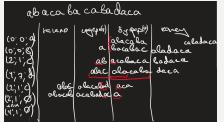
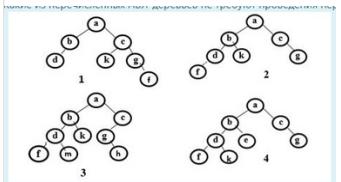


ВОПРОС:	ВАРИАНТЫ ОТВЕТОВ	ОТВЕТ	КОММЕНТАРИЙ		
Предфиксный код в теории кодирования - это:	а. кодовое слово фиксированной длины, с которого не может начинаться другое кодовое слово б. кодовое слово переменной длины, с которого не может начинаться другое кодовое слово того же кода с. кодовое слова переменной длины, с которого может начинаться другое кодовое слово д. кодовое слово фиксированной длины, с которого может начинаться другое кодовое слово	b	точно не а и не д		
Что определяет структура данных:	а. Множество данных и отношений между ними б. Множество отношений между данными в форме операций над ними в. Множество данных г. Множество данных и множество операций над ними	a	approved by Tony		
Какой зависимостью описывается функция вычислительной сложности алгоритма быстрой сортировки методом Хоара (quicksort) в среднем наилучшем случае?	а. $f(n)=\Theta(n)$ б. $f(n)=\Theta(n^2)$ в. $f(n)=\Theta(\log(n))$ г. $f(n)=\Theta(n \log(n))$	d	в википедии написанно $O(n \log n)$, но надо будет перепроверить		
Какой код будет получен из входной строки символов 1111223444 при использовании алгоритма группового кодирования RLE (Run Length Encoding),	а. 4 1 2 2 1 3 3 4 б. (4)1 (2)2 (1)3 (3)4 в. 1 4 2 2 3 1 4 3 г. 0 0 0 0 0 1 0 1 10 11 11 11	a	тест 38 из 50		
Какой зависимостью описывается функция вычислительной сложности алгоритма поиска в тексте по образцу методом Кнута-Морриса-Пратта в лучшем случае?	а. $f(n,m)=\Theta(\log(n+m))$ б. $f(n)=\Theta(n)$ в. $f(n)=\Theta(n^2)$ г. $f(n, m)=\Theta(\log(n*m))$	b	Алгоритм Кнута, Морриса и Пратта (КМП-алгоритм) является алгоритмом, который фактически требует только $O(n)$ сравнений даже в самом худшем случае. ВРОДЕ ТАК НО ЛУЧШЕ ПРОВЕРЬТЕ (extremal approved by Tony)		
В какой последовательности располагаются вершины дерева при прямом обходе (preorder):	а. 8, 9, 4, 10, 11, 5, 2, 12, 13, 6, 14, 15, 7, 3, 1 б. 8, 4, 9, 2, 10, 5, 11, 1, 12, 6, 13, 3, 14, 7, 15 в. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 г. 1, 2, 4, 8, 9, 5, 10, 11, 3, 6, 12, 13, 7, 14, 15	d	точно не а и не б так как прямой обход начинается с корня. Скорее всего ответ d ибо надо сначала пройтись по левому поддереву. Да, ответ точно d. (approved by Antony)		
Как определить есть ли в неориентированном связном графе Эйлеров цикл?	а. в графе все вершины имеют нечетную степень б. в графе только две вершины, которые имеют четную степень в. в графе все вершины имеют четную степень г. в графе все вершины, которые имеют нечетную степень	c	Эйлеров цикл в связном неориентированном графе G (X, E) существует только тогда, когда все его вершины имеют четную степень. Источник: https://studopedia.ru/3_104227_eylerovi-tsperi-i-tsikli.html		
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), (1,3,2,5,4), (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)	а. Простого выбора б. Простой вставки в. Шелла г. Простого обмена	d			
Какой зависимостью описывается функция вычислительной сложности алгоритма быстрой сортировки методом Шелла в наихудшем случае:	а. $f(n)=O(n \log(n))$ б. $f(n)=O(\log(n^0))$ в. $f(n)=O(n)$ г. $f(n)=O(n^2)$	d	Из теста (38/50) ответ $f(n) = O(n^2)$. Скорее всего правильно (approved by Tony)		
Что будет выведено при прямом обходе дерева?	а. d b e a f c б. b d e c f 1 в. a b d e c f г. a b c d e f	c	В пункте b почему то в конце "1", по идеи там должно быть: "b d e c f a" (approved by Tony)		

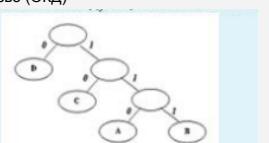
ВОПРОС:	ВАРИАНТЫ ОТВЕТОВ	ОТВЕТ	КОММЕНТАРИЙ		
Рекурсия имеет место, если решение задачи сводится к разделению её на меньшие подзадачи, выполняемые с помощью одного и того же алгоритма. Когда должен завершиться прави...	a. Когда завершится ввод исходных данных b. Когда достигаются простейшее возможное решение c. Когда он будет остановлен пользователем d. До заполнения стека	a	Скриншот задачи обрезан :(
К графу на рисунке 1 применен алгоритм обхода в ширину. Какое из оставших деревьев соответствует этому алгоритму обхода с вершины 3?	a. 5 b. 4 c. 3 d. 2	c	Вроде оно		
Укажите правильную аналогию между асимптотическим сравнением двух функций f и g для обозначения $f(n) = O(g(n))$ и сравнением двух действительных чисел a и b :	a. $a = b$ b. $a < b$ c. $a \geq b$ d. $a \leq b$	d			
Какой зависимостью описывается функция вычислительной сложности алгоритма сортировки методом простого обмена (с условием Айверсона) в наилучшем случае:	a. $f(n)=O(n^2)$ b. $f(n)=O(\log(n))$ c. $f(n)=O(n \log(n))$ d. $f(n)=O(n)$	d			
Укажите асимптотическую сложность операции удаления элемента в позиции i из массива размером n :	a. $O(n^2)$ b. $O(\log n)$ c. $O(1)$ d. $O(n)$	d	https://habr.com/ru/post/188010/		
Какие основные операции следует учитывать при оценке временной (вычислительной) сложности алгоритмов сортировки:	a. Все имеющиеся операции в алгоритме b. Операции сравнения данных c. Операции сравнения и перемещения данных d. Операции перемещения данных	c			
Пример какой структуры данных изображён на рисунке?	a. стек b. очередь c. кольцо d. дек	d	вот такую картинку выдал браузер на запрос что такое дек		
Что такое инвариант цикла:	a. Логическое выражение, истинное после первого прохода тела цикла и в конце выполнения цикла, зависящее от переменных, изменяющихся в теле цикла b. Логическое выражение, истинное после каждого прохода тела цикла и перед началом выполнения цикла, зависящее констант, изменяющихся в теле цикла c. Логическое выражение, истинное после первого прохода тела цикла, и в конце выполнения цикла, зависящее от переменных, изменяющихся в теле цикла d. Логическое выражение, истинное перед началом цикла, после каждого прохода тела цикла, и в конце выполнения цикла зависящее от переменных, изменяющихся в теле цикла	b	Логическое выражение, истинное после каждого прохода тела цикла и перед началом выполнения цикла, зависящее констант, изменяющихся в теле цикла		
Какое из перечисленных АВЛ-деревьев требует балансировки?	a. 4 b. 1 c. 2 d. 3	d			

ВОПРОС:	ВАРИАНТЫ ОТВЕТОВ	ОТВЕТ	КОММЕНТАРИЙ																																																																																	
Дана следующая рекурсивная функция: <pre>int fib(int n) { if(n < 2) { return 1; } else { return fib(n-2) + fib(n-1); } }</pre> Какова глубина рекурсии при вызове fib(5)?	a. 5 b. 4 c. 7 d. 3	b	в ответах указано С, но если считать ручками, то получается ответ А. но в итоге если верить конспектам скворцовой за прошлый семестр получается (n-1) = 4																																																																																	
Каков главный недостаток хэш-таблиц?	a. Фиксированный размер, при неизвестном окончательном числе элементов b. Фиксированное число элементов, располагающихся в таблице c. В трудеёмкой функции хэширования d. Отказ от повторного хэширования	a	было написано С, но на функцию должно быть пофиг, это всего лишь функция, полагаю, что это фиксированный размер при неизвестном окончательном числе элементов, ибо логичнее и практичесче создать таблицу из всех элементов, чем потом ебаться с рехешированием																																																																																	
Какой зависимостью описывается функция вычислительной сложности алгоритма поиска по бинарному дереву поиска (binary search tree, BST) в лучшем случае?	a. $f(n)=O(n \log(n))$ b. $f(n)=O(n^2)$ c. $f(n)=O(\log(n))$ d. $f(n)=O(n)$	c																																																																																		
Что хранит матрица смежности графа?	a. Список ребер и их направления b. Длины путей между вершинами графа c. Список вершин d. Только направления ребер	a	по определению матрицы смежности графа, в ней хранятся длины рёбер между вершинами, а длина пути между вершинами графа - это длина ребра. но по матрице смежности можно определить ориентированность (допустим, путь из A в B существует, а из B в A - нет). формулировка ебнутая, нужно перепроверить																																																																																	
Что означает утверждение, что алгоритм X асимптотически более эффективен, чем алгоритм Y?	a. X будет лучшим выбором для любых входных данных, кроме очень больших b. X будет лучшим выбором для любых данных, начиная с некоторого граничного значения c. X будет лучшим выбором для любых входных данных d. X будет лучшим выбором для любых входных данных до некоторого граничного значения	b																																																																																		
Какой зависимостью описывается функция вычислительной сложности алгоритма поиска в ширину в неориентированном графе, построенном на списке смежных вершин, если V - количество вершин в графе, а E - количество ребер.	a. $f(n)=O(V)$ b. $f(n)=O(\log(E))$ c. $f(n)=O(V^2)$ d. $f(n)=O(E)$	a	BFS Algorithm Complexity The time complexity of the BFS algorithm is represented in the form of $O(V + E)$, where V is the number of nodes and E is the number of edges. The space complexity of the algorithm is $O(V)$.																																																																																	
Система байтового кодирования символов ASCII кодирует символы входной последовательности кодами:	a. фиксированной длины в один байт b. переменной длины свыше одого байта c. переменной длины до одного байта d. хэш-кодами	a	ну, символ по аски обычно кодируется 7 битами (8 бит служит для отслеживания ошибок или другой какой-то чернухи - пишет википедия), поэтому, наверное, ответ А - длина каждого символа фиксирована и составляет 1 байт (мы же не знаем наверняка, сколько битов понадобится для кодирования единичного символа, поэтому мы берём максимум)																																																																																	
В теории вычислимости важную роль играет функция Аккермана $A(m, n)$, определённая следующим образом:	-	7	Тут нет вариантов ответа, тут нужно получить свой ответ и ввести его в формуочку. Проверено в компиляторе.																																																																																	
			<table border="1"><tr><td>A</td><td> </td><td>n=0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr><tr><td>m=0</td><td> </td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td></tr><tr><td>1</td><td> </td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td></tr><tr><td>2</td><td> </td><td>3</td><td>5</td><td>7</td><td>9</td><td>11</td><td>13</td><td>15</td><td>17</td><td>19</td><td>21</td><td>23</td></tr><tr><td>3</td><td> </td><td>5</td><td>13</td><td>29</td><td>61</td><td>125</td><td>253</td><td>509</td><td>1021</td><td>2045</td><td>4093</td><td>8189</td></tr><tr><td>4</td><td> </td><td>13</td><td>65533</td><td>big</td><td>really big...</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	A		n=0	1	2	3	4	5	6	7	8	9	10	m=0		1	2	3	4	5	6	7	8	9	10	11	1		2	3	4	5	6	7	8	9	10	11	12	2		3	5	7	9	11	13	15	17	19	21	23	3		5	13	29	61	125	253	509	1021	2045	4093	8189	4		13	65533	big	really big...										
A		n=0	1	2	3	4	5	6	7	8	9	10																																																																								
m=0		1	2	3	4	5	6	7	8	9	10	11																																																																								
1		2	3	4	5	6	7	8	9	10	11	12																																																																								
2		3	5	7	9	11	13	15	17	19	21	23																																																																								
3		5	13	29	61	125	253	509	1021	2045	4093	8189																																																																								
4		13	65533	big	really big...																																																																															
Основное требование, предъявляемое к массиву для возможности выполнения двоичного поиска:	a. Статический массив b. Массив целочисленных значений c. Массив небольшого размера d. Упорядоченность массива	d																																																																																		
Как называется алгоритм, который выполняет 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)	a. Простого обмена b. Пирамидальная c. Простой вставки d. Простого выбора	c																																																																																		
Линейные списки находят применение в технологиях:	a. распределенных систем блокчейн b. организация доменов Active Directory c. реляционных баз данных d. символьной адресации узлов DNS	a	хз где они там находят применение, поскольку это очень неудобная хуйня, где невозможно напрямую по индексу обратиться к элементам списка (только лишь пройдя N элементов, мы сможем вытащить конкретно нам нужный элемент)																																																																																	

ВОПРОС:	ВАРИАНТЫ ОТВЕТОВ	ОТВЕТ	КОММЕНТАРИЙ		
Прямой доступ к элементу данных в памяти делает возможным	a. индикатор элемента в программе b. порядковый номер элемента в структуре c. любая списочная структура d. адрес элемента в ОЗУ	b			
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?	a. 4 b. 5 c. 7 d. 6	d			
Определите теоретическую вычислительную сложность (функцию роста времени) алгоритма: <pre>int count = 1; while(count < n) { count = count*2; /* Последовательность шагов программы с временной сложностью O (1) */}</pre>	a. линейная b. константная c. квадратичная d. логарифмическая	b			
В AVL-дерево, содержащее ключи 8, 10, 11, 14, был вставлен ключ 12. Потребовалась ли перестройка дерева, если да, то каким способом?	a. да, двойной LR b. нет c. да, одинарный правый d. да, двойной RL	d	https://www.cs.usfca.edu/~qalles/visualization/AVLtree.html		
Выберите код, являющийся результатом сжатия по методу скользящего окна текста "abacabacabada". Примечание: кавычки ограничивают текст и не кодируются.	a. (0,0,a)(0,0,b)(2,1,c)(4,7,d)(2,1,c)(4,1,пусто) b. (0,0,a)(0,0,b)(2,1,c)(4,4,d)(2,1,c)(2,1,пусто) c. (0,0,a)(0,0,b)(2,1,c)(4,7,d)(2,1,c)(2,1,пусто) d. (0,0,a)(0,0,b)(2,1,c)(3,7,d)(2,1,c)(2,1,пусто)	c		формально, здесь 2 правильных ответа (a c), но хз Точно с.	https://neerc.ifmo.ru/wiki/index.php?title=Алгоритмы_L77_и_L78
Какая операция считается недопустимой для линейного односвязанного списка:	a. Вставка нового элемента после заданного элемента b. Удаление последнего элемента c. Замена содержимого информационной части заданного элемента d. Вставка нового элемента перед заданным элементом	d			
Имеется идеально сбалансированное двоичное дерево (не являющееся деревом поиска), содержащее целые числа. Просмотр дерева слева-направо (in-order обход) даёт следующий результат: 2,4,6,8,10,12,14. Какойузел является корнем дерева?	a. 8 b. 10 c. 2 d. 6	a	из теста 38 50 ответ a, из того же документа есть шанс что ответ с (2); правильный ответ А проверено симметричным обходом		
Есть ли ошибка в коде функции f, которая осуществляет получение ссылки на последнийузел линейного односвязанного 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>	a. Ошибка оператора Tnode *q = L - должно быть Tnode q=L b. Ошибки нет c. Ошибка в записи оператора q=q->next d. Ошибка в условии while(q) - должно быть while(q->next)	d	Из теста 38 50		
Как называется алгоритм, который напрямую или через другие вспомогательные алгоритмы вызывает сам себя:	a. Линейный b. Разветвляющийся c. Рекурсивный d. Циклический	c			
В каких случаях алгоритм Кнута-Морриса-Пратта дает подлинный выигрыш по сравнению с другими алгоритмами поиска в тексте по образцу?	a. На больших алфавитах b. Когда образец предварительно адаптирован c. Когда текст предварительно адаптирован d. Когда неудачному сравнению образца с текстом предшествовало некоторое число совпадений	d			

