)$ $f(n) = O(V)$ $f(n) = O(\log(E))$</p>
116.	<p>Структура данных, работа с элементами которой организована по принципу FIFO (первый пришел - первый ушел), - это:</p>	<p>очередь</p>	<p>а. дек б. стек в. массив д. очередь</p>
117.	<p>Для оценки порядка роста функций, описывающих вычислительную сложность алгоритмов, используются асимптотические обозначения (символики) или нотации. Что обозначает запись $f(n) = O(g(n))$:</p>	<p>Множество всех функций, порядок роста которых при достаточно больших n не больше (меньше или равен) некоторой константы c, умноженной на значение функции $g(n)$</p> <p>$g(n)$ - это, например, n^3</p>	<p>а. Множество всех функций, порядок роста которых при достаточно больших n не меньше (больше или равен) некоторой константы c, умноженной на значение функции $g(n)$ б. Множество всех функций, порядок роста которых не ограничен сверху и снизу функцией $g(n)$ с точностью до постоянных множителей c_1 и c_2 соответственно</p>

			<p>с. Множество всех функций, порядок роста которых при достаточно больших n не больше(меньше или равен) некоторой константы c, умноженной на значение функции $g(n)$</p> <p>d. Множество всех функций, порядок роста которых ограничен сверху и снизу функцией $g(n)$ с точностью до постоянных множителей c_1 и c_2 соответственно возможно</p>
118.	При создании бинарного дерева поиска ключи поступали в следующей последовательности: 25, 12, 13, 10, 44, 11, 7, 8, 42. Какова высота этого дерева?	Скорее всего, 4 - высота - длина наибольшего пути от листа к корню, а в дереве 5ть уровней	5 3 4 2
119.	Какой зависимостью описывается функция вычислительной сложности алгоритма поиска хешированием в лучшем случае?	$f(n)=O(1)$	$f(n)=O(1)$ $f(n)=O(\log(n))$ $f(n)=O(n\log(n))$ $f(n)=O(n^2)$
120.	Что необходимо делать с большим объемом информации, связанной с каждым ключом:	Не хранить ее в хеш-таблице	<p>a. Не хранить ее в хеш-таблице</p> <p>b. Хранить ее в хеш-таблице</p> <p>c. Чаше рехешировать</p> <p>d. Увеличить хэш-таблицу</p>
121.	Когда достигается максимальная эффективность алгоритма Бойера-Мура?	Если образец длинный, а мощность алфавита достаточно велика	<p>a. Если образец короткий, а мощность алфавита достаточно велика</p> <p>b. Если образец длинный, а мощность алфавита достаточно велика</p> <p>c, Если образец длинный, а мощность алфавита достаточно низка</p> <p>d. Если образец короткий, а мощность алфавита достаточно низка</p>
122.	Какие из перечисленных AVL-деревьев не требуют проведения перестройки?	ответ: 1.2.4	1,3,4 1,2,3 1,2,4 2,3,4

123.	<p>Исходный текст "abcdabcdaaacc" был сжат, в результате был получен код abcd(4,4)(8,1)(1,1)(1,1)(5,1). Какой принцип сжатия текстовой информации при этом использовался?</p>	<p>уменьшение объема текста за счет замены повторно встретившейся цепочки символов ссылкой на ранее встреченную цепочку</p>	<p>а. объема текста за счет формирования словаря кодов цепочек и замены цепочек на код б. уменьшение объема текста за счет замены повторно встретившейся цепочки символов ссылкой на ранее встреченную цепочку с. частотное кодирование д. уменьшение объема текста за счет замены каждой повторно встреченной цепочки символов комбинацией: (количество знаков в цепочке, цепочка символов)</p>
124.	<p>Чем может быть вызвано переполнение стека при выполнении рекурсивной функции с небольшой глубиной рекурсии:</p>	<p>Неправильно сформулированным условием завершения рекурсии</p>	<p>а. Неправильно оформленным выводом результата б. Неправильно сформулированным условием завершения рекурсии с. Неправильно оформленным блоком входа в рекурсию д. Неправильным выполнением вызова функции</p>
125.	<p>Как называется определённая последовательность вычислительных шагов, преобразующих входные величины в выходные?</p>	<p>Алгоритм</p>	<p>а. Множество данных б. Множество допустимых операций над данными с. Алгоритм д. Структура данных</p>
126.	<p>Понятие "глубина рекурсии" для рекурсивной функции определяет:</p>	<p>Наибольшее одновременное количество</p>	<p>а. Количество внешних вызовов функции в программе б. Наибольшее количество</p>

		рекурсивных обращений функции	операторов в рекурсивной функции с. Количество операторов вызова функции в самой функции d. Наибольшее одновременное количество рекурсивных обращений функции
127.	Какой код будет получен из входной строки символов ABCCCDDDDD при использовании алгоритма Хаффмана?	110 111 10 10 10 0 0 0 0 0 0	110 111 10 10 10 0 0 0 0 0 0 1 2 3 3 3 4 4 4 4 4 00 01 10 10 10 11 11 11 11 11 A1 B1 C3 D5
128.	Выберите все характеристики хеш-функции в алгоритме поиска Рабина-Карпа:	<u>Полиномиальный хеш</u> <u>Модульная арифметика</u> <u>кольцевой</u> <u>("скользящий",</u> <u>"летающий")</u>	Отсутствие коллизий Полиномиальный хеш Модульная арифметика кольцевой ("скользящий", "летающий")
129.	Какой из видов линейных списков лучше использовать при реализации задачи по проверке баланса круглых скобок (соответствие открывающей и закрывающей скобок: (()()) - баланс) ?	Стек	Линейный двусвязный список очередь стек линейный односвязный список
130.	какой зависимостью описывается функция вычислительной сложности алгоритма сортировки шейкерным методом (с условием Айверсона) в наихудшем случае:	O(n²)	O(n ³) O(n²) O(logn) O(nlogn)
131.	Префиксный код в теории кодирования - это:	Кодовое слово переменной длины, с которого не может начинаться другое кодовое слово того же кода	Кодовое слово переменной длины, с которого не может начинаться другое кодовое слово того же кода Кодовое слово фиксированной длины, с которого не может начинаться другое кодовое слово Кодовое слово фиксированной длины, с которого может начинаться другое кодовое слово кодовое слово переменной длины, с которого может начинаться другое кодовое слово
132.	Какой алгоритм из перечисленных не основан на жадном подходе?	Алгоритм нахождения кратчайшего пути Беллмана-Форда	a. Алгоритм кодирования Хаффмана b. Алгоритм нахождения кратчайшего пути Дейкстры

			<p>с. Алгоритм построения минимального остовного дерева Крускала</p> <p>d. Алгоритм нахождения кратчайшего пути Беллмана-Форда</p>
133.	Когда один алгоритм считается эффективнее другого:	Если его вычислительная сложность в наихудшем случае имеет более низкий порядок роста	<p>а. Если его вычислительная сложность в наихудшем случае имеет более высокий порядок роста</p> <p>б. Если его вычислительная сложность в наилучшем случае имеет более низкий порядок роста</p> <p>с. Если его вычислительная сложность в наилучшем случае имеет более высокий порядок роста</p> <p>d. Если его вычислительная сложность в наихудшем случае имеет более низкий порядок роста</p>
134.	Для каких узлов необходимо проверить коэффициент балансировки (баланс-фактор) после добавления узла в AVL-дерево?	с. для всех предков нового узла вплоть до корня дерева	<p>а. Если его вычислительная сложность в наихудшем случае имеет более высокий порядок роста</p> <p>б. Если его вычислительная сложность в наилучшем случае имеет более низкий порядок роста</p> <p>с. Если его вычислительная сложность в наилучшем случае имеет более высокий порядок роста</p> <p>d. Если его вычислительная сложность в наихудшем случае имеет более низкий порядок роста</p>
135.	Для каких узлов необходимо проверить коэффициент балансировки (баланс-фактор) после добавления узла в AVL-дерево?	с. для всех предков нового узла вплоть до корня дерева	<p>а. Только для нового добавленного узла</p> <p>б. для родителя нового узла</p> <p>с. для всех предков нового узла вплоть до корня дерева</p> <p>d. для всех узлов дерева</p>
136.	Имеется узел AVL-дерева для которого $ hl-hr =0$, где hl , hr высота левого и правого поддеревя соответственно. В левое поддерево вставляется новый ключ, после этого:	d. не требуется	<p>а. требуется перестройка дерева</p> <p>б. требуется перестройка малым правым поворотом</p> <p>с. требуется перестройка малым левым поворотом</p> <p>d. не требуется перестройка дерева</p>

137.	Сложность рекурсивного алгоритма вычисления N-ого числа Фибоначчи можно снизить с $O(2^n)$ до $O(n)$, т.е. оптимизировать процесс вычисления, если сохранять промежуточные значения решения подзадач. Какой метод построения алгоритма здесь задействован?	b. Динамическое программирование	<p>a. "Жадный" алгоритм</p> <p>b. Динамическое программирование</p> <p>c. Разделяй и властвуй</p> <p>d. Метод рекурсии</p>
138.	Что измеряется для оценки временной (вычислительной) сложности алгоритма:	c. Зависимость количества выполняемых основных операций от объёма входа	<p>a. Зависимость объема требуемой памяти от размера обрабатываемых данных</p> <p>b. Зависимость количества итераций от размера обрабатываемых данных</p> <p>c. Зависимость количества выполняемых основных операций от объёма входа</p> <p>d. Зависимость количества выполняемых основных операторов от размера доступной памяти</p>
139.	При создании бинарного дерева поиска слюжи поступали в следующей последовательности: 25, 15, 18, 10, 11, 7, 8, 17, 16. Из дерева удалили узел с ключом 18. Узел с каким значением был выбран замещающим?	c. 17	<p>a. 16</p> <p>b. 15</p> <p>c. 17</p> <p>d. 18</p>
140.	Какой зависимостью описывается функция вычислительной сложности алгоритма интерполяционного поиска в среднем случае?	$f(n)=O(\log(\log(n)))$	<p>a. $f(n)=O(\log(\log(n)))$</p> <p>b. $f(n)=O(n \log(n))$</p> <p>c. $f(n)=O(n)$</p> <p>d. $f(n)=O(\log(n))$</p>
141.	Имеется линейный односвязный список из n ($n > 2$) узлов. Структура узла списка: <pre>struct Tnode { Tdata data; Tnode* next; }</pre> L - указатель на его начало. Укажите группу операторов, которые обеспечат корректное удаление двух узлов из начала списка L:	c. $q1=L$; $q2=L->next$; $L=L->next->next$; delete $q1$; delete $q2$;	<p>a. $q1=L$; $L=L->next->next$; $q2=L$; delete $q1$; delete $q2$;</p> <p>b. $q1=L$; $q2=L$; $L=q2->next$; delete $q1$; delete $q2$;</p> <p>c. $q1=L$; $q2=L->next$; $L=L->next->next$; delete $q1$; delete $q2$;</p> <p>d. $q1=L$; delete $q1$; $q2=q1$; $L=q2->next$; delete $q2$;</p>
142.	Какой зависимостью описывается функция вычислительной сложности алгоритма поиска в ширину на списке смежных вершин, если V - количество вершин в графе, а E - количество ребер	скорее всего $O(V^2)$, но вообще $O(V+E)$	<p>$f(n)=O(\log(E))$</p> <p>$f(n)=O(V)$</p> <p>$f(n)=O(E)$</p> <p>$f(n)=O(V^2)$ - имеется в виду $O(V^2)$</p>

143.	При создании бинарного дерева поиска ключи поступали в следующей последовательности: 25, 12, 13, 10, 11, 7, 8, 44, 42. В это дерево был вставлен узел с ключом 6. В какое поддерево и какого узла был вставлен узел ?	В левое поддерево узла 7	В левое поддерево узла 42 В левое поддерево узла 13 В левое поддерево узла 11 В левое поддерево узла 7
144.	Дан рекурсивный алгоритм: <pre>void F (int n){ cout << n << endl; if (n < 5) { F(n+1); F(n+2); F(n*3); } }</pre> Найдите сумму чисел, которые будут выделены при вызове F(2). (В поле ответа - только число)	79	
145.	Структура данных массив отображается на физическом уровне в структуру хранения:	Скорее всего вектор Очень маловероятно, но возможно, что список, хотя у него, как и других неправильных вариантов есть связи между элементами, которые отсутствуют в массиве	вектор граф сеть список
146.	Что такое алгоритм:	Последовательность вычислительных шагов, преобразующих входные величины/данные в выходные	<ul style="list-style-type: none"> • Последовательность вычислительных шагов, преобразующих входные величины в выходные * • Любая вычислительная процедура • Последовательность вычислительных шагов, преобразующих выходные величины во входные • Инструмент, предназначенный для постановки вычислительной задачи
147.	Как называется последовательность вычислительных шагов, преобразующих входные величины в выходные:	Алгоритмы	<ul style="list-style-type: none"> • Структуры • Алгоритмы * • Данные • Информация
148.	Что определяет структура данных	Множество данных и отношений между ними Первая презентация Красникова	<ul style="list-style-type: none"> • Множество данных • Множество отношений между данными • Множество данных и

