

Документ подписан простой электронной подписью Информация о владельце: ФИО: Таскаев Сергей Валерьевич Должность: Ректор Дата подписания: 26.06.2026 11:04:24	МИНОБНАУКИ РОССИИ Федеральное государственное бюджетное образовательное учреждение высшего образования «Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)	
Уникальный идентификатор документа: 04c19ed8b109b1506c07a48809a078888521525	Фонд оценочных средств для промежуточной аттестации по дисциплине "Алгоритмы и анализ сложности" по направлению подготовки (специальности) "09.03.04 Программная инженерия" направленности (профилю) "Разработка программно-информационных систем ФГБОУ ВО «ЧелГУ»"	стр. 1

**Фонд оценочных средств для промежуточной аттестации по дисциплине
Алгоритмы и анализ сложности**

Направление подготовки (специальность)

09.03.04 Программная инженерия

Направленность (профиль)

Разработка программно-информационных систем

Присваиваемая квалификация (степень)

бакалавр

Форма обучения

очная форма обучения

Год(ы) набора 2026

Челябинск 2026 г.

09.03.04 Программная инженерия профиль Разработка программно-информационных систем, дисциплина Алгоритмы и анализ сложности, 2026 год набора, очная форма обучения

Фонд оценочных средств дисциплины (модуля) одобрен и рекомендован:

Проректор по учебной работе утверждено 27.02.2026 А.А. Саламатов

Ученым советом института информационных технологий

Протокол заседания № 7 от 26.02.2026

Председатель Ученого совета
института информационных
технологий

согласовано

Ю.В. Петриченко

Заседанием кафедры информационных технологий и экономической информатики

Протокол заседания №7 от 26.02.2026

Заведующий кафедрой

согласовано

С.А. Скрипов

Автор (составитель)

И.Е. Николаев

Структура фондов оценочных средств соответствует приказу ректора ФГБОУ ВО «ЧелГУ» от 27 сентября 2022 № 573-1



Содержание

1. Паспорт фонда оценочных средств	3
2. Перечень формируемых компетенций	4
3. Содержание оценочных средств по дисциплине	6
3.1. Виды оценочных средств	6
3.2. Содержание оценочных средств	7
4. Порядок проведения и критерии оценивания промежуточной аттестации	56
4.1. Порядок проведения промежуточной аттестации	56
4.2. Критерии оценивания промежуточной аттестации по видам оценочных средств	56
4.3. Результаты промежуточной аттестации и уровни сформированности компетенций	57



1. Паспорт фонда оценочных средств

Направление подготовки: 09.03.04 Программная инженерия

Направленность: Разработка программно-информационных систем

Дисциплина: Алгоритмы и анализ сложности

Семестры: 3.

Форма промежуточной аттестации: экзамен.

Для оценивания результатов обучения используется балльно-рейтинговая система.



2. Перечень формируемых компетенций

Изучение дисциплины «Алгоритмы и анализ сложности» направлено на формирование компетенций, приведённых в 1.

Таблица 1. Результаты обучения по дисциплине.

Коды компетенции и согласно ФГОС (ОПОП ВО)	Содержание компетенций согласно ФГОС (ОПОП ВО)	Индикаторы достижения компетенции согласно ОПОП	Перечень планируемых результатов обучения по дисциплине
1	2	3	4
ОПК-6	Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов;	ОПК-6.1. Демонстрирует знание основ информатики, теории алгоритмов, методологии и технологии программирования ОПК-6.2. Демонстрирует умения разрабатывать алгоритмические и программные решения, проводить проектирование, конструирование и тестирование программных продуктов ОПК-6.3. Имеет практический опыт использования технологий разработки программного обеспечения	Знать:основные сведения о методах и способах построения алгоритмов для различных технических задач; основные и наиболее популярные программные продукты, позволяющие проектировать и разрабатывать алгоритмы. Уметь:выбирать соответствующие условиям поставленной задачи структуры представления данных, а также алгоритмы обработки информации; выбирать наиболее подходящий алгоритм в рамках конкретной задачи и типа данных Владеть:навыками использования программных средств для проектирования и разработки, а также анализа разработанных алгоритмов
ОПК-7	Способен применять в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой;	ОПК-7.1. Демонстрирует знание основных концепций, принципов, теорий и фактов, связанных с информатикой ОПК-7.2. Демонстрирует умения применять на практике основные концепции, принципы и теории из области информатики при решении стандартных задач ОПК-7.3. Имеет практический опыт решения задач профессиональной деятельности с использованием основ информатики	Знать:основные сведения о методах и способах построения алгоритмов для различных технических задач; актуальные и эффективные алгоритмы обработки информации, представленной в различном виде; основные и наиболее популярные программные продукты, позволяющие проектировать и разрабатывать алгоритмы. Уметь:производить анализ сложности алгоритма и находить пути упрощения полученных алгоритмов. Владеть:навыками использования программных средств для проектирования и разработки, а также анализа



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)

Фонд оценочных средств для промежуточной аттестации по дисциплине (модулю) «Алгоритмы и анализ сложности» по направлению подготовки 09.03.04 «Программная инженерия» направленности «Разработка программно-информационных систем» ФГБОУ ВО «ЧелГУ»

стр. 5

разработанных алгоритмов.



3. Содержание оценочных средств по дисциплине

3.1. Виды оценочных средств

Таблица 2. Виды оценочных средств.

№ п/п	Код компетенции/ планируемые результаты обучения	Контролируемые темы/ разделы	Наименование оценочного средства для текущего контроля	Наименование оценочного средства на промежуточной аттестации/№ задания
1	ОПК-6.1. Демонстрирует знание основ информатики, теории алгоритмов, методологии и технологии программирования Знать: основные сведения о методах и способах построения алгоритмов для различных технических задач; основные и наиболее популярные программные продукты, позволяющие проектировать и разрабатывать алгоритмы.	Теория алгоритмов Анализ сложности алгоритмов	Опрос. Проверка практических работ.	Экзаменационный билет и практические работы
2	ОПК-6.2. Демонстрирует умения разрабатывать алгоритмические и программные решения, проводить проектирование, конструирование и тестирование программных продуктов Уметь: выбирать соответствующие условиям поставленной задачи структуры представления данных, а также алгоритмы обработки информации; выбирать наиболее подходящий алгоритм в рамках конкретной задачи и типа данных	Теория алгоритмов Анализ сложности алгоритмов	Опрос. Проверка практических работ.	Экзаменационный билет и практические работы
3	ОПК-6.3. Имеет практический опыт использования технологий разработки программного обеспечения Владеть: навыками использования программных средств для проектирования и разработки, а также анализа разработанных алгоритмов	Теория алгоритмов Анализ сложности алгоритмов	Опрос. Проверка практических работ.	Экзаменационный билет и практические работы
4	ОПК-7.1. Демонстрирует знание основных концепций, принципов, теорий и фактов,	Теория алгоритмов Анализ сложности алгоритмов	Опрос. Проверка практических работ.	Экзаменационный билет и практические работы



	связанных с информатикой Знать:основные сведения о методах и способах построения алгоритмов для различных технических задач; актуальные и эффективные алгоритмы обработки информации, представленной в различном виде; основные и наиболее популярные программные продукты, позволяющие проектировать и разрабатывать алгоритмы.			
5	ОПК-7.2. Демонстрирует умения применять на практике основные концепции, принципы и теории из области информатики при решении стандартных задач Уметь:производить анализ сложности алгоритма и находить пути упрощения полученных алгоритмов.	Теория алгоритмов Анализ сложности алгоритмов	Опрос. Проверка практических работ.	Экзаменационный билет и практические работы
6	ОПК-7.3. Имеет практический опыт решения задач профессиональной деятельности с использованием основ информатики Владеть:навыками использования программных средств для проектирования и разработки, а также анализа разработанных алгоритмов.	Теория алгоритмов Анализ сложности алгоритмов	Опрос. Проверка практических работ.	Экзаменационный билет и практические работы

Типовые задания, критерии и показатели оценивания в рамках текущего контроля представлены в рабочей программе дисциплины (модуля). Полные комплекты оценочных средств и контрольно-измерительных материалов хранятся на кафедре.

3.2.Содержание оценочных средств

База тестовых вопросов

№ п/п	Формулировка вопроса	Варианты ответов (полужирным шрифтом – верные варианты)
1.	Выберите основные недостатки открытой адресации:	а. Каждый раз, когда встречается коллизия, увеличивается шанс возникновения новой коллизии б. Время поиска элемента при открытой адресации в среднем равно $O(1)$



		с. Время поиска элемента стремится к $O(n)$ d. Время поиска элемента стремится к $O(n/k)$, где k – размер хеш-таблицы
2.	Что будет, если повторно занести ключ в хеш-таблицу?	a. Произойдет дубликация ключей, все новые данные по ключу будут ссылаться на новый ключ b. Старая связь ключ-значение будет перезаписана новыми данными c. Ключ останется один, а данные запишутся в цепочку (список)
3.	Можно ли вывести элементы хеш-таблицы в определенном порядке?	a. Да, потому что в хеш-таблицу данные заносятся как в массив, следовательно, мы можем отсортировать элементы в заданном порядке b. Нет, так как хеш-таблица является неупорядоченной коллекцией
4.	Дана хеш-таблица, содержащая следующие индексы: [1, 3, 5, 7, 9]. При каком значении "x" хеш-функция $h(x, i) = (x + i*2) \bmod 10$ будет возвращать каждый индекс один раз, подставляя i от 1 до 9?	a. 2 b. 5 c. 4 d. 3 e. Такого значения нет
5.	Дана хеш-таблица, содержащая следующие индексы: [8, 1, 4, 7, 0, 3, 6, 9, 2]. При каком значении "x" хеш-функция $h(x, i) = (x + 3i) \bmod 10$ будет возвращать каждый индекс один раз, подставляя i от 1 до 9?	a. 3 b. 2 c. 5 d. 4 e. Такого значения нет
6.	Что такое коэффициент заполнения хеш-таблицы?	a. Отношения числа ключей, которое возвращает хеш-функция, к общему числу ключей хеш-таблицы b. Отношение числа ключей, которые хранятся в хеш-таблице, к размеру хеш-таблицы c. Отношение текущего числа элементов, привязанных к определенному ключу хеш-таблицы, к максимальному допустимому



7.	Чем хеш-таблица отличается от обычного массива, в котором в качестве индексов использовались бы хеш-значения?	a. Хеш-таблица существенно больше массива b. Возникновением коллизий c. Такого массива не существует, так как в качестве индексов нельзя использовать хеш-значения d. Ничем
8.	Что такое уникальный ключ?	a. Если разность хеш-значений двух элементов равна ключу b. Ключ, вызывающий коллизию c. Ключ, при обращении по которому получаем только один элемент
9.	Что из перечисленного может быть ключом хеш-таблицы?	a. Строка b. Указатель c. Число d. Ничего из перечисленного. Он задается автоматически
10.	Что такое коллизия?	a. Совпадение значений хеш-функции для двух разных ключей b. Ошибка инициализации ключа c. Использование недопустимых типов данных для ключа d. Совпадение ключей e. Совпадение значений хеш-функции, при обращении к одному и тому же ключу
11.	Что такое универсальное хеширование?	a. Хеширование, при котором хеш-функция может вычислять «хеш» как остаток от деления входных данных b. Хеширование, при котором хеш-функция может выполнять деление входных данных на полином по модулю два. c. Вид хеширования, при котором хеш-функция выбирается случайно для каждой хэш-таблицы и не зависит от набора ключей, с которыми работает. d. Вид хеширования, при

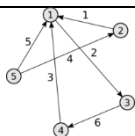
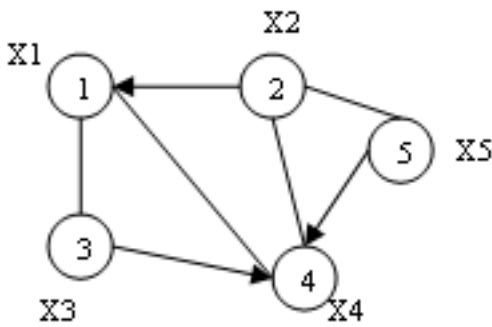
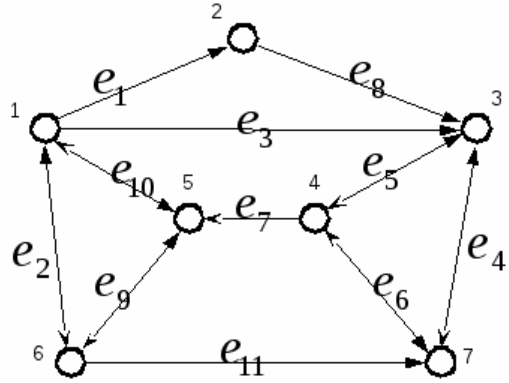
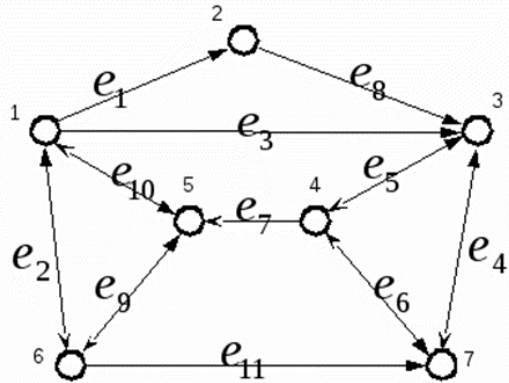