ВОПРОС:	ВАРИАНТЫ ОТВЕТОВ	ОТВЕТ	КОММЕНТАРИЙ			
Какую операцию можно использовать в качестве хэш-функции для целочисленных ключей?	a. Операцию умножения ключа на размерность массива b. Операцию взятия остатка от деление размерности массива на ключ c. Операцию деления ключа на размерность массива d. Операцию взятия остатка от деления ключа на размерность массива	d	ну, обычно берут в качестве хеш-функции следующую формулу: $\text{hash}(X) = X \% \text{TABLE_SIZE}$, так что думаю, что правильный ответ - d (операция взятия остатка от деления ключа на размерность массива); нужен аппрув другого человека, вдруг я не прав (+ согласен, с тем что это d)			
В чем смысл анализа алгоритма?	a. Определить требования к интерфейсу b. Обеспечить безопасность данных c. Предсказать формат структур данных и внешние связи между ними d. Предсказать требуемые для его выполнения вычислительные ресурсы	d				
Какая структура данных относятся к категории линейных списков?	a. Массив b. Множество c. Бинарное дерево d. Дек	d				
К недостаткам рекурсивного метода можно отнести?	<input type="checkbox"/> а. возможность переполнения стековой памяти программного процесса <input type="checkbox"/> б. меньшая точность результата вычислений <input type="checkbox"/> в. расход времени на выделение и очистку стекового кадра в памяти <input type="checkbox"/> г. возможность переполнения динамически распределяемой памяти программного процесса	a,c	Множественный выбор			
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 какой последовательности расположатся вершины пути в структуре?	a. Упорядоченно по возрастанию вершин b. Последовательно от вершины b до a c. В любом порядке d. Последовательно от вершины a до b		 ХИЙ СОН	Пиздец какой то а не вопрос, никто даже в душе не ёбет за такие вопросы(тыкайте на рандом)		
Какой вычислительной сложностью характеризуется алгоритм "грубой силы" (brute force attack) прямого поиска по образцу?	a. $O(n \log m)$ b. $O(n^m)$ c. $O(n+m)$ d. $O(n^2)$	d				
Что необходимо делать с большим объёмом информации, связанной с каждым ключом:	a. Чаще рехешировать b. Не хранить ее в хеш-таблице c. Увеличить хеш-таблицу d. Хранить ее в хеш-таблице	c	из теста 38 50			
Какое из перечисленных АВЛ-деревьев требует проведения перестройки?	a. 1, 2, 4 b. 1, 2, 3 c. 2, 3, 4 d. 1, 3, 4	b				
						
Дано описание функции: int f(int x) { return (x==0)? 1 : x * f(x-1); } Данная функция вычисляет:	a. Степень числа n b. n-ое простое число c. факториал числа n d. n-ое число Фибоначчи	c				
Массив длиной n сортируется пузырьковым алгоритмом по неубыванию. За сколько проходов по массиву наименьшее значение массива окажется "вверху"?	a. n b. n-1 c. n^2 d. Один	c				

ВОПРОС:	ВАРИАНТЫ ОТВЕТОВ	ОТВЕТ	КОММЕНТАРИЙ		
Какой зависимостью описывается функция вычисления сложности алгоритма пирамидальной сортировки (Heapsort) в среднем случае?	a. $f(n)=O(n)$ b. $f(n)=O(\log(n))$ c. $f(n)=O(n \log(n))$ d. $f(n)=O(n^2)$	c	Пирамидальная сортировка — алгоритм сортировки, работающий в худшем, в среднем и в лучшем случае (то есть гарантированно) за O(n log n) операций при сортировке n элементов. <i>точно C, поскольку сложности в среднем и наихудшем случаях у пирамидальной сортировки совпадают</i>		
Структура данных, работа с элементами которой организована по принципу FIFO (первый пришел - первый ушел), - это:	a. очередь b. дек c. стек d. массив	a			
Для динамической структуры данных (переменной) справедливо утверждение:	a. создается на этапе выполнения программы и размер может быть изменен на этапе выполнения b. создается на этапе компиляции кода и размер может быть изменен во время выполнения программы c. создается на этапе компиляции кода и размер не может быть изменен во время выполнения программы d. создается на этапе выполнения программы и размер не может быть изменен на этапе выполнения	a			
К графу на рисунке 1 применен алгоритм обхода в глубину. Какое из остовных деревьев соответствует обходу в глубину с вершины 3?	a. 3 b. 5 c. 4 d. 2	b			
Какой зависимостью описывается функция вычисления сложности алгоритма быстрый сортировки методом Хоара (quicksort) в наихудшем случае?	a. $f(n) = O(\log(n))$ b. $f(n) = O(n^2)$ c. $f(n) = O(n \log(n))$ d. $f(n) = O(n)$	b	Из теста (38/50) ответ: b		
Отличительная особенность алгоритма Бойера-Мура	a. Сравнение символов производится начиная с конца образца b. После каждого неудачного сравнения производится сдвиг образца вправо на одну позицию c. Сравнение символов производится начиная с конца текста d. Производится посимвольное сравнение образца с текстом	a	Из теста (38/50) ответ: a		
Когда достигается максимальная эффективность алгоритма Бойера-Мура:	a. Если образец длинный, а мощность алфавита достаточно велика b. Если образец короткий, а мощность алфавита достаточно низка c. Если образец короткий, а мощность алфавита достаточно велика d. Если образец длинный, а мощность алфавита достаточно низка	a	Из теста (38/50) ответ: a		
Имеется описание структуры узла линейного односвязанного списка struct Tnode{Tdata data; XXXX next; } Какое определение должно быть на месте XXXX:	a. void b. Tnode c. *Tnode d. next	c	Из теста 38 50, уверен что правильно.		
Для каких узлов необходимо проверить коэффициент балансировки после добавления узла в АВЛ-дерево:	a. Для родителя нового узла b. Для всех предков нового узла вплоть до корня дерева c. Только для нового добавленного узла d. Для всех узлов дерева	b	из теста 38 50; <i>точно b</i>		
При каком значении показателя сбалансированности b (разницы высот правого и левого поддеревьев) необходимо выполнить перестройку АВЛ-дерева:	a. $ b = 2$ b. $ b = 1$ c. $ b > 2$ d. $ b > 1$	d	из теста		

ВОПРОС:	ВАРИАНТЫ ОТВЕТОВ	ОТВЕТ	КОММЕНТАРИЙ		
	а. Указатель на начало списка ключей с одним и тем же значением хеш функции б. Адрес следующего элемента списка с. Ключ элемента д. Значение хеш-функции	a	из теста		
Сколько байт бинарного файла занимает число 0 в двоичном формате:	а. 8 байт б. 4 байта в. 1 байт г. 0 байт	c			
Что такое структурированный протокол:	а. Тип сохраненного кластера б. Простейший способ записи кластеров данных в файл в. Простой формат файлов для доступа из любой среды г. Сетевые правила для миграции бинарных файлов	b			
Алгоритм Прима - это:	а. Алгоритм поиска минимального остовного дерева возвешенном неориентированном связном графе б. Алгоритм для нахождения кратчайших путей от одной из вершин графа до всех остальных в. Алгоритм для нахождения кратчайших путей всеми вершинами взвешенного ориентированного графа г. Алгоритм, указывающий несколько путей обработки одних и тех же входных данных, - без какого-либо уточнения, какой именно вариант будет выбран	a			
Какая пространственная сложность описывает матрицу смежности графа как функции от числа n вершин и числа m ребёр:	а. O(n+m) б. O(n) в. O(n*m) г. O(n^2)	d			
Остовное дерево - это:	а. Подграф, полученный путем кратчайшего обхода всех вершин графа б. Дерево, полученное путем удаления вершин, связанных с более чем половиной всех вершин в графе в. Подграф, полученный путем удаления максимального числа ребер, без нарушения связности г. Подграф, полученный путем удаления вершин, связанных со всеми остальными	c			
Имеется следующее оптимальное кодовое дерево (ОКД)	 Выберите исходную строку, которая была закодирована в битовый код 11100101001100100 по алгоритму Хаффмана с использованием приведенного ОКД:	b			
Какая вычислительная сложность характерна алгоритмам со стратегией "разделяй и властвуй" (divide and conquer approach):	а. O(1) б. O(n logn) в. O(n^2) г. O(n)	b			
Что будет результатом применения алгоритма RLE ("групповое кодирование") к сжатию текста	а. 19A5B3A7B1C б. (22)A(12)B(1)C в. ABABC(19,5,3,7,1) г. ABC(19,5,3,7,1)	a	из левого теста (может быть не правильно)		

ВОПРОС:	ВАРИАНТЫ ОТВЕТОВ	ОТВЕТ	КОММЕНТАРИЙ			
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?	a. 15 b. 18 c. 16 d. 17	d	из левого теста (может быть не правильно) Проверено на сайте: https://www.cs.usfca.edu/~galles/visualization/BST.html			
Нелинейными называются структуры данных, в которых:	a. связи между элементами не зависят от выполнения какого-либо условия b. связи между элементами не зависят от линейной упорядоченности элементов c. связи между элементами зависят от выполнения определенного условия d. связи между элементами не зависят от упорядоченности значений элементов	c	из левого теста (может быть не правильно)			
Из числа приведенных сортировок выберите наименее эффективную на больших массивах:	a. выбором b. пирамидальная c. пузырьковая d. вставками	c	из левого теста (может быть не правильно)			
Как определить, что функция реализует каскадную (множественную) рекурсию:	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	a	из левого теста (может быть не правильно)			
Какие отношения между элементами неважно поддерживает структура данных массив:	a. Нелинейные b. Иерархические c. Наследование d. Линейные	d	из левого теста (может быть не правильно)			
Какие отношения между элементами поддерживает структура данных бинарное дерево?	a. Сложные b. Бинарные c. Двоичные d. Иерархические	d	из левого теста (может быть не правильно)			
Количество ребёр в остовном дереве (n - количество вершин графа) равно:	a. n-2 b. n-1 c. n+1 d. n	b	из левого теста (может быть не правильно)			
Укажите свойство, характеризующее структуру данных Дек:	a. Вставка нового элемента возможна в заданную позицию b. Линейный список с заданным количеством узлов c. Удаление элемента возможно из заданной позиции d. Обладает двумя первыми элементами	d	из левого теста (может быть не правильно)			
Что определяет термин "чёрная высота" узла в структуре красно-чёрного дерева?	a. Количество узлов на пути от корня до ближайшего чёрного узла b. Длина пути от корня до ближайшего чёрного узла c. Количество чёрных узлов на пути от корня до листового узла d. Количество чёрных узлов в красно-чёрном дереве	c	из левого теста (может быть не правильно)			
Какие основные операции следует учитывать при оценке временной (вычислительной) сложности алгоритмов поиска:	a. Операции перемещения данных b. Операции сравнения и перемещения данных c. Операции сравнения данных d. Все имеющиеся операции в алгоритме	b				
Какой зависимостью описывается функция вычисления сложности алгоритма бинарного (двоичного) поиска в худшем случае?	a. $f(n)=\Theta(n^2)$ b. $f(n)=\Theta(\log(n))$ c. $f(n)=\Theta(n)$ d. $f(n)=\Theta(n \log(n))$	b				

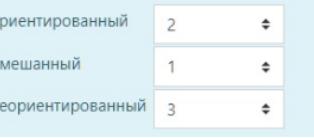
ВОПРОС:	ВАРИАНТЫ ОТВЕТОВ	ОТВЕТ	КОММЕНТАРИЙ			
Какой зависимостью описывается функция вычислительной сложности алгоритма поиска ключа в бинарном дереве поиска в наихудшем случае?	a. $f(n)=O(n)$ b. $f(n)=O(n \cdot \log(n))$ c. $f(n)=O(\log(n))$ d. $f(n)=O(1)$	a	https://ru.wikipedia.org/wiki/Двоичное_дерево_поиска			
Суть алгоритма сортировки методом Шелла заключается в:	a. Чередовании проходов по сортируемому массиву слева-направо и справа-направо b. Отдельной сортировке частей массива относительно опорного элемента c. Отдельной сортировке элементов, отстоящих друг от друга на расстоянии h , уменьшающимся на каждом проходе массива до значения 1 d. Попарным сравнением элементов массива с целью выбора наименьшего	c	Идея метода заключается в сравнение разделенных на группы элементов последовательности, находящихся друг от друга на некотором расстоянии. Изначально это расстояние равно d или $N/2$, где N — общее число элементов. На первом шаге каждая группа включает в себя два элемента расположенных друг от друга на расстоянии $N/2$; они сравниваются между собой, и, в случае необходимости, меняются местами. На последующих шагах также происходит проверка и обмен, но расстояние d сокращается на $d/2$, и количество групп, соответственно, уменьшается. Постепенно расстояние между элементами уменьшается, и на $d=1$ проход по массиву происходит в последний раз. Источник: https://forkettle.ru/vidioteka/programmirovaniye-i-set/algoritmy-i-struktury-dannykh/108-sorirovka-i-poisk-dlya-chajnikov/1008-sorirovka-shella			
Как называется алгоритм поиска минимального остовного дерева во взвешенном неориентированном связном графе?	a. Алгоритм Крускала b. Алгоритм Белмана-Форда c. Алгоритм Дейкстры d. Алгоритм Флойда-Уоршала	a	https://ru.wikipedia.org/wiki/Минимальное_остовное_дерево			
Выберите неверный способ представления графа	a. Матрицей смежности; b. Матрицей инцидентности; c. Списком смежности; d. Матрицей замыкания; e. Массивом дуг;	d	Из файла			
Классический алгоритм Дейкстры предназначен для	a. нахождения кратчайшего расстояния между двумя вершинами ациклического графа при отсутствии дуг отрицательного веса b. нахождения кратчайшего расстояния между двумя вершинами в ациклическом взвешенном графе c. нахождения кратчайшего расстояния между двумя вершинами только в циклическом взвешенном графе d. нахождения кратчайшего расстояния между двумя вершинами в произвольном взвешенном графе при отсутствии дуг отрицательного веса e. нахождения расстояния между всеми вершинами в произвольном взвешенном графе	d	Из файла			
Задача нахождения кратчайшего расстояния между двумя вершинами графа в случае отсутствия дуг отрицательного веса решается	a. используя алгоритм Дейкстры b. используя алгоритм Флойда c. 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)	c	Из файла			
Выберите неверный термин (тот, который не применим к графикам)	a. Ориентированный граф; b. Двудольный граф; c. Ациклический граф; d. Все термины верны; e. Циклический граф;	d	Из файла			

ВОПРОС:	ВАРИАНТЫ ОТВЕТОВ	ОТВЕТ	КОММЕНТАРИЙ			
Центром орграфа $G = (V, E)$ называется (множественный выбор)	a. вершина, для которой максимальное расстояние (длина пути) до (от) других вершин минимально b. дуга с максимальным эксцентриситетом. c вершина с максимальным эксцентриситетом. d. вершина с минимальным эксцентриситетом. e. дуга с минимальным эксцентриситетом. f. вершина, для которой минимальное расстояние (длина пути) до (0/1) других вершин максимально	a, d	Из файла			
Укажите свойства свободных деревьев (множественный выбор)	a. При добавлении в свободное дерево нового ребра образуется цикл. b. Свободное дерево представляет собой дерево без корня, ориентированное к листьям c. Каждое свободное дерево является полносвязанным графом d. Каждое свободное дерево с числом вершин n , $n > 1$, имеет в точности $n - 1$ ребер. e. Сумма весов ребер свободного дерева наименьшая на множестве ребер графа	a, d	Из файла			
В глубинном оством лесу, построенном при обходе орграфа $G = (V, E)$, различают (выделяют) (множественный выбор)	a. сильносвязанные компоненты глубинного оствомого леса b. обратные дуги c. поперечные дуги d. прямые дуги e. дуги дерева f. паросочетания дуг глубинного оствомого леса	b, c, d, e	Из файла			
Какое требование предъявляется к спецификации алгоритма:	a. Спецификация алгоритма должна предоставлять вычисление до сотой составляющей b. Спецификация алгоритма должна предоставлять точное вычисление c. Спецификация алгоритма должна предоставлять точное описание процедуры, которую требуется выполнить d. Спецификация алгоритма должна предоставлять примерное описание процедуры, которую требуется выполнить	c	файл тест 38 из 50			
Что проверяет условие Айверсона в алгоритме сортировки методом простого обмена:	a. Наличие сравнений в текущем проходе массива b. Наличие сравнений и обменов в текущем проходе массива c. Количество сравнений и обменов в текущем проходе массива d. Наличие обменов в текущем проходе массива	d	файл тест 38 из 51			
Какие операции необходимо реализовать при разработке программы управления отдельным элементом линейного списка:	a. Уничтожение структуры b. Сортировка по различным полям структуры c. Добавление нового элемента в заданную позицию, удаление элемента из заданной позиции, доступ к элементу в заданной позиции d. Инициализация структуры	c				
Что является результатом теоретического подхода к определению временной сложности алгоритма:	a. количество основных операторов, выполняемых алгоритмом b. количество операторов цикла c. процесорное время выполнения алгоритма	a				
Лист дерева это узел, который	a. имеет более двух потомков b. имеет степень 0 c. является внутренним узлом дерева d. имеет только правого потомка	b				
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:	a. Разделей и властвуй b. "Жадный" алгоритм c. Метод перебора возвратом d. Метод рекурсии	b				