			<p>отношений между ними *</p> <ul style="list-style-type: none"> • Множество данных и множество операций над ними
149.	Какие существуют уровни уровня представления данных (согласно American National Standards Institute - ANSI):	Пользовательский, концептуальный, физический	<ul style="list-style-type: none"> • Внешний, внутренний • Абстрактный, конкретный • Пользовательский, концептуальный, физический • Программный, аппаратный
150.	Когда алгоритм считается корректным	Когда для каждого ввода результатом его работы является корректный вывод	<ul style="list-style-type: none"> • Когда для каждого вывода результатом его работы является корректный вывод • Когда для каждого вывода результатом его работы является корректный ввод • Когда он может выдать ответ, отличный от ожидаемого • Когда для каждого ввода результатом его работы является корректный вывод
151.	Какое требование предъявляется к спецификации алгоритма	<u>Спецификация алгоритма должна предоставлять точное описание процедуры, которую требуется выполнить</u>	<ul style="list-style-type: none"> • Спецификация алгоритма должна предоставлять точное описание процедуры, которую требуется выполнить * • Спецификация алгоритма должна предоставлять примерное описание процедуры, которую требуется выполнить • Спецификация алгоритма должна предоставлять точное вычисление • Спецификация алгоритма должна предоставлять вычисление до сотой составляющей
152.	Как представляются данные в памяти компьютера	В виде последовательности битов	<ul style="list-style-type: none"> • В виде последовательности инструкций • В виде последовательности битов * • В виде последовательности операций

			<ul style="list-style-type: none"> • В виде последовательности операндов
153.	Что называют структурой данных	Множество элементов данных и внутренних связей между ними	<ul style="list-style-type: none"> • Множество элементов данных и внешних связей между ними • Множество алгоритмов и внутренних связей между ними • Множество элементов данных и внутренних связей между ними * • Множество данных алгоритмов и внешних связей между ними
154.	Как называют множество элементов данных и внутренних связей между ними?	Структура данных	<ul style="list-style-type: none"> • Система "клиент-сервер" • Алгоритм • Спецификация алгоритма • Структура данных *
155.	Что такое модель?	Система, исследование которой служит средством для получения информации о другой системе, представление некоторого реального процесса, устройства или концепции	<ul style="list-style-type: none"> • Система, исследование которой служит средством для получения информации о другой системе, представление некоторого реального процесса, устройства или концепции * • Система, исследование которой служит средством для получения информации о другой системе, представление некоторого абстрактного процесса, устройства или концепции • Более эффективная система • Система, в которой один из алгоритмов считается эффективнее другого
156.	Что измеряется для оценки вычислительной сложности алгоритма	• Зависимость количества выполняемых основных операций от размера обрабатываемых данных	<ul style="list-style-type: none"> • Зависимость выполняемых операторов от размера обрабатываемых данных • Зависимость объема требуемой памяти от размера обрабатываемых данных • Зависимость количества

			<p>выполняемых основных операций от размера обрабатываемых данных *</p> <ul style="list-style-type: none"> • Зависимость количества шагов цикла от размера обрабатываемых данных
157.	<p>Временная (вычислительная) сложность алгоритма определяется количеством входных данных. Для простоты входные данные представляются параметром n. Этот параметр пропорционален величине обрабатываемого набора данных и может обозначать</p>	<p>Размер массива или файла</p>	<ul style="list-style-type: none"> • Размер массива или файла* • Количество функций в программе • Размер занимаемой памяти при сортировке или поиске • Количество основных операций
158.	<p>Какие основные операции следует учитывать при оценке временной (вычислительной) сложности алгоритмов поиска:</p>	<ul style="list-style-type: none"> • Операции сравнения * 	<ul style="list-style-type: none"> • Операции сравнения и перемещения • Операции сравнения * • Операции сравнения и сложения • Операции сложения и перемещения
159.	<p>Какие существуют случаи при выполнении алгоритмов:</p>	<ul style="list-style-type: none"> • Наилучший, средний и наихудший 	<ul style="list-style-type: none"> • Общий, частный и оптимальный • Эффективный и неэффективный • Наилучший, средний и наихудший * • Средний и наихудший
160.	<p>Для оценки порядка роста функций, описывающих вычислительную сложность алгоритмов, используются асимптотические обозначения (символики) или нотации. Что обозначает запись $f(n)=\Omega(g(n))$</p>	<p>Множество всех функций, порядок роста которых при достаточно больших n не меньше (больше или равен) некоторой константы c, умноженной на значение функции $g(n)$</p>	<ul style="list-style-type: none"> • Множество всех функций, порядок роста которых при достаточно больших n не меньше (больше или равен) некоторой константы c, умноженной на значение функции $g(n)$ • Множество всех функций, порядок роста которых при достаточно больших n не превышает (меньше или равен) некоторую константу c, умноженную на значение функции $g(n)$ • Множество всех функций, порядок роста которых ограничен сверху и снизу функцией $g(n)$ с точностью до постоянных множителей c_1 и c_2 соответственно

			<ul style="list-style-type: none"> • Множество всех функций, порядок роста которых не ограничен сверху и снизу функцией $g(n)$ с точностью до постоянных множителей c_1 и c_2 соответственно
161.	Для оценки порядка роста функций, описывающих вычислительную сложность алгоритмов, используются асимптотические обозначения (символики) или нотации. Что обозначает запись $f(n)=\Omega(g(n))$		
162.	Укажите правильную аналогию между асимптотическим сравнением двух функций f и g для обозначения $f(n)=\Theta(g(n))$ и сравнением двух действительных чисел a и b	$a = b$	<ul style="list-style-type: none"> • $a \leq b$ • $a \geq b$ • $a = b$ * • $a < b$
163.	Как называется алгоритм, который напрямую или через другие вспомогательные алгоритмы вызывает сам себя:	Рекурсивный	<ul style="list-style-type: none"> • Линейный • Разветвляющийся • Циклический • Рекурсивный *
164.	Что определяет понятие «глубина рекурсии» для рекурсивной функции:	<ul style="list-style-type: none"> • Наибольшее одновременное количество рекурсивных обращений функции * 	<ul style="list-style-type: none"> • Количество вызовов внешних функции • Количество операторов вызова функции в самой функции • Количество операторов в рекурсивной функции • Наибольшее одновременное количество рекурсивных обращений функции *
165.	Что определяет понятие "глубина рекурсии" для рекурсивной функции:		
166.	Рекурсия имеет место, если решение задачи сводится к разделению ее на меньшие подзадачи, выполняемые с помощью одного и того же алгоритма. Когда должен завершиться процесс разбиения задачи на подзадачи	Когда достигается простейшее возможное решение	<ul style="list-style-type: none"> • Когда результат не будет получен • Когда достигается простейшее возможное решение * • Когда возникнет переполнение стека • Когда завершиться ввод исходных данных
167.	Что такое инвариант цикла:	Правильный ответ: Логическое выражение, истинное после каждого прохода тела цикла и перед началом выполнения цикла, зависящее от переменных,	<ul style="list-style-type: none"> • Логическое выражение, истинное после первого прохода тела цикла и в конце выполнения цикла, зависящее от переменных, изменяющихся в теле цикла • Логическое выражение, истинное перед началом, после каждого прохода тела

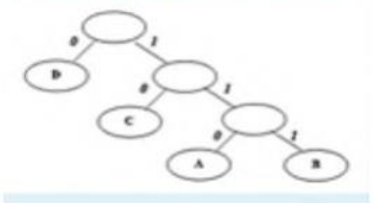
		изменяющихся в теле цикла ПО ИДЕЕ И ПОСЛЕ, И ПЕРЕД	цикла и в конце выполнения цикла, зависящее от переменных, изменяющихся в теле цикла * • Логическое выражение, истинное после каждого прохода тела цикла и перед началом выполнения цикла, зависящее констант, изменяющихся в теле цикла • Логическое выражение, истинное после первого прохода тела цикла и в конце выполнения цикла, зависящее от переменных, изменяющихся в теле цикла
168.	Для чего нужен инвариант цикла	Для проверки корректности алгоритма	<ul style="list-style-type: none"> • Для обеспечения правильного жизненного цикла • Для повышения эффективности алгоритма • Для проверки корректности алгоритма * • Для одновременной работы с несколькими массивами
169.	Если на вход подается последовательность (31,41,59,26,41,58), то какой должен быть вывод алгоритма сортировки:	(26,31,41,41,58,59)	<ul style="list-style-type: none"> • (31,26,41,41,58,59) • (41,31,41,26,58,59) • (58,31,41,41,26,59) • (26,31,41,41,58,59) *
170.	Что проверяет условие Айверсона в алгоритме сортировки методом простого обмена	Наличие обменов в текущем проходе массива	<ul style="list-style-type: none"> • Наличие сравнений и обменов в текущем проходе массива • Количество обменов в текущем проходе массива • Наличие обменов о в текущем проходе массива • Наличие сравнений в текущем проходе массива
171.	Какой зависимостью описывается функция вычислительной сложности алгоритма сортировки подсчетом (Counting sort) в среднем и наилучшем случаях:	$f(n)=O(n+k)$	<ul style="list-style-type: none"> • $f(n)=O(n)$ • $f(n)=O(n^2)$ • $f(n)=O(n+k)$ • $f(n)=O(\log(n))$
172.	Какой зависимостью описывается функция вычислительной сложности алгоритма сортировки шейкерным методом (с условием Айверсона) в наилучшем случае:	$f(n)=O(n)$	<ul style="list-style-type: none"> • $f(n)=O(\log(n))$ • $f(n)=O(n^2)$ • $f(n)=O(n)$ • $f(n)=O(n \log(n))$
173.	Какой зависимостью описывается функция вычислительной сложности алгоритма быстрой сортировки методом Шелла в наихудшем случае:	• $f(n)=O(n^2) *$	<ul style="list-style-type: none"> • $f(n)=O(\log(n))$ • $f(n)=O(n^2)$ • $f(n)=O(n)$ • $f(n)=O(n \log(n))$

174.	Суть алгоритма сортировки методом Шелла заключается в:	<ul style="list-style-type: none"> • отдельной сортировке элементов, отстоящих друг от друга на расстоянии h, уменьшающейся на каждом проходе массива до значения 1 * 	<ul style="list-style-type: none"> • отдельной сортировке элементов, отстоящих друг от друга на расстоянии h, уменьшающейся на каждом проходе массива до значения 1 * • чередовании проходов по сортируемому массиву слева-направо и справа-налево • отдельной сортировке частей массива относительно опорного элемента • попарным сравнением элементов массива с целью выбора максимального из них
175.	Какой зависимостью описывается функция вычислительной сложности алгоритма пирамидальной сортировки (Heapsort) в среднем и наихудшем случаях:	$f(n)=\Theta(n \log(n))$ *	<ul style="list-style-type: none"> • $f(n)=\Theta(n \log(n))$ * • $f(n)=\Theta(\log(n))$ • $f(n)=\Theta(n^2)$ • $f(n)=\Theta(n)$
176.	Алгоритм сортировки подсчетом (counting sort) имеет следующие характеристики эффективности:	Временная (вычислительная) сложность - $f(n)=\Theta(n)$, дополнительная память - $f(k)=\Theta(k)$	<ul style="list-style-type: none"> • Временная (вычислительная) сложность - $f(n)=\Theta(n)$, дополнительная память - $f(n)=\Theta(1)$ • Временная (вычислительная) сложность - $f(n)=\Theta(n \log(n))$, дополнительная память - $f(k)=\Theta(k)$ • Временная (вычислительная) сложность - $f(n)=\Theta(n \log(n))$, дополнительная память - $f(n)=\Theta(n^2)$ • Временная (вычислительная) сложность - $f(n)=\Theta(n)$, дополнительная память - $f(k)=\Theta(k)$ *
177.	Какой зависимостью описывается функция вычислительной сложности алгоритма сортировки слиянием (Mergesort) в среднем и наихудшем случаях	$f(n)=\Theta(n \log(n))$	<ul style="list-style-type: none"> • $f(n)=\Theta(n \log(n))$ • $f(n)=\Theta(\log(n))$ • $f(n)=\Theta(n^2)$ • $f(n)=\Theta(n)$
178.	Какой зависимостью описывается функция вычислительной сложности алгоритма бинарного (двоичного) поиска в среднем и худшем случаях:	$f(n)=\Theta(\log(n))$	$f(n)=\Theta(n \log(n))$ $f(n)=\Theta(\log(n))$ $f(n)=\Theta(n^2)$ $f(n)=\Theta(n)$
179.	Какой зависимостью описывается функция вычислительной сложности алгоритма бинарного (двоичного) поиска в худшем случае:		
180.	Какой зависимостью описывается функция вычислительной сложности	$f(n)=\Theta(n)$	<ul style="list-style-type: none"> • $f(n)=\Theta(n \log(n))$ • $f(n)=\Theta(\log(n))$