		котором хеш-функция отображает каждый ключ из набора во множество целых чисел без коллизий.
12.	Какие виды Хеш-таблиц (из нижеперечисленных) существуют?	a. Хеш-таблица с открытой адресацией b. Хеш-таблица с цепочками c. Хеш-таблица с закрытой адресацией d. С прямой адресацией e. С опциональной адресацией f. С замкнутой адресацией g. С невозможной адресацией
13.	Какие факторы могут замедлить операции хеш-таблицы?	a. Вычисление сложной хеш-функции при каждой операции b. Хеш-коллизии c. Организация таблицы как массива цепочек d. Ничего из перечисленного
14.	При каком методе разрешения коллизий хеш-таблица может оказаться заполненной, делая невозможной вставку новых элементов?	a. открытая адресация b. метод внешних цепочек c. оба вышеперечисленных
15.	Какие требования предъявляются к хорошей хэш-функции:	a. Минимизировать количество коллизий b. Исключение коллизий c. Распределение ключей по закону Гаусса d. Стохастический характер функции e. Случайный характер хэш-функции f. Функция обязательно должна быть детерминистической
16.	В каком диапазоне находится коэффициент α при организации хеш-таблицы с помощью метода открытой адресации?	a. $(-\infty; +\infty)$ b. $(0; 1)$ c. $(0; +\infty)$ d. $[1; +\infty)$ e. $[0; 1]$
17.	Дана хеш-функция $h(x, i) = (x + 3i) \bmod 10$. Напишите через запятую индексы (без пробелов), которые будет выдавать эта функция при $x=5$, подставляя i от 0 до 9	a. 5,8,1,4,7,0,3,6,9,2
18.	Выберите правильные утверждения	a. размер динамической структуры данных ограничивается только доступным объемом

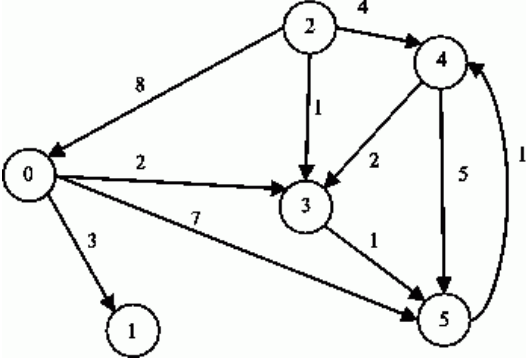


		<p>памяти компьютера b. размер динамической структуры данных фиксирован и задается во время её инициализации c. при изменении логической последовательности элементов динамической структуры (удалении, добавлении элемента), создается новая структура данных, а старая удаляется, высвобождая память компьютера</p>
19.	Какие из перечисленных структур данных являются динамическими?	<p>a. Деревья b. Стек c. Массивы d. Множества</p>
20.	При объявлении одномерного массива постоянной длины определяется	<p>a. тип элементов, имя массива b. тип элементов, нижнюю границу массива, верхнюю границу массива, имя массива, индекс массива c. тип элементов, нижнюю границу массива, верхнюю границу массива, имя массива, шаг для индекса массива d. тип элементов, количество элементов, имя массива</p>
21.	Первым шагом разработки алгоритма является	<p>a. Выбор абстрактного типа данных b. Выбор математической модели c. Выбор структуры данных d. Выбор языка программирования</p>
22.	Что характерно для статического типа данных?	<p>a. В процессе выполнения программы может меняться характер взаимосвязи между элементами структуры b. Не требуется дополнительной памяти c. Количество элементов структуры может не фиксироваться d. Количество элементов структуры заранее определено e. Характер взаимосвязи</p>

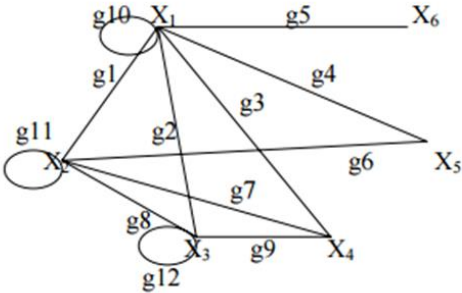


		между элементами структуры не изменен
23.	Какой граф изображён на рисунке? 	a. Смешанный граф b. Неориентированный граф c. Ориентированный граф d. Граф с петлёй
24.	Какой граф изображён на рисунке? 	a. Смешанный граф b. Неориентированный граф c. Ориентированный граф d. Граф с петлёй
25.	Какой из представленных вариантов будет считаться ориентированным маршрутом? 	a. e11, e3, e8, e1, e2 b. e5, e7, e10, e3, e11 c. e2, e11, e6, e7, e10, e2 d. E4, e1, e8, e2, e7, e11
26.	Какой из представленных вариантов будет считаться орцепью (Ориентированной цепью) 	a. e11, e3, e8, e1, e2 b. e5, e7, e10, e3, e11 c. e2, e11, e6, e7, e10, e2 d. E4, e1, e8, e2, e7, e11 e. e8, e4, e6, e7, e10, e1

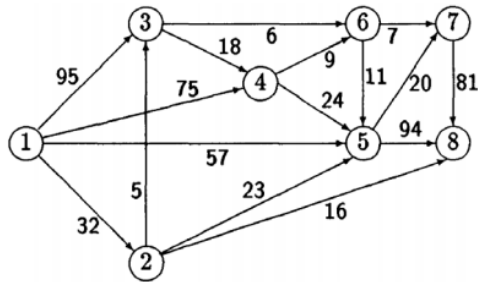


27.	 <p>Дан граф</p>	<p>a. Смешанный граф b. Неориентированный граф c. Ориентированный граф d. Граф с петлёй e. Графом со взвешенными дугами</p>
28.	<p>Определите какой шаг пропущен для поиска в глубину: 1. Всем вершинам графа присваивается значение не посещенная. Выбирается первая вершина и помечается как посещенная. 2. ??? 3. Повторить шаг 2 до тех пор, пока все вершины не будут помечены как посещенные. Варианты ответов:</p>	<p>a. Посещается первая вершина из очереди (если она не помечена как посещенная). Все ее соседние вершины заносятся в очередь. После этого она удаляется из очереди. b. Выбирается вторая вершина и помечается как не посещенная. Все ее соседние вершины заносятся в очередь. После этого она удаляется из очереди. c. Для последней помеченной как не посещенная вершина выбирается смежная вершина, являющаяся первой помеченной как посещенная, и ей присваивается значение не посещенная. Если таких вершин нет, то берется предыдущая помеченная вершина. d. Для последней помеченной как посещенная вершины выбирается смежная вершина, являющаяся первой помеченной как не посещенная, и ей присваивается значение посещенная. Если таких вершин нет, то берется предыдущая помеченная вершина.</p>
29.	<p>Определите какой шаг пропущен для поиска в ширину. 1. Всем вершинам графа присваивается значение не посещенная. Выбирается первая вершина и помечается как посещенная (и заносится в очередь). 2. ??? 3. Повторить шаг 2 до тех пор, пока</p>	<p>a. Выбирается следующая вершина и помечается как не посещенная. Все ее соседние вершины</p>



	<p>все вершины не будут помечены как посещенные. Варианты ответов:</p>	<p>заносятся в очередь. После этого она удаляется из очереди. б. Посещается первая вершина из очереди (если она не помечена как посещенная). Все ее соседние вершины заносятся в очередь. После этого она удаляется из очереди. с. Для последней помеченной как посещенная вершины выбирается смежная вершина, являющаяся первой помеченной как не посещенная, и ей присваивается значение посещенная. Если таких вершин нет, то берется предыдущая помеченная вершина.</p>
30.	<p>Какая из матриц инцидентности является правильной для графа, показанного на рисунке. рисунок)</p> 	<p>a. А b. Б c. В d. Г</p>