ВОПРОС:	ВАРИАНТЫ ОТВЕТОВ	ОТВЕТ	КОММЕНТАРИЙ			
Структура данных массив отображается на физическом уровне в структуру хранения:	a. вектор b. сеть c. массив d. список	d				
Какая идея положена в основу построения алгоритма методом "жадного алгоритма"	a. На каждом этапе осуществляется разбиение задачи на подзадачи, которые могут быть оптимальными в данный момент b. На каждом этапе осуществляется выбор решения, которое кажется оптимальным в данный момент c. На каждом этапе осуществляется выбор решения, которое является не оптимальным в данный момент d. На каждом этапе осуществляется разбиение задачи на подзадачи, которые могут быть не оптимальным в данный момент	b				
Алгоритм Прима поиска (построения) оствновного дерева минимальной стоимости связзанного взвешенного не ориентированного графа	a. Строит оствновое дерево, применяя для поиска минимальных по стоимости ребер алгоритм Флойда и собирает из них оствновое дерево b. Строит оствновое дерево, применяя для поиска минимальных по стоимости ребер алгоритм Дейкстры и собирает из них оствновое дерево c. Строит оствновое 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 оствновое дерево, путем создания набора связанных подграфов и объединения их в оствновое дерево	c	by pashecka :)			
Алгоритм вставки ключа в рандомизированное дерево	a. определяет вставить ключ в лист дерева, или в корень b. ключ срачу вставляется в корень, а затем дерево перестраивается c. если ключа нет в дереве, то он вставляется по правилу бинарного дерева поиска и дерево не перестраивается d. если ключ уже есть в дереве, дерево не перестраивается	a				
Что будет результатом применения алгоритма RLE - "Групповое кодирование" к сжатию текста ААААААААААААААААААААБСАВСАСССССС?	a. A(19)BCA(2)C(6) b. 19A-2BCA6C c. A19ABCABC6C d. 19A-6BCABCABC	d				
Какое из условий проверяется при определении сбалансированности красно-черного дерева?	a. Любой путь от корня дерева к листу содержит одно и то же число черных узлов b. Высота левого и правого поддерева равны c. Любой путь от корня дерева к листу содержит одно и то же число красных узлов d. Количество красных или чёрных узлов и левом и правом поддеревьях равны	a				
Какой зависимостью описывается функция вычислительной сложности алгоритма турнирной сортировки в наихудшем случае:	a. $f(n) = O(n \log(n))$ b. $f(n) = O(n)$ c. $f(n) = O(n^2)$ d. $f(n) = O(\log(n))$	a	approved by tony & oksana			
Какой зависимостью описывается функция вычислительной сложности алгоритма линейного (последовательного) поиска в худшем случае?	a. $f(n) = O(\log(n))$ b. $f(n) = O(n^2)$ c. $f(n) = O(n)$ d. $f(n) = O(n \log(n))$	c				
Дан рекурсивный алгоритм: <pre>int F(int n) { if(n>2) return F(n-1)+F(n-2)+F(n-3) else return n; }</pre> Найдите значение F(5)	-	11				
Имеется бинарное дерево поиска, содержащее целые числа от 1 до 7. Каким будет результат вывода при обратном обходе дерева (postorder):	a. 4,1,2,3,4,5,6 b. 1,2,3,4,5,6,7 c. 7,6,5,4,3,2,1 d. 4,1,5,2,6,3,7	c	90%			

ВОПРОС:	ВАРИАНТЫ ОТВЕТОВ	ОТВЕТ	КОММЕНТАРИЙ			
Какой зависимостью описывается функция вычислительной сложности алгоритма поиска в глубину на списке смежных вершин, если V - количество вершин в графе, а E - количество ребер:	a. $f(n) = O(V^2)$ b. $f(n) = O(\log(E))$ c. $f(n) = O(V)$ d. $f(n) = O(E)$	c				
Для структуры данных дек справедливо:	a. Вставка нового элемента возможно в начало или в конец списка b. Удаление элемента возможно из произвольной позиции c. Линейный список с произвольным количеством вершин d. Вставка нового элемента возможна только в начало списка	a				
Как называют множество элементов данных и внутренних связей между ними:	a. Структура данных b. Класс c. Алгоритм d. Программа	a				
Укажите асимптотическую сложность операции удаления элемента, на который имеется указатель (например, ptr), из однонаправленного списка размером n:	a. $O(\log n)$ b. $O(n^2)$ c. $O(1)$ d. $O(n)$		Тоже тяжелый вопрос, мы не знаем что это такое, так что тыкайте на рандомикс			
Нелинейными называются структуры данных, в которых:	a. связи между элементами не зависят от упорядоченности значений элементов b. связи между элементами не зависят от выполнения какого-либо условия c. связи между элементами не зависят от линейной упорядоченности элементов d. связи между элементами зависят от выполнения определенного условия	d				
Какой зависимостью описывается функция вычислительной сложности алгоритма поиска хэшированием в лучшем случае?	a. $f(n)=O(n^2)$ b. $f(n)=O(1)$ c. $f(n)=O(\log(n))$ d. $f(n)=O(n \log(n))$	b				
Структура хранения, в которой элементы данных фактически размещаются в последовательных ячейках памяти строго один за другим:	a. векторная b. списочная c. иерархическая d. сетевая	b				
Время выполнения рекурсивной функции f(n) в наихудшем случае определяется рекуррентным соотношением $f(n) = \Theta(1) \text{ при } n=1$ $T(n) = \begin{cases} 2T(n/2) + \Theta(n) & \text{при } n > 1 \end{cases}$	a. $O(\log 2n)$ b. $O(n)$ c. $O(2n)$ d. $O(n \log 2n)$	d	by tony			
ТО ЧТО ЯВЛЯЕТСЯ РЕШЕНИЕ ЭТОГО СООТНОШЕНИЯ:						
В каком типе деревьев лист не должен содержать значение?	a. бинарное дерево поиска b. В-дерево c. красно-черное дерево d. АВЛ-дерево	c				
Какой зависимостью описывается функция вычислительной сложности алгоритма интерполяционного поиска в среднем случае?	a. $f(n)=O(\log(n))$ b. $f(n)=O(\log(\log(n)))$ c. $f(n)=O(n)$ d. $f(n)=O(n \log(n))$	b	by tony			
Что помещается в стек рекурсивных вызовов при вызове рекурсивных функций?	a. Значения всех переменных функции при текущем вызове до возврата в точку вызова b. Адрес точки возврата, значения всех переменных функции текущего вызова c. Имена локальных переменных и параметров функции d. Описание всех переменных функции при текущем вызове	b	by tony			

ВОПРОС:	ВАРИАНТЫ ОТВЕТОВ	ОТВЕТ	КОММЕНТАРИЙ		
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 выведены при вызове F(2):		79			
Какая структура данных используется для хранения элементов данных в хэш-таблице, при реализации разрешения коллизий по методу цепочек (цепного хэширования)?	a. Двоичное дерево b. Однонаправленный список c. Очередь d. Дек	b			
Какое из определений соответствует структуре данных граф?	a. Иерархическая, динамическая структура данных b. Линейная, динамическая структура данных c. Нелинейная, многосвязанная динамическая структура данных d. Нелинейная, статическая, упорядоченная структура данных	c			
Какое количество узлов в полном бинарном дереве высотой 4?	a. 48 b. 64 c. 32 d. 31	d			
Какой зависимостью описывается функция вычислительной сложности алгоритма сортировки методом простого выбора в наилучшем случае:	a. $f(n)=O(n)$ b. $f(n)=O(n \log(n))$ c. $f(n)=O(n^2)$ d. $f(n)=O(\log(n))$	c	90% + oksana & tony approved		
Выберите все характеристики хэш-функции, используемой в алгоритме поиска Рабина-Карпа:	a. кольцевой ("скользящий", "летящий") b. полиномиальный хэш c. отсутствие коллизий d. модульная арифметика	a b d	множественный выбор		
Определите теоретическую вычислительную сложность (функцию роста времени) алгоритма: <pre>for(int i = 0; i < n;i++) for(int j = 0; j < n; j++) cout << "hello";</pre>	a. константная b. логарифмическая c. линейная d. квадратичная	d			
При каком значении баланс-фактора b (т.е разницы высот правого и левого поддеревьев) необходимо выполнить перестройку АВЛ-дерева?	a. $ b > 2$ b. $ b = 0$ c. $ b = 1$ d. $ b = 2$	d			
Чему равна сумма степеней вершин графа?	a. Удвоенному весу ребер b. Числу его ребер c. Сумме веса ребер d. Удвоенному числу его ребер	d			
Какая проблема может возникнуть в хеш-таблице с открытым адресом, после удаления ключа первичного кластера ключей, хешированных с одним индексом?	a. Невозможность найти ключ в первичном кластере b. Неоднозначность результата поиска ключа c. Ключи поменяют индексы, с которыми были ххшированы при вставке в таблицу d. Невозможно вставить ключ в первичный кластер	a			
Сколько указателей используется в структуре узла стека? (Введите только число):	-	скорее всего 1	Ручной ввод		
Какой зависимостью описывается функция вычислительной сложности алгоритма слиянием (Mergesort) в среднем и наихудшем случаях:	a. $f(n)= O(n\log(n))$ b. $f(n)= O(n^2)$ c. $f(n)= O(n)$ d. $f(n)=O(\log(n))$	a			
Сложность рекурсивного алгоритма вычисления N-ого числа Фибоначчи можно снизить с $O(2^n)$ до $O(n)$, т.е оптимизировать процесс вычисления, если сохранить значения решения подзадач. Какой метод построения алгоритма был использован при решении указанной задачи	a. Разделяй и властвуй b. "Жадный" алгоритм c. Метод рекурсии d. Динамическое программирование	d	90%		

ВОПРОС:	ВАРИАНТЫ ОТВЕТОВ	ОТВЕТ	КОММЕНТАРИЙ		
При построении хеш-таблиц возможно появление коллизий. Коллизия это такая ситуация, когда:	МНЕ ЛЕНЬ ПИСАТЬ ВСЕ ВАРИАНТЫ ОВТЕТА, ТУТ ТОЧНО ОТВЕТ: для разных ключей хэш-функция может принимать одно и тоже значения $h(k_i) = h(k_j)$	для разных ключей хэш-функция может принимать одно и тоже значения $h(k_i) = h(k_j)$			
Какой зависимостью описывается функция вычислительной сложности алгоритма поиска в тексте по образцу Бойера-Мура в лучшем случае:	a. $f(n, m) = O(\log(n/m))$ b. $f(n, m) = O(\log(n+m))$ c. $f(n, m) = O(\log(n^*m))$ d. $f(n) = O(n)$	a	90% by tony + oksana		
Иерархическая структура данных -	a. это множество элементов (узлов) связи между которыми определяются на основе линейной зависимости b. это множество элементов (узлов), связи между которыми определяются на основе уровней c. это множество элементов (узлов), в которой реализуются связи между элементами одного уровня d. это множество элементов (узлов), связи между которыми определяются на основе зависимости многие-ко-многим	b	Скорее всего b, ваш верный слуга Павел		
Свойство однородной структуры данных	a. элементы могут быть разного типа b. все элементы одного типа c. все элементы разного типа	b	by tony		
Укажите вид каждого графа представленного матрицами смежности		1 - Смешанный 2 - Ориентированый 3 - Неориентированный	by tony		
Когда алгоритм считается корректным:	a. Когда для каждого вывода результатом его работы является корректный ввод b. Когда для каждого входа результатом его работы является корректный выход c. Когда для каждого вывода результатом его работы является корректный выход d. Когда он может выдать ответ, отличный от ожидаемого	b			
Алгоритм двойного RL поворота	a. правый поворот вокруг правого сына узла с фактором балансировки +1, затем левый вокруг корневого узла b. правый поворот вокруг узла, затем левый вокруг узла с фактором балансировки -1 c. правый поворот вокруг правого сына узла с фактором балансировки -1, затем левый вокруг корневого узла d. правый поворот вокруг левого сына узла с фактором балансировки -1, затем левый вокруг корневого узла	d			
Для оценки порядка роста функции, описывающих вычислительную сложность алгоритмов, используется асимптотические обозначения (символики) или нотации. Что обозначает запись $f(n)=\Omega(g(n))$	<input type="radio"/> множество всех функций, порядок роста которых при достаточно больших n не меньше (больше или равен) некоторой константы c , умноженной на значение функции $g(n)$ <input checked="" type="radio"/> множество всех функций, порядок роста которых ограничен сверху и снизу функцией $g(n)$ с точностью до постоянных множителей $c1$ и $c2$ соответственно <input type="radio"/> множество всех функций, порядок роста которых не ограничен сверху и снизу функцией $g(n)$ с точностью до постоянных множителей $c1$ и $c2$ соответственно <input type="radio"/> множество всех функций, порядок роста которых при достаточно больших n не превышает (меньше или равен) некоторую константу c , умноженную на значение функции $g(n)$	знали бы вы как мне лень эту хуйню писать, вы бы расплакались			
Дисциплина обсуживания (порядок выполнения операций) стека:	a. LIFO b. OFIL c. FIFO d. LOFI	a			
В основе алгоритма Бойера-Мура используется	a. Таблица префиксов b. Хэш-таблица c. Таблица смешений ("стоп-символов") d. Условия Айверсона	c			
Какой код будет получен из входной строки символов ABCCCDDDD при использовании алгоритма Хаффмана:	a. 00 01 10 10 11 11 11 11 b. 110 111 10 10 0 0 0 0 0 c. 1 2 3 3 4 4 4 4 4 d. A1 B1 C3 D5	b			
Укажите асимптотическую сложность операции удаления элемента в позиции i из массива размером n :	a. $O(1)$ b. $O(n^2)$ c. $O(\log n)$ d. $O(n)$	d			