	алгоритма интерполяционного поиска в худшем случае:		<ul style="list-style-type: none"> • $f(n)=\Theta(\log(\log(n)))$ • $f(n)=\Theta(n)$
181.	Идея алгоритма бинарного поиска основана на:	делении области поиска на две части	<ul style="list-style-type: none"> • выборе новой области поиска по расстоянию между ключом и текущим значением элемента • делении области поиска на две части • сравнении каждого элемента с искомым • использовании хеш-таблицы
182.	Какой зависимостью описывается функция вычислительной сложности алгоритма поиска хешированием в худшем случае:	<ul style="list-style-type: none"> • $f(n)=\Theta(n)$ 	<ul style="list-style-type: none"> • $f(n)=\Theta(\log(n))$ • $f(n)=\Theta(n)$ • $f(n)=\Theta(n^2)$ • $f(n)=\Theta(1)$
183.	Какой зависимостью описывается функция вычислительной сложности алгоритма прямого поиска в тексте по образцу в худшем случае:	<p>Вообще $\Theta(n*m)$, но тут такого нет :)</p> <ul style="list-style-type: none"> • $f(n, m)=\Theta(\log(n*m))$ 	<ul style="list-style-type: none"> • $f(n)=\Theta(n)$ • $f(n)=\Theta(n^2)$ • $f(n, m)=\Theta(\log(n+m))$ • $f(n, m)=\Theta(\log(n*m))$
184.	Какой зависимостью описывается функция вычислительной сложности алгоритма поиска в тексте по образцу методом Кнута-Морриса-Пратта в лучшем случае:	<ul style="list-style-type: none"> • $f(n)=\Theta(n)$ <p>Но на самом деле $O(n+m)$</p>	<ul style="list-style-type: none"> • $f(n)=\Theta(n)$ • $f(n)=\Theta(n^2)$ • $f(n, m)=\Theta(\log(n+m))$ • $f(n, m)=\Theta(\log(n*m))$
185.	В каких случаях алгоритм Кнута-Морриса-Пратта дает подлинный выигрыш по сравнению с другими алгоритмами поиска в тексте по образцу:	<ul style="list-style-type: none"> • Когда неудачному сравнению образца с текстом предшествовало некоторое число совпадений 	<ul style="list-style-type: none"> • Когда неудачному сравнению образца с текстом предшествовало некоторое число совпадений • Когда текст предварительно отсортирован • Когда образец предварительно отсортирован • Когда полное сравнение образца с текстом произошло на первом этапе сравнения
186.	В основе алгоритма Кнута-Морриса-Пратта используется:	Таблица префиксов	<ul style="list-style-type: none"> • Условие Айверсона • Хэш-таблица • Таблица смещений («стоп-символов») • Таблица префиксов
187.	В основе алгоритма Рабина-Карпа используется:	Хэш-таблица (хэш-функция)	<ul style="list-style-type: none"> • Условие Айверсона • Хэш-таблица • Таблица смещений («стоп-символов») • Таблица префиксов
188.	Укажите, какими свойствами должна обладать структура данных,	Однородная	<ul style="list-style-type: none"> • Сортированная • Несортированная

	предназначенная для хранения среднесуточной температуры воздуха за каждый день месяца, если при ее использовании в программе требуется обращаться и ко всем элементам сразу, и к каждому элементу в отдельности:		<ul style="list-style-type: none"> • Однородная • Неоднородная
189.	Какими свойствами можно характеризовать динамическую структуру данных:	Размер может изменяться во время работы программы	<ul style="list-style-type: none"> • Должна быть линейной • Формат элементов может изменяться во время работы программы • Доступ к элементам произвольный • Размер может изменяться во время работы программы
190.	Какие операции необходимо реализовать при разработке программы управления отдельным элементом структуры данных:	Добавление нового элемента	<ul style="list-style-type: none"> • Сортировка по различным полям структуры • Добавление нового элемента • Уничтожение структуры • Инициализация структуры
191.	Укажите свойство, характеризующее структуру данных Стек:	Линейный список с одной вершиной	<ul style="list-style-type: none"> • Линейный список с одной вершиной • Линейный список с двумя вершинами • Доступ возможен к произвольному элементу • Удаление (выборка) возможно произвольного элемента
192.	Имеется описание структуры узла линейного односвязного списка struct Tnode{ Tdata data; XXXX next;}; Какое определение должно быть на месте XXXX:	*Tnode	<ul style="list-style-type: none"> • Tnode • *Tnode • void • int
193.	Укажите свойства, характеризующие структуру данных Дек:	Вставка нового элемента возможна в начало и в конец списка	<ul style="list-style-type: none"> • Линейный список с произвольным количеством вершин • Вставка нового элемента возможна в начало и в конец списка • Вставка нового элемента возможна только в начало списка • Удаление элемента возможно из произвольной позиции
194.	Какой линейный список подойдет лучше для реализации структуры	Дек	<ul style="list-style-type: none"> • Линейный односвязный список

	данных, в которой требуется часто перемещаться как слева направо, так и справа налево:		<ul style="list-style-type: none"> • Очередь • Стек • Дек
195.	Имеется линейный однонаправленный список из n ($n > 1$) узлов. Структура узла списка <code>struct Tnode{ Tdata data; Tnode* next;};</code> L - указатель на его вершину. Укажите группу операторов, которые обеспечат корректную вставку в вершину списка L нового узла, адрес которого хранит указатель qq:	<code>qq->next=L; L=qq;</code>	<ul style="list-style-type: none"> • <code>qq->next=L; L=qq;</code> • <code>L=qq; qq->next=L;</code> • <code>qq=L;</code> • <code>qq->next=L; L=qq->next;</code>
196.	Имеется указатель q на узел в середине линейного однонаправленного списка со структурой узла <code>struct Tnode{ Tdata data; Tnode* next;};</code> Требуется удалить узел, ссылку на который хранит указатель q. Какую последовательность операторов необходимо выполнить, чтобы корректно выполнялась операция удаления узла:	<code>q1=q->next; *q=*q1; delete q1;</code>	<ul style="list-style-type: none"> • <code>(*q)=*(q->next); delete (q->next);</code> • <code>delete (q->next); (*q)=*(q->next);</code> • <code>q1=q->next; *q=*q1; delete q1;</code> • <code>*(q->next)=*(q); delete (q->next);</code>
197.	Как лучше реализовать линейный связный список, для задачи, в которой часто требуется добавлять узел в конец списка:	Очередь	<ul style="list-style-type: none"> • Стек • Очередь • Линейный односвязный список • Дек
198.	Какое минимальное количество операторов потребуется выполнить алгоритму при исключении (не удалении) узла, на который указывает указатель q, из линейного двусвязного списка:	2	<ul style="list-style-type: none"> • 1 • 2 • 3 • 4
199.	Какую операцию в линейном двунаправленном списке выполняют следующие операторы <code>q->prev->next=q->next; q->next->prev=q->prev;</code> ? Где q указатель на отдельный узел списка, расположенный в серединесписка:	Удаление узла по указателю q	<ul style="list-style-type: none"> • Вставка нового узла по указателю q • Вставка нового узла после элемента, ссылку на который хранит указатель q • Обмен значениями двух узлов • Удаление узла по указателю q
200.	Какая экономная по памяти структура данных используется для реализации кольцевой (циклической) очереди:	Массив	<ul style="list-style-type: none"> • Линейный двунаправленный список • Линейный однонаправленный список

			<ul style="list-style-type: none"> • Бинарное дерево • Массив
201.	<p>Имеется некоторая структура данных, в которую заносятся упорядоченные по возрастанию символы. Считывание данных из этой структуры даёт результат: F, E, D, C, B, A. Чем является эта структура данных:</p>	Стек	<ul style="list-style-type: none"> • Очередь • Дерево • Стек • Связный список
202.	<p>Имеется следующее оптимального кодовое дерево (ОКД). Закодировано 111 0 0 10 10 0 110 0 10 0</p> 	BDDCCDADCD	BDDCCDADCD
203.	<p>Какой код будет получен из входной строки символов 11112234444 при использовании алгоритма RLE</p>	4 1 2 2 1 3 4 4????	4 1 2 2 1 3 3 4????
204.	<p>Какой алгоритм реализует стратегию «разделяй и властвуй»</p>	*quick sort	*quick sort
205.	<p>Сколько байт бинарного файла занимает число 0 в двоичном формате</p>	*1 байт	*1 байт
206.	<p>Что такое структурированный протокол</p>	скорее всего простейший способ записи кластеров данных в файл, но хз	<p>a. Тип сохраненного кластера b. Простейший способ записи кластеров данных в файл c. Сетевые правила для миграции бинарных файлов d. Простой формат файлов для доступа из любой среды</p>
207.	<p>Каков главный недостаток хеш-таблиц</p> <p>Есть такой же вопрос с другими вариантами ответа, где нет "статического распределения памяти", но есть про фиксированное количество элементов</p>		
208.	<p>Имеется идеально сбалансированное двоичное дерево (не явл. деревом</p>		

	поиска). Обход слева-направо inorder: 2 4 6 8 10 12 14.		
209.	Что такое бит?	<p>единица измерения количества информации. 1 бит информации — символ или сигнал, который может принимать два значения: включено или выключено, да или нет, высокий или низкий, заряженный или незаряженный; в двоичной системе исчисления это 1 (единица) или 0 (ноль)</p>	
210.	Почему при записи элемента в бинарное дерево в параметры передается указатель на указатель на узел?	<p>Чтобы можно было его изменить</p>	
211.	Почему у ддп (двоичное дерево поиска) высокая эффективность поиска и сортировки	<p>Эффективность поиска по дереву напрямую связана с его сбалансированностью, то есть с максимальной разницей между глубиной левого и правого поддеревя среди всех вершин. Имеется два крайних случая – сбалансированное бинарное дерево (где каждый уровень имеет полный набор вершин) и вырожденное дерево, где на каждый уровень приходится по одной вершине. Вырожденное дерево эквивалентно связанному списку.</p>	
212.	Почему расстояние от самой короткой ветки до самой длинной в кчд (красно-черное дерево) различается не более чем в 2 раза?	<p>в короткой только чёрные, а в длинное ещё столько же красных</p>	
213.	Что хорошо работает на кчд (красно-черное дерево)?	<p>Ассоциативный массив? МБ multiset, map, multimap?</p>	