	<p>А.</p> <table border="1"> <thead> <tr><th></th><th>g1</th><th>g2</th><th>g3</th><th>g4</th><th>g5</th><th>g6</th><th>g7</th><th>g8</th><th>g9</th><th>g10</th><th>g11</th><th>g12</th></tr> </thead> <tbody> <tr><td>X1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>X2</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>X3</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>X4</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>X5</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>X6</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr> </tbody> </table> <p>Б.</p> <table border="1"> <thead> <tr><th></th><th>g1</th><th>g2</th><th>g3</th><th>g4</th><th>g5</th><th>g6</th><th>g7</th><th>g8</th><th>g9</th><th>g10</th><th>g11</th><th>g12</th></tr> </thead> <tbody> <tr><td>X1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>X2</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>X3</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>X4</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>X5</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>X6</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr> </tbody> </table> <p>В.</p> <table border="1"> <thead> <tr><th></th><th>g1</th><th>g2</th><th>g3</th><th>g4</th><th>g5</th><th>g6</th><th>g7</th><th>g8</th><th>g9</th><th>g10</th><th>g11</th><th>g12</th></tr> </thead> <tbody> <tr><td>X1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>X2</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>X3</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>X4</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>X5</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>X6</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </tbody> </table> <p>Г.</p> <table border="1"> <thead> <tr><th></th><th>g1</th><th>g2</th><th>g3</th><th>g4</th><th>g5</th><th>g6</th><th>g7</th><th>g8</th><th>g9</th><th>g10</th><th>g11</th><th>g12</th></tr> </thead> <tbody> <tr><td>X1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>X2</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>X3</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>X4</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>X5</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>X6</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>		g1	g2	g3	g4	g5	g6	g7	g8	g9	g10	g11	g12	X1	1	1	1	0	1	0	0	0	0	0	0	0	X2	1	0	1	0	0	1	1	1	1	0	0	0	X3	0	1	0	1	0	0	0	1	1	0	0	1	X4	0	0	1	0	0	0	1	0	1	0	0	0	X5	0	1	0	1	0	1	0	0	1	0	0	0	X6	0	0	0	0	1	0	0	0	0	0	1	0		g1	g2	g3	g4	g5	g6	g7	g8	g9	g10	g11	g12	X1	1	1	1	1	1	0	0	0	0	1	0	0	X2	1	0	0	0	0	1	1	1	0	0	1	0	X3	0	1	0	0	0	0	0	1	1	0	0	1	X4	0	0	1	0	0	0	1	0	1	0	0	0	X5	0	0	0	1	0	1	0	0	0	0	0	0	X6	0	0	0	0	1	0	0	0	0	1	0	0		g1	g2	g3	g4	g5	g6	g7	g8	g9	g10	g11	g12	X1	1	1	1	1	1	0	0	0	0	1	0	0	X2	1	0	0	0	0	1	1	1	0	0	1	0	X3	0	1	0	0	0	0	0	1	1	0	0	1	X4	0	0	1	0	0	0	1	0	1	0	0	1	X5	0	0	0	1	0	1	0	0	0	0	0	0	X6	0	0	0	0	1	0	0	0	0	0	0	0		g1	g2	g3	g4	g5	g6	g7	g8	g9	g10	g11	g12	X1	1	1	1	1	1	0	0	0	0	1	0	0	X2	1	0	0	0	0	1	1	1	0	0	1	0	X3	0	1	0	0	0	0	0	1	1	0	0	1	X4	0	0	1	0	0	0	1	0	1	0	0	1	X5	0	0	0	1	0	1	0	0	0	0	0	0	X6	0	0	0	0	1	0	0	0	0	0	0	0	
	g1	g2	g3	g4	g5	g6	g7	g8	g9	g10	g11	g12																																																																																																																																																																																																																																																																																																																																																																		
X1	1	1	1	0	1	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																		
X2	1	0	1	0	0	1	1	1	1	0	0	0																																																																																																																																																																																																																																																																																																																																																																		
X3	0	1	0	1	0	0	0	1	1	0	0	1																																																																																																																																																																																																																																																																																																																																																																		
X4	0	0	1	0	0	0	1	0	1	0	0	0																																																																																																																																																																																																																																																																																																																																																																		
X5	0	1	0	1	0	1	0	0	1	0	0	0																																																																																																																																																																																																																																																																																																																																																																		
X6	0	0	0	0	1	0	0	0	0	0	1	0																																																																																																																																																																																																																																																																																																																																																																		
	g1	g2	g3	g4	g5	g6	g7	g8	g9	g10	g11	g12																																																																																																																																																																																																																																																																																																																																																																		
X1	1	1	1	1	1	0	0	0	0	1	0	0																																																																																																																																																																																																																																																																																																																																																																		
X2	1	0	0	0	0	1	1	1	0	0	1	0																																																																																																																																																																																																																																																																																																																																																																		
X3	0	1	0	0	0	0	0	1	1	0	0	1																																																																																																																																																																																																																																																																																																																																																																		
X4	0	0	1	0	0	0	1	0	1	0	0	0																																																																																																																																																																																																																																																																																																																																																																		
X5	0	0	0	1	0	1	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																		
X6	0	0	0	0	1	0	0	0	0	1	0	0																																																																																																																																																																																																																																																																																																																																																																		
	g1	g2	g3	g4	g5	g6	g7	g8	g9	g10	g11	g12																																																																																																																																																																																																																																																																																																																																																																		
X1	1	1	1	1	1	0	0	0	0	1	0	0																																																																																																																																																																																																																																																																																																																																																																		
X2	1	0	0	0	0	1	1	1	0	0	1	0																																																																																																																																																																																																																																																																																																																																																																		
X3	0	1	0	0	0	0	0	1	1	0	0	1																																																																																																																																																																																																																																																																																																																																																																		
X4	0	0	1	0	0	0	1	0	1	0	0	1																																																																																																																																																																																																																																																																																																																																																																		
X5	0	0	0	1	0	1	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																		
X6	0	0	0	0	1	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																		
	g1	g2	g3	g4	g5	g6	g7	g8	g9	g10	g11	g12																																																																																																																																																																																																																																																																																																																																																																		
X1	1	1	1	1	1	0	0	0	0	1	0	0																																																																																																																																																																																																																																																																																																																																																																		
X2	1	0	0	0	0	1	1	1	0	0	1	0																																																																																																																																																																																																																																																																																																																																																																		
X3	0	1	0	0	0	0	0	1	1	0	0	1																																																																																																																																																																																																																																																																																																																																																																		
X4	0	0	1	0	0	0	1	0	1	0	0	1																																																																																																																																																																																																																																																																																																																																																																		
X5	0	0	0	1	0	1	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																		
X6	0	0	0	0	1	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																		
31.	Найти кратчайшие пути в орграфе от первой вершины ко всем остальным, используя алгоритм Дейкстры. Какое из деревьев кратчайших путей верно?	<p>а. (1,3), (3,6), (6,7), (7,8), (8,5), (5,4), (4,3)</p> <p>б. (1,2) (2,5), (5,4), (4,3), (3,6), (6,7), (7,8),</p> <p>с. (1,2), (2,5), (5,4), (4,3), (5,6), (6,7), (7,8)</p> <p>д. (1,2), (2,8), (8,5), (5,4), (4,3), (3,6), (6,7)</p>																																																																																																																																																																																																																																																																																																																																																																												
32.	Какими алгоритмами из перечисленных решается задача нахождения кратчайшего пути?	<p>а. Беллмана–Форда</p> <p>б. Дейкстры</p> <p>с. Форда-Фалкерсона</p> <p>д. Буковки</p> <p>е. Крускала</p> <p>ф. Прима</p>																																																																																																																																																																																																																																																																																																																																																																												
33.	Какой из перечисленных алгоритмов допускает работу с отрицательными весами рёбер при поиске кратчайшего пути.	<p>а. Беллмана–Форда</p> <p>б. Дейкстры</p> <p>с. Волновой алгоритм</p>																																																																																																																																																																																																																																																																																																																																																																												
34.	<p>Найдите максимальный поток в транспортной сети, используя алгоритм Форда–Фалкерсона. Источник – вершина 1, сток – вершина 8.</p> 	<p>а. 130</p> <p>б. 129</p> <p>с. 126</p> <p>д. 128</p>																																																																																																																																																																																																																																																																																																																																																																												
35.	Найдите какие вершины образуют минимальный разрез в транспортной сети, используя алгоритм Форда–Фалкерсона. Источник – вершина 1, сток – вершина 8	<p>а. [2,5,6,7,8]</p> <p>б. [3,5,6,4,2]</p> <p>с. [3,4,6,7]</p> <p>д. [4,5,6,7,8]</p>																																																																																																																																																																																																																																																																																																																																																																												

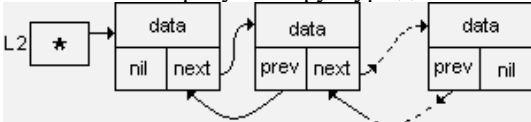


36.	<p>Найдите максимальный поток в транспортной сети, используя алгоритм Форда–Фалкерсона. Источник – вершина x_4, сток – вершина x_5.</p>	a. 70 b. 80 c. 62 d. 78
37.	<p>Найдите какие вершины образуют минимальный разрез в транспортной сети, используя алгоритм Форда–Фалкерсона. Источник – вершина x_4, сток – вершина x_5.</p>	a. $[x_3]$ b. $[x_1, x_5, x_2]$ c. $[x_1, x_4, x_3]$ d. $[x_5, x_3, x_2]$
38.	<p>Постройте остовное дерево минимального веса, используя алгоритм Прима или Краскала. Определите вес дерева.</p>	a. 13 b. 10 c. 9 d. 11
39.	<p>Неориентированное дерево это: (выбрать верные утверждения)</p>	а. связный граф, содержащий n вершин и $n-1$ ребер



		<p>в. связный граф, не имеющий циклов с. граф, в котором каждая пара вершин соединена несколькими простыми цепями d. граф без циклов, в котором полустепень захода каждой вершины, за исключением одной (начальной вершины x_1), равна единице.</p>
40.	В каком случае поток будет называться полным?	<p>a. Если хотя бы две дуги насыщены b. Если все дуги насыщены c. Если только одна дуга насыщена d. Если насыщена хотя бы одна дуга</p>
41.	Выберите наиболее полное и верное определение потока в транспортной сети.	<p>a. Неотрицательная вещественная функция, определенная на множестве дуг. b. Отрицательная вещественная функция, определенная на множестве дуг, удовлетворяющая условиям: ограниченности, сохранения. c. Неотрицательная вещественная функция, определенная на множестве дуг, удовлетворяющая условию сохранения. d. Неотрицательная вещественная функция, определенная на множестве дуг, удовлетворяющая условиям: ограниченности, сохранения.</p>
42.	На каком из перечисленных ниже алгоритмов основан волновой(переборный) алгоритм.	<p>a. Алгоритм поиска в глубину b. Алгоритм Дейкстры c. Алгоритм поиска в ширину d. Алгоритм Форда–Фалкерсона e. Алгоритм Прима</p>
43.	Что такое отрицательный цикл?	<p>a. Когда вес одного из ребёр в цикле отрицателен b. Цикл, сумма весов рёбер</p>



		которого отрицательна с. Цикл, в котором хотя бы один вес ребра имеет отрицательное значение d. Цикл, произведение весов ребёр которого отрицательно
44.	Каким условия перечисленным ниже должна удовлетворять транспортная сеть	a. нет петель b. существует хотя бы одна вершина, не имеющая ни одного прообраза c. существует хотя бы одна вершина, не имеющая ни одного образа d. существует только одна вершина, не имеющая ни одного образа
45.	Выберите правильные утверждения	a. длина списка постоянна и задается при его инициализации b. списки относятся к динамической структуре данных c. вся информация о списке хранится в динамической памяти d. в списке указатели могут ссылаться как на предыдущее, так и на последующее звенья списка
46.	Какие операции применимы к списку?	a. Проверка на пустоту b. Добавление\Удаление элементов c. Поиск конкретного элемента
47.	Показанная на рисунке структура данных является 	a. 2-связным линейным списком b. Стеком c. 2-связным кольцевым списком d. 1-связным линейным списком e. 1-связным кольцевым списком
48.	Из каких позиций списка можно удалять звенья (выделенного ведущего звена нет)?	a. Из любой позиции, кроме ведущего звена b. Из любой позиции, кроме последнего звена c. Из любой позиции d. Только из конца списка e. Только из ведущего звена



49.	Какие виды списков существуют?	a. линейный b. ненаправленный c. цилиндрический d. пирамидальный e. кольцевой
50.	Из каких позиций стека можно извлекать элементы?	a. Только со дна стека b. Из любой позиции, кроме дна стека c. Из любой позиции d. Только из вершины e. Из любой позиции, кроме вершины стека
51.	По какому принципу работает стек?	a. LIFO b. FIFO c. OIFO d. Как в очереди
52.	Какие операции из перечисленных являются стандартными для стека?	a. insert, pop b. push, delete c. push, pop d. put, extract e. add, delete
53.	В чем особенность структуры данных «стек»?	a. Открыт с обеих сторон на вставку и удаление b. Доступен любой элемент c. Открыт с одной стороны на вставку и удаление
54.	<p>Что останется в стеке после выполнения ряда операций? Push(S); Pop(); Push(A); Top(); Pop();</p> <p>Варианты ответов:</p> <p>1 2 </p> <p>3 4 </p>	a. 1 b. 2 c. 3 d. 4
55.	По какому принципу работает очередь?	a. OIFO b. FIFO c. LIFO d. Как в стеке

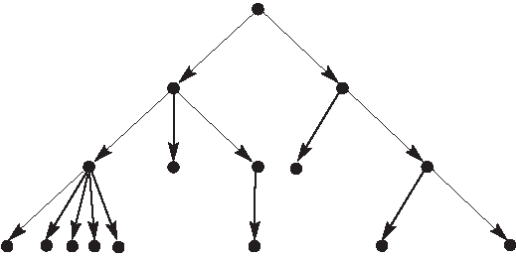


56.	В чем особенность структуры данных «очереди»?	a. Открыта с обеих сторон b. Открыта с одной стороны на вставку и удаление c. Доступен любой элемент
57.	Какие операции из перечисленных являются стандартными для очереди?	a. insert, pop b. push, delete c. push, pop d. enqueue, dequeue e. add, delete
58.	Сколько и какие указатели необходимы при инициализации очереди?	a. 1 – top b. 2 – head, tail c. 3 – head, top, tail d. 0 – ничего не нужно
59.	С какими проблемами можно столкнуться при реализации очереди с помощью списка?	a. «хвост» съедает «голову» b. выделение дополнительной памяти для каждого элемента c. переполнение очереди d. необходима связь каждого элемента очереди с последующим и предыдущим элементами
60.	Если последовательность в сортировке вставкой – связный список, то вставка приводит...	a. к переопределению указателей b. к сдвигу элементов в массиве c. к отсортировке в обратном порядке d. при сортировке вставкой невозможно использование связного списка
61.	Алгоритм, где на каждом проходе производится увеличение в два раза частично упорядоченных последовательностей, называется...	a. Сортировка слиянием b. Шейкерная сортировка c. Пузырьковая сортировка d. Сортировка выбором
62.	На каждом проходе выбирается минимальный элемент в сортировке, которая называется...	a. Шейкерная сортировка b. Пузырьковая сортировка c. Сортировка вставкой d. Сортировка выбором
63.	Бинарный поиск может применяться в алгоритме...	a. Шейкерная сортировка b. Пузырьковая сортировка c. Сортировка вставкой d. Сортировка выбором
64.	Шейкерная сортировка является модификацией...	a. Пузырьковой сортировки b. Сортировки вставкой c. Сортировки выбором



65.	Вычислительная сложность – это...	а. количественная характеристика алгоритма, которая определяется максимальным количеством памяти, необходимым для реализации алгоритма на вычислительной системе б. это количественная характеристика алгоритма, характеризующая количеством шагов и времени выполнения каждого шага с. максимальное количество шагов, которое производит алгоритм на всех наборах входных данных, соответствующих мере N
66.	сть $T(n) = \Omega(g(n))$, где Ω – это...	а. Верхняя оценка функции б. Порядок роста функции с. Нижняя оценка функции
67.	Алгоритмы какого класса имеют бо`льшую скорость роста?	а. Экспоненциальный б. Логарифмический с. Полиномиальный d. Линейный логарифм
68.	Алгоритм перебора элементов массива относится к классу сложности...	а. Логарифмический б. Полиномиальный с. Линейный логарифм d. Линейный
69.	Время выполнения операторов присваивания, чтения, сравнения считают равным...	а. $O(n)$ б. $O(c)$ с. $O(1)$ d. $c \cdot O(1)$
70.	Что может привести к переполнению стека при использовании быстрой сортировки?	а. Рекурсивный вызов функции б. Выбор эталонного элемента с. Перераспределение элементов на группы d. Переполнение стека невозможно при быстрой сортировке
71.	В каком алгоритме происходит увеличение упорядоченной серии чисел ровно в 2 раза?	а. Алгоритм сортировки слиянием б. Алгоритм пирамидальной сортировки с. Алгоритм быстрой сортировки