ВОПРОС:	ВАРИАНТЫ ОТВЕТОВ	ОТВЕТ	КОММЕНТАРИЙ			
Какой бы вид графа вы выбрали при реализации задачи поиска маршрута с минимальным временем поездки в московском метро	a. Ориентированный не взвешенный граф b. Неориентированный взвешенный граф c. Ориентированный ациклический граф d. Ориентированный взвешенный граф	b				
Какое минимальное количество операторов потребуется выполнить алгоритму при исключении (не удалении) узла, на который указывает указатель q, из линейного двусвязанного списка	a. 3 b. 2 c. 4 d. 1	b				
В узлах бинарного дерева поиска разместили список значений. Какой алгоритм обхода бинарного дерева поиска позволяет вывести эту последовательность в упорядоченном по возрастанию порядке	a. Прямой b. Последовательный c. Обратный d. Симметричный	d				
Бинарное дерево поиска построено из ключей, который поступали в следующей последовательности: 25,15,10,12,7,44,50,42,30,44,11,17. Было выполнено удаление узла с ключем 15. После удаления были выведеныключи дерева обходом в ширину: 25, 12, 44, 10, 17, 42, 55, 7, 11. Какой из следующих алгоритмов поиска замещающего узла был применен?	a. Поиск в правом поддереве удаляемого узла самого правого узла b. Поиск в правом поддереве удаляемого узла самого левого узла c. Поиск в левом поддереве удаляемого узла самого левого узла d. Поиск в левом поддереве удаляемого узла самого правого узла	b				
Иерархическая структура данных (дерево)	a. это статическая нелинейная структура b. это динамическая нелинейная структура c. это скалярная нелинейная структура d. это динамическая линейная структура	b				
Чем может быть вызвано переполнение стека при выполнении рекурсивной функции с небольшой глубиной рекурсии:	a. Неправильно сформулированным условием завершения рекурсии b. Неправильным выполнением вызова функции c. Неправильным оформлением боком входа в рекурсию d. Неправильным оформлением выводом результата	a				
Какой из приведенных кодов является префиксным (коды отдельных кодовых слов перечислены через запятую)?	a. 1, 12, 31 b. 1, 10, 100, 1000 c. 0, 10, 11, 100, 110 d. 0, 100, 101, 110, 111	d				
Имеется бинарное дерево поиска, содержащее целые числа. Просмотр дерева даёт следующий результат: 2, 4, 6, 8, 10, 12, 14. Какой способ обхода дерева при этом используется:	a. Обратный b. Прямой c. Симметричный d. В ширину	c				
Высота (глубина) бинарного дерева поиска из N узлов в наилучшем случае определяется по формуле:	a. $h = \log_2(N+1)$ b. $h = \log_2(N+1)-1$ c. $h = \log_2 N$ d. $h = N$	b				
Линейными называются структуры данных, в которых:	a. связи между элементами не зависят от линейно упорядоченности элементов b. связи между элементами зависят от упорядоченности значений элементов c. связи между элементами не зависят от выполнения какого-либо условия d. связи между элементами зависят от выполнения какого-либо условия	c				
Имеется идеально сбалансированное двоичное дерево, содержащее 31 узел. Какова высота этого дерева?	a. 5 уровней b. 4 уровня c. 6 уровней d. 7 уровней	b				
Какой алгоритм из перечисленных не основан на жадном подходе?	a. Алгоритм построение минимального остовного дерева Крускала b. Алгоритм нахождения кратчайшего пути Беллмана-Форда c. Алгоритм кодирования Хаффмана d. Алгоритм нахождения кратчайшего пути Дейкстры	b		https://webdevblog.ru/zhadnye-algoritmy-chast-1-algoritm-dejkstry/		
Величина сдвига при поиске образца в тексте определяется наибольшим значением из трёх, вычисленных, соответственно, по трём правилам (эвристикам) в алгоритме:	a. Бойера-Мура с турбосдвигом b. Кнута-Морриса-Прата c. Бойера-Мура d. Бойера-Мура-Хорспула	c				

Тема 1. Введение в дисциплину. Основные понятия

Что такое алгоритм:

- 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

Как называется последовательность вычислительных шагов, преобразующих входные величины в выходные:

- Структуры
- Алгоритмы *
- Данные
- Информация

Что определяет структура данных:

- Множество данных
- Множество отношений между данными
- Множество данных и отношений между ними *
- Множество данных и множество операций над ними

Какие отношения между элементами поддерживает структура данных массив:

- Простые
- Сложные
- Линейные *
- Иерархические

Какие существуют уровни уровня представления данных (согласно American National Standards Institute - ANSI):

- Внешний, внутренний
- Абстрактный, конкретный
- Пользовательский, концептуальный, физический *
- Программный, аппаратный

Когда алгоритм считается корректным:

- 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 *

Какое требование предъявляется к спецификации алгоритма:

- 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

Как представляются данные в памяти компьютера:

- В виде последовательности инструкций
- В виде последовательности битов *
- В виде последовательности операций
- В виде последовательности операндов

Что называют структурой данных:

- 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

Как называют множество элементов данных и внутренних связей между ними?

- Система "клиент-сервер"
- Алгоритм
- Спецификация алгоритма
- Структура данных *

Что такое модель?

- 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

Тема 2. Анализ вычислительной сложности алгоритмов

Показатели эффективности алгоритмов (или программ) включают:

- Количество выполняемых операций и объем памяти, требуемой для выполнения алгоритма *
- Размер (количество операторов) программы
- Корректность и достоверность программы
- Уровень отлаженности программ

В чем заключается анализ алгоритма:

- Предсказать форматы структур данных и внешние связи между ними
- Предсказать требуемые для его выполнения ресурсы *
- Предсказать форматы структур и внутренние связи между ними
- Обеспечить безопасность данных

Что измеряется для оценки вычислительной сложности алгоритма:

- Зависимость выполняемых операторов от размера обрабатываемых данных
- Зависимость объема требуемой памяти от размера обрабатываемых данных
- Зависимость количества выполняемых основных операций от размера обрабатываемых данных *
- Зависимость количества шагов цикла от размера обрабатываемых данных

Когда один алгоритм считается эффективнее другого:

- 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
- Если время его работы в наихудшем случае имеет более низкий порядок роста *

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:

- Размер массива или файла*
- Количество функций в программе
- Размер занимаемой памяти при сортировке или поиске
- Количество основных операций

Какие основные операции следует учитывать при оценке временной (вычислительной) сложности алгоритмов поиска:

- Операции сравнения и перемещения
- Операции сравнения *
- Операции сравнения и сложения
- Операции сложения и перемещения

Какие основные операции следует учитывать при оценке временной (вычислительной) сложности алгоритмов сортировки:

- Операции сравнения
- Операции сравнения и перемещения *
- Операции сравнения и сложения
- Операции сложения и перемещения

Какие существуют случаи при выполнении алгоритмов:

- Общий, частный и оптимальный
- Эффективный и неэффективный
- Наилучший, средний и наихудший *
- Средний и наихудший

Для оценки порядка роста функций, описывающих вычислительную сложность алгоритмов, используются асимптотические обозначения (символики) или нотации. Что обозначает запись $f(n)=O(g(n))$:

- 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 при достаточно больших n не меньше (больше или равен) некоторой константы c , умноженной 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
- 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

Для оценки порядка роста функций, описывающих вычислительную сложность алгоритмов, используются асимптотические обозначения (символики) или нотации. Что обозначает запись $f(n)=\Omega(g(n))$:

- 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
- 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

Для оценки порядка роста функций, описывающих вычислительную сложность алгоритмов, используются асимптотические обозначения (символики) или нотации. Что обозначает запись $f(n)=\Theta(g(n))$:

- 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
- 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 при достаточно больших n не меньше (больше или равен) некоторой константы c , умноженной 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

Укажите правильную аналогию между асимптотическим сравнением двух функций f и g для обозначения $f(n)=O(g(n))$ и сравнением двух действительных чисел a и b :

- $a \leq b$ *
- $a \geq b$
- $a = b$
- $a < b$

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 :

- $a \leq b$
- $a \geq b$ *
- $a = b$
- $a < b$

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 :

- $a \leq b$
- $a \geq b$
- $a = b$ *
- $a < b$

Тема 3. Рекурсивные алгоритмы

Как называется алгоритм, который напрямую или через другие вспомогательные алгоритмы вызывает сам себя:

- Линейный
- Разветвляющийся
- Циклический
- Рекурсивный *

Какая функция реализует линейную рекурсию:

- Содержит два последовательных вызова самой себя
- Содержит только один вызов самой себя *
- Содержит два не последовательных вызова самой себя
- Содержит множественный параллельный вызов самой себя

Что определяет понятие «глубина рекурсии» для рекурсивной функции:

- Количество вызовов внешних функций
- Количество операторов вызова функции в самой функции
- Количество операторов в рекурсивной функции
- Наибольшее одновременное количество рекурсивных обращений функции *

Какие из приведённых строк кода

```
int fib(int n)
{
    if (n < 2)
        return 1;
    else {
    }
}
```

необходимо записать после `else` в рекурсивной функции вычисления n -ного числа Фибоначчи:

- $\text{fib}(n - 1) + \text{fib}(n - 2)$
- $\text{return fib}((n - 1) + (n - 2))$
- $\text{return fib}(n - 1) + \text{fib}(n - 2)$ *
- $\text{return } (n - 1) + (n - 2)$

Как определить, что функция реализует каскадную (множественную) рекурсию:

- Функция вызывает несколько внешних функций по любой из всех возможных ветвей алгоритма

- Функция вызывает последовательно несколько внешних функций
- Функция вызывает параллельно несколько внешних функций
- Вызов функции по любой из всех возможных ветвей алгоритма встречается более одного раза *

Что помещается в стек рекурсивных вызовов при вызове рекурсивной функции:

- Значения всех переменных функции при текущем вызове
- Описание всех переменных функции при текущем вызове
- Адрес точки возврата, значения всех переменных функции текущего вызова *
- Имена локальных переменных и параметров функции

Какова глубина рекурсии следующей функции

```
int fib(int n)
{
    if (n < 2) { return 1; }
    else { return fib(n - 2) + fib(n - 1); }
}
```

при вызове $fib(5)$:

- 3
- 4
- 5
- 7 *

Чем может быть вызвано переполнение стека при выполнении рекурсивной функции с небольшой глубиной рекурсии:

- Выполнением вызова функции
- Неправильно сформулированным условием завершения рекурсии *
- Неправильно оформленным блоком входа в рекурсию
- Неправильно оформленным выводом результата

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 результат не будет получен
- Когда достигается простейшее возможное решение *
- Когда возникнет переполнение стека
- Когда завершиться ввод исходных данных

Если время выполнения рекурсивной функции $f(n)$ в наихудшем случае определяется рекуррентным соотношением

$$\Theta(1) \text{ при } n=1$$

$T(n)=\{$

$$2T(n/2)+\Theta(n) \text{ при } n>1$$

то что является решение этого соотношения:

- $O(n)$
- $O(\log_2 n)$
- $O(2^n)$
- $O(n \log_2 n)$ *

Тема 4. Простые алгоритмы сортировки

Что такое инвариант цикла:

- Логическое выражение, истинное после первого прохода тела цикла и в конце выполнения цикла, зависящее от переменных, изменяющихся в теле цикла
- Логическое выражение, истинное после каждого прохода тела 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

Для чего нужен инвариант цикла:

- Для обеспечения правильного жизненного цикла
- Для повышения эффективности алгоритма
- Для проверки корректности алгоритма *
- Для одновременной работы с несколькими массивами

Если на вход подается последовательность (31,41,59,26,41,58), то какой должен быть вывод алгоритма сортировки:

- (31,26,41,41,58,59)
- (41,31,41,26,58,59)
- (58,31,41,41,26,59)
- (26,31,41,41,58,59) *

Как называется алгоритм, который выполнит сортировку исходного массива (3,1,5,2,4) после следующей последовательностью проходов

(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)

- Простого обмена *
- Простого выбора
- Простой вставки
- Шелла

Как называется алгоритм, который выполнит сортировку исходного массива (3,1,5,2,4) после следующей последовательностью проходов

(3,1,4,2,5), (3,1,2,4,5), (2,1,3,4,5), (1,2,3,4,5)

- Простого обмена
- Простого выбора *
- Простой вставки
- Шелла

Как называется алгоритм, который выполнит сортировку исходного массива (3,1,5,2,4) после следующей последовательностью проходов

(1,3,5,2,4), (1,3,5,2,4), (1,2,3,5,4), (1,2,3,4,5)

- Простого обмена
- Простого выбора
- Простой вставки *
- Шелла

Что проверяет условие Айверсона в алгоритме сортировки методом простого обмена:

- 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

Какой зависимостью описывается функция вычислительной сложности алгоритма сортировки методом простой вставки в наилучшем случае:

- $f(n)=O(\log(n))$
- $f(n)=O(n^2)$
- $f(n)=O(n) *$
- $f(n)=O(n \log(n))$

Какой зависимостью описывается функция вычислительной сложности алгоритма сортировки методом простого обмена (с условием Айверсона) в наилучшем случае:

- $f(n)=O(\log(n))$
- $f(n)=O(n^2)$
- $f(n)=O(n) *$
- $f(n)=O(n \log(n))$

Какой зависимостью описывается функция вычислительной сложности алгоритма сортировки методом простого выбора в наилучшем случае:

- $f(n)=O(\log(n))$
- $f(n)=O(n^2) *$
- $f(n)=O(n)$
- $f(n)=O(n \log(n))$

Какой зависимостью описывается функция вычислительной сложности алгоритма сортировки подсчетом (Counting sort) в среднем и наилучшем случаях:

- $f(n)=O(n)$
- $f(n)=O(n^2)$
- $f(n)=O(n+k) *$
- $f(n)=O(\log(n))$

Тема 5. Нетривиальные алгоритмы сортировки

Какой зависимостью описывается функция вычислительной сложности алгоритма сортировки шейкерным методом (с условием Айверсона) в наилучшем случае:

- $f(n)=O(\log(n))$
- $f(n)=O(n^2)$
- $f(n)=O(n) *$
- $f(n)=O(n \log(n))$

Какой зависимостью описывается функция вычислительной сложности алгоритма сортировки шейкерным методом (с условием Айверсона) в наихудшем случае:

- $f(n)=O(\log(n))$
- $f(n)=O(n^2) *$
- $f(n)=O(n)$
- $f(n)=O(n \log(n))$

Какой зависимостью описывается функция вычислительной сложности алгоритма быстрой сортировки методом Хоара (quicksort) в среднем и наилучшем случаях:

- $f(n)=O(\log(n))$
- $f(n)=O(n^2)$
- $f(n)=O(n)$
- $f(n)=O(n \log(n)) *$

Какой зависимостью описывается функция вычислительной сложности алгоритма быстрой сортировки методом Шелла в наилучшем случае:

- $f(n)=O(\log(n))$
- $f(n)=O(n^2)$
- $f(n)=O(n) *$
- $f(n)=O(n \log(n))$

Какой зависимостью описывается функция вычислительной сложности алгоритма быстрой сортировки методом Шелла в наихудшем случае:

- $f(n)=O(\log(n))$

- $f(n)=O(n^2)$ *
- $f(n)=O(n)$
- $f(n)=O(n \log(n))$

Суть алгоритма сортировки методом Шелла заключается в:

- 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

Какой зависимостью описывается функция вычислительной сложности алгоритма быстрой сортировки методом Хоара (quicksort) в наихудшем случае:

- $f(n)=O(\log(n))$
- $f(n)=O(n^2)$ *
- $f(n)=O(n)$
- $f(n)=O(n \log(n))$

Какой зависимостью описывается функция вычислительной сложности алгоритма пирамидальной сортировки (Heapsort) в среднем и наихудшем случаях:

- $f(n)=\Theta(n \log(n))$ *
- $f(n)=\Theta(\log(n))$
- $f(n)=\Theta(n^2)$
- $f(n)=\Theta(n)$

Какой зависимостью описывается функция вычислительной сложности алгоритма турнирной сортировки в среднем и наихудшем случаях:

- $f(n)=\Theta(\log(n))$
- $f(n)=\Theta(n \log(n))$ *
- $f(n)=\Theta(n^2)$
- $f(n)=\Theta(n)$

Алгоритм сортировки подсчетом (counting sort) имеет следующие характеристики эффективности:

- 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 - $f(k)=\Theta(k)$ *

Какой зависимостью описывается функция вычислительной сложности алгоритма сортировки слиянием (Mergesort) в среднем и наихудшем случаях:

- $f(n)=\Theta(n \log(n))$ *
- $f(n)=\Theta(\log(n))$
- $f(n)=\Theta(n^2)$
- $f(n)=\Theta(n)$

Тема 6. Алгоритмы поиска в массивах

Какой зависимостью описывается функция вычислительной сложности алгоритма бинарного (двоичного) поиска в среднем и худшем случаях:

- $f(n)=\Theta(n \log(n))$
- $f(n)=\Theta(\log(n))$ *
- $f(n)=\Theta(n^2)$
- $f(n)=\Theta(n)$

Какой зависимостью описывается функция вычислительной сложности алгоритма интерполяционного поиска в среднем случае:

- $f(n)=\Theta(n \log(n))$
- $f(n)=\Theta(\log(n))$
- $f(n)=\Theta(\log(\log(n))) *$
- $f(n)=\Theta(n)$

Какой зависимостью описывается функция вычислительной сложности алгоритма интерполяционного поиска в худшем случае:

- $f(n)=\Theta(n \log(n))$
- $f(n)=\Theta(\log(n))$
- $f(n)=\Theta(\log(\log(n)))$
- $f(n)=\Theta(n) *$