214.	Как может рассматриваться каждый элемент хэш-таблицы в организации ключей претерпевших коллизию с точки зрения производительности?	С точки зрения производительности, каждый элемент хэш-таблицы, претерпевший коллизию, можно рассматривать как элемент однонаправленного списка, так как для поиска по ключу нужно будет перебирать следующие записи по порядку до совпадения, начиная с индекса, возвращённого хэш-функцией	Возможные варианты ответа: a. Как переполненный связанный список *** b. Как элемент, расположенный по методу открытой адресации c. Как элемент, пропущенный через двойное хеширование. d. Как корень пустого дерева
215.	Каково наибольшее число ребер у графа (без треугольников) с n вершинами ?	Граф, состоящий из N вершин содержит не более $N(N-1)/2$ ребер. Что такое треугольники?	
216.	Сколько места требуется матрице смежности графа как функции от числа n вершин и числа m ребер?	n^2	
217.	Какая пространственная сложность описывает матрицу смежности графа как функции от числа n вершин и числа m ребер:		
218.	Какое из следующих времен выполнения лучше всего описывает простую реализацию алгоритма Дейкстры для графов, представленных в виде списков смежности? n и m обозначают соответственно число вершин и ребер входного графа.	$O(n^2+m)$ или $O(n^2)$ в самом простом случае В лучшем случае $O(m \log(n))$	
219.	В каких случаях в массивах меняется порядок хранения двоичных файлов по сравнению с их хранением в памяти компьютера?		
220.	Почему не рекомендуется открывать общий доступ к структурированному протоколу?	Если вы не будете знать тип кластера, файл будет чрезвычайно сложно декодировать. По этой причине файлы структурированного протокола не рекомендуется	

		использовать для совместного доступа к данным или для хранения данных в больших организациях, где возможна потеря информации о типе кластера.	
221.	Для чего созданы файлы объединяющие текстовые(XML) и битовые форматы?	Благодаря этому используются преимущества двоичных файлов, такие как эффективное использование памяти и высокая скорость записи. Кроме этого, в XML формате сохраняется структура данных и информация о них. Это позволяет очень легко считывать и находить информацию в файле. Обычно двоичные и XML данные разделены на два файла, файл с расширением .tdm для XML данных и файл с расширением .tdx для двоичных данных.	
222.	Для чего созданы файлы объединяющие текстовые(XML) и битовые форматы?	Благодаря этому используются преимущества двоичных файлов, такие как эффективное использование памяти и высокая скорость записи. Кроме этого, в XML формате сохраняется структура данных и информация о них. Это позволяет очень легко считывать и находить информацию в файле. Обычно двоичные и XML данные разделены на два файла, файл с расширением .tdm для XML данных и файл с расширением .tdx для двоичных данных.	Предположение: чтобы метаданные не были в ещё одном файле

223.	От чего зависит эффективность алгоритма Прима?		Производительность алгоритма Прима зависит от выбранной реализации приоритетной очереди. Асимптотика алгоритма зависит от способа хранения графа и способа хранения вершин, не входящих в дерево
224.	В чем проигрывает алгоритм Крускала алгоритму Прима?		Алгоритм Прима лучше работает с плотным графом, имеющим гораздо больше ребер, чем вершин.
225.	Как строить минимальное связанное дерево алгоритм Прима?		<p>На вход алгоритма подаётся связный неориентированный граф. Для каждого ребра задаётся его стоимость. Сначала берётся произвольная вершина и находится ребро, инцидентное данной вершине и обладающее наименьшей стоимостью. Найденное ребро и соединяемые им две вершины образуют дерево. Затем, рассматриваются рёбра графа, один конец которых — уже принадлежащая дереву вершина, а другой — нет; из этих рёбер выбирается ребро наименьшей стоимости. Выбираемое на каждом шаге ребро присоединяется к дереву. Рост дерева происходит до тех пор, пока не будут исчерпаны все вершины исходного графа. Результатом работы алгоритма является остовное дерево минимальной стоимости.</p>

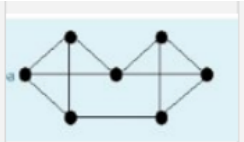
226.	Как построить минимальное связанное дерево алгоритм Крускала?		В начале текущее множество рёбер устанавливается пустым. Затем, пока это возможно, проводится следующая операция: из всех рёбер, добавление которых к уже имеющемуся множеству не вызовет появления в нём цикла, выбирается ребро минимального веса и добавляется к уже имеющемуся множеству. Когда таких рёбер больше нет, алгоритм завершён. Подграф данного графа, содержащий все его вершины и найденное множество рёбер, является его остовным деревом минимального веса.
227.	Где работает быстрее алгоритм Крускала в отличие от алгоритма Прима?		Алгоритм Крускала лучше работает с разреженными графами.
228.	Выберите неверный способ представления графа	Матрицей замыкания	а. Матрицей смежности; б. Матрицей инцидентности; в. Списком смежности; д. Матрицей замыкания; е. Массивом дуг;
229.	Классический алгоритм Дейкстры предназначен для	нахождения кратчайшего расстояния между двумя вершинами в произвольном взвешенном графе при отсутствии дуг отрицательного веса	а. нахождения кратчайшего расстояния между двумя вершинами ациклического графа при отсутствии дуг отрицательного веса б. нахождения кратчайшего расстояния между двумя вершинами в ациклическом взвешенном графе в. нахождения кратчайшего расстояния между двумя вершинами только в циклическом взвешенном графе д. нахождения кратчайшего расстояния между двумя

			вершинами в произвольном взвешенном графе при отсутствии дуг отрицательного веса е нахождения расстояния между всеми вершинами в произвольном взвешенном графе
230.	Выберите неверный термин (тот, который неприменим к графам)	Все термины верны;	а. Ориентированный граф; б. Двудольный граф; с. Ациклический граф; d. Все термины верны; е. Циклический граф;
231.	Центром орграфа $G = (V, E)$ называется (множественный выбор)	Центр графа — это множество всех вершин с минимальным эксцентриситетом. То есть множество всех вершин A , для которой максимальное расстояние $d(A, B)$ до других вершин B минимально. Эквивалентно, это множество вершин с эксцентриситетом, равным радиусу графа.	а. вершина, для которой максимальное расстояние (длина пути) до (от) других вершин минимально б. дуга с максимальным эксцентриситетом. с вершина с максимальным эксцентриситетом. d. вершина с минимальным эксцентриситетом. е. дуга с минимальным эксцентриситетом. f. вершина, для которой минимальное расстояние (длина пути) до (01) других вершин максимально
232.	Укажите свойства свободных деревьев (множественный выбор)	Свободное дерево или некорневое дерево — это связный неориентированный граф без циклов. Вершины, имеющие одного соседа, называются листьями дерева, остальные вершины называются внутренними узлами.	а. При добавлении в свободное дерево нового ребра образуется цикл. б. Свободное дерево представляет собой дерево без корня, ориентированное к листьям с. Каждое свободное дерево является полностью связанным графом d. Каждое свободное дерево с числом вершин n , $n > 1$, имеет в точности $n - 1$ ребер. е. Сумма весов ребер свободного дерева наименьшая на множестве ребер графа
233.	В глубинном остоном лесу, построенном при обходе орграфа $G =$	<u>b. обратные дуги</u> <u>с. поперечные дуги</u>	а. сильносвязанные компоненты глубинного

	(V, E), различают (выделяют) (множественный выбор)	<u>d. прямые дуги</u> <u>e. дуги дерева</u>	остовного леса b. обратные дуги c. поперечные дуги d. прямые дуги e. дуги дерева f. паросочетания дуг глубинного остовного леса
234.	Укажите неверное высказывание: "Поиск в ширину на взвешенном неориентированном графе предназначен (может быть использован) для	определения кратчайшего расстояния между парой узлов графа	a. определения циклов на графе b. определения кратчайшего расстояния между парой узлов графа c. отыскания какого-либо цикла на заданном графе d. получения глубинного остовного леса e. систематического обхода всех вершин (узлов) графа
235.	"Максимальное множество вершин графа, в котором существуют пути из любой вершины в любую другую вершину" - это ...	a. сильно связанная компонента графа?? Очень сомнительно	a. сильно связанная компонента графа b. минимальное покрытие графа c. редуцированный орграф d. максимальное покрытие графа e. максимальное паросочетание графа
236.	Пусть дан связанный неориентированный граф $G=(V, E)$. Отметьте те алгоритмы (полные варианты, возможно с некоторой адаптацией алгоритма), которые можно использовать для решения следующей задачи: построить остовное дерево минимальной стоимости. (множественный выбор)	a. алгоритм Крускала d. алгоритм Прима	a. алгоритм Крускала b. алгоритм Флойда c. метод поиска в глубину d. алгоритм Прима e. %-100%алгоритм Дейкстры f. метод поиска в ширину g. алгоритм Уоршелла
237.	Пусть дан связанный орграф $G=(V, E)$, не имеющий петель. Отметьте те алгоритмы (полные варианты, возможно с некоторой адаптацией алгоритма), которые можно использовать для решения следующей задачи: требуется определить, является ли граф циклическим (множественный выбор)	a. алгоритм Уоршелла e. метод поиска в глубину	a. алгоритм Уоршелла b. алгоритм Прима c. метод поиска в ширину d. алгоритм Флойда e. метод поиска в глубину f. алгоритм Крускала g. %-100%алгоритм Дейкстры
238.	Пусть дан связанный неориентированный взвешенный	e. алгоритм Прима f. алгоритм Крускала	a. алгоритм Уоршелла b. метод поиска в глубину

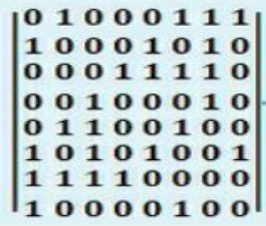

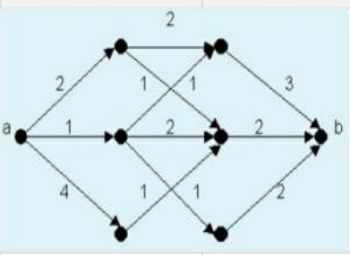
	граф $G=(V, E)$. Отметьте те алгоритмы (полные варианты, возможно с некоторой адаптацией алгоритма), которые можно использовать для решения следующей задачи определить такой подграф $G=(V, E')$, чтобы сумма весов дуг, принадлежащих E' , была бы минимальна. (множественный выбор)		<ul style="list-style-type: none"> c. метод поиска в ширину d. алгоритм Дейкстры e. алгоритм Прима f. алгоритм Крускала g. алгоритм Флойда
239.	Пусть дан связанный орграф $G=(V, E)$, причем все дуги имеют неорицательный вес. Отметьте те алгоритмы (полные варианты, возможно с некоторой адаптацией алгоритма), которые можно использовать для решения следующей задачи: Требуется определить кратчайшее расстояние от вершины до вершины y . (множественный выбор)	<ul style="list-style-type: none"> d. алгоритм Дейкстры f. алгоритм Флойда 	<ul style="list-style-type: none"> a. метод поиска в ширину b. алгоритм Уоршелла c. алгоритм Прима d. алгоритм Дейкстры e. метод поиска в глубину f. алгоритм Флойда g. алгоритм Крускала
240.	Пусть дан связанный невзвешенный (и непомеченный орграф $G=(V, E)$. Отметьте те алгоритмы (полные варианты, возможно с некоторой адаптацией алгоритма), которые можно использовать для решения следующей задачи: Требуется определить кратчайшее расстояние между вершинами x и y . (множественный выбор)	<ul style="list-style-type: none"> b. алгоритм Флойда e. метод поиска в ширину f. алгоритм Дейкстры 	<ul style="list-style-type: none"> a. алгоритм Прима b. алгоритм Флойда c. метод поиска в глубину d. алгоритм Уоршелла e. метод поиска в ширину f. алгоритм Дейкстры g. алгоритм Крускала
241.	Пусть дан связанный неориентированный граф $G=(V, E)$, в виде матрицы смежности. Отметьте те алгоритмы, которые имеют трудоемкость порядка $O(N^3)$, где N – число вершин графа.	<ul style="list-style-type: none"> c. алгоритм Уоршелла d. метод поиска в ширину. алгоритм Флойда 	<ul style="list-style-type: none"> a. метод поиска в глубину b. алгоритм Прима c. алгоритм Уоршелла d. метод поиска в ширину. алгоритм Флойда f. алгоритм Крускала. e. Алгоритм Дейкстры
242.	Связный граф, не имеющий точек сочленения, называется (является) ... (множественный выбор) (укажите 2 правильных ответов)	<p>Точкой сочленения в теории графов называется вершина графа, при удалении которой количество компонент связности возрастает.</p> <ul style="list-style-type: none"> a. k-связным, причем $k \geq 2$ c. двусвязным 	<ul style="list-style-type: none"> a. k-связным, причем $k \geq 2$ b. покрытием графа $G = (V, E)$ c. двусвязным d. паросочетанием графа $G - (V, E)$ e. k связным, причем $k = 1$ f. если между любой парой вершин и существует не менее разных путей ($k > 1$), таких, что, за исключением вершин v и w, ни одна из