72.	У вас имеется сортирующее дерево, которое хранится в массиве. Какое число может быть в этой последовательности вместо звездочек: 25, 14, 24, ***, 6, 21,22, 3,4,5?	a. 23 b. 16 c. 9 d. 2
73.	Алгоритм какой сортировки может быть разделен на две фазы?	a. Алгоритм сортировки слиянием b. Алгоритм пирамидальной сортировки c. Алгоритм быстрой сортировки
74.	В алгоритме быстрой сортировки количество сравнений во время каждого прохода равно n в...	a. лучшем случае b. худшем случае c. произвольном случае d. в любом случае
75.	Листом дерева называется....	a. Верхняя вершина b. Вершина, имеющая ровно 1 потомка c. Вершина, не имеющая потомков d. Произвольная вершина
76.	 <p>На рисунке представлено k-позиционное дерево, где $k=...$</p>	a. 2 b. 3 c. 4 d. 5
77.	Если корневое k -позиционное дерево хранить в массиве, то для вершины с индексом s потомками будут вершины с индексами	a. $k, k+1, \dots, s$ b. $k^s, k^{s+1}, \dots, k^{s+k}$ c. $ks+1, ks+2, \dots, ks+k$ d. $k+s, k+s+1, \dots, k+s+s$
78.	Если в коде Прюффера N чисел, то количество вершин в исходном дереве...	a. N b. $N+1$ c. $N+2$ d. $N-1$ e. $N-2$
79.	Недостатками реализации корневого дерева с помощью списков являются...	a. Простой памяти b. Нехватка памяти c. Дополнительная память для указателей d. Быстрая скорость прохода по дереву e. Гибкость структуры
80.	Алгоритм карманной сортировки эффективен...	a. если в массиве большое количество одинаковых



		чисел b. если все числа в массиве различны c. если в массиве содержатся указатели на внешние структуры d. если массив формируется динамически
81.	Во время работы алгоритма карманной сортировки заводится столько же вспомогательных стеков, сколько и...	a. максимальное количество разрядов b. количество сортируемых чисел c. основание системы счисления чисел
82.	Пусть имеются для карманной сортировки данные с параметрами $N=1000$, $m=5$, $k=7$. Сколько итераций (проходов) будет совершено во время работы алгоритма?	a. 5 b. 7 c. 12 d. 1000
83.	Пусть имеются данные с параметрами $N=1000$, $m=100$, $k=10$. Сколько в рамках одной итерации будет совершено операций переключивания из основного стека во вспомогательные?	a. 100 b. 1000 c. 10000 d. 100000
84.	Карманная сортировка также называется...	a. почисленной сортировкой b. сортировкой дубликатов c. поразрядной сортировкой d. тубусной сортировкой
85.	Внешняя сортировка...	a. применяется к данным, которые расположены на внешних носителях и целиком не помещаются в оперативную память b. должна быть построена на принципе произвольного доступа c. на практике эффективнее сортировки слиянием
86.	Алгоритмы внутренней сортировки работают с данными, которые	a. располагаются во внешних базах данных b. полностью помещаются в оперативную память c. характеризуются последовательным доступом к элементам
87.	Последовательность элементов, упорядоченную по ключу называют...	a. фазой b. окном c. серией d. пирамидой



88.	Алгоритмом внешней сортировки является	a. алгоритм Хоара b. шейкерный алгоритм c. алгоритм естественный слиянием d. алгоритм сортировки выбором
89.	Алгоритмы внешней сортировки отличает ... доступ к элементам последовательности:	a. произвольный b. смешанный c. последовательный d. систематический
90.	В памяти компьютера двоичное дерево удобно представлять в виде:	a. Связанных линейных списков b. Массивов c. Связанных нелинейных списков
91.	Имеется двоичное дерево (не являющееся деревом поиска), содержащее целые числа. Восходящий просмотр дерева даёт следующий результат: 2, 4, 6, 8, 10, 12, 14. Какой узел является корнем дерева?	a. 10 b. 14 c. 8 d. 6 e. 2
92.	Имеется идеально сбалансированное двоичное дерево поиска, содержащее целые числа. Просмотр дерева даёт следующий результат: 2, 4, 6, 8, 10, 12, 14. Какой способ просмотра дерева использовался?	a. Восходящий b. Поуровневый c. Последовательный d. Поузловой e. Нисходящий
93.	Какие из операций над двоичными деревьями поиска наиболее затратны по памяти компьютера? (оценивается сложность алгоритма)	a. FindKey(key) b. Add(key) c. Delete(key) d. Все операции имеют одинаковую сложность
94.	Вершина с максимальным значением ключа является...	a. Корнем дерева b. Самой левой вершиной дерева c. Самой правой вершиной дерева d. Самой правой вершиной в левом поддереве корня
95.	В таблице с прямой адресацией её размер совпадает с...	a. размером всего множества ключей b. размером множества реальных ключей c. количеством ненулевых указателей
96.	Время выполнения операция в таблице прямой адресации:	a. для операции поиска и вставки $O(1)$, для удаления – $O(\log N)$ b. для всех операций - $O(1)$ c. для всех операций -



		$O(\log N)$ d. для операции поиска $O(1)$ для удаления и вставки - $O(\log N)$
97.	Главная цель хэш-функции:	a. уменьшение рабочего диапазона индексов массива b. увеличение рабочего диапазона индексов массива c. минимизировать количество коллизий d. хэшировать элементы по ячейкам равномерно и независимо
98.	Размер хэш-таблицы совпадает с...	a. размером всего множества ключей b. количеством коллизий c. размером множества реальных ключей d. количеством ненулевых указателей
99.	Пусть имеется ключ $k_1 = 26$ и хэш-функция для хэш-таблицы вычисляется по формуле $h(x) = x \bmod 11$. При каком минимальном значении ключа k_2 возникнет коллизия на ключах k_1 и k_2 ?	a. 4
100.	В красно-черном дереве фиктивные потомки могут быть следующих цветов...	a. Только красного цвета b. Только черного цвета c. Красного и черного цветов d. Фиктивные потомки не имеют цвета
101.	В красно-черном дереве должно выполняться	a. Равенство «красных» высоты b. Равенство «черных» высот c. «Красная» и «черная» высота могут различаться не более чем на 1 d. «Красная» и «черная» могут различаться не более чем в 2 раза
102.	Дано красно-черное дерево:	a. 1 b. 2 c. 3 d. 4



	<p>Какой примет вид дерево после добавления</p> <p>1 2 3 4 </p> <p>50?</p>	
103.	Какие операции в КЧ-дереве могут привести к его перестройке?	a. Min() b. Add() c. FindNext() d. FindKey() e. Delete()
104.	Операция «правого/левого вращения» служит для ...	a. Нахождения самого правого/левого узла b. Выравнивания высот поддеревьев c. Перекрашивания вершин d. Удаления вершины
105.	Средняя длина цепочки в хеш-таблице размера m , в которую добавлено n элементов будет составлять:	a. 1 b. n/m



		c. m d. m/n
106.	Все операции в хэш-таблице с цепочками в среднем будут выполняться за время:	a. O(1) b. O(m) c. O(log m) d. у операций различное время выполнения
107.	Универсальным хэшированием называется такой подход, когда...	a. хэш-функция выбирается случайно для каждой хэш-таблицы и не должна зависеть от набора ключей, с которыми работает b. хэш-функция выбирается для каждой хэш-таблицы в зависимости от набора ключей, с которыми работает c. помещаемые в хэш-таблицу данные являются зависимыми d. помещаемые в хэш-таблицу данные являются независимыми
108.	Какой метод вычисления хэш-значений по ключу существует?	a. метод вычитания b. метод взятия остатка c. метод умножения d. метод деления
109.	Пусть имеется хэш-таблица размерностью на 1000 элементов. В нее добавлено 5466 элементов. Какая возможна минимальная длина цепочки элементов, ассоциированной с некоторой ячейкой хэш-таблицы?	a. 0
110.	Какой алгоритм сортировки является фактически худшим?	a. Шейкерная сортировка b. Пузырьковая сортировка c. Быстрая сортировка d. Сортировка выбором
111.	Некоторый массив размером N был отсортирован за время, пропорциональное $N \cdot \log_2 N$. По какому алгоритму выполнялась сортировка?	a. Перестановками b. Быстрая сортировка c. Шелла d. Выбором e. Вставками
112.	Производится пузырьковая сортировка (всплывает «легкий» элемент) массива из 5 элементов, причём массив упорядочен в порядке возрастания. Сколько будет выполнено перестановок?	a. 5 b. 28 c. 6 d. 15 e. 10
113.	В процессе сортировки весь сортируемый массив и каждая его часть делятся на две части. По какому алгоритму выполняется эта сортировка?	a. Пузырьковая b. Вставками c. Шелла

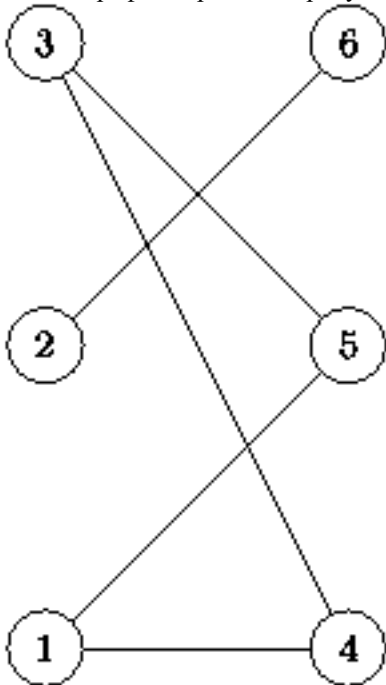


		d. Быстрая e. Выбором
114.	Имеется неупорядоченный массив целых чисел из 8 элементов. Сколько операций сравнения потребуется для нахождения искомого ключа, если он находится в конце массива?	a. 7 b. 0 c. 1 d. 4 e. 8
115.	Увеличение объема хэш-таблицы в методе открытой адресации достигается...	a. отказом от указателей b. использованием указателей c. использованием внешней памяти d. неограниченным размером таблицы
116.	При методе открытой адресации коэффициент заполнения будет:	a. <1 b. ≤ 1 c. $=1$ d. ≥ 1 e. >1
117.	Проверку хэш-таблицы на возможность вставки объекта с ключом k называют:	a. фазой b. итерацией c. серией d. исследованием
118.	Вторичная кластеризация характерна для...	a. линейного исследования b. квадратичного исследования c. двойного хеширования.
119.	Пусть имеется хэш-таблица, элементы в которую добавляются согласно методу линейного исследования ($m=15$). В какую ячейку хэш-таблицы (при условии, что она свободная) добавится элемент с ключом $k=63$ при третьей попытке записи в таблицу?	a. 5
120.	Назовите два основных варианта хэш-таблицы:	a. С цепочками b. Кольцевая c. С двоичным деревом d. С открытой адресацией
121.	При каком методе разрешения коллизий хэш-таблица не может оказаться заполненной, делая невозможной вставку новых элементов?	a. открытая адресация b. метод внешних цепочек c. оба вышеперечисленных
122.	Параметр α хэш-таблицы называется...	a. Коэффициентом пустоты b. Коэффициентом заполнения c. Мощностью множества реальных ключей d. Параметром функции хеширования
123.	Какие существуют методы вычисления хэш-значений?	a. Метод ветвления b. Метод умножения