Идея алгоритма интерполяционного поиска основана на:

- выборе новой области поиска по расстоянию между ключом и текущим значением элемента *
- делении области поиска на две части
- сравнении каждого элемента с искомым
- учёте знака разности между ключом и текущим значением элемента

При построении хеш-таблиц возможно появление коллизий. Коллизия это такая ситуация, когда:

- способ нахождения промежуточных значений определяется величиной по имеющемуся дискретному набору значений
- определение области поиска производится с помощью учета знака разности между ключом и текущим значением элемента
- для разных ключей хеш-функция может принимать разные значения $h(k_i) \neq h(k_j)$
- для разных ключей хеш-функция может принимать одно и тоже значение $h(k_i) = h(k_j) *$

Идея алгоритма бинарного поиска основана на:

- 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

Какой зависимостью описывается функция вычислительной сложности алгоритма поиска по бинарному дереву (binary search tree, BST) в лучшем случае:

- $f(n)=\Theta(n \log(n))$
- $f(n)=\Theta(n^2)$
- $f(n)=\Theta(n)$
- $f(n)=\Theta(\log(n)) *$

Какой зависимостью описывается функция вычислительной сложности алгоритма поиска по бинарному дереву в худшем случае (несбалансированное бинарное дерево):

- $f(n)=\Theta(n \log(n))$
- $f(n)=\Theta(n^2)$
- $f(n)=\Theta(n) *$
- $f(n)=\Theta(\log(n))$

Какой зависимостью описывается функция вычислительной сложности алгоритма поиска хешированием в лучшем случае:

- $f(n)=\Theta(n \log(n))$
- $f(n)=\Theta(\log(n))$
- $f(n)=\Theta(n^2)$
- $f(n)=\Theta(1) *$

Какой зависимостью описывается функция вычислительной сложности алгоритма поиска хешированием в худшем случае:

- $f(n)=\Theta(\log(n))$
- $f(n)=\Theta(n)$ *
- $f(n)=\Theta(n^2)$
- $f(n)=\Theta(1)$

Какой зависимостью описывается функция вычислительной сложности алгоритма линейного (последовательного) поиска в худшем случае:

- $f(n)=\Theta(\log(n))$
- $f(n)=\Theta(n \log(n))$
- $f(n)=\Theta(n)$ *
- $f(n)=\Theta(n^2)$

Тема 7. Алгоритмы поиска по образцу

Какой зависимостью описывается функция вычислительной сложности алгоритма прямого поиска в тексте по образцу в худшем случае:

- $f(n)=\Theta(n)$
- $f(n)=\Theta(n^2)$
- $f(n, m)=\Theta(\log(n+m))$
- $f(n, m)=\Theta(\log(n*m))$ *

Какой зависимостью описывается функция вычислительной сложности алгоритма поиска в тексте по образцу методом Кнута-Морриса-Пратта в лучшем случае:

- $f(n)=\Theta(n)$
- $f(n)=\Theta(n^2)$
- $f(n, m)=\Theta(\log(n+m))$ *
- $f(n, m)=\Theta(\log(n*m))$

В каких случаях алгоритм Кнута-Морриса-Пратта дает подлинный выигрыш по сравнению с другими алгоритмами поиска в тексте по образцу:

- 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

В основе алгоритма Кнута-Морриса-Пратта используется:

- Условие Айверсона
- Хэш-таблица
- Таблица смещений («стоп-символов»)
- Таблица префиксов *

В основе алгоритма Бойера-Мура используется:

- Условие Айверсона
- Хэш-таблица
- Таблица смещений («стоп-символов») *
- Таблица префиксов

Когда достигается максимальная эффективность алгоритма Бойера-Мура:

- Если образец длинный, а мощность алфавита достаточно велика *
- Если образец короткий, а мощность алфавита достаточно велика
- Если образец длинный, а мощность алфавита достаточно низка

- Если образец короткий, а мощность алфавита достаточно низка

Отличительная особенность алгоритма Бойера-Мура:

- Сравнение символов производится начиная с конца текста
- Производится посимвольное сравнение образца с текстом
- Сравнение символов производится начиная с конца образца *
- После каждого неудачного сравнения производится сдвиг образца вправо на одну позицию

Отличительная особенность алгоритма Бойера-Мура:

- После каждого неудачного сравнения производится сдвиг образца вправо в соответствии с таблицей префиксов
- После каждого неудачного сравнения производится сдвиг образца вправо на количество позиций, равное количеству предшествующих удачных сравнений
- После каждого неудачного сравнения производится сдвиг образца вправо на одну позицию
- После каждого неудачного сравнения производится сдвиг образца вправо в соответствии с таблицей смещений («стоп-символов») *

В основе алгоритма Рабина-Карпа используется:

- Условие Айверсона
- Хэш-таблица *
- Таблица смещений («стоп-символов»)
- Таблица префиксов

Какой зависимостью описывается функция вычислительной сложности алгоритма поиска в тексте по образцу методом Бойера-Мура в лучшем случае:

- $f(n)=\Theta(n)$
- $f(n, m)=\Theta(\log(n/m))$ *
- $f(n, m)=\Theta(\log(n+m))$
- $f(n, m)=\Theta(\log(n*m))$

Тема 8. Линейные списки

Какая операция считается недопустимой для линейного односвязного списка:

- Удаление последнего элемента
- Вставка нового элемента после заданного элемента
- Вставка нового элемента перед заданным элементом *
- Замена содержимого информационной части заданного элемента

Укажите, какими свойствами должна обладать структура данных, 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:

- Сортированная
- Несортированная
- Однородная *
- Неоднородная

Какими свойствами можно характеризовать динамическую структуру данных:

- Должна быть линейной
- Формат элементов может изменяться во время работы программы
- Доступ к элементам произвольный
- Размер может изменяться во время работы программы *

Какие операции необходимо реализовать при разработке программы управления отдельным элементом структуры данных:

- Сортировка по различным полям структуры

- Добавление нового элемента *
- Уничтожение структуры
- Инициализация структуры

Какая структура данных относится к категории линейных списков:

- Бинарное дерево
- Дек *
- Последовательность
- Множество

Укажите свойство, характеризующее структуру данных Стек:

- Линейный список с одной вершиной *
- Линейный список с двумя вершинами
- Доступ возможен к произвольному элементу
- Удаление (выборка) возможно произвольного элемента

Имеется описание структуры узла линейного односвязного списка

struct Tnode{ Tdata data; XXXX next;};

Какое определение должно быть на месте XXXX:

- Tnode
- *Tnode *
- void
- int

Укажите свойство, характеризующее структуру данных Очередь:

- Вставка нового элемента возможна в начало и в конец списка
- Удаление элемента возможно из начала и из конца списка
- Доступ возможен к произвольному элементу
- Удаление элемента возможно с одной из сторон списка *

Укажите свойства, характеризующие структуру данных Дек:

- 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

Какой линейный список подойдет лучше для реализации структуры данных, в которой требуется часто перемещаться как слева направо, так и справа налево:

- Линейный односвязный список
- Очередь
- Стек
- Дек *

Есть ли ошибка в коде функции *f*, которая осуществляет получение ссылки на последний узел линейного односвязного списка со структурой узла

struct Tnode{ Tdata data; Tnode next;};*

*Tnode *f(Tnode *L)*

{

```
    Tnode *q=L;
    while(q){ q=q->next; }
    return q;
```

}

- Ошибки нет
- Ошибка в записи оператора *Tnode *q=L* - должно быть *Tnode q=L*
- Ошибка в условии *while (q)* - должно быть *while (q->next) **

- Ошибка в записи оператора $q=q->next$

Имеется указатель q на узел в середине линейного одностороннего списка со структурой узла `struct Tnode{ Tdata data; Tnode* next;};` Требуется вставить новый узел, ссылку на который хранит указатель qq (узел содержит данные), в позицию, в которой находится узел q . Какую последовательность операторов необходимо выполнить, чтобы корректно выполнилась операция вставки:

- $(*qq)=(*q);$
- $qq->next=q->next; q->next=qq; swap(qq->data, q->data); *$
- $q->next=qq; qq->next=q; swap(qq->data, q->data);$
- $qq=q;$

Имеется линейный односторонний список из $n (n>1)$ узлов. Структура узла списка `struct Tnode{ Tdata data; Tnode* next;};` L - указатель на его вершину. Укажите группу операторов, которые обеспечат корректную вставку в вершину списка L нового узла, адрес которого хранит указатель qq :

- $qq->next=L; L=qq; *$
- $L=qq; qq->next=L;$
- $qq=L;$
- $qq->next=L; L=qq->next;$

Имеется линейный односторонний список из $n (n>2)$ узлов. Структура узла списка `struct Tnode{ Tdata data; Tnode* next;};` L - указатель на его вершину. Укажите группу операторов, которые обеспечат корректное удаление двух узлов из вершины списка L :

- $q1=L; L=L->next->next; q2=L; delete q1; delete q2;$
- $q1=L; q2=L->next; L=L->next->next; delete q1; delete q2; *$
- $q1=L; q2=L; L=q2->next; delete q1; delete q2;$
- $q1=L; delete q1; q2=q1; L=q2->next; delete q2;$

Имеется указатель q на узел в середине линейного одностороннего списка со структурой узла `struct Tnode{ Tdata data; Tnode* next;};` Требуется удалить узел, ссылку на который хранит указатель q . Какую последовательность операторов необходимо выполнить, чтобы корректно выполнилась операция удаления узла:

- $(*q)=*(q->next); delete (q->next);$
- $delete (q->next); (*q)=*(q->next);$
- $q1=q->next; *q=*q1; delete q1; *$
- $(*(q->next))=*(q); delete (q->next);$

Какой вы выберите способ реализации линейного списка, для задачи, в которой требуется обеспечить наискорейшее выполнение операций вставки и удаления над любым элементом:

- Массив
- Линейный односторонний список *
- Стек
- Очередь

Какой из видов линейных списков лучше использовать при реализации задачи по проверке баланса круглых скобок (соответствие открывающей и закрывающей скобок: $((())$) – баланс):

- Линейный односвязный список
- Линейный двусвязный список
- Очередь
- Стек *

Как лучше реализовать линейный связный список, для задачи, в которой часто требуется добавлять узел в конец списка:

- Стек
- Очередь *
- Линейный односвязный список

- Дек

Какое минимальное количество операторов потребуется выполнить алгоритму при исключении (не удалении) узла, на который указывает указатель q , из линейного двусвязного списка:

- 1
- 2 *
- 3
- 4

Какую операцию в линейном двунаправленном списке выполняют следующие операторы $q->prev->next=q->next$; $q->next->prev=q->prev$; ? Где q указатель на отдельный узел списка, расположенный в середине списка:

- 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 *

Какая экономная по памяти структура данных используется для реализации кольцевой (циклической) очереди:

- Линейный двунаправленный список
- Линейный односторонний список
- Бинарное дерево
- Массив *

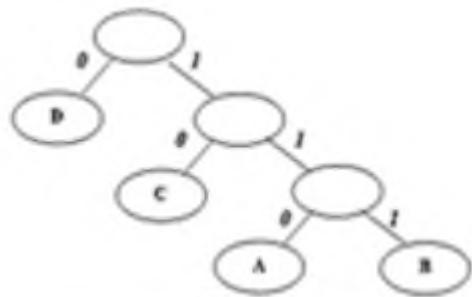
Имеется некоторая структура данных, в которую заносятся упорядоченные по возрастанию символы. Считывание данных из этой структуры даёт результат: F, E, D, C, B, A . Чем является эта структура данных:

- Очередь
- Дерево
- Стек *
- Связный список

Зачем нужен заглавный элемент в двусвязном кольцевом списке

*для связи последнего и первого

Имеется следующее оптимального кодовое дерево (ОКД). Закодировано 111 0 0 10 10 0 110 0 10 0



Ответ: BDDCCDFDCD

Какой код будет получен из входной строки символов 11112234444 при использовании алгоритма RLE
*4 1 2 2 1 3 3 4

Какой алгоритм реализует стратегию «разделяй и властвуй»

*quick sort

Как называется алгоритм для нахождения кратчайшего пути между двумя вершинами графа

*алгоритм Дейкстры

* $O(n^2)$ (не точно)

Количество ребер в основном дереве равно

* $n-1$

Сколько байт бинарного файла занимает число 0 в двоичном формате

*1 байт

Что такое структурированный протокол

*Простейший способ записи кластеров данных в файл (не точно)

Известно, что при построении хеш-таблиц возможно появление коллизий. Коллизия это...

* $h(k_i) = h(k_j)$

Каков главный недостаток хеш-таблиц

*В статическом распределении памяти

Какое из перечисленных АВЛ-деревьев требует перестройки

Какое из перечисленных АВЛ-деревьев требует проведения перестройки:



*1 (не точно)

Для каких узлов необходимо проверить коэффициент балансировки после добавления узла в АВЛ-дерево

*для всех предков нового узла вплоть до корня

Имеется идеально сбалансированное двоичное дерево, содержащее 31 узел. Высота

*5 уровней

Имеется идеально сбалансированное двоичное дерево (не явл. деревом поиска). Обход слева-направа in-order: 2 4 6 8 10 12 14.

*2

Какая операция считается недопустимой для линейного односвязного списка:

*вставка нового элемента перед заданным

В основе алгоритма Кнута-Морриса-Пратта используется:

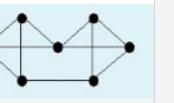
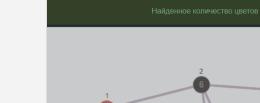
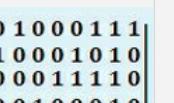
*таблица префиксов

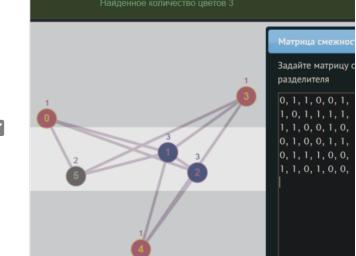
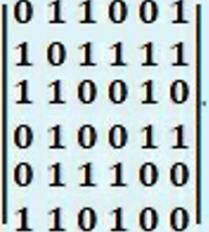
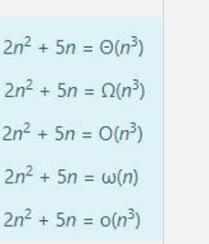
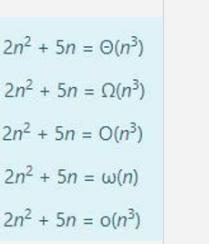
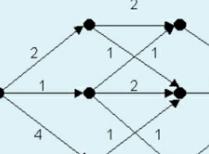
Какав	Вопрос	Варианты ответа (или скрин вариантов)	Ответ	Да?	Комментарий	Скрин задания	Источник ответа
	Возможные вопросы:						
1	Что необходимо делать с большим объемом информации, связанной с каждым ключом?			<input type="checkbox"/>			
2	Что такое бит?			<input type="checkbox"/>	Построить хеш-таблицу на основе целного хеширования или двойного хеширования для быстрого поиска 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).		Какой код будет получен из входной строки символов
3	Почему при записи элемента в бинарное дерево в параметры передается указатель на указатель на узел?			<input type="checkbox"/>	чтобы можно было его изменить		
4	Почему у ддп (двоичное дерево поиска) высокая эффективность поиска и сортировки			<input type="checkbox"/>	Эффективность поиска по дереву напрямую связана с его сбалансированностью, то есть с максимальной разницей между глубиной левого и правого поддерева 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 связному списку.		
5	Почему расстояние от самой короткой ветки до самой длинной в кчд (красно-черное дерево) различается не более чем в 2 раза?			<input type="checkbox"/>	в короткой только чёрные, а в длинное ещё столько же красных		
6	Что хорошо работает на кчд (красно-черное дерево)?			<input type="checkbox"/>	ассоциативный массив		
7	Как может рассматриваться каждый элемент хеш-таблицы в организации ключей претерпевших коллизию с точки зрения производительности?			<input type="checkbox"/>	С точки зрения производительности, каждый элемент хеш-таблицы, претерпевший коллизию, можно рассматривать как элемент одностороннего списка, так как для поиска по ключу нужно будет перебирать следующие записи по порядку до совпадения, начиная с индекса, возвращенного хеш-функцией		
8	Зачем нужен заголовок в двунаправленном циклическом списке?			<input type="checkbox"/>	возможность быстрого доступа к началу и концу списка для вставки нового звена как в начало списка (после заголовка), так и в его конец; для предотвращения "зацикливания" при просмотре списка		
9	Каково наибольшее число ребер у графа (без тругольников) с n вершинами ?			<input type="checkbox"/>	к-во ребер = сумме степеней всех вершин деленое на 2. Что такое степень вершины графа - Википедия . Граф, состоящий из N вершин содержит не более N(N-1)/2 ребер.		
10	Сколько места требуется матрице смежности графа как функции от числа n вершин и числа m ребер?			<input type="checkbox"/>	n^2		
11	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.			<input type="checkbox"/>	$O(n^2+mn)$ или $O(n^2)$ в самом простом случае		
12	В каких случаях в массивах меняется порядок хранения двоичных файлов по сравнению с их хранением в памяти компьютера?			<input type="checkbox"/>			
13	Что такое структурированный протокол?	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		<input type="checkbox"/>	https://ru.wikipedia.org/wiki/SOAP; https://ru.wikipedia.org/wiki/Protocol_Buffers;		
14	Почему не рекомендуется открывать общий доступ к структурированному протоколу?			<input type="checkbox"/>			
15	Для чего созданы файлы объединяющие текстовые(XML) и битовые форматы?			<input type="checkbox"/>	Предположение: чтобы метаданные не были в ещё одном файле		
16	От чего зависит эффективность алгоритма Прима?			<input type="checkbox"/>	Производительность алгоритма Прима зависит от выбранной реализации приоритетной очереди. Асимптотика алгоритма зависит от способа хранения графа и способа хранения вершин, не входящих в дерево.		
17	В чем проигрывает алгоритм Крускала алгоритму Прима?			<input type="checkbox"/>	Алгоритм Прима лучше работает с плотным графом, имеющим гораздо больше ребер, чем вершин.		