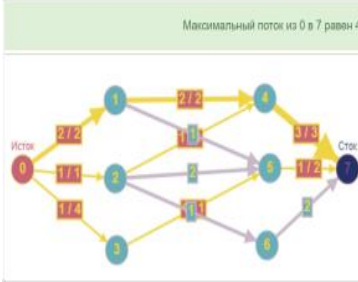
			<p>вершин, входящих в один путь, не входит ни в какой другой из этих путей. g. свободным деревом</p>
243.	<p>Какому алгоритму соответствует следующий фрагмент: "На каждом шаге алгоритма находится ребро наименьшей стоимости (u, v) такое, что $u \in U$ и $v \in V \setminus U$, затем вершина v переносится из множества $V \setminus U$ в множество U. Этот процесс продолжается до тех пор, пока множество U не станет равным множеству V."</p>	<p>а. алгоритму Прима</p>	<p>а. алгоритму Прима b. обходу графа методом поиска в ширину c. решение задачи коммивояжера d. алгоритму Крускала e. обходу графа методом поиска в глубину f. алгоритму Дейкстра g. вычислению эксцентриситета графа</p>
244.	<p>Укажите описание алгоритма, которое может являться фрагментом построения остовного дерева минимальной стоимости алгоритмом Прима.</p>	<p>б. "На каждом шаге алгоритма находится ребро наименьшей стоимости (u, v) такое, что $u \in U$ и $v \in V \setminus U$, затем вершина v переносится из множества $V \setminus U$ в множество U. Этот процесс продолжается до тех пор, пока множество U не станет равным множеству V."</p>	<p>а. "На каждом шаге алгоритма находится ребро наименьшей стоимости (u, v) такое, что $u \in U$ и $v \in V \setminus U$, и оно еще не принимало участие в алгоритме. После проверки того, что это ребро не образует цикл на подграфе, содержащем все узлы и отобранные к текущему моменту дуги, данное ребро добавляется к данному подмножеству дуг, образующих решение задачи. Этот процесс продолжается до тех пор, пока число отобранных дуг не станет равным $n - 1$ (где n - число узлов графа)"</p> <p>б. "На каждом шаге алгоритма находится ребро наименьшей стоимости (u, v) такое, что $u \in U$ и $v \in V \setminus U$, затем вершина v переносится из множества $V \setminus U$ в множество U. Этот процесс продолжается до тех пор, пока множество U не станет равным множеству V."</p> <p>с. "На каждом шаге рассматриваются в лексикографическом порядке дуги, исходящие из текущей вершины. Если дуга ведет к еще не посещенному узлу, то остальные еще не посещенные узлы, связанные дугой с текущим, помещаются</p>

			<p>в стек, а алгоритм переходит к рассмотрению аналогичным образом такого узла, помечая как посещенный. Если у текущего узла связанных не посещенных узлов нет, то из стека "выталкивается" верхний элемент и алгоритм переходит к рассмотрению такого узла. Данный процесс продолжается до тех пор, пока в стеке содержится хотя бы один элемент.</p> <p>d. "На каждом шаге рассматриваются в лексикографическом порядке дуги, исходящие из текущей вершины. Еще не посещенные узлы, связанные дугой с текущим, помечаются и помещаются в очередь. Для обработки следующего узла из очереди "извлекается" очередной узел и обрабатывается аналогичным образом. Данный процесс продолжается до тех пор, пока в очереди содержится хотя бы один элемент.</p> <p>e. "На каждом шаге к множеству U добавляется та из оставшихся вершин, расстояние до которой от источника меньше, чем для других оставшихся вершин. Если стоимости всех дуг неотрицательны, то можно быть уверенным, что кратчайший путь от источника к конкретной вершине проходит только через вершины множества U. Когда множество U будет содержать все вершины орграфа, тогда вспомогательный массив будет содержать длины кратчайших путей от источника к каждой вершине.</p>
245.	<p>Определите хроматическое число графа</p> 	<p>е. 3</p> <p>Хроматическое число графа G — минимальное число цветов, в которые можно раскрасить вершины графа G так,</p>	<p>a. 7 b. 4 c. 6 d. 5 e. 3 f. 2</p>

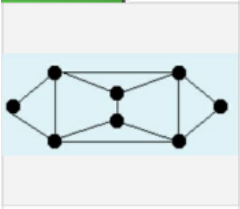
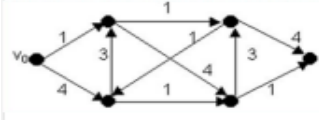
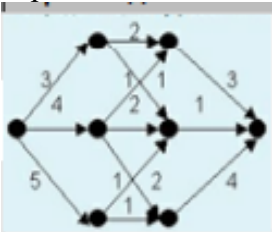
		чтобы концы любого ребра имели разные цвета.	
246.	Определите задачу, решение которой предполагает использование алгоритма Крускала	<p>а. Проектирование такой топологии связей между компьютерами в компьютерной сети, при которой обеспечивался минимальный расход сетевого кабеля без учета топологических особенностей местности</p>	<p>а. Проектирование такой топологии связей между компьютерами в компьютерной сети, при которой обеспечивался минимальный расход сетевого кабеля без учета топологических особенностей местности</p> <p>б. Проектирование такой топологии связей между компьютерами в компьютерной сети, при которой обеспечивалась повышенная надежность соединений в следствие наличия избыточных каналов связи</p> <p>с. Проектирование топологии типа "кольцо", при которой обеспечивался минимальный расход сетевого кабеля</p> <p>д. Проектирование такой топологии сети типа "звезда", при которой обеспечивалась минимальный расход сетевого кабеля</p> <p>е. Оптимизация трафика в компьютерной сети путем нахождения кратчайшего расстояния между двумя компьютерами в компьютерной сети</p>
247.	Класс сортировок слиянием основан	д. На объединении двух или более упорядоченных последовательностей	<p>а. На нахождении наибольшего (наименьшего) из оставшихся неупорядоченных элементов</p> <p>б. На нахождении места между меньшим (или равном) и большим (или равном) ключами в уже</p>

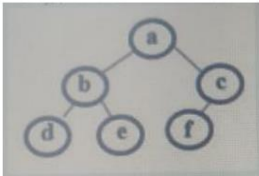
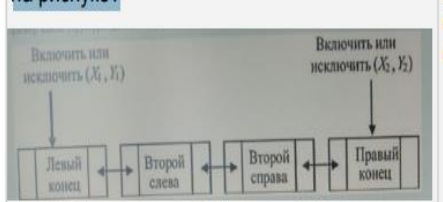
			<p>упорядоченной последовательности</p> <p>с. На подсчете элементов меньших данному</p> <p>d. На объединении двух или более упорядоченных последовательностей</p> <p>е. На обмене элементом местами, если они неупорядочены</p>
248.	Времени работы алгоритма соответствует	<p>???</p> <p>b. наиболее благоприятный случай, когда алгоритм работает меньше (быстрее) всего по времени, которое можно определить с помощью встроенного в компьютер таймером</p> <p>d. наименее неблагоприятный случай, когда алгоритм работает меньше (быстрее) всего по времени, которое можно определить с помощью встроенного в компьютер таймером</p>	<p>а. случай, когда после компиляции данного алгоритма в машинный код время работы программы будет минимальным</p> <p>b. наиболее благоприятный случай, когда алгоритм работает меньше (быстрее) всего по времени, которое можно определить с помощью встроенного в компьютер таймером</p> <p>с. наименее вероятная цепочка исполнения команд алгоритма</p> <p>d. наименее неблагоприятный случай, когда алгоритм работает меньше (быстрее) всего по времени, которое можно определить с помощью встроенного в компьютер таймером</p> <p>е. случай, когда после компиляции данного алгоритма в машинный код будет выполнено минимальное число машинных команд и, как следствие, время выполнения программы будет минимальным</p> <p>f. случай, когда алгоритм выполняет меньшее число операций</p> <p>g. случай, когда после компиляции данного алгоритма в машинный код программа будет содержать минимальное число машинных команд</p>

249.	<p>Пусть дан граф $G=(V, E)$, $V=(V_1, \dots, V_8)$, с матрицей смежности: Определите хроматическое число графа</p> 	е. 3	<p>a. 8 b. 6 c. 7 d. 4 e. 3 f. 5</p>
250.	<p>Что из указанного образуется подмножеством узлов графа (множественный выбор)</p>	<p>???</p> <p>В таблице еще b,c,f В душе не ебу, по идее только б, ведь граф может быть пустым</p>	<p>a. поток b. независимое множество ... графа c. путь d. эксцентриситет e. точка сочленения графа f. остовное дерево</p>
251.	<p>Укажите арифметическое выражение в постфиксной форме записи, которое полностью соответствует следующему арифметическому выражению в префиксной форме записи: $+1+-2+3-+4*5+6\ 7\ 8\ 9$</p>	<p>P = 1 2 3 4 5 6 7 + * + 8 - + 9 ++</p>	<p>a. P = 1 2 * 3 4 5 * + 6 7 + 8 9 + * + b. P = 1 2 * 3 4 5 * + + 6 7 + 8 9 + * c. P = 1 2 3 4 5 6 * 7 + + 8 - + 9 ++ d. P = 1 2 3 - + 4 5 + 6 7 8 9 + - * + + e. P = 1 2 3 4 5 6 7 + * + 8 - + 9 ++ f. P = 1 2 3 4 5 + 6 7 8 9 + - * + + +</p>
252.	<p>Пусть дан граф $G=(V, E)$, $V=(V_1, \dots, V_6)$, с матрицей смежности: Определите хроматическое число графа.</p> 	4	<p>a. 3 b. 6 c. 4 d. 5 e. 2</p>
253.	<p>Укажите верные соотношения (множественный выбор)</p>	<p>$2n^2+5n=w(n)$ $2n^2+5n=O(n^3)$</p>	<p>a. $2n^2 + 5n = \Theta(n^3)$ b. $2n^2 + 5n = \Omega(n^3)$ c. $2n^2 + 5n = O(n^3)$ d. $2n^2 + 5n = \omega(n)$ e. $2n^2 + 5n = o(n^3)$</p>
254.	<p>Дана сеть $N=(G, \alpha)$, $G=(V, E)$, $\alpha(e)$ пропускная способность дуги e. Определите максимальный поток через сеть N</p> 	с. 4	<p>a. 9 b. 5 c. 4 d. 7 e. 3 f. 8 g. 6</p>

<p>255.</p>	<p>Какому алгоритму соответствует следующий фрагмент: “На каждом шаге рассматриваются в лексикографическом порядке дуги, исходящие из текущей вершины. Если дуга ведет к еще не посещенному узлу, то остальные еще не посещенные узлы, связанные дугой с текущим, помещаются в стек, алгоритм переходит к рассмотрению аналогичным образом такого узла, помечая как посещенный. Если у текущего узла связанных непосещенных узлов нет, то из стека “выталкивается” верхний элемент и алгоритм переход к рассмотрению такого узла. Данный процесс продолжается до тех пор, пока в стеке содержится хотя бы один элемент”.</p>	<p>f. обходу графа методом поиска в глубину</p>	 <p>Максимальный поток из 0 в 7 равен 4</p>
<p>256.</p>	<p>Что из указанного является скалярной величиной для графа (множественный выбор)</p>	<p>a. пропускная способность d. плотность графа</p>	<p>a. пропускная способность b. поток c. сильно связанная компонента d. плотность графа e. цикл f. точка сочленения графа g. разрез</p>
<p>257.</p>	<p>Укажите описание алгоритма, которое может являться фрагментом алгоритма обхода методом поиска в ширину</p>	<p>e. "На каждом шаге рассматриваются в лексикографическом порядке дуги, исходящие из текущей вершины. Еще не посещенные узлы, связанные дугой с текущим, помечаются и помещаются в очередь. Для обработки следующего узла из очереди "извлекается" очередной узел и обрабатывается аналогичным образом. Данный процесс продолжается до тех пор, пока в очереди содержится хотя бы один элемент.</p>	<p>a. "На каждом шаге алгоритма находится ребро наименьшей стоимости (u, v) такое, что $u \in U$ и $v \in V \setminus U$, затем вершина v переносится из множества $V \setminus U$ в множество U. Этот процесс продолжается до тех пор, пока множество U не станет равным множеству V." b. "На каждом шаге рассматриваются в лексикографическом порядке дуги, исходящие из текущей вершины. Если дуга ведет к еще не посещенному узлу, то остальные еще не посещенные узлы, связанные дугой с текущим, помещаются в стек, а алгоритм переходит к рассмотрению аналогичным образом такого узла, помечая как посещенный. Если у текущего узла связанных не посещенных узлов нет, то из стека</p>