		<p>c. Метод сложения d. Метод отделения e. Метод деления</p>
124.	Какая основная проблема имеется при применении линейного исследования в методе открытой адресации?	<p>a. Первичная кластеризация b. Вторичная кластеризация c. Большая вычислительная сложность d. Требуется много памяти для хранения указателей</p>
125.	В графе через V обозначают:	<p>a. множество ребер b. множество вершин c. множество весов d. подмножество уникальных дуг</p>
126.	Ориентированной цепью называется такой путь, в котором...	<p>a. каждая дуга используется не больше одного раза b. каждая дуга используется не менее одного раза c. каждая дуга используется строго один раз d. каждая дуга является начальной вершиной следующей дуги</p>
127.	Взвешенный граф – это...	<p>a. граф, каждому ребру которого поставлено в соответствие некое значение b. граф, каждой вершине которого поставлено в соответствие некое значение c. граф, каждая вершина которого содержит одинаковое количество дуг d. граф, каждая вершина которого содержит одинаковое значение</p>
128.	Максимальная глубина обхода графа с помощью алгоритма поиска в глубину:	<p>a. всегда меньше либо равна высоте графа b. может быть больше высоты графа c. не превосходит значения $N \cdot \log N$, где N – это количество вершин d. не превосходит значения N, где N – это количество вершин</p>
129.	Алгоритм обхода с помощью поиска в ширину использует динамическую структуру данных:	<p>a. стек b. очередь</p>



		с. дек d. бинарное дерево
130.	Какой граф изображён на рисунке? 	a. Смешанный граф b. Неориентированный граф с. Ориентированный граф d. Граф с петлёй e. Графом со взвешенными дугами
131.	Какой из перечисленных алгоритмов допускает работу с отрицательными весами рёбер при поиске кратчайшего пути.	a. Беллмана–Форда b. Дейкстры с. Волновой алгоритм
132.	Какой контур будет называть гамильтоновым	a. Контур, проходящий через все вершины графа b. Контур, проходящий хотя бы через одну вершину графа с. Контур, проходящий через все вершины графа, кроме тех, где есть петли. d. Контур, проходящий только через три вершины графа
133.	Что такое остовное дерево, выбрать верное утверждение.	a. Граф, который является ориентированным деревом с корнем в вершине x; b. Остовный подграф графа, являющийся деревом; с. Остовной граф, являющийся деревом; d. Подграф ориентированного или неориентированного графа.
134.	Основой транспортной сети является граф:	a. неориентированный, невзвешенный



		<p>b. неориентированный, взвешенный c. ориентированный, невзвешенный d. ориентированный, взвешенный</p>
135.	Для потока, протекающего по любому ребру в графе верна формула:	<p>a. $f(u,v) < c(u,v)$ b. $f(u,v) \leq c(u,v)$ c. $f(u,v) = c(u,v)$ d. $f(u,v) \geq c(u,v)$ e. $f(u,v) > c(u,v)$</p>
136.	Теорема Форда-Фалкерсона гласит: в любой сети величина максимального потока равна пропускной способности...	<p>a. вершины стока b. максимального разреза c. минимального разреза d. ребру истока</p>
137.	В транспортной сети параметр c называют...	<p>a. пропускной способностью ребра b. величиной максимального потока ребра c. показателем минимального потока ребра d. величиной разреза</p>
138.	Разрез транспортной сети бывает...	<p>a. минимальным b. максимальным c. оптимальным d. совершенным</p>
139.	Выберите правильные утверждения	<p>a. адрес элемента динамической структуры может быть вычислен из адресов соседних элементов b. при изменении логической последовательности элементов динамической структуры (удалении, добавлении элемента) требуется только коррекция указателей c. размер динамической структуры данных не зависит от числа элементов, он фиксирован и задается при инициализации</p>
140.	Выберите правильные утверждения	<p>a. размер динамической структуры данных ограничивается только доступным объемом памяти компьютера b. адрес элемента динамической структуры</p>



		может быть вычислен из адресов соседних элементов с. при изменении логической последовательности элементов динамической структуры (удалении, добавлении элемента) требуется только коррекция указателей
141.	Какие из перечисленных структур данных являются динамическими?	a. Массивы b. Графы c. Множества d. Списки e. Очередь
142.	Выберите правильные утверждения	a. списки относятся к статической структуре данных b. длина списка постоянна и задается при его инициализации c. в списке указатели могут ссылаться как на предыдущее, так и на последующее звенья списка d. вся информация о списке хранится в динамической памяти
143.	Линейный список, в котором доступен только последний элемент, называется	a. Стеком b. Очередью c. Последовательностью d. Массивом e. Деревом
144.	В связном представлении разреженной матрицы, голова списка столбцов хранит	a. Указатель на первый узел списка столбцов b. Указатель на голову следующего столбца c. Указатель на последний узел списка столбцов d. Номер столбца e. Всю перечисленную информацию
145.	data - циклический массив из N элементов и last - индекс в этом массиве, какая формула индекса следующего после last элемента?	a. $(last \% 1) + N$ b. $(last + 1) \% N$ c. $last \% (1 + N)$ d. $last + (1 \% N)$
146.	Какая структура данных позволяет производить вставку и удаление элемента с обоих концов?	a. Дек b. Очередь c. Стек

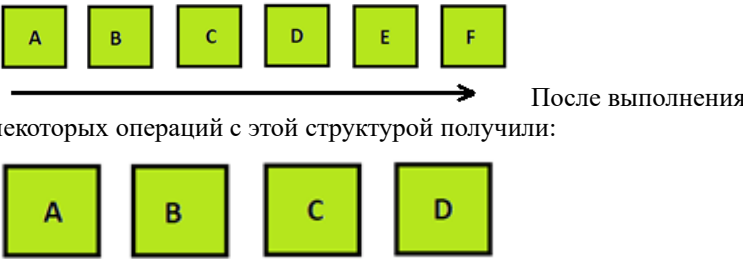
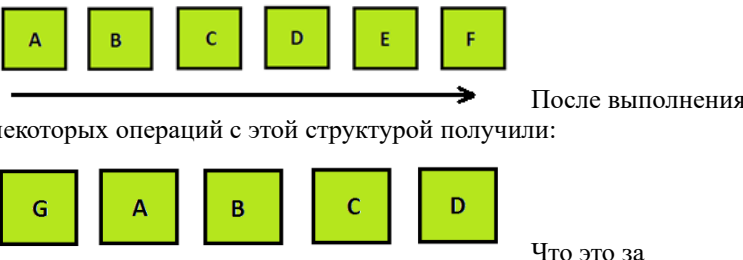
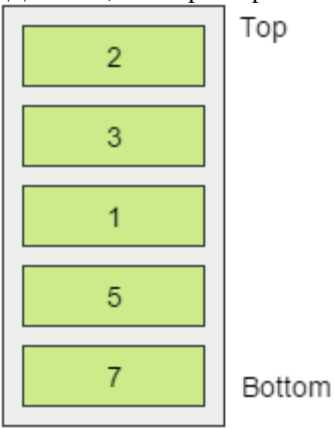


147.	С какими проблемами можно столкнуться при реализации очереди с помощью циклического массива?	a. выделение дополнительной памяти под указатели b. «хвост» съедает «голову» c. простой памяти в случае слабого заполнения очереди d. операция «поиск элемента» выполняется долго
148.	Какие из перечисленных операций менее затратны (по времени) для реализации списка через массив относительно реализации через динамически создаваемый связный список?	a. Обход b. Удаление c. Поиск d. Вставка
149.	Дан список $a = ['b', 'c', 'd', 'e']$. Что выведет программа в результате следующих операций? $b = a[1]$ $a.delete(0)$ $a.delete(1)$ $a[1] = b$ $print(a)$	a. ['c', 'e'] b. ['c', 'b', 'd', 'e'] c. ['d', 'c', 'e'] d. ['d', 'c'] e. ['c', 'c'] f. Ошибку
150.	Дан список $a = ['c', 's', 'u', '7', '4']$. Что выведет программа в результате следующих операций? $x = a[2]$ $a.delete(2)$ $a[3] = x$ $a.delete(2)$	a. ['7', '4'] b. ['c', 's', '7', '4'] c. ['c', 's', 'u'] d. ['c', 's', '4'] e. ['c', 's', '7'] f. Ошибку
151.	Просмотр списка, содержащего символы, даёт следующий результат: 6, 5, 4, 3, 2, 1. Известно, что данные заносились в начало списка. Как они были упорядочены перед вводом в список?	a. Данные были упорядочены случайно b. В шахматном порядке c. Данные не были упорядочены d. По возрастанию e. По убыванию
152.	Какие основные операции над элементами характерны для списков?	a. Занесение нового элемента в список, удаление элемента из списка, просмотр списка, поиск элемента в списке, сортировка списка. b. Сортировка элементов списка, занесение элемента в список, извлечение элемента из списка и удаление списка. c. Создание ведущего звена, вставка нового звена, удаление звена, поиск. d. Просмотр списка, поиск элемента в списке и

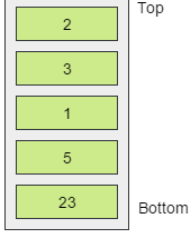
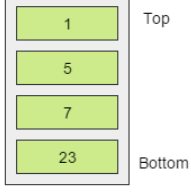


		сортировка списка. е. Занесение нового элемента в список и извлечение элемента из списка.
153.	Из каких позиций очереди можно извлекать элементы?	a. Из любой позиции, кроме конца очереди b. Только из конца очереди c. Только из начала очереди d. Из любой позиции e. Только из начала или конца очереди
154.	Дан список $a = ['x', 'r', 'i', 'k', '3', 't']$. Что выведет программа в результате следующих операций? $b = a[0]$ $a.delete(1)$ $b.append(a[1])$ $a.delete(1)$ $a[1] = b[3]$ $print(a)$	a. $['r', 'k', '3', 't']$ b. $['x', 'i', '3', 'k', '3', 't']$ c. $['x', 'i', 'i', 'k', '3', 't']$ d. $['r', 'i', '3', 't']$ e. $['x', 'x']$ f. ошибку
155.	В какие позиции стека можно добавлять элементы?	a. Только в дно стека b. Из любой позиции, кроме дна стека c. В любую позицию d. Только в вершину e. Из любой позиции, кроме вершины стека
156.	Какая структура данных соответствует принципу: FIFO?	a. Очередь b. Стек c. В-дерево d. Односвязные циклические списки
157.	Если символы '1', '2', '3', '4' помещены в очередь по порядку и затем будут по одному удалены, в каком порядке это произойдет?	a. 4321 b. 1234 c. 1243 d. 4231
158.	Какая структура данных соответствует принципу: LIFO?	a. Очередь b. Стек c. В-дерево d. Односвязные циклические списки
159.	В какой структуре данных вставка и удаление происходит на одном конце?	a. Массив b. Стек c. Очередь d. Дек e. Дерево f. Связный список
160.	Какие операции над элементами характерны для очередей и стеков?	a. Занесение элемента и извлечение элемента



		<p>b. Занесение элемента, извлечение элемента и очистка c. Занесение элемента, извлечение элемента и просмотр d. Занесение элемента, извлечение элемента, просмотр, сортировка и удаление текущего элемента e. Поиск элемента и сортировка</p>
161.	<p>Известны элементы некоторой структуры данных:</p>  <p>После выполнения некоторых операций с этой структурой получили:</p> <p>Что это за структура данных?</p>	<p>a. Стек b. Очередь c. Дерево d. Такой структуры не существует e. FIFO</p>
162.	<p>Известны элементы некоторой структуры данных:</p>  <p>После выполнения некоторых операций с этой структурой получили:</p> <p>Что это за структура данных?</p>	<p>a. Стек b. Очередь c. Дерево d. Такой структуры не существует e. LIFO</p>
163.	<p>Дан стек, в котором хранятся следующие значения:</p>  <p>Что останется в стеке после выполнения ряда операций? Pop(); Push(23); Push(3); Top();</p>	<p>a. 1 b. 2 c. 3 d. 4</p>

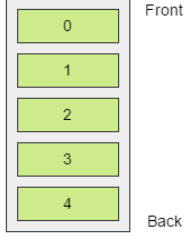
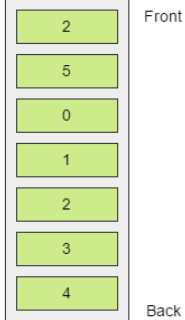
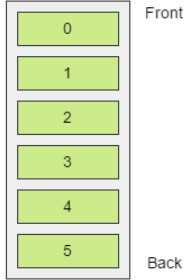
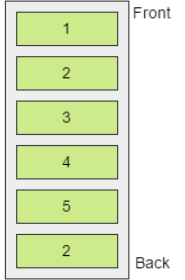
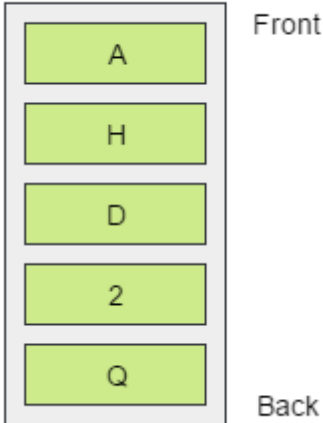