Какав	Вопрос	Варианты ответа (или скрин вариантов)	Ответ	Да?	Комментарий	Скрин задания	Источник ответа
18	Как строить минимальное связное дерево алгоритм Прима?			<input type="checkbox"/>	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 соединенные им две вершины образуют дерево. Затем, рассматриваясь ребра графа, один конец которых — уже принадлежащая дереву вершина, а другой — нет; из этих ребер 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 оственное дерево минимальной стоимости.		
19	Как строить минимальное связное дерево алгоритм Крускала?			<input type="checkbox"/>	В начале текущее множество рёбер устанавливается пустым. Затем, пока это возможно, проводится следующая операция: из всех ребёр, добавление которых к уже имеющемуся множеству не вызовет появление 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 оственным деревом минимального веса.		
20	Где работает быстрее алгоритм Крускала в отличии от алгоритма Прима?			<input type="checkbox"/>	https://evileg.com/uk/post/523#header_%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%BB%D1%82%D0%BC_%D0%9A%D1%80%D0%B0%D1%81%D0%BA%D0%B0%D0%BB%D0%BB_0_vs_%D0%9F%D1%80%D0%BB%D0%BC%D0%BD		
Вопросы/ответы из студизбы							
-2	Выберите неверный способ представления графа	a. Матрицей смежности; b. Матрицей инцидентности; c. Списком смежности; d. Матрицей замыкания; e. Массивом дуг;	d	<input checked="" type="checkbox"/>			
-1	Классический алгоритм Дейкстры предназначен для	a. нахождения кратчайшего расстояния между двумя вершинами ациклического графа при отсутствии дуг отрицательного веса b. нахождения кратчайшего расстояния между двумя вершинами в ациклическом взвешенном графе c. нахождения кратчайшего расстояния между двумя вершинами только в циклическом взвешенном графе d. нахождения кратчайшего расстояния между двумя вершинами в произвольном взвешенном графе при отсутствии дуг отрицательного веса e. нахождения расстояния между всеми вершинами в произвольном взвешенном графе	d	<input checked="" type="checkbox"/>			
0	Задача нахождения кратчайшего расстояния между двумя вершинами графа в случае отсутствия дуг отрицательного веса решается	a. используя алгоритм Дейкстры b. используя алгоритм Флойда c. 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)	c	<input type="checkbox"/>			
1	Выберите неверный термин (тот, который не применим к графам)	a. Ориентированный граф; b. Двудольный граф; c. Ациклический граф; d. Все термины верны; e. Циклический граф;	d	<input checked="" type="checkbox"/>	https://neerc.ifmo.ru/wiki/index.php?title=%D0%94%D0%BB%D1%83%D0%BD%D0%BE%D0%BB%D1%8C%D0%BD%D1%8B%D0%BD%D1%84%D1%8B		
Реальные вопросы							
-2	Центром орграфа $G = (V, E)$ называется (множественный выбор)	a. вершина, для которой максимальное расстояние (длина пути) до (от) других вершин минимально b. дуга с максимальным эксцентриситетом. c. вершина с максимальным эксцентриситетом. d. вершина с минимальным эксцентриситетом. e. дуга с минимальным эксцентриситетом. f. вершина, для которой минимальное расстояние (длина пути) до (01) других вершин максимально	a, d	<input checked="" type="checkbox"/>	Центром орграфа G называется вершина с минимальным эксцентриситетом, т.е. это вершина, для которой максимальное расстояние (длина пути) до других вершин минимально. (а, д) https://studopedia.ru/7_114623_nahozhdenie-tsentrata-orientirovannogo-grafra.html		

Какак	Вопрос	Варианты ответа (или скрин вариантов)	Ответ	Да?	Комментарий	Скрин задания	Источник ответа
-1	Укажите свойства свободных деревьев (множественный выбор)	a. При добавлении в свободное дерево нового ребра образуется цикл. b. Свободное дерево представляет собой дерево без корня, ориентированное к листьям c. Каждое свободное дерево является полносвязанным графом d. Каждое свободное дерево с числом вершин n , имеет в точности $n - 1$ ребер. e. Сумма весов ребер свободного дерева наименьшая на множестве ребер графа	a, d	<input type="checkbox"/>			
0	В глубинном оством лесу, построенным при обходе орграфа $G = (V, E)$, различают (выделяют) (множественный выбор)	a. сильносвязанные компоненты глубинного оствомого леса b. обратные дуги c. поперечные дуги d. прямые дуги e. дуги дерева f. паросочетания дуг глубинного оствомого леса	b, c, d, e	<input type="checkbox"/>	(b, c, d, e) В процессе обхода ориентированного графа методом поиска в глубину только определенные дуги ведут к вершинам, которые ранее не посещались. Такие дуги, ведущие к новым вершинам, называются дугами дерева и формируют для данного графа глубинный оствомый лес, построенный методом поиска в глубину. Кроме дуг дерева, существуют еще три типа дуг, определяемых в процессе обхода орграфа методом поиска в глубину. Это обратные, прямые и поперечные дуги: https://viki.dalka.ru/3-47377.html		
1	Укажите неверное высказывание: "Поиск в ширину на взвешенном неориентированном графе предназначен (может быть использован) для	a. определения циклов на графе b. определения кратчайшего расстояния между парой узлов графа c. отыскания какого-либо цикла на заданном графе d. получения глубинного оствомого леса e. систематического обхода всех вершин (узлов) графа	b	<input type="checkbox"/>	https://studizba.com/files/show/doc/203435-1-saod_2_kurs_grafy_2.html		
2	"Максимальное множество вершин графа, в котором существуют пути из любой вершины в любую другую вершину" - это ...	a. сильно связная компонента графа b. минимальное покрытие графа c. редуцированный орграф d. максимальное покрытие графа e. максимальное паросочетание графа	a	<input checked="" type="checkbox"/>	https://www.e-olymp.com/ru/blogs/posts/49		
3	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 оствомое дерево минимальной стоимости. (множественный выбор)	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	a, d	<input type="checkbox"/>			
4	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 b. алгоритм Прима c. метод поиска в ширину d. алгоритм Флойда e. метод поиска в глубину f. алгоритм Крускала g. %-100%алгоритм Дейкстры	a, e	<input type="checkbox"/>			
5	Пусть дан связанный неориентированный взвешенный 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)	a. алгоритм Уоршелла b. метод поиска в глубину c. метод поиска в ширину d. %-100%алгоритм Дейкстры e. алгоритм Прима f. алгоритм Крускала g. алгоритм Флойда	e, f	<input type="checkbox"/>			
6	Пусть дан связанный орграф $G=(V, E)$, причем все дуги имеют неориентальный вес. Отметьте те алгоритмы (полные варианты, возможно с некоторой адаптацией алгоритма), которые можно использовать для решения следующей задачи: Требуется определить кратчайшее расстояние от вершины хдо вершины у. (множественный выбор)	a. метод поиска в ширину b. алгоритм Уоршелла c. алгоритм Прима d. %-50%алгоритм Дейкстры e. метод поиска в глубину f. алгоритм Флойда g. алгоритм Крускала	d, f	<input type="checkbox"/>			
7	Укажите неверное высказывание: "Поиск в ширину на не взвешенном неориентированном графе предназначен (может быть использован) для ..."	a. определения кратчайшего расстояния между парой узлов графа b. определения центра графа c. получения глубинного оствомого леса d. систематического обхода всех вершин (узлов) графа e. отыскания какого-либо цикла на заданном графе	a	<input type="checkbox"/>			
8	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 х и у. (множественный выбор)	a. алгоритм Прима b. алгоритм Флойда c. метод поиска в глубину d. алгоритм Уоршелла e. метод поиска в ширину f. %-33.33%алгоритм Дейкстры g. алгоритм Крускала	b, e, f	<input type="checkbox"/>			

Какав	Вопрос	Варианты ответа (или скрин вариантов)	Ответ	Да?	Комментарий	Скрин задания	Источник ответа
9	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.	a. метод поиска в глубину b. алгоритм Прима c. алгоритм Уоршелла d. метод поиска в ширину e. алгоритм Флойда f. алгоритм Крускала g. %-100%алгоритм Дейкстры	c, e	<input checked="" type="checkbox"/>	https://ru.wikipedia.org/wiki/Алгоритм_Флойда_—_Уоршелла		
10	Связный граф, не имеющий точек сочленения, называется (является) ... (множественный выбор) (укажите 2 правильных ответов)	a. к-связным, причем $k > 2$ b. покрытием графа $G = (V, E)$ c. двусвязным d. паросочетанием графа $G - (V, E)$ e. к 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 вершин, входящих в один путь, не входит ни в какой другой из этих путей. g. свободным деревом	c	<input type="checkbox"/>	Ациклический связный граф называется деревом. Остовное дерево графа - определение 2 - Связный граф, содержащий N вершин и $N-1$ ребер.		Лекции скворцовой
11	Какому алгоритму соответствует следующий фрагмент: "На каждом шаге алгоритма находится ребро наименьшей стоимости (u, v) такое, что u и $v \in U$, 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 ."	a. 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 Дейкстры g. вычислению экцентричеситета графа	b	<input checked="" type="checkbox"/>			
12	Укажите описание алгоритма, которое может являться фрагментом построения остовного дерева минимальной стоимости алгоритмом Прима.	a. "На каждом шаге алгоритма находится ребро наименьшей стоимости (u, v) такое, что u и $v \in V$, 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)" b. "На каждом шаге алгоритма находится ребро наименьшей стоимости (u, v) такое, что $u \in V \setminus U$ и $v \in U$, 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 множество U не станет равным множеству V ." c. "На каждом шаге рассматривается в лексикографическом порядке дуги, исходящие из текущей вершины. Если дуга ведет к еще не посещенному узлу, то оставльные 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 "выталкивается" верхний элемент и алгоритм переходит к рассмотрению такого узла. Данный процесс продолжается до тех пор, пока в стеке содержится хотя бы один элемент. d. "На каждом шаге рассматриваются в лексикографическом порядке дуги, исходящие из текущей вершины. Еще не посещенные узлы, связанные дугой с текущим, помещаются 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. 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 ографа, тогда вспомогательный массив будет содержать длины кратчайших путей от источника к каждой вершине."	d	<input checked="" type="checkbox"/>	На этом свойстве основан алгоритм Прима. В этом алгоритме строится множество вершин U , из которого "вырастает" остовное дерево. Пусть $V=\{1, 2, \dots, n\}$. Сначала $U=\{1\}$. 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 . Процесс продолжается до тех пор, пока множество U не станет равным множеству V .		
13	"Классический" алгоритм Дейкстры предназначен для ...	a. нахождения расстояния между всеми вершинами в произвольном взвешенном графе b. нахождения кратчайшего расстояния между двумя вершинами в произвольном взвешенном графе при отсутствии дуг отрицательного веса c. нахождения кратчайшего расстояния между двумя вершинами ациклического графа при отсутствии дуг отрицательного веса d. нахождения кратчайшего расстояния между двумя вершинами только в циклическом взвешенном графе e. нахождения кратчайшего расстояния между двумя вершинами в ациклическом взвешенном графе	b	<input checked="" type="checkbox"/>	https://studizba.com/files/show/doc/203435-1-saod_2_kurs_grafy_2.html		

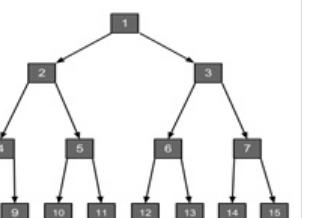
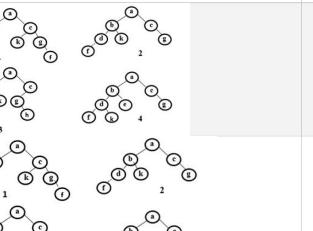
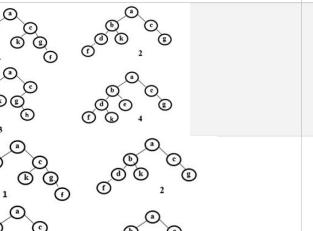
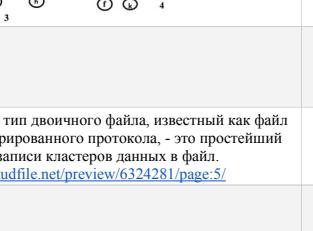
Какак	Вопрос	Варианты ответа (или скрин вариантов)	Ответ	Да?	Комментарий	Скрин задания	Источник ответа
14	Определите хроматическое число графа (смотреть фото в коментах)	a. 7 b. 4 c. 6 d. 5 e. 3 f. 2	4	<input checked="" type="checkbox"/>			
15	Определите задачу, решение которой предполагает использование алгоритма Крускала	a. Проектирование такой топологии связей между компьютерами в компьютерной сети, при которой обеспечивалась минимальный расход сетевого кабеля без учета топологических особенностей местности b. Проектирование такой топологии связей между компьютерами в компьютерной сети, при которой обеспечивалась повышенная надежность соединений в следствие наличия избыточных связей c. Проектирование топологии типа "кольцо", при которой обеспечивалась минимальный расход сетевого кабеля d. Проектирование такой топологии сети типа "звезда", при которой обеспечивалась минимальный расход сетевого кабеля e. Оптимизация трафика в компьютерной сети путем нахождения кратчайшего расстояния между двумя компьютерами в компьютерной сети	a	<input checked="" type="checkbox"/>	<p>Определите задачу, решение которой предполагает использование алгоритма Крускала</p> <p>Ответы:</p> <ul style="list-style-type: none">- Оптимизация трафика в компьютерной сети путем нахождения кратчайшего расстояния между двумя компьютерами в компьютерной сети- Проектирование топологии сети типа «звезда», при которой обеспечивалась минимальный расход сетевого кабеля- Проектирование такой топологии связей между компьютерами в компьютерной сети, при которой обеспечивалась повышенная надежность соединений в следствие наличия избыточных каналов связи- Проектирование такой топологии связей между компьютерами в компьютерной сети, при которой обеспечивалась минимальный расход сетевого кабеля без учета топологических особенностей местности- Проектирование топологии типа «кольцо», при которой обеспечивалась минимальный расход сетевого кабеля		
16	Класс сортировок слиянием основан	a. На нахождении наибольшего (наименьшего) из оставшихся неупорядоченных элементов b. На нахождении места между меньшим (или равном) и большим (или равном) ключами в уже упорядоченной последовательности c. На подсчете элементов меньших данному d. На объединении двух или более упорядоченных последовательностей e. На обмене элементом местами, если они неупорядочены	d	<input checked="" type="checkbox"/>	https://studopedia.org/7-48391.html		
17	Времени работы алгоритма соответствует	a. случай, когда после компиляции данного алгоритма в машинный код время работы программы будет минимальным b. наиболее благоприятный случай, когда алгоритм работает меньше (быстрее) всего по времени, которое можно определить с помощью встроенного в компьютер таймером c. наименее вероятная цепочка исполнения команд алгоритма d. наименее неблагоприятный случай, когда алгоритм работает меньше (быстрее) всего по времени, которое можно определить с помощью встроенного в компьютер таймером e. случай, когда после компиляции данного алгоритма в машинный код будет выполнено минимальное число машинных команд и, как следствие, время выполнения программы будет минимальным f. случай, когда алгоритм выполняет меньшее число операций g. случай, когда после компиляции данного алгоритма в машинный код программа будет содержать минимальное число машинных команд	b, d	<input type="checkbox"/>			
18	Пусть дан граф $G=(V, E)$, $V=\{V_1, \dots, V_8\}$, с матрицей смежности: (смотреть фото в коментах) Определите хроматическое число графа.	a. 8 b. 6 c. 7 d. 4 e. 3 f. 5	a	<input checked="" type="checkbox"/>			
19	Что из указанного образуется подмножеством узлов графа (множественный выбор)	a. поток b. независимое множество ... графа c. путь d. эксцентриситет e. точка сопления графа f. остовное дерево	b, c, f	<input checked="" type="checkbox"/>			
20	Укажите арифметическое выражение в постфиксной форме записи, которое полностью соответствует следующему арифметическому выражению в префиксной форме записи: $+1+2*3-4^5+6\ 7+8\ 9+-+$	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. $P = 1\ 2\ 3\ 4\ 5\ +\ 6\ 7\ 8\ 9\ +\ -$	e	<input checked="" type="checkbox"/>	Prefix: $+1+2+3-4^5+6\ 7+8\ 9+-+$ Postfix: $1234567+*8+9+-$	Prefix to Postfix	