			<p>"вытаскивается" верхний элемент и алгоритм переход к рассмотрению такого узла. Данный процесс продолжается до тех пор, пока в стеке содержится хотя бы один элемент.</p> <p>с. "На каждом шаге алгоритма находится ребро наименьшей стоимости (u, v) такое, что u (хз) V и v (хз) V, и оно еще не принимало участие в алгоритме. После проверки того, что это ребро не образует цикл на подграфе, содержащем все узлы и отобранные к текущему моменту дуги, данное ребро добавляется к данному подмножеству дуг, образующих решение задачи. Этот процесс продолжается до тех пор, пока число отобранных дуг не станет равным $n - 1$ (где n - число узлов графа)"</p> <p>d. "На каждом шаге к множеству U добавляется та из оставшихся вершин, расстояние до которой от источника меньше, чем для других оставшихся вершин. Если стоимости всех дуг неотрицательны, то можно быть уверенным, что кратчайший путь от источника к конкретной вершине проходит только через вершины множества U. Когда множество U будет содержать все вершины орграфа, тогда вспомогательный массив будет содержать длины кратчайших путей от источника к каждой вершине.</p> <p>е. "На каждом шаге рассматриваются в лексикографическом порядке дуги, исходящие из текущей вершины. Еще не посещенные узлы, связанные дугой с текущим, помечаются и помещаются в очередь. Для обработки следующего узла из очереди "извлекается" очередной узел и обрабатывается аналогичным образом. Данный процесс продолжается до тех пор, пока в очереди содержится хотя бы один элемент.</p>
258.	"Жадные" алгоритмы (схемы поиска решений) ...	е. являются эвристическими алгоритмами	а. включают этап возврата к предыдущему успешному шагу, в случае если текущий шаг

			<p>поиска решений привел к неудаче</p> <p>b. получили такое название из-за повышенных требований к памяти</p> <p>c. получили такое название из-за повышенных требований к ресурсам ЭВМ</p> <p>d. являются алгоритмами с полным перебором всех вариантов ответа</p> <p>e. являются эвристическими алгоритмами</p>
259.	<p>Определите хроматическое число графа</p> 	f. 3	<p>a. 7</p> <p>b. 2</p> <p>c. 4</p> <p>d. 6</p> <p>e. 5</p> <p>f. 3</p>
260.	<p>С помощью алгоритма Дейкстры определите наикратчайшее расстояние от вершины v_0 до наиболее удаленной вершины графа</p> 	b. 5	<p>a. 4</p> <p>b. 5</p> <p>c. 6</p> <p>d. 7</p> <p>e. 3</p>
261.	<p>Дана сеть $N=(G, \alpha)$, $G=(V, E)$, $\alpha(e)$ пропускная способность дуги e. Определите максимальный поток через сеть N</p> 	b. 9	<p>a. 6</p> <p>b. 9</p> <p>c. 7</p> <p>d. 8</p> <p>e. 5</p> <p>f. 4</p>
262.	<p>На вход алгоритма построения дерева двоичного гюиска (изначально дерево было пустое) была подана следующая последовательность чисел: 18, 20, 14, 10, 1, 16, 3, 17, 5, 40, 35, 45. Необходимо удалить корневой</p>	b. 17	<p>a. 20</p> <p>b. 17</p> <p>c. 14</p> <p>d. 18</p> <p>e. 16</p> <p>f. 35</p> <p>g. 1</p>

	элемент. Определите элемент, который после этого будет корнем дерева.(множественный выбор)		
263.	Как определить есть ли в неориентированном связном графе Эйлера цикл?	с. в графе все вершины имеют четную степень	<p>a. в графе все вершины имеют нечетную степень</p> <p>b. в графе только две вершины, которые имеют четную степень</p> <p>с. в графе все вершины имеют четную степень</p> <p>d. в графе все вершины, которые имеют нечетную степень</p>
264.	<p>Что будет выведено при прямом обходе дерева?</p> 	с. a b d e c f	<p>a. d b e a f c</p> <p>b. b d e c f 1</p> <p>с. a b d e c f</p> <p>d. a b c d e f</p>
265.	Укажите асимптотическую сложность операции удаления элемента в позиции i из массива размером n :	d. $O(n)$	<p>a. $O(n^2)$</p> <p>b. $O(\log n)$</p> <p>c. $O(1)$</p> <p>d. $O(n)$</p>
266.	<p>Пример какой структуры данных изображён на рисунке?</p> 	d. дек	<p>a. стек</p> <p>b. очередь</p> <p>c. кольцо</p> <p>d. дек</p>
267.	Что хранит матрица смежности графа?	<p>предположительно d</p> <p>Так-то</p> <p>Количество ребер между вершинами для каждой пары вершин</p> <p>В другом файле выделено b</p>	<p>a. Список ребер и их направления</p> <p>b. Длину пути между вершинами графа</p> <p>c. Список вершин</p> <p>d. Только направления ребер</p>
268.	Какую операция можно использовать в качестве хэш-функции для целочисленных ключей?	d. Операцию взятия остатка от деления ключа на размерность массива	<p>a. Операцию умножения ключа на размерность массива</p> <p>b. Операцию взятие остатка от деление размерности массива на ключ</p> <p>c. Операцию деления ключа на размерность</p>

			массива d. Операцию взятия остатка от деления ключа на размерность массива
269.	К недостаткам рекурсивного метода можно отнести?	а. возможность переполнения стековой памяти программного процесса с. расход времени на выделение и очистку стека	а. возможность переполнения стековой памяти программного процесса б. меньшая точность результата вычислений с. расход времени на выделение и очистку стекового кадра в памяти d. возможность переполнения динамически распределяемой памяти программного процесса
270.	Остовное дерево - это:	с. Подграф, полученный путём удаления максимального числа ребёр без нарушения связности	а. Подграф, полученный путем кратчайшего обхода всех вершин графа б. Дерево, полученное путем удаления вершин, связанных с более чем половиной всех вершин в графе с. Подграф, полученный путем удаления максимального числа ребер, без нарушения СВЯЗНОСТИ d. Подграф, полученный путем удаления вершин, связанных со всеми остальными
271.	Нелинейными называются структуры данных, в которых:	с (b)	а. связи между элементами не зависят от выполнения какого-либо условия б. связи между элементами не зависят от линейной упорядоченности элементов с. связи между элементами зависят от выполнения определенного условия d. связи между элементами

			не зависят от упорядоченности значений элементов
272.	Количество ребёр в остовном дереве (n - количество вершин графа) равно:	b. n-1	a. n-2 b. n-1 c. n+1 d. n
273.	Что определяет термин "чёрная высота" узла в структуре красно-чёрного дерева?	с. Количество чёрных узлов на пути от корня до листового узла	a. Количество узлов на пути от корня до ближайшего чёрного узла b. Длина пути от корня до ближайшего черного узла с. Количество чёрных узлов на пути от корня до листового узла d. Количество чёрных узлов в красно-чёрном дереве
274.	Какой зависимостью описывается функция вычислительной сложности алгоритма Дейкстры в графе, построенному на матрице смежности, если V - количество вершин в графе, а E - количество ребёр:	b. $f(n)=O(V^2)$	a. $f(n)=O(E)$ b. $f(n)=O(V^2)$ c. $f(n)=O(\log(E))$ d. $f(n)=O(V)$
275.	Выберите неверный способ представления графа	d. Матрицей замыкания;	a. Матрицей смежности; b. Матрицей инцидентности; c. Списком смежности; d. Матрицей замыкания; e. Массивом дуг;
276.	Задача нахождения кратчайшего расстояния между двумя вершинами графа в случае отсутствия дуг отрицательного веса решается	с. используя алгоритм Дейкстры или алгоритм Флойда	a. используя алгоритм Дейкстры b. используя алгоритм Флойда с. используя алгоритм Дейкстры или алгоритм Флойда d. последовательно, сначала алгоритмом Дейкстры, затем алгоритмом Флойда e. используя алгоритм Флойда, он же алгоритм Дейкстры (т. е. алгоритм Флойда и алгоритм Дейкстры - это два различных

			названия одного и того же алгоритма)
277.	Лист дерева — это узел, который	b. имеет степень 0	<p>a. имеет более двух потомков</p> <p>b. имеет степень 0</p> <p>c. является внутренним узлом дерева</p> <p>d. имеет только правого потомка</p>
278.	Требуется написать задачу о перемещении в лабиринте (поле из клеток с препятствиями) из заданной клетки в другую заданную клетку путем последовательного перемещения по клеткам. Какой метод позволяет сократить число переборов:	b. "Жадный" алгоритм A*Search	<p>a. Разделяй и властвуй</p> <p>b. "Жадный" алгоритм</p> <p>c. Метод перебора возвратом</p> <p>d. Метод рекурсии</p>
279.	Какая идея положена в основу построения алгоритма методом "жадного алгоритма"	b. На каждом этапе осуществляется выбор решения, которое кажется оптимальным в данный момент	<p>a. На каждом этапе осуществляется разбиение задачи на подзадачи, которые могут быть оптимальными в данный момент</p> <p>b. На каждом этапе осуществляется выбор решения, которое кажется оптимальным в данный момент</p> <p>c. На каждом этапе осуществляется выбор решения, которое является не оптимальным в данный момент</p> <p>d. На каждом этапе осуществляется разбиение задачи на подзадачи, которые могут быть не оптимальным в данный момент</p>
280.	Алгоритм вставки ключа в рандомизированное дерево	a. определяет вставить ключ в лист дерева, или в корень	<p>a. определяет вставить ключ в лист дерева, или в корень</p> <p>b. ключ сразу вставляется в корень, а затем дерево перестраивается</p> <p>c. если ключа нет в дереве, то он вставляется по правилу</p>

			бинарного дерева поиска и лерево не перестраивается d. если ключ уже есть в дереве, дерево не перестраивается
281.	Алгоритм Прима поиска (построения) остовного дерева минимальной стоимости связанного взвешенного не ориентированного графа	с. Строит остовное дерево, добавляя в него по одному ребру за раз, отыскивая на каждом шаге ребро с минимальной стоимостью и присоединяет его к единственному дереву, растущему на каждом шаге	а. Строит остовное дерево, применяя для поиска минимальных по стоимости ребер алгоритм Флойда и собирает из них остовное дерево b. Строит остовное дерево, применяя для поиска минимальных по стоимости ребер алгоритм Дейкстры и собирает из них остовное дерево с. Строит остовное дерево, добавляя в него по одному ребру за раз, отыскивая на каждом шаге ребро с минимальной стоимостью и присоединяет его к единственному дереву, растущему на каждом шаге d. Строит остовное дерево, путем создания набора связанных подграфов и объединения их в остовное дерево
282.	Свойство однородной структуры данных	b. все элементы одного типа	а. элементы могут быть разного типа b. все элементы одного типа с. все элементы разного типа
283.	Укажите вид каждого графа представленного матрицами смежности		
284.	Алгоритм двойного RL поворота	d. правый поворот вокруг левого сына узла с фактором балансировки	а. правый поворот вокруг правого сына узла с фактором балансировки +1, затем