	<p>1</p>  <p>2</p>  <p>3</p>  <p>4</p>  <p>Pop();</p>	
164.	<p>Дан стек, в котором хранятся следующие значения:</p>  <p>Что останется в стеке после выполнения ряда операций? if IsEmpty(): Push(5); Push(2); Top();</p>	<p>a. 1 b. 2 c. 3 d. 4</p>

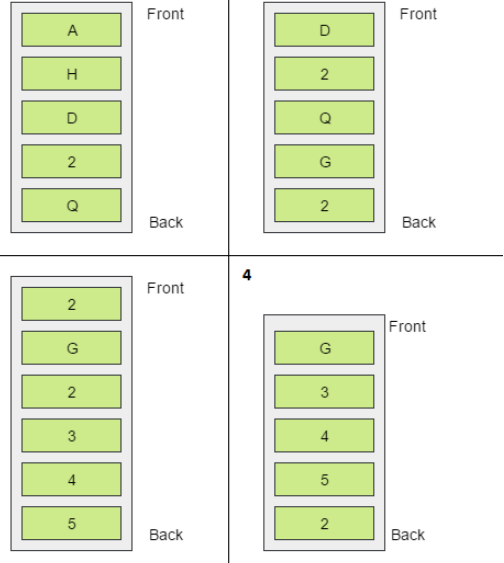
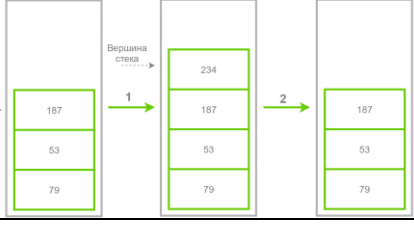
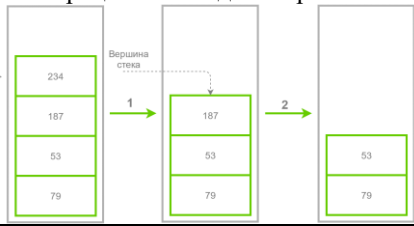



	<p>1</p> <p>2</p> <p>3</p> <p>4</p>	
165.	<p>Дана очередь, в которой в определенный момент времени хранится следующая информация:</p> <p>Что останется в очереди после выполнения ряда операций? Push(5); Push(2); Top(); Pop(); if</p>	<p>a. 1 b. 2 c. 3 d. 4</p>



	<p>1</p>  <p>2</p>  <p>3</p>  <p>4</p>  <p>IsEmpty(): Pop();</p>	
166.	<p>Дана очередь, в которой в определенный момент времени хранится следующая информация:</p>  <p>Что останется в очереди после выполнения ряда операций? Pop(); Push(G); Push(2); Pop();</p>	a. 1 b. 2 c. 3 d. 4



		
167.	Какие операции необходимо произвести над стеком? 	a. insert, pop b. push, pop c. put, extract d. add, delete e. push, delete f. нет правильных вариантов
168.	Какие операции необходимо произвести над стеком? 	a. insert, pop b. pop, pop c. extract, extract d. add, delete e. delete, delete f. push, push g. нет правильных вариантов
169.	Какие операции необходимо произвести над стеком? 	a. insert, pop b. pop, pop c. extract, extract d. add, delete e. delete, delete f. push, push g. нет правильных вариантов
170.	Пусть имеется исходный граф с n вершинами и m ребрами, тогда в остовном графе вершин будет:	a. n b. m c. $n-1$ d. $m-1$
171.	Пусть имеется исходный граф с n вершинами и m ребрами, тогда в остовном графе ребер будет:	a. n b. m c. $n-1$ d. $m-1$
172.	Какой операции нет в алгоритме Краскала:	a. Операции выбора ребра



		минимального веса b. Операции проверки на появление цикла c. Операции объединения множества вершин d. Операции разбиения множества вершин
173.	В алгоритме Буровка производится итераций:	a. $ E $ b. $ V $ c. $\text{Log} E $ d. $\text{Log} V$ e. $ E \cdot \log V $
174.	В алгоритме Прима к исходной вершине добавляем ребра...	a. нулевого веса b. минимального веса c. «безопасные» d. исходящие
175.	Алгоритм Дейкстры находит кратчайшее расстояние:	a. от одной вершины графа до другой b. от одной вершины до всех остальных c. от всех вершин до всех других
176.	Алгоритм Форда-Беллмана работает со взвешенным графом, у которого все веса ребер:	a. положительные b. неотрицательные c. отрицательные d. ненулевые e. любые
177.	Волновой алгоритм основан на...	a. поиске корня дерева b. поиске в ширину c. поиске в глубину d. поиске смысла жизни
178.	Чтобы свести задачу поиска кратчайшего расстояния до задачи поиска минимального количества ребер между вершинами, необходимо полагать веса ребер:	a. отличными от нуля b. равными нулю c. равными единице d. равными любым неотрицательным числам
179.	Какой алгоритм поиска кратчайшего расстояния существует:	a. A* b. B* c. C* d. D*
180.	Возможно ли отсортировать произвольный массив за время $O(N)$?	a. Да. Массив любого размера и любого типа. b. Нет. Это невозможно ни для какого массива. c. Это возможно для массивов определённых размеров. d. Это возможно для



		массивов некоторых типов.
181.	Каким выражением характеризуется эффективность поиска в двоичном дереве поиска?	a. $O(N/2)$ b. $O(1)$ c. $O(N)$ d. $O(N^2)$ e. $O(\log 2N)$
182.	Некоторый массив размером N был отсортирован за время, пропорциональное N^2 . По какому алгоритму выполнялась сортировка?	a. Пузырьковая b. Ветвлением c. Быстрой d. Замыканием
183.	Каким выражением определяется количество перестановок для пузырьковой сортировки в худшем случае?	a. $N(N-1)/4$ b. 0 c. $N-1$ d. N^2 e. $N(N-1)/2$
184.	Каким выражением определяется количество перестановок для пузырьковой сортировки в лучшем случае?	a. 0 b. $N-1$ c. N^2 d. $N(N-1)/2$ e. $N(N-1)/4$
185.	Какое из следующих высказываний наилучшим образом характеризует быструю сортировку?	a. Относится к самым быстрым в среднем случае из универсальных алгоритмов b. Считается самой простой c. Не подходит для 1-мерных массивов d. Ищет наименьший или наибольший элемент e. Выполняет наименьшее число операций
186.	Какова сложность алгоритма "Быстрая сортировка" в худшем случае.	a. $O(n \cdot \log(n))$ b. $O(2 \cdot n \cdot \log(n))$ c. $O(n \cdot \log(n^2))$ d. $O(n^2)$
187.	Какой поиск эффективнее?	a. линейный b. бинарный c. одинаково
188.	Пусть $T(n) = O(g(n))$, где O – это...	a. Верхняя оценка функции b. Порядок роста функции c. Нижняя оценка функции
189.	Пусть $T(n) = \Theta(g(n))$, где Θ – это...	a. Верхняя оценка функции b. Порядок роста функции c. Нижняя оценка функции



190.	Укажите два наилучших алгоритма по критерию трудоемкости	a. алгоритм с логарифмической скоростью роста b. алгоритм с линейной скоростью роста c. алгоритм с линейно-логарифмической скоростью роста d. алгоритм с квадратичной скоростью роста
191.	Задан массив $X[1..N]$. Определите временную сложность алгоритма: <code>function R(N: integer): integer; begin R:=0; if N > 3 then R:=R+R(N-1)+2*Rec(N-2); end;</code>	a. $O(\log N)$ b. $O(N)$ c. $O(N^2)$ d. $O(N^3)$ e. $O(2^N)$
192.	Задан массив $X[1..N]$. Определите временную сложность алгоритма: <code>for i:=1 to N-1 do for j:=N-1 downto i do if A[j]>A[j+1] then Swap(A[j], A[j+1]);</code>	a. $O(\log N)$ b. $O(N)$ c. $O(N^2)$ d. $O(N^3)$ e. $O(2^N)$
193.	Какое максимальное расстояние между двумя узлами в идеально сбалансированном двоичном дереве из n элементов?	a. около $n/2$ b. около $4 \cdot \log_2 n$ c. около $2 \cdot \log_2 n$ d. около n e. около $\log_2 n$
194.	Какое положение будет верно относительно B-дерева (B-Tree)?	a. Все записи узла больше чем или равняются записям узлов-потомков b. Все узлы имеют одинаковое количество записей c. Все листья находятся на нижнем уровне d. Все нелистовые узлы имеют одинаковое число потомков
195.	Какая из указанных структур данных имеет сбалансированное состояние?	a. Стек b. Двусвязный список c. Двоичное дерево поиска d. Дек (двусторонняя очередь)
196.	Имеется двоичное дерево (не являющееся деревом поиска), содержащее целые числа. Нисходящий просмотр дерева даёт следующий результат: 2, 4, 6, 8, 10, 12, 14. Какой узел является корнем дерева?	a. 14 b. 10 c. 6 d. 8 e. 2
197.	Имеется двоичное дерево (не являющееся деревом поиска), содержащее произвольные символы. Восходящий просмотр дерева даёт следующий результат: B, +, #, 3, f, k. Какой узел	a. B b. + c. 3

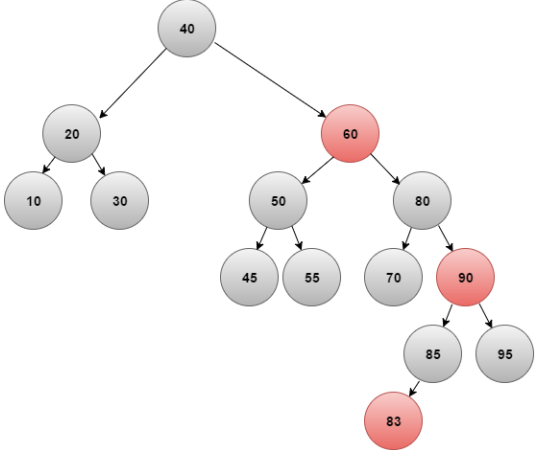


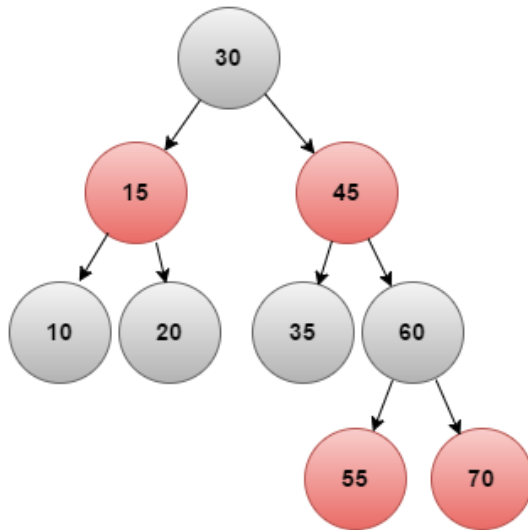
	является корнем дерева?	d. k e. #
198.	Имеется идеально сбалансированное двоичное дерево, содержащее 31 узел. Какова высота этого дерева?	a. 2 уровня b. 5 уровней c. 4 уровня d. 1 уровень e. 31 уровень f. 3 уровня
199.	Имеется идеально сбалансированное двоичное дерево, узлы которого размещены на 6-и уровнях. Какое максимальное число узлов может быть в этом дереве?	a. 127 b. 7 c. 63 d. 64 e. 6
200.	Имеется идеально сбалансированное двоичное дерево, содержащее 64 узла. Какова высота этого дерева?	a. 5 уровней b. 3 уровня c. 64 уровня d. 7 уровней e. 6 уровней
201.	Имеется идеально сбалансированное двоичное дерево, содержащее 8 узлов. Какова высота этого дерева?	a. 3 уровня b. 5 уровней c. 4 уровня d. 8 уровней e. Такого дерева не существует
202.	Имеется идеально сбалансированное двоичное дерево, узлы которого размещены на 5-и уровнях. Какое максимальное число узлов может быть в этом дереве?	a. 49 b. 63 c. 31 d. 56 e. 32
203.	Имеется идеально сбалансированное двоичное дерево, узлы которого размещены на 7-и уровнях. Выберите ответы, в которых число узлов может быть в этом дереве?	a. 127 b. 63 c. 7 d. 512 e. 64 f. 128
204.	Имеется двоичное дерево поиска, содержащее целые числа от 1 до 7. Результаты последовательного и нисходящего просмотров дерева оказались одинаковыми. В каком состоянии находится дерево?	a. Байера b. Вырожденное c. Сбалансированное d. Мультивариантное e. Красно-чёрное
205.	К каким структурам данных в общем случае относится дерево?	a. К динамическим нелинейным b. К статическим нелинейным c. К статическим линейным d. К динамическим линейным e. К кольцевым