Какая	Вопрос	Варианты ответа (или скрин вариантов)	Ответ	Да?	Комментарий	Скрин задания	Источник ответа																																			
21	Пусть дан граф $G=(V, E)$, $V=(V_1, \dots, V_6)$, с матрицей смежности: Определите хроматическое число графа.	a. 3 b. 6 c. 4 d. 5 e. 2	a	<input checked="" type="checkbox"/>	 <p>Найденное количество цветов 3</p> <p>Матрица смежности</p> <p>Задайте матрицу смежности</p> <table border="1"><tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td></tr></table>	0	1	1	0	0	1	1	0	1	1	1	1	1	1	0	0	1	0	0	1	0	0	1	1	0	1	1	1	0	0	1	1	0	1	0	0	
0	1	1	0	0	1																																					
1	0	1	1	1	1																																					
1	1	0	0	1	0																																					
0	1	0	0	1	1																																					
0	1	1	1	0	0																																					
1	1	0	1	0	0																																					
22	Укажите верные соотношения (множественный выбор)	смотреть фото в скрине задания	d, e	<input type="checkbox"/>																																						
23	Дана сеть $N=(G, a)$, $G=(V, E)$, $a(e)$ – пропускная способность дуги e . Определите максимальный поток через сеть N (смотрите фото в комментарии)	a. 9 b. 5 c. 4 d. 7 e. 3 f. 8 g. 6	c	<input checked="" type="checkbox"/>	 <p>Максимальный поток из 0 в 7 равен 4</p>																																					
24	Какому алгоритму соответствует следующий фрагмент: "На каждом шаге рассматриваются в лексикографическом порядке дуги, исходящие из текущей вершины. Если дуга ведёт к еще не посещенному узлу, то оставные еще не посещенные узлы, связанные другой с текущим, помещаются в стек, а 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".	a. решение задачи коммивояжера b. обходу графа методом поиска в ширину c. алгоритму Прима d. вычислению эксцентриситета графа e. алгоритму Дейкстра f. обходу графа методом поиска в глубину g. алгоритму Крускала	f	<input checked="" type="checkbox"/>	 <p>https://studizba.com/files/show/doc/203437-1-saod_2_kurs_grafy_4.html</p>		https://studizba.com/files/show/doc/203437-1-saod_2_kurs_grafy_4.html																																			
25	Что из указанного является скалярной величиной для графа (множественный выбор)	a. пропускная способность b. поток c. сильно связная компонента d. плотность графа e. цикл f. точка сочленения графа g. разрез	a, d	<input checked="" type="checkbox"/>	<p>Плотность графа – это величина, значение которой равно отношению суммы весов всех рёбер графа к максимально возможному весу, который мог бы иметь рассматриваемый граф.</p> <p>Пропускная способность сечения $D \subseteq E$ называется неориентированное число.</p>																																					

Какав	Вопрос	Варианты ответа (или скрин вариантов)	Ответ	Да?	Комментарий	Скрин задания	Источник ответа
26	Укажите описание алгоритма, которое может являться фрагментом алгоритма обхода методом поиска в ширину	<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".</p> <p>b. "На каждом шаге рассматривается в лексикографическом порядке дуги, исходящие из текущей вершины. Если дуга ведет к еще не посещенному узлу, то оставные 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 "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>c. "На каждом шаге алгоритма находится ребро наименьшей стоимости (u, v) такое, что $u \in V \setminus U$ и $v \in V$, 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 уверенными, что кратчайший путь от источника к конкретной вершине проходит только через вершины множества U. Когда множество U будет содержать все вершины ортрафа, тогда вспомогательный массив будет содержать длины кратчайших путей от источника к каждой вершине.</p> <p>e. "На каждом шаге рассматривается 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>	e	<input checked="" type="checkbox"/>	https://prog.bobrodobro.ru/8083		
27	"Жадные" алгоритмы (схемы поиска решений) ...	<p>a. включают этап возврата к предыдущему успешному шагу, в случае если текущий шаг поиска решения 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>	d	<input checked="" type="checkbox"/>	<h3>Жадные алгоритмы</h3> <p>Жадный алгоритм — однопроходный итерационный алгоритм. Строит решение, добавляя на каждом шаге к текущему частичному решению новый элемент. Добавляемый элемент выбирается на основе локального оптимума («наилучший на текущем шаге»).</p> <p>Жадные алгоритмы:</p> <ul style="list-style-type: none"> • Дают точное решение для задач на матридах (с аддитивной целевой функцией) • Для некоторых задач дают приближенное решение с гарантированной (не обязательно константной) оценкой приближения • В общем случае используются как эвристики <p>Рассмотрим жадные эвристики для «Задачи коммивояжера».</p>		http://edu.mmcs.sfu.edu.ru/pluginfile.php/15567/mod_resource/content/7/13%20D%AD%D0%B2%D1%80%D0%B8%D1%81%D1%82%D0%BB%D1%87%D0%BC%D1%81%D0%BA%D0%B8%D0%BC%D5%20%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC%D1%8B.pdf
28	Определите хроматическое число графа (смотреть фото в комментарии)	<p>a. 7 b. 2 c. 4 d. 6 e. 5 f. 3</p>	a	<input checked="" type="checkbox"/>			
29	С помощью алгоритма Дейкстры определите наименее расстояние от вершины v_0 до наиболее удаленной вершины графа (смотреть фото в комментарии)	<p>a. 4 b. 5 c. 6 d. 7 e. 3</p>	c	<input checked="" type="checkbox"/>	<p>Расстояние между вершинами: 0: 0-1-3-2-4-5</p>		

Какав	Вопрос	Варианты ответа (или скрин вариантов)	Ответ	Да?	Комментарий	Скрин задания	Источник ответа
30	Дана сеть $N=(G, a)$, $G=(V, E)$, $a(e)$ – пропускная способность дуги e . Определите чему равен минимальный разрез сети N	a. 6 b. 9 c. 7 d. 8 e. 5 f. 4	b	<input checked="" type="checkbox"/>	Из графа 7 выбираем минимальное значение для всех работ, оканчивающихся на 8. Продолжительность минимального пути: 5		https://math.semestr.ru/setm/minput.php
31	На вход алгоритма построения дерева двоичного поиска (изначально дерево было пустое) была подана следующая последовательность чисел: 18, 20, 14, 10, 1, 16, 3, 17, 5, 40, 35, 45. Необходимо удалить корневой элемент. Определите элемент, который после этого будет корнем дерева. (множественный выбор)	a. 20 b. 17 c. 14 d. 18 e. 16 f. 35 g. 1	a	<input checked="" type="checkbox"/>			https://www.cs.usfca.edu/~galles/visualization/BST.html
32	Вычислить арифметическое выражение в постфиксной форме записи: $+*1\ 3-*5^*+924+68$	письменный ответ	-99	<input checked="" type="checkbox"/>	Prefix: $+*1\ 3-*5^*+924+68$ Infix: $((1*3)+(((5-7)*(9+2)*4)-(6+8)))$ $(1 * 3) + (((5 - 7) * ((9 + 2) * 4)) - (6 + 8))$ -99		
				<input type="checkbox"/>			
1	Какие операции необходимо реализовать при разработке программы управления отдельным элементом структуры данных:	a. Уничтожение структуры b. Сортировка по различным полям структуры c. Добавление нового элемента d. Инициализация структуры	c	<input type="checkbox"/>			
2	Укажите правильную аналогию между асимптотическим сравнением двух функций f и g для обозначения $f(n)=O(g(n))$ и сравнением двух действительных чисел a и b :	a. $a = b$ b. $a < b$ c. $a \gg b$ d. $a \geq b$	d	<input checked="" type="checkbox"/>	$f(n) = O(g(n))$ is like $a \ll b$ $f(n) = \Omega(g(n))$ is like $a \gg b$ $f(n) = \Theta(g(n))$ is like $a \approx b$ $f(n) = o(g(n))$ is like $a < b$ $f(n) = \omega(g(n))$ is like $a > b$		Аналогия между асимптотически сравнимыми для функций f и g и сравнимыми для действительных чисел a и b : $f(n) = O(g(n)) \approx a \leq b$, $f(n) = \Omega(g(n)) \approx a \geq b$, $f(n) = \Theta(g(n)) \approx a = b$, $f(n) = o(g(n)) \approx a < b$, $f(n) = \omega(g(n)) \approx a > b$.
3	Для оценки порядка роста функций, описывающих вычислительную сложность алгоритмов, используются асимптотические обозначения (символики) или нотации. Что обозначает запись $f(n)=O(g(n))$:	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 $g(n)$	d	<input checked="" type="checkbox"/>	Найдется константа C , что для любого n , начиная с некоторого \bar{n} , справедливо $ f(n) \leq C g(n) $		 В O -обозначении функция асимптотически ограничивается сверху и снизу. Если же достаточно определить только асимптотическую верхнюю границу, то для этого используется O -обозначение. Для функции $g(n)$ ($O(g(n))$) предполагается "о большем от g и n " или просто "о g и n " означает множество функций, таких что $O(g(n)) = \{ f(n) : \text{существуют положительные константы } c \text{ и } n_0 \text{ такие что } 0 \leq f(n) \leq c g(n) \text{ для всех } n \geq n_0\}$. О-обозначение применяется, когда нужно учесть различные функции, которые отличаются только постоянными множителями. Понятие представления о O -обозначении позволяет получить рис. 6. Для всех n , лежащих справа от n_0 , значение функции $f(n)$ не превышает значения функции $g(n)$. Если $f(n) = O(g(n))$, то функция $f(n)$ принадлежит множеству $O(g(n))$. Но $f(n) = O(g(n))$ следует $f(n) = O(g(n))$, поскольку O -обозначения более сплошные, чем O -обозначения. В обозначениях теории множеств $O(g(n)) \subset O(g(n))$.
4	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:	a. Когда завершиться ввод исходных данных b. Когда возникнет переполнение стека c. Когда результат не будет получен d. Когда достигается простейшее возможное решение	a	<input checked="" type="checkbox"/>			
5	Как называется алгоритм, который выполнит сортировку исходного массива $(3,1,5,2,4)$ после следующей последовательности проходов $(1,3,5,2,4)$, $(1,3,5,2,4)$, $(1,3,2,5,4)$, $(1,3,2,4,5)$, $(1,2,3,4,5)$, $(1,2,3,4,5)$	a. Простого выбора b. Простого обмена c. Простой вставки d. Шелла	b	<input checked="" type="checkbox"/>			
6	Как называется алгоритм, который выполнит сортировку исходного массива $(3,1,5,2,4)$ после следующей последовательности проходов $(1,3,5,2,4)$, $(1,3,5,2,4)$, $(1,2,3,5,4)$, $(1,2,3,4,5)$	a. Простой вставки b. Простого выбора c. Простого обмена d. Шелла	a	<input checked="" type="checkbox"/>			
7	Какой зависимостью описывается функция вычислительной сложности алгоритма быстрой сортировки методом Шелла в наилучшем случае:	a. $f(n)=O(n)$ b. $f(n)=O(n^2)$ c. $f(n)=O(n \log(n))$ d. $f(n)=O(\log(n))$	a	<input checked="" type="checkbox"/>			
8	Какой зависимостью описывается функция вычислительной сложности алгоритма поиска хешированием в лучшем случае:	a. $f(n)=\Theta(1)$ b. $f(n)=\Theta(n^2)$ c. $f(n)=\Theta(\log(n))$ d. $f(n)=\Theta(n \log(n))$	a	<input checked="" type="checkbox"/>			

Какав	Вопрос	Варианты ответа (или скрин вариантов)	Ответ	Да?	Комментарий	Скрин задания	Источник ответа
9	Какой зависимостью описывается функция вычислительной сложности алгоритма интерполяционного поиска в худшем случае:	a. $f(n)=\Theta(\log(\log(n)))$ b. $f(n)=\Theta(n)$ c. $f(n)=\Theta(\log(n))$ d. $f(n)=\Theta(n \log(n))$	b	<input checked="" type="checkbox"/>			
10	Что определяет понятие "глубина рекурсии" для рекурсивной функции:	a. Количество операторов вызова функции в самой функции b. Наибольшее количество операторов в рекурсивной функции c. Количество вызовов внешних функций d. Наибольшее одновременное количество рекурсивных обращений функции	d	<input checked="" type="checkbox"/>			
11	В основе алгоритма Бойера-Мура используется:	a. Условие Айверсона b. Таблица смещений ("стоп-символов") c. Хэш-таблица d. Таблица префиксов	a	<input checked="" type="checkbox"/>			
12	В основе алгоритма Кнута-Морриса-Пратта используется:	a. Таблица смещений ("стоп-символов") b. Хэш-таблица c. Условие Айверсона d. Таблица префиксов	c	<input checked="" type="checkbox"/>			
13	Имеется указатель q на узел в середине линейного односвязного списка со структурой узла struct Tnode{ Tdata data; Tnode* next; }; 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 выполнилась данная операция вставки:	a. $q->next=qq; qq->next=q; swap(qq->data,q->data);$ b. $(*qq)=(*q);$ c. $qq=q;$ d. $qq->next=q->next; q->next=qq; swap(qq->data, q->data);$	a	<input checked="" type="checkbox"/>			
14	Какой зависимостью определяется функция вычислительной сложности алгоритма поиска в тексте по образцу методом Кнута-Морриса-Пратта в лучшем случае		$O(n+m)^2$	<input checked="" type="checkbox"/>	$O(n+m)^2$ коммент верный		
15	Есть ли ошибка в коде функции f, которая осуществляет получение ссылки на последний узел линейного односвязного списка со структурой узла struct Tnode(Tdata data; Tnode* next;); Tnode *f(Tnode *L) { Tnode *q=L; while(q){ q=q->next; } return q; }	a. Ошибка в записи оператора $q=q->next$ b. Ошибка в условии while (q) - должно быть while ($q->next$) c. Ошибки нет d. Ошибка в записи оператора $Tnode *q=L$ - должно быть $Tnode q=L$	b	<input checked="" type="checkbox"/>			
16	Имеется линейный односвязный список из $(n>2)$ узлов. Структура узла списка struct Tnode(Tdata data; Tnode* next;); L - указатель на его начало. Укажите группу операторов, которые обеспечивают корректное удаление двух узлов из начала списка L:	a. $q1=L; delete q1; q2=q1; L=q2->next; delete q2;$ b. $q1=L; q2=L->next; L=L->next->next; delete q1; delete q2;$ c. $q1=L; L=L->next->next; q2=L; delete q1; delete q2;$ d. $q1=L; q2=L; L=q2->next; delete q1; delete q2;$	c	<input checked="" type="checkbox"/>			
17	Имеется бинарное дерево поиска, содержащее целые числа от 1 до 7. Каким будет результат вывода при обходе дерева снизу-вверх (обратный/postorder обходе):	a. 7, 6, 5, 4, 3, 2, 1 b. 4, 1, 5, 2, 6, 3, 7 c. 4, 1, 2, 3, 5, 6, 7 d. 1, 2, 3, 4, 5, 6, 7	a	<input checked="" type="checkbox"/>			https://www.cs.usfca.edu/~galles/visualization/BST.html