		-1, затем левый вокруг корневого узла	<p>левый вокруг корневого узла</p> <p>b. правый поворот вокруг узла, затем левый вокруг узла с фактором балансировки -1</p> <p>c. правый поворот вокруг правого сына узла с фактором балансировки -1, затем левый вокруг корневого узла</p> <p>d. правый поворот вокруг левого сына узла с фактором балансировки -1, затем левый вокруг корневого узла</p>
285.	Дисциплина обслуживания (порядок выполнения операций) стека:	a. LIFO	<p>a. LIFO</p> <p>b. OFIL</p> <p>c. FIFO</p> <p>d. LOFI</p>
286.	В основе алгоритма Бойера-Мура используется	c. Таблица смещений ("стоп-символов")	<p>a. Таблица префиксов</p> <p>b. Хэш-таблица</p> <p>c. Таблица смещений ("стоп-символов")</p> <p>d. Условией Айверсона</p>
287.	В узлах бинарного дерева поиска разместили список значений. Какой алгоритм обхода бинарного дерева поиска позволяет вывести эту последовательность в упорядоченном по возрастанию порядке	d. Симметричный	<p>a. Прямой</p> <p>b. Последовательный</p> <p>c. Обратный</p> <p>d. Симметричный</p>
288.	Иерархическая структура данных (дерево)	b. это динамическая нелинейная структура	<p>a. это статическая нелинейная структура</p> <p>b. это динамическая нелинейная структура</p> <p>c. это скалярная нелинейная структура</p> <p>d. это динамическая линейная структура</p>
289.	<p>Вычислить арифметическое выражение в постфиксной форме записи: +*1 3-*5 7*+9 2 4+6 8</p> <p>Но тут префиксная нотация - Возможно:</p> <p>Вычислить арифметическое</p>	-99	

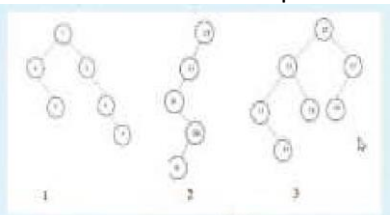
	выражение в префиксной форме записи: +*1 3-* -5 7*+9 2 4+6 8		
290.	<p>Какое из перечисленных AVL-деревьев требует проведения перестройки:</p>	с. 3	<p>a.4 b. 1 с. 3 d. 2</p>
291.	Алгоритм Прима - это:	с. Алгоритм поиска минимального остовного дерева во взвешенном неориентированном связном графе	<p>a. Алгоритм, указывающий несколько путей обработки одних и тех же входных данных, - без какого-либо уточнения, какой именно вариант будет выбран b. Алгоритм для нахождения кратчайших путей от одной из вершин графа до всех остальных с. Алгоритм поиска минимального остовного дерева во взвешенном неориентированном связном графе d. Алгоритм для нахождения кратчайших путей всеми вершинами взвешенного ориентированного графа</p>
292.	Как называется алгоритм для нахождения кратчайших путей между всеми вершинами взвешенного ориентированного графа:	d. Алгоритм Флойда-Уоршала Обратный этому вопросу верен по СДО (строка 279)	<p>a. Алгоритм Крускала b. Алгоритм Прима c. Алгоритм Дейкстры d. Алгоритм Флойда-Уоршала</p>
293.	Какой алгоритм реализует стратегию "жадных алгоритмов" (greedy algorithm):	a. алгоритм Дейкстры	<p>a. алгоритм Дейкстры b. алгоритм Бойера-Мура c. алгоритм Флойда-Уоршера d. алгоритм быстрой сортировки Quick-sort</p>
294.	Временная (вычислительная) сложность алгоритма определяется	с - Размер массива или файла	a- Количество основных операций

	количеством входных данных n . Этот параметр пропорционален величине обрабатываемого набора данных и может обозначать:		b - Количество функций в программе c - Размер массива или файла d - Размер занимаемой памяти при сортировке или поиске
295.	Какое понятие не относится к характеристикам алгоритмов хеширования:	<u>Мощность</u>	Разрядность Вычислительная сложность Криптостойкость Мощность
296.	Алгоритм обхода графа - это:	b. Алгоритм, устанавливающий переход от одной его вершины к другой в поисках свойств связей этих вершин	a . Алгоритм, указывающий несколько путей обработки одних и тех же входных данных, без какого-либо уточнения, какой именно вариант будет выбран b. Алгоритм, устанавливающий переход от одной его вершины к другой в поисках свойств связей этих вершин c . Алгоритм, устанавливающий движение от начальной вершины в определенном направлении (по определенному пути) до тех пор, пока не будет достигнут конец пути или заданная вершина d . Алгоритм, устанавливающий переход от одной его вершины к другой с целью нахождения кратчайшего пути
297.	Какая вычислительная сложность описывает реализацию алгоритма Дейкстры для графов, представленных в виде списков смежности (n и m обозначают соответственно число вершин и ребер исходного графа)		$O(m \log(n))$ $O(m+n)$ $O(mn)$ $O(n^2)$
298.	Какой зависимостью описывается функция вычислительной сложности алгоритма Дейкстры в графе, построенном на списке смежных вершин, если V - количество вершин в графе, а E - количество ребер:		

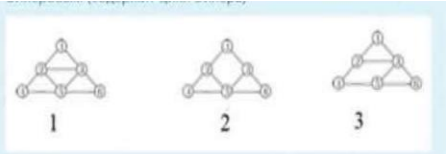
299.	Выберите определение бинарного дерева поиска:	а. Бинарное дерево, у которого каждый узел может содержать либо два дочерних узла, либо может содержать один дочерний узел, либо может не содержать ни одного узла	<p>а. Бинарное дерево, у которого каждый узел может содержать либо два дочерних узла, либо может содержать один дочерний узел, либо может не содержать ни одного узла</p> <p>б. Бинарное дерево, у которого каждый узел может содержать либо два дочерних узла, либо может не содержать ни одного узла</p> <p>с. Бинарное дерево, элементы которого являются двоичными числами</p> <p>д. Бинарное дерево, у которого каждый узел обязательно должен содержать два дочерних узла</p>
300.	Имеется следующее оптимальное кодовое дерево (ОКД) Выберите исходную строку, которая была закодирована в битовый код 111 00 10 10 0 110 0 10 0 по алгоритму Хаффмана с использованием приведенного окд:	ABCCDDDDDD	<p>BDDCCDADCD</p> <p>ABDCDCDCDD</p> <p>DDDDCCCCBA</p> <p>ABCCDDDDDD</p>
301.	Структура хранения, обеспечивающая прямой доступ к каждому элементу по адресу	вектор - если в C++	<p>граф</p> <p>сеть</p> <p>вектор</p> <p>список</p>
302.	Какой зависимостью описывается функция вычислительной сложности алгоритма Дейкстры в графе, построенном на списке смежных вершин, если V - количество вершин в графе, а E - количество ребер:	<p>$f(n)=O(V^2)$</p> <p>Но вообще $O(V + E)$</p>	<p>$f(n)=O(V^2)$</p> <p>$f(n)=O(E)$</p> <p>$f(n)=O(E*\log(V))$</p> <p>$f(n)=O(V)$</p>
303.	Какая структура данных позволяет, используя минимальный объем памяти для хранения данных с ключами, значения которых больше количества значений в структуре, обеспечить прямой доступ к любому элементу данных по ключу за константное время	Хеш-таблица	<p>Идеально-сбалансированное дерево поиска</p> <p>Хеш-таблица</p> <p>Сбалансированное дерево поиска</p> <p>Линейный</p> <p>однонаправленный список</p>
304.	Какими значениями должны быть заполнены узлы бинарного дерева	строго упорядоченными	строго упорядоченными случайными

	поиска, чтобы при поиске ключа возник наилучший случай		
305.	В дереве Фибоначчи при заданной высоте количество узлов	минимально Дерево Фибоначчи — AVL-дерево с наименьшим числом вершин при заданной высоте (глубине)	максимальное среднее равно половине высоты дерева минимально
306.	Имеется узел AVL-дерева для которого $hl < hr$, где hl , hr высота левого и правого поддеревья соответственно. В левое поддерево его правого сына вставляется новый ключ, после этого:	Условие недостаточно Но будем считать, что не требуется	а) не требуется перестройка дерева б) требуется перестройка с двойным поворотом LR или одним правым поворотом в) требуется перестройка с одним правым поворотом г) требуется перестройка с одним левым поворотом или двойным поворотом RL
307.	Какую сложность обеспечит применение хеш-таблицы в алгоритме поиска записи с ключом в файле, если элемент хеш-таблицы будет содержать адрес места размещения записи с этим ключом в файле	O(1)	O(1) O(logn) O(nlogn) O(n) где n количество ключей в файле
308.	K-арное дерево (упорядоченно слева направо по порядку размещения узлов) с $N > 0$ узлами и метками от 1 до N реализовано списком сыновей. Дерево реализовано на структуре <code>struct Node{int Label; Node *next;};</code> - структура узла списка; <code>struct Tree{ Node *List[N+1]; int N;};</code> - структура дерева. Алгоритм нахождения самого левого сына заданного узла M	В файле выделено с, но это не точно Надо проверить	а. получить ссылку (значение узла List[M]) на список сыновей в массиве указателей на список, вернуть метку первого узла в списке б. получить значение метки узла List[M] с. получить ссылку (значение узла List[M]) на список сыновей в массиве указателей на список, вернуть метку последнего узла в списке d. получить значение узла List[M]
309.	Асимптотическую сложность операции обращения к i-ому элементу в массиве размером n по нотации O	O(1)	O(n ²) O(1) O(n) O(logn)

310.	Реализация линейно упорядоченного, по порядку следования, множества значений задачи, может быть выполнена в программе множественный ответ	на однонаправленном списке на линейном списке	на однонаправленном списке на линейном списке на двумерном массиве на структуре данных множество
311.	K-арное дерево (упорядоченно слева направо по порядку размещения узлов) с $N > 0$ узлами и метками от 1 до N, реализуется на массиве родителей. Задан узел $M > 0$ ($M \leq N$) (метка или номер узла). Алгоритм поиска правого сына заданного узла, при условии, что он есть у узла	найти элемент со значением равными M-1 и вернуть индекс следующего узла	a. найти элемент со значением равными 0 и вернуть индекс следующего узла b. найти элемент со значением M и вернуть индекс этого узла c. найти элемент со значением равными M-1 и вернуть индекс следующего узла d. найти элемент со значением равными M и вернуть индекс следующего узла со значением равным M
312.	Какой из методов поиска используется при поиске кратчайшего пути между любыми двумя вершин во взвешенном неориентированном графе	d. Поиск методом в ширину	a. Поиск методом в глубину b. Линейный поиск c. Бинарный поиск d. Поиск методом в ширину
313.	Степень вершины графа	Вообще-то различают две степени вершины: степень выхода и степень входа. В неорграфе - количество смежных вершин - хотя там нет разделения на исходящие/входящие ребра Если речь про неор, то количество смежных вершин	a. количество исходящих ребёр b. количество входящих ребер c. количество смежных вершин
314.	При реализации алгоритма по определению того, что обрабатываемый текст содержит палиндром, были использованы две линейные структуры	с. стек и очередь	a. очередь и дерево b. стек и дерево c. стек и очередь d. очередь и граф
315.	Какие операции необходимо реализовать при разработке		a. Добавление нового элемента в заданную позицию, удаление

	программы управления отдельным элементом линейного списка		элемента из заданной позиции, доступ к элементу в заданной позиции b. Инициализация структуры c. Сортировка по различным полям структуры d. Уничтожение структуры																																								
316.	Какими значениями должны быть заполнены узлы бинарного дерева поиска, чтобы при поиске ключа возник наилучший случай	с. случайными	a. строго упорядоченными по убыванию b. строго упорядоченными по возрастанию с. случайными																																								
317.	Для реализации бинарное дерево поиска применяется структура хранения, обеспечивающая логарифмическую сложность алгоритму поиска	Вроде b. с указателями на сыновей Но надо проверить	a. массив родителей b. с указателями на сыновей c. список сыновей d. список сыновей и правых братьев																																								
318.	Какие из представленных деревьев - деревья поиска Множественный выбор 	Числа не видно на картинке, но отмечены варианты 2 и 3	2 3 1																																								
319.	В AVL дерево с последовательностью ключей 2,1,3,5 был вставлен ключ 4. Потребовалась ли перестройка дерева и если да, то каким алгоритмом	На самом корее всего двойной LR - double left	a. да, двойной RL b. да, одинарный правый с. да, двойной LR d. нет																																								
320.	?К-арное дерево (упорядоченно слева направо по порядку размещения узлов) с N=9 узлами и метками в виде букв реализовано таблицей (список левых сыновей и правых братьев), представленной на рисунке. Определите сыновей узла с меткой А. <small>К-арное дерево (упорядоченно слева направо по порядку размещения узлов) с N=9 узлами и метками в виде букв реализовано таблицей (список левых сыновей и правых братьев), представленной на рисунке. Стандартом сыновей sons и братьев A.</small> <table border="1" data-bbox="263 1870 558 2072"> <thead> <tr> <th>Индекс элемента</th> <th>Метка узла</th> <th>Левый сын</th> <th>Правый брат</th> </tr> </thead> <tbody> <tr><td>1</td><td>F</td><td>0</td><td>0</td></tr> <tr><td>2</td><td>B</td><td>6</td><td>4</td></tr> <tr><td>3</td><td>A</td><td>2</td><td>0</td></tr> <tr><td>4</td><td>C</td><td>7</td><td>5</td></tr> <tr><td>5</td><td>D</td><td>0</td><td>8</td></tr> <tr><td>6</td><td>E</td><td>0</td><td>1</td></tr> <tr><td>7</td><td>J</td><td>0</td><td>0</td></tr> <tr><td>8</td><td>K</td><td>9</td><td>0</td></tr> <tr><td>9</td><td>O</td><td>0</td><td>0</td></tr> </tbody> </table>	Индекс элемента	Метка узла	Левый сын	Правый брат	1	F	0	0	2	B	6	4	3	A	2	0	4	C	7	5	5	D	0	8	6	E	0	1	7	J	0	0	8	K	9	0	9	O	0	0	В C D K	A B E F B E F A B C K B C D K
Индекс элемента	Метка узла	Левый сын	Правый брат																																								
1	F	0	0																																								
2	B	6	4																																								
3	A	2	0																																								
4	C	7	5																																								
5	D	0	8																																								
6	E	0	1																																								
7	J	0	0																																								
8	K	9	0																																								
9	O	0	0																																								