206.	Элемент дерева j , на который нет ссылок называется	a. Корнем b. Промежуточным элементом c. Терминальным (лист)
207.	Дерево называется полным бинарным, если степень исходов вершин равна:	a. 2 или 0 b. 2 c. N или 0 d. N
208.	При обходе дерева слева направо получаем последовательность...	a. Отсортированную по убыванию b. Неотсортированную c. Отсортированную по возрастанию
209.	Элемент дерева, который не ссылается на другие элементы, называется:	a. Корнем b. Узлом c. Листом d. Промежуточным
210.	Элемент дерева, на который не ссылаются другие элементы, называется:	a. Корнем b. Листом c. Промежуточным d. Узлом
211.	Элемент дерева, который имеет предка и потомков, называется:	a. Корнем b. Листом c. Узлом d. Промежуточным
212.	Что такое «высота дерева»?	a. Максимальное количество узлов b. Максимальное количество связей c. Максимальное количество листьев d. Максимальная длина пути от корня до листа
213.	Степенью дерева называется:	a. Максимальная степень всех узлов b. Максимальное количество уровней его узлов c. Максимальное количество узлов d. Максимальное количество связей e. Максимальное количество листьев
214.	Как определяется длина пути дерева?	a. Как сумма длин путей всех его узлов b. Как количество ребер от узла до вершины

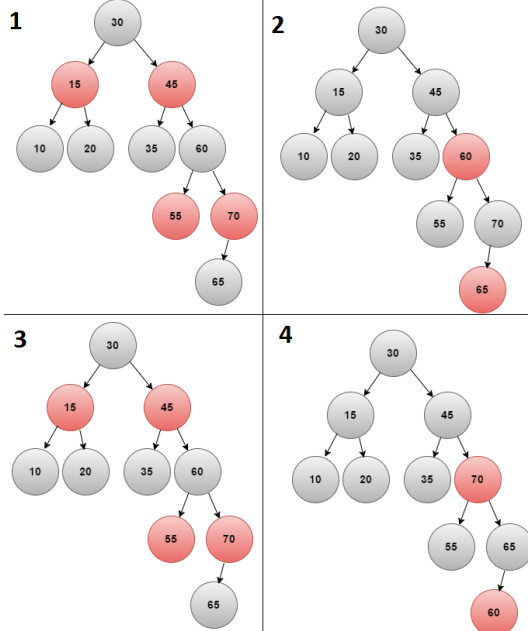


		<p>с. Как количество ребер от листа до вершины d. Как максимальное количество листьев е. Как максимальное количество ребер f. Как длина самого длинного пути от ближнего узла до какого-либо листа</p>
215.	Дерево называется бинарным, если:	<p>a. Каждый узел имеет не менее двух предков b. От корня до листа не более двух уровней с. Количество узлов может быть либо пустым, либо состоять из корня с двумя другими бинарными поддеревьями d. От корня до листа не менее двух уровней</p>
216.	Бинарное дерево можно представить с помощью...	<p>a. Массивов b. Указателей с. Индексов d. Хеш-таблиц е. Правильного ответа нет</p>
217.	Какая максимальная высота данного красно-черного дерева? 	<p>a. 4 b. 3 с. 5 d. 2 е. 14 f. 6</p>
218.	Дано красно-черное дерево:	<p>a. 1 b. 2 с. 3 d. 4</p>

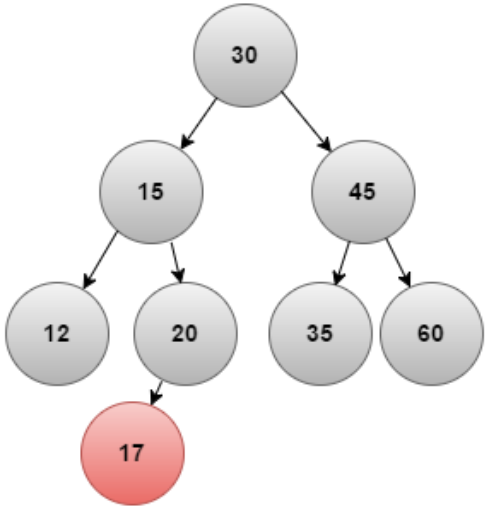
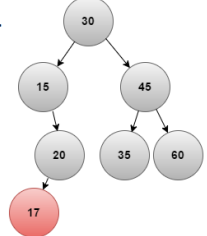
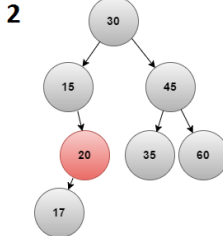
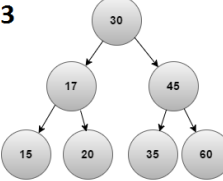
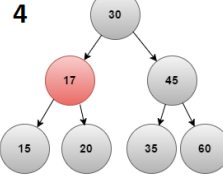
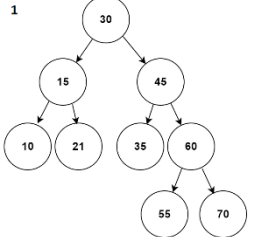
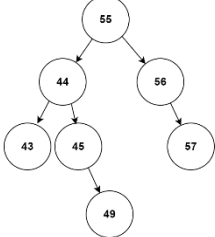
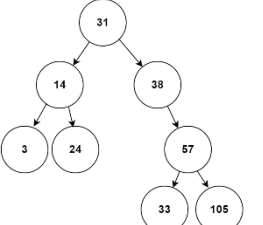
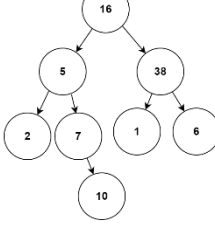


дерево после добавления 65?

Какой примет вид





219.	 <p>Дано дерево: Какой примет вид дерево после удаления 12?</p> <p>1 </p> <p>2 </p> <p>3 </p> <p>4 </p>	a. 1 b. 2 c. 3 d. 4
220.	<p>Какие из перечисленных деревьев удовлетворяют требованиям двоичного дерева поиска?</p> <p>1 </p> <p>2 </p> <p>3 </p> <p>4 </p>	a. 3 b. 2 c. 4 и 1 d. 3 и 4 e. 3 и 2 f. 1 и 2 g. 1 h. Правильного варианта нет
221.	<p>Какие из перечисленных деревьев удовлетворяют требованиям двоичного дерева поиска?</p>	a. 4 b. 3 и 2



		<p>с. 4 и 1 d. 3 е. 2 f. 1 и 2 g. 1 h. Правильного варианта нет</p>
222.	<p>Какие из перечисленных деревьев удовлетворяют требованиям двоичного дерева поиска?</p>	<p>a. 2 b. 3 с. 4 и 1 d. 3 и 2 е. 4 f. 3 и 4 g. 1 h. Правильного варианта нет</p>
223.	<p>Какие из перечисленных деревьев удовлетворяют требованиям двоичного дерева поиска?</p>	<p>a. 1 b. 2 с. 4 d. 3 и 4 е. 3 f. 1 и 2 g. 3 и 4 h. Правильного варианта нет</p>
224.	<p>На какой сортировке основана сортировка Шелла?</p>	<p>a. Вставками b. Выбором с. Пирамидальный d. Карманная е. Перестановками</p>



225.	Имеются 2 одинаковых ЭВМ - А и Б, в памяти каждой из которых содержится упорядоченный массив из 2000 целых неповторяющихся чисел от 1 до 2000. ЭВМ А ищет некоторое число при помощи последовательного поиска, а ЭВМ Б ищет число 24 при помощи двоичного поиска. Какое из следующих утверждений является истинным?	a. Обе заикнутся b. ЭВМ Б найдёт число за не более чем 2000 операций c. ЭВМ Б найдёт число за не более чем 12 операций d. В обоих случаях будет выявлена ошибка e. ЭВМ А найдёт число за не более чем 12 операций
226.	В процессе сортировки сравниваются элементы, отстоящие друг от друга на некоторое, возможно большое, число позиций. По какому алгоритму выполняется эта сортировка?	a. Пузырьковая b. Выбором c. Шелла d. Быстрая e. Вставками
227.	Какое из следующих высказываний наилучшим образом характеризует сортировку Выбором?	a. Считается самой простой b. Не подходит для 1-мерных массивов c. Ищет наименьший или наибольший элемент d. Считается самой быстрой e. Выполняет наименьшее число операций
228.	Какое из следующих высказываний наилучшим образом характеризует пузырьковую сортировку?	a. Считается самой простой b. Не подходит для 1-мерных массивов c. Ищет наименьший или наибольший элемент d. Считается самой быстрой e. Выполняет наименьшее число операций
229.	Производится пузырьковая сортировка (всплывает «легкий» элемент) массива из 9 элементов, причём массив уже упорядочен в порядке возрастания. Сколько будет выполнено перестановок?	a. 8 b. 0 c. 18 d. 9 e. 36
230.	Производится пузырьковая сортировка массива из 8 элементов. Сколько будет выполнено операций сравнения?	a. 7 b. 28 c. 8 d. 15 e. 0
231.	В процессе сортировки выполняется поиск наименьшего элемента. По какому алгоритму выполняется эта сортировка?	a. Пузырьковая b. Вставками c. Быстрая d. Выбором e. Шелла
232.	В процессе сортировки весь сортируемый массив и каждая его часть делятся на две части. По какому алгоритму выполняется эта	a. Пузырьковая b. Вставками



	сортировка?	с. Шелла d. Быстрая е. Выбором
233.	В процессе сортировки сравниваются соседние элементы. По какому алгоритму выполняется эта сортировка?	а. Быстрая b. Пузырьковая с. Шелла d. Выбором е. Вставками
234.	В процессе сортировки возможно перемещение по массиву большого числа элементов. По какому алгоритму выполняется эта сортировка?	a. Вставками b. Шелла с. Пузырьковая d. Выбором е. Быстрая
235.	В процессе сортировки возможно перемещение по массиву большого числа элементов. По какому алгоритму выполняется эта сортировка?	а. Пузырьковая b. Вставками с. Пирамидальная d. Выбором е. Быстрая
236.	Какая сортировка из следующих является самой эффективной?	a. Быстрая b. Пузырьковая с. Выбором d. Шелла е. Вставками
237.	Какое из следующих высказываний наилучшим образом характеризует быструю сортировку?	а. Не подходит для 1-мерных массивов b. Считается самой простой с. Относится к самым быстрым в среднем случае из универсальных алгоритмов d. Ищет наименьший или наибольший элемент е. Выполняет наименьшее число операций
238.	Имеется неупорядоченный массив целых чисел из 8 элементов. Сколько операций сравнения потребуется для нахождения искомого ключа, если он находится в конце массива?	а. 7 b. 0 с. 1 d. 4 е. 8
239.	Имеется упорядоченный массив целых чисел из 15 элементов. Сколько операций сравнения потребуется при двоичном поиске для установления факта отсутствия искомого данных в этом массиве?	а. 14 b. 5 с. 10 d. $\log_2(15)$ е. 1
240.	Имеется упорядоченный массив целых чисел. Для нахождения ключа используется двоичный поиск. Гарантируется ли в этом случае истинность результата поиска?	а. Нет b. Да с. Гарантируется при условии, что значение ключа



		не превышает размера массива d. Гарантируется при условии, что в процедуре поиска используется цикл for e. Гарантируется при условии, что в процедуре поиска используется цикл while
241.	Имеется неупорядоченный массив целых чисел. Для нахождения ключа используется последовательный поиск. Гарантируется ли в этом случае истинность результата поиска?	a. Гарантируется при условии, что в процедуре поиска используется цикл while b. Гарантируется при условии, что значение ключа не превышает размера массива c. Нет d. Гарантируется при условии, что в процедуре поиска используется цикл for e. Да
242.	Имеются 2 одинаковых ЭВМ - А и Б, в памяти каждой из которых содержится упорядоченный массив из 2000 целых неповторяющихся чисел от 1 до 2000. ЭВМ А ищет число 24 при помощи последовательного поиска, а ЭВМ Б ищет число 24 при помощи двоичного поиска. Какая из машин раньше закончит поиск?	a. Обе закинутся b. А c. Обе одновременно d. В обоих случаях будет выявлена ошибка e. Б
243.	Какая сортировка из следующих является самой неэффективной?	a. Вставками b. Выбором c. Пузырьковая d. Быстрая
244.	Имеется упорядоченный массив целых чисел из 9 элементов. Сколько операций сравнения потребуется при двоичном поиске для нахождения искомого ключа, если он находится в точно в середине массива?	a. 5 b. 1 c. 0 d. 8 e. 9
245.	Сколько дополнительных переменных нужно в пузырьковой сортировке помимо массива, содержащего элементы?	a. 0 (не нужно) b. всего 1 элемент c. n переменных (столько, сколько элементов в массиве)
246.	Как рассортировать массив быстрее, пользуясь пузырьковым методом?	a. одинаково b. по возрастанию элементов c. по убыванию элементов
247.	Какой из критериев эффективности сортировки определяется формулой $M=0,01*n*n+10*n$?	a. число сравнений b. время, затраченное на