Какая	Вопрос	Варианты ответа (или скрин вариантов)	Ответ	Да?	Комментарий	Скрин задания	Источник ответа
18	В какой последовательности располагаются вершины дерева при обходе "сверху-вниз" (прямой/preorder): (смотрите фото в комментарии)	a. 1, 2, 4, 8, 9, 5, 10, 11, 3, 6, 12, 13, 7, 14, 15 b. 8, 4, 9, 2, 10, 5, 11, 1, 12, 6, 13, 3, 14, 7, 15 c. 8, 9, 4, 10, 11, 5, 2, 12, 13, 6, 14, 15, 7, 3, 1 d. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15	a	<input checked="" type="checkbox"/>			 <p>Рис. 4.7. Различные направления обхода дерева.</p>
19	Какое из перечисленных АВЛ-деревьев требует проведения перестройки:	a.4 b. 1 c. 3 d. 2	c	<input checked="" type="checkbox"/>			Бинарное дерево является сбалансированн
20	Какие из перечисленных АВЛ-деревьев не требуют проведения перестройки:	a. 1, 3, 4 b. 2, 3, 4 c. 1, 2, 4 d. 1, 2, 3	c	<input checked="" type="checkbox"/>			Бинарное дерево является сбалансированн
21	Какие поля должны содержать записи хеш-таблицы при использовании метода цепочек:	a. Адрес следующего элемента списка b. Значение хеш-функции c. Указатель на начало списка ключей с одним и тем же значением хеш-функции d. Ключ элемента	c	<input checked="" type="checkbox"/>			
22	Что такое структурированный протокол:	a. Простейший способ записи кластеров данных в файл b. Сетевые правила для миграции бинарных файлов c. Тип сохраненного кластера d. Простой формат файлов для доступа из любой среды	a	<input checked="" type="checkbox"/>	Особый тип двоичного файла, известный как файл структурированного протокола, - это простейший способ записи кластеров данных в файл. https://studfile.net/preview/6324281/page:5/		
23	Сколько байт бинарного файла занимает число 0 в двоичном формате:	a. 0 байт b. 8 байт c. 4 байта d. 1 байт	d	<input checked="" type="checkbox"/>			
24	Алгоритм Прима - это:	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 c. Алгоритм поиска минимального остовного дерева во взвешенном неориентированном связном графе d. Алгоритм для нахождения кратчайших путей всеми вершинами взвешенного ориентированного графа	c	<input checked="" type="checkbox"/>	—алгоритм построения минимального оствового дерева взвешенного связного неориентированного графа.		
25	Как называется алгоритм для нахождения кратчайших путей между всеми вершинами взвешенного ориентированного графа:	a. Алгоритм Крускала b. Алгоритм Прима c. Алгоритм Дейкстры d. Алгоритм Флойда-Уоршала	d	<input checked="" type="checkbox"/>	Алгоритм Флойда — Уоршала — динамический алгоритм для нахождения кратчайших расстояний между всеми вершинами взвешенного ориентированного графа. Разработан в 1962 году Робертом Флойдом и Стивеном Уоршлом.		
26	Как называется алгоритм поиска минимального оствового дерева во взвешенном неориентированном связном графе:	a. Алгоритм Дейкстры b. Алгоритм Крускала c. Алгоритм Белмана-Форда d. Алгоритм Флойда-Уоршала	b	<input checked="" type="checkbox"/>	https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%8B%D1%82%D0%BC_%D0%9A%D1%80%D0%BD%D1%81%D0%BA%D0%BD%D0%BB%D0%BD		
27	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:	a. ABCCCCDDDD b. BDDCCDADCD c. DDDDDCCCCBA d. ABDCDCDCDD	b	<input checked="" type="checkbox"/>			

Какав	Вопрос	Варианты ответа (или скрин вариантов)	Ответ	Да?	Комментарий	Скрин задания	Источник ответа
52	Чем может быть вызвано переполнение стека при выполнении рекурсивной функции с небольшой глубиной рекурсии:	a. Неправильным выполнением вызова функции вопрос b. Неправильно оформленным выводом результата c. Неправильно оформленным блоком входа в рекурсию d. Неправильно сформулированным условием завершения рекурсии	d	<input checked="" type="checkbox"/>			
53	Какой код будет получен из входной строки символов ABCCCDDDD при использовании алгоритма Хаффмана:	a. 110 111 10 10 10 0 0 0 0 0 b. 00 01 10 10 11 11 11 11 11 c. 1 2 3 3 3 4 4 4 4 d A1 B1 C3 D5	d	<input checked="" type="checkbox"/>			
54	Какая структура данных относятся к категории линейных списков:	a. Бинарное дерево b. Массив c. Дек d. Множество	c	<input checked="" type="checkbox"/>	Дек — это линейный список, в котором можно добавлять и удалять элементы как с одного, так и с другого конца.		
55	Что такое инвариант цикла:	a. Логическое выражение, истинное после каждого прохода тела цикла и перед началом выполнения цикла, зависящее от констант, изменяющихся в теле цикла b. Логическое выражение, истинное после каждого прохода тела цикла и в конце выполнения цикла, зависящее от переменных, изменяющихся в теле цикла c. Логическое выражение, истинное после первого прохода тела цикла и в конце выполнения цикла, зависящее от переменных, изменяющихся в теле цикла d. Логическое выражение, истинное после первого прохода тела цикла и в конце выполнения цикла, от переменных, изменяющихся в теле цикла	a	<input checked="" type="checkbox"/>	https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D0%B2%D0%80%D1%80%D0%BD%D0%BD%D1%82%D1%86%D0%BB%D0%BA%D0%BB%D0%B0		
56	Какие основные операции следует учитывать при оценке временной (вычислительной) сложности алгоритмов поиска:	a. Операции сравнения b. Операции сравнения и перемещения c. Операции сложения и перемещения d. Операции перемещения	b	<input checked="" type="checkbox"/>	https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D0%BD%D0%BC%D0%BD%D0%BD%D1%8F%D1%81%D0%BB%D0%BE%D0%BD%D0%BD%D0%BD%D1%82%D1%81%D1%82%D1%8C%D0%BD%D0%BB%D0%B3%D0%BE%D1%80%D0%BB%D1%82%D0%BC%D0%B0		
57	Что такое алгоритм:	a. Последовательность вычислительных шагов, преобразующих промежуточные данные во входные, охватывающие все случаи хода вычислительного b. Последовательность вычислительных шагов, преобразующих входные данные в выходные c. Любая вычислительная процедура d. Средство, предназначенное для постановки вычислительной задачи	b	<input checked="" type="checkbox"/>			
58	Укажите правильную аналогию между асимптотическим сравнением двух функций и для обозначения $f(n)=O(g(n))$ и сравнением двух действительных чисел a и b :	a. $a > b$ b. $a \geq b$ c. $a < b$ d. $a = b$	d	<input checked="" type="checkbox"/>	$f(n) = O(g(n))$ is like $a < b$ $f(n) = \Omega(g(n))$ is like $a \geq b$ $f(n) = \Theta(g(n))$ is like $a == b$ $f(n) = o(g(n))$ is like $a < b$ $f(n) = \omega(g(n))$ is like $a > b$		Аналогия между асимптотически сравниваемыми двух функций и в сравнении двух действительных чисел a и b : $f(n) = O(g(n)) \approx a \leq b$, $f(n) = \Omega(g(n)) \approx a \geq b$, $f(n) = \Theta(g(n)) \approx a = b$, $f(n) = o(g(n)) \approx a < b$, $f(n) = \omega(g(n)) \approx a > b$.
59	Когда один алгоритм считается эффективнее другого:	a. Если его вычислительная сложность в наихудшем случае имеется b. Если его вычислительная сложность в наилучшем случае имеется c. Если его вычислительная сложность в наихудшем случае имеется d. Если его вычислительная сложность в наилучшем случае имеется	a	<input checked="" type="checkbox"/>			
60	Как называется алгоритм, который напрямую или через другие вспомогательные алгоритмы вызывает сам себя:	a. Циклический b. Развезывающийся c. Линейный d. Рекурсивный	d	<input checked="" type="checkbox"/>	Алгоритм обратности задачи Рекурсия называется алгоритмом, который возвращает результат выполнения задачи обратно в исходное состояние. Алгоритм обратности задачи Рекурсия называется алгоритмом, который возвращает результат выполнения задачи обратно в исходное состояние. Алгоритм обратности задачи Рекурсия называется алгоритмом, который возвращает результат выполнения задачи обратно в исходное состояние. Алгоритм обратности задачи Рекурсия называется алгоритмом, который возвращает результат выполнения задачи обратно в исходное состояние.		
61	Какой зависимостью описывается функция вычислительной сложности алгоритма сортировки методом простой вставки в наилучшем случае:	a. $f(n)=O(\log(n))$ b. $f(n)=O(n^2)$ c. $f(n)=O(n \log(n))$ d. $f(n)=O(n)$	d	<input checked="" type="checkbox"/>			
62	Как называется алгоритм, который выполняет сортировку исходного массива $(3,1,5,2,4)$ после следующего последовательности проходов $(1,3,5,2,4), (1,3,5,2,4), (1,3,2,5,4), (1,3,2,4,5), (1,3,2,4,5), (1,2,3,4,5), (1,2,3,4,5)$	a. Шелла b. Простого выбора c. Простого обмена d. Простой вставки	c	<input checked="" type="checkbox"/>			
63	Какой зависимостью описывается функция вычислительной сложности алгоритма сортировки слиянием (Mergesort) в среднем и наихудшем случаях:	a. $f(n)=O(\log(n))$ b. $f(n)=\Theta(n^2)$ c. $f(n)=O(n)$ d. $f(n)=\Theta(n \log(n))$	b	<input checked="" type="checkbox"/>			
64	Как вычислительная сложность описывает реализацию алгоритма Дейкстры для графов, представленных в виде списков смежности (p и m обозначают соответственно число вершин и ребер исходного графа)	a. $O(m \log n)$ b. $O(m + n)$ c. $O(mn)$ d. $O(n^2)$	d	<input checked="" type="checkbox"/>	https://foxford.ru/wiki/informatika/algoritm-deykstry		
65	Основное требование, предъявляемое к массиву для возможности выполнения двойичного поиска:	a. Упорядоченность массива b. Массивы целочисленных значений c. Нет особых требований d. Массивы относительно небольшого размера	a	<input checked="" type="checkbox"/>	Работает только в отсортированных массивах		

Какав	Вопрос	Варианты ответа (или скрин вариантов)	Ответ	Да?	Комментарий	Скрин задания	Источник ответа
105	Какой зависимостью описывается функция вычислительной сложности алгоритма поиска в тексте по образцу методом Кнута-Моррисса-Пратта в лучшем случае	a. $f(n, m) = \Theta(\log(n+m))$ b. $f(n) = O(n^2)$ c. $f(n, m) = \Theta(\log(n'm))$ d. $f(n) = \Theta(n)$	a	<input checked="" type="checkbox"/>	По табличке ответов из 2 сема		
106	Какие основные операции следует учитывать при оценке временной (вычислительной) сложности алгоритмов сортировки	a. Операции сравнения и перемещения b. Операции сложения и перемещения c. Операции сравнения d. Операции сравнения и сложения	a	<input checked="" type="checkbox"/>	Спасибо, Сырок		
107	Временная (вычислительная) сложность алгоритма определяется количеством входных данных n. Этот параметр пропорционален величине обрабатываемого набора данных и может обозначать	a. Количество функций в программе b. Количество основных операций c. Размер занимаемой памяти при сортировке или поиске d. Размер массива или файла	a	<input type="checkbox"/>			
108	Основное дерево - это:	a. Подграф, полученный путем кратчайшего обхода всех вершин графа b. Подграф, полученный путем удаления вершин, связанных со всеми остальными c. Дерево, полученное путем удаления вершин, связанных с более чем половиной всех вершин в графе d. Подграф, полученный путем удаления максимального числа ребер, без нарушения связности	d	<input checked="" type="checkbox"/>			
109	Какой зависимостью описывается функция вычислительной сложности алгоритма сортировки подсчетом (Counting sort) в среднем и наилучшем случаях:	a. $f(n) = O(n^2)$ b. $f(n) = O(\log(n))$ c. $f(n) = O(n+k)$ d. $f(n) = O(n)$	c	<input checked="" type="checkbox"/>	См 102		https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D1%80%D1%82%D0%B6%D1%80%D0%BE%D0%B2%D0%BA%D0%B0%D0%BF%D0%BE%D0%BD%D1%81%D1%87%D1%91%D1%82%D0%BE%D0%BC%D0%90%D0%90%D0%BD%D0%BD%D0%BD%D0%BB%D0%BD%D0%88%D0%BD%D0%BA
110	Какой зависимостью описывается функция вычислительной сложности алгоритма интерполяционного поиска в среднем случае:	a. $f(n) = \Theta(\log(n))$ b. $f(n) = \Theta(n \log(n))$ c. $f(n) = \Theta(n)$ d. $f(n) = \Theta(\log(\log(n)))$	d	<input checked="" type="checkbox"/>			https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%BF%D1%80%D0%BF%D0%BE%D0%BD%D1%8F%D1%86%D0%BF%D0%BD%D0%BB%D1%81%D0%BD%D1%8B%D0%90%D0%90%D0%BD%D0%BD%D0%BF%D0%BE%D0%BD%D0%BB%D1%81%D0%BD%D0%BA
111	Имеется описание структуры узла линейного односвязного списка struct Tnode{ Tdata data; XXXX next; }; Какое определение должно быть на месте XXXX:	a. Tnode b. void c. next d. *Tnode	d	<input type="checkbox"/>	По табличке ответов из 2 сема		
112	При каком значении показателя сбалансированности b (разницы высот правого и левого поддеревьев) необходимо выполнить перестройку АВЛ-дерева	a. $ b = 2$ b. $ b = 1$ c. $ b > 1$ d. $ b > 2$	c	<input checked="" type="checkbox"/>			https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%BF%D1%80%D0%BF%D0%BE%D0%BD%D1%8F%D1%86%D0%BF%D0%BD%D0%BB%D1%81%D0%BD%D1%8B%D0%90%D0%90%D0%BD%D0%BD%D0%BF%D0%BE%D0%BD%D0%BB%D1%81%D0%BD%D0%BA
113				<input type="checkbox"/>			
114	Имеется идеально сбалансированное двоичное дерево, содержащее 31 узел. Какова высота этого дерева	a. 5 уровней b. 6 уровней c. 7 уровней d. 4 уровня	a	<input type="checkbox"/>	$2^5 = 32$ $16 < 31 < 32$		
115				<input type="checkbox"/>			
116				<input type="checkbox"/>			
117				<input type="checkbox"/>			
118				<input type="checkbox"/>			
119				<input type="checkbox"/>			
120				<input type="checkbox"/>			
121				<input type="checkbox"/>			
122				<input type="checkbox"/>			
123				<input type="checkbox"/>			
124				<input type="checkbox"/>			
125				<input type="checkbox"/>			
126				<input type="checkbox"/>			
127				<input type="checkbox"/>			
128				<input type="checkbox"/>			
129				<input type="checkbox"/>			
130				<input type="checkbox"/>			
131				<input type="checkbox"/>			
132				<input type="checkbox"/>			
133				<input type="checkbox"/>			
134				<input type="checkbox"/>			