321.	Какой метод построения алгоритма использован в алгоритме Дейкстры "Поиск кратчайшего пути в графе":	"Жадный алгоритм"	"Жадный алгоритм" Метод полного перебора Метод рекурсии Разделяй и властвуй
322.	Линейные списки находят применение в технологии:	распределённые системы блокчейн	реляционных баз данных, распределённые системы блокчейн, организации доменов Active Directory, символьный адресации DNS
323.	Определите теоретическую сложность функции роста времени алгоритма: for (int i=0;i<n;i++) for (int j=0; j<n; j++) cout << "hello"	квадратичная	квадратичная линейная логарифмическая константа
324.	Свойство алгоритма, обеспечивающее возможность его использовать для решения множества однотипных задач (применимость к различным наборам входных данных), называется:	Массовость	Дискретность Результативность Завершаемость Массовость
325.	Какая структура данных используется для хранения элементов данных в хеш-таблице, при реализации разрешения коллизий по методу цепочек (цепное хеширование)	Однонаправленный список	Однонаправленный список Очередь Дек Двоичное дерево
326.	Перестройка AVL дерева выполняется за счет двойного RL поворота, если после вставки ключа	а. фактор балансировки узла AVL дерева равен +2, фактор балансировки его правого сына -1	а. фактор балансировки узла AVL дерева равен +2, фактор балансировки его правого сына -1 б. фактор балансировки узла AVL дерева равен +2, фактор балансировки его правого сына +1 с. фактор балансировки узла AVL дерева равен -2, фактор балансировки его левого сына +1 д. фактор балансировки узла AVL дерева равен -2, фактор балансировки его правого сына -1
327.	Простой путь в графе	с. последовательность вершин графа, в которой каждая вершина	а. последовательность вершин графа, в которой каждая вершина

		соединена с предыдущей ребром и все вершины различны	соединена с предыдущей ребром b. последовательность вершин графа, в которой все вершины различны c. последовательность вершин графа, в которой каждая вершина соединена с предыдущей ребром и все вершины различны
328.	Какой бы вид графа вы выбрали при реализации задачи поиска маршрута с минимальным временем поездки в московском метро	b. Неориентированный взвешенный граф	a. Ориентированному ациклическому графу b. Неориентированный взвешенный граф c. Ориентированный взвешенный граф d. Ориентированный не взвешенный граф
329.	Имеется узел AVL-дерева для которого $hl > hr$, где hl, hr высота левого и правого поддерева соответственно. В левое поддерево левого сына вставляется новый ключ, после этого:	с. Требуется перестройка с одним правым поворотом	a. Не требуется перестройка дерева b. Требуется перестройка с одним левым поворотом с. Требуется перестройка с одним правым поворотом d. Требуется перестройка с двойным поворотом RL
330.	Для какого ориентированного графа алгоритм поиска в глубину сформирует глубинный лес	с. Когда после очередного вызова алгоритм поиска в глубину в графе еще остались не посещенные вершины	a. Когда граф связный b. Когда после очередного вызова алгоритм поиска в глубину в графе не осталось не посещенных вершин с. Когда после очередного вызова алгоритм поиска в глубину в графе еще остались не посещенные вершины d. Когда после очередного вызова алгоритм поиска в глубину в графе все вершины посещены
331.	Какой из представленных связных неориентированных графов является Эйлеровым (содержит цикл Эйлера) 	b. 1	a. 2 b. 1 c. 3

332.	Перестройка AVL дерева выполняется за счет только правого поворота		<p>a. фактор балансировки узла дерева, в которое вставляется ключ -1 и ключ вставляется в его левое поддереве</p> <p>b. фактор балансировки узла дерева, в которое вставляется ключ $+1$ и ключ вставляется в его левое поддереве</p> <p>c. фактор балансировки узла дерева, в которое вставляется ключ $+1$ и ключ вставляется в его правое поддереве</p> <p>d. фактор балансировки узла дерева, в которое вставляется ключ -1 и ключ вставляется в его правое поддереве</p>																																								
333.	В узлах бинарного дерева поиска ключи поступали в следующей последовательности: 25 12, 10, 8, 7, 5. Какой алгоритм обхода бинарного дерева поиска позволит вывести эту последовательность в упорядоченном по возрастанию порядке	d. Обратный	<p>a. Симметричный</p> <p>b. Прямой</p> <p>c. обход в ширину</p> <p>d. Обратный</p>																																								
334.	<p>K-арное дерево (упорядоченно слева направо по порядку размещения узлов) с $M > 0$ узлами и метками в виде букв реализовано таблицей (список левых сыновей и правых братьев), представленной на рисунке. Вывести простой путь при прямом обходе дерева методом в глубину с вершины с меткой A</p> <table border="1" data-bbox="263 1529 598 1758"> <thead> <tr> <th>Индекс элемента</th> <th>Метка узла</th> <th>Левый сын</th> <th>Правый брат</th> </tr> </thead> <tbody> <tr><td>1</td><td>F</td><td>0</td><td>0</td></tr> <tr><td>2</td><td>B</td><td>6</td><td>4</td></tr> <tr><td>3</td><td>A</td><td>2</td><td>0</td></tr> <tr><td>4</td><td>C</td><td>7</td><td>5</td></tr> <tr><td>5</td><td>D</td><td>0</td><td>8</td></tr> <tr><td>6</td><td>E</td><td>0</td><td>1</td></tr> <tr><td>7</td><td>J</td><td>0</td><td>0</td></tr> <tr><td>8</td><td>K</td><td>9</td><td>0</td></tr> <tr><td>9</td><td>O</td><td>0</td><td>0</td></tr> </tbody> </table>	Индекс элемента	Метка узла	Левый сын	Правый брат	1	F	0	0	2	B	6	4	3	A	2	0	4	C	7	5	5	D	0	8	6	E	0	1	7	J	0	0	8	K	9	0	9	O	0	0	d. A B E F C J D K O	<p>a. E B F A J C D O K</p> <p>b. E F B A J C D O K</p> <p>c. A B C D K E F J O</p> <p>d. A B E F C J D K O</p>
Индекс элемента	Метка узла	Левый сын	Правый брат																																								
1	F	0	0																																								
2	B	6	4																																								
3	A	2	0																																								
4	C	7	5																																								
5	D	0	8																																								
6	E	0	1																																								
7	J	0	0																																								
8	K	9	0																																								
9	O	0	0																																								
335.	В AVL дерево, содержащее ключи 8,10,14,11, был вставлен ключ 12. Потребовалась ли перестройка дерева и если да, то каким способом?	b. да, двойной RL	<p>a. нет</p> <p>b. да, двойной RL</p> <p>c. да, двойной LR</p> <p>d. да, одинарный правый</p>																																								
336.	Структура хранения данных, каждый элемент (ячейка) в которой явно	b. сеть	<p>a. список</p> <p>b. сеть</p>																																								

	хранит несколько связей с другими элементами структуры		с. массив d. вектор
337.	Что определяет поле "фактор балансировки" в структуре узла AVL дерева	с. разность высот левого и правого поддеревьев узла	a. глубину дерева с корнем в этом узле b. сумму высот левого и правого поддеревьев узла с. разность высот левого и правого поддеревьев узла d. Модуль разности высот левого и правого поддеревьев узла
338.	Что будет результатом применения алгоритма RLE - "Групповое кодирование" к сжатию текста AAAAAAAAAAAAAAAAAABBVVVAAAVV VVVVVC	b. 19A5B3A7B1C	a. (19)A(5)B(3)A(7)B(1)C b. 19A5B3A7B1C c. A19B5A3B7C1 d. 19A-5B3A-7B1C
339.	K-арное дерево (упорядоченно слева направо по порядку размещения узлов) с $N > 0$ узлами и метками от 1 до N реализовано списком сыновей. Список сыновей должен быть реализован	однонаправленный динамический список ОЧЕНЬ НЕТОЧНО	одномерный статический массив двумерный динамический массив однонаправленный динамический список одномерный динамический массив
340.	Что предусматривает метод динамического программирования	Переформулирование сложной задачи в виде рекурсивной последовательности более простых подзадач Метод динамического программирования сверху — это простое запоминание результатов решения тех подзадач, которые могут повторно встретиться в дальнейшем. Динамическое программирование снизу включает в себя переформулирование сложной задачи в виде рекурсивной последовательности более простых подзадач.	Переформулирование сложной задачи в виде рекурсивной последовательности более простых подзадач Получение решения исходно задачи путем комбинирования рекурсивных решений подзадач Оптимальное решение подзадач меньшего размера может быть использовано для решения исходной задачи Запоминание результатов решения тех подзадач, которые могут повторно использоваться
341.	Иерархическая структура данных	Это множество элементов (узлов) связи между которыми	это множество элементов (узлов), связи между которыми определяется на основе зависимости

		определяются на основе уровней	многие-ко-многим это множество элементов (узлов), в которой реализуются связи между элементами одного уровня Это множество элементов (узлов) связи между которыми определяются на основе уровней это множество элементов (узлов) связи между которыми определяются на основе линейной зависимости
342.	Когда один алгоритм считается эффективнее другого	если время его работы в наихудшем случае имеет более низкий порядок роста	если время его работы в наихудшем случае имеет более низкий порядок роста если время его работы в наилучшем случае имеет более низкий порядок роста если время его работы в наихудшем случае имеет более высокий порядок роста если время его работы в наилучшем случае имеет более высокий порядок роста
343.	Чтобы алгоритм Дейкстры сформировал помимо длины кратчайшего пути от вершины a к вершине b и сам путь в графе, предусматривают запись вершин в линейную структуру. В каком последовательности расположатся вершины пути в структуре	Последовательно от вершины b до a	Последовательно от вершины a до b Последовательно от вершины b до a Упорядоченно по возрастанию В случайном порядке
344.	Бинарное дерево поиска построено из ключей, которые поступали в следующей последовательности: 25, 12, 13, 10, 11, 7, 44, 50, 42, 30, 40. Алгоритм обхода бинарного дерева вывел следующую последовательность ключей: 25, 12, 44, 10, 13, 42, 50, 7, 11, 30, 40. Какой алгоритм обхода бинарного дерева поиска при этом выводе был использован	Обход в ширину	Прямой обход в глубину Обход в ширину Обратный обход в глубину Обход в глубину симметричный

345.	Внутренний узел дерева — это узел который	Имеет сыновей	Имеет двух родителей Имеет степень 0 Имеет сыновей
346.	Количество путей в дереве определяется	Степенью узла?	Количеством предков дерева Количеством левых братьев Количеством внутренних узлов Степенью узла дерева
347.	Какое количество узлов в полном бинарном дереве высотой 4?	31	48 31 64 32
348.	Какую операцию можно использовать в качестве хеш-функции для целочисленных ключей?	Операцию взятия остатка от деления ключа на размерность массива	Операцию деления ключа на размерность массива Операцию взятия остатка от деления размерности массива на ключ Операцию умножения ключа на размерность массива Операцию взятия остатка от деления ключа на размерность массива