		написание программы с. количество перемещений d. время, затраченное на сортировку															
248.	Как называется сортировка, происходящая в оперативной памяти?	a. сортировка таблицы адресов b. полная сортировка c. сортировка прямым включением d. внутренняя сортировка e. внешняя сортировка															
249.	Как можно сократить затраты машинного времени при сортировке большого объема данных?	a. производить сортировку в таблице адресов ключей b. производить сортировку на более мощном компьютере c. разбить данные на более мелкие порции и сортировать их															
250.	Существуют следующие методы сортировки. Найдите ошибку.	a. строгие b. улучшенные c. динамические															
251.	Улучшенные методы имеют значительное преимущество:	a. при большом количестве сортируемых элементов b. когда массив обратно упорядочен c. при малых количествах сортируемых элементов d. во всех случаях															
252.	Что из перечисленных ниже понятий является одним из типов сортировки?	a. внутренняя сортировка b. сортировка по убыванию c. сортировка данных d. сортировка по возрастанию															
253.	Есть исходный файл, сортируемый простым слиянием. Исходный файл: f: 9 5 4 8 1 6 23 41 13 2 <table border="1"><thead><tr><th></th><th>Распределение</th><th>Слияние</th></tr></thead><tbody><tr><td>N проход</td><td>f1: 9 4 1 23 13 F2: 5 8 6 41 2</td><td>F: 5,9 4,8 1,6 23,41 2, 13</td></tr><tr><td>N+1 проход</td><td>F1: 5,9 1,6 2,13 F2: 4,8 23,41</td><td>4 5 8 9, 1 6 23 41, 2 13</td></tr><tr><td>N+2</td><td>F1: 4 5 8 9, 2 13 F2: 1 6 23 41</td><td>1 4 5 6 8 9 23 41, 2 13</td></tr><tr><td>N+3</td><td>F1: 1 4 5 6 8 9 23 41 F2: 2 13</td><td>1 2 4 5 6 8 9 13 23 41</td></tr></tbody></table> Сколько потребуется проходов для достижения конечного результата?		Распределение	Слияние	N проход	f1: 9 4 1 23 13 F2: 5 8 6 41 2	F: 5,9 4,8 1,6 23,41 2, 13	N+1 проход	F1: 5,9 1,6 2,13 F2: 4,8 23,41	4 5 8 9, 1 6 23 41, 2 13	N+2	F1: 4 5 8 9, 2 13 F2: 1 6 23 41	1 4 5 6 8 9 23 41, 2 13	N+3	F1: 1 4 5 6 8 9 23 41 F2: 2 13	1 2 4 5 6 8 9 13 23 41	a. 2 b. 3 c. 4 d. 5
	Распределение	Слияние															
N проход	f1: 9 4 1 23 13 F2: 5 8 6 41 2	F: 5,9 4,8 1,6 23,41 2, 13															
N+1 проход	F1: 5,9 1,6 2,13 F2: 4,8 23,41	4 5 8 9, 1 6 23 41, 2 13															
N+2	F1: 4 5 8 9, 2 13 F2: 1 6 23 41	1 4 5 6 8 9 23 41, 2 13															
N+3	F1: 1 4 5 6 8 9 23 41 F2: 2 13	1 2 4 5 6 8 9 13 23 41															
254.	Есть исходный файл, сортируемый простым слиянием. Исходный файл: f: 9 5 4 8 1 6 23 41 13 2 Как будут распределены элементы этого файла при первом проходе?	a. f1: 1 2 4 5 6 8 f2: 9 13 23 41 b. f1: 41 23 13 9 f2: 1 2 4 5 6 8 c. f1: 8 6 5 4 2 1 f2: 41 23 13 9															



		d. f1: 9 4 1 23 13 f2: 5 8 6 41 2															
255.	Есть исходный файл, сортируемый простым слиянием. Исходный файл: f: 9 5 4 8 1 6 23 41 13 2 Как будет выглядеть слияние элементов этого файла после третьего прохода?	a. 1 4 5 6 8 9 23 41, 2 13 b. 41 4 5 23 8 9 6 1, 13 2 c. 41 5 4 23 1 9 6 8, 13 2 d. 41 5, 4 23, 1 9, 6 8, 13 2															
256.	Есть исходный файл, сортируемый естественным слиянием. Исходный файл: f: 9 5 4 8 1 6 23 41 13 2 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th>Распределение</th> <th>Слияние</th> </tr> </thead> <tbody> <tr> <td>N проход</td> <td>f1: 9 4 8 13 F2: 5 1 6 23 41 2</td> <td>F: 5 9 1 4 6 8 23 41 2 13</td> </tr> <tr> <td>N+1 проход</td> <td>F1: 4 5 8 9 2 13 F2: 1 6 23 41</td> <td>4 5 8 9, 1 6 23 41, 2 13</td> </tr> <tr> <td>N+2</td> <td>F1: 4 5 8 9, 2 13 F2: 1 6 23 41</td> <td>1 4 5 6 8 9 23 41, 2 13</td> </tr> <tr> <td>N+3</td> <td>1 4 5 6 8 9 23 41 2 13</td> <td>1 2 4 5 6 8 9 13 23 41</td> </tr> </tbody> </table> <p>Сколько потребуется проходов для достижения конечного результата?</p>		Распределение	Слияние	N проход	f1: 9 4 8 13 F2: 5 1 6 23 41 2	F: 5 9 1 4 6 8 23 41 2 13	N+1 проход	F1: 4 5 8 9 2 13 F2: 1 6 23 41	4 5 8 9, 1 6 23 41, 2 13	N+2	F1: 4 5 8 9, 2 13 F2: 1 6 23 41	1 4 5 6 8 9 23 41, 2 13	N+3	1 4 5 6 8 9 23 41 2 13	1 2 4 5 6 8 9 13 23 41	a. 2 b. 3 c. 4 d. 5
	Распределение	Слияние															
N проход	f1: 9 4 8 13 F2: 5 1 6 23 41 2	F: 5 9 1 4 6 8 23 41 2 13															
N+1 проход	F1: 4 5 8 9 2 13 F2: 1 6 23 41	4 5 8 9, 1 6 23 41, 2 13															
N+2	F1: 4 5 8 9, 2 13 F2: 1 6 23 41	1 4 5 6 8 9 23 41, 2 13															
N+3	1 4 5 6 8 9 23 41 2 13	1 2 4 5 6 8 9 13 23 41															
257.	Есть исходный файл, сортируемый естественным слиянием. Исходный файл: f: 9 5 4 8 1 6 23 41 13 2 Как будут распределены элементы этого файла при втором проходе?	a. f1: 1 2 4 5 6 8 f2: 9 13 23 41 b. f1: 41 23 13 9 f2: 1 2 4 5 6 8 c. f1: 8 6 5 4 2 1 f2: 41 23 13 9 d. f1: 4 5 8 9 2 13 f2: 1 6 23 41															
258.	Есть исходный файл, сортируемый естественным слиянием. Исходный файл: f: 9 5 4 8 1 6 23 41 13 2 Как будет выглядеть слияние элементов этого файла после первого прохода?	a. 5 9 1 4 6 8 23 41 2 13 b. 41 4 5 23 8 9 6 1, 13 2 c. 41 5 4 23 1 9 6 8, 13 2 d. 41 5, 4 23, 1 9, 6 8, 13 2															

Примеры практических работ:

Практическая работа №1: Эмпирический анализ временной сложности алгоритмов

Цель: научиться оценивать и сравнивать теоретическую и практическую временную сложность алгоритмов.

Содержание:

- Реализация функций: константная, сумма/произведение элементов, вычисление полинома (наивный метод и метод Горнера)
- Реализация алгоритмов сортировки: пузырьком, быстрая, Timsort
- Алгоритмы возведения в степень (простой, рекурсивный, быстрый)
- Умножение матриц размера $n \times n$
- Замер времени выполнения для $n = 1..2000$ (5 запусков на каждое значение)
- Построение графиков зависимости времени от n , сравнение с теоретической оценкой O-нотации

Ожидаемый результат: Отчёт с графиками, таблицами замеров и обоснованием соответствия эмпирических данных теоретической сложности.

Практическая работа №2: Рекурсивные алгоритмы

Цель: освоить принципы рекурсии и научиться применять её для решения задач.

Содержание:



- Реализация отрисовки фракталов (например, дерево Пифагора, снежинка Коха) с использованием рекурсии и графического интерфейса
- Реализация головоломки «Ханойские башни» с визуализацией шагов
- *Доп. задание:* Нерекурсивная реализация «Ханойских башен» с использованием стека

Ожидаемый результат: Рабочее приложение с графическим интерфейсом, демонстрирующее работу рекурсивных алгоритмов.

Практическая работа №3: Динамические структуры данных

Цель: изучить и реализовать базовые динамические структуры: стек, очередь, связный список.

Содержание:

- **Часть 1:** Реализация стека на собственном связном списке (Push, Pop, Top, isEmpty, Print); обработка команд из файла; вычисление выражений в постфиксной записи; перевод инфиксной записи в постфиксную
- **Часть 2:** Реализация очереди (на списке и через стандартный класс); замеры времени для разных сценариев
- **Часть 3:** Поиск и реализация примеров применения структур данных (Список, Стек, Очередь, Дерево) в реальных задачах
- **Часть 4:** Реализация 12 алгоритмов работы со связными списками (разворот, удаление дубликатов, вставка в упорядоченный список и др.)

Ожидаемый результат: Модульная библиотека структур данных с тестами, отчёт с анализом сложности операций.

Практическая работа №4: Алгоритмы сортировки

Цель: сравнить эффективность внутренних и внешних алгоритмов сортировки.

Содержание:

- **Задание 1:** Визуализация двух алгоритмов внутренней сортировки (один из группы квадратичных, один усовершенствованный); пошаговое отображение, логирование сравнений/перестановок, настройка задержки
- **Задание 2:** Реализация внешней сортировки слиянием (прямое / естественное / многопутевое — на выбор); сортировка записей из файла по ключевому атрибуту
- **Задание 3:** Сортировка слов текста в лексикографическом порядке двумя алгоритмами (один базовый/усовершенствованный + radix/ABC-сортировка); подсчёт частоты слов; эксперименты на текстах разной длины с замерами времени

Ожидаемый результат: Интерактивная программа с визуализацией, таблица сравнения производительности, отчёт с выводами.

Практическая работа №5: Алгоритмы на графах

Цель: реализовать и визуализировать ключевые алгоритмы теории графов.

Содержание:

- **Задание 1:** Графический редактор графов (добавление/удаление вершин и рёбер, загрузка/сохранение); визуализация обхода в ширину (BFS) и глубину (DFS)
- **Задание 2:** Моделирование транспортной сети; визуализация алгоритма поиска максимального потока (Форда-Фалкерсона / Эдмондса-Карпа)
- **Задание 3:** Визуализация построения минимального остовного дерева (алгоритм Крускала или Прима)



- **Задание 4:** Поиск кратчайшего пути между двумя вершинами (алгоритм Дейкстры) с пошаговой демонстрацией

Ожидаемый результат: Интерактивное приложение с графическим интерфейсом, демонстрирующее работу алгоритмов «по шагам».

Практическая работа 6: Хеш-таблицы

Цель: Изучить методы хеширования и стратегии разрешения коллизий.

Содержание:

- **Задание 1:** Хеш-таблица с цепочками ($m=1000$); реализация Search/Insert/Delete; сравнение хеш-функций (деление, умножение, 3–4 собственных метода); генерация 100 000 ключей, анализ заполнения и длины цепочек
- **Задание 2:** Хеш-таблица с открытой адресацией ($m=10\ 000$); реализация линейного/квадратичного зондирования, двойного хеширования, 2 собственных метода; обработка переполнения, анализ кластеризации

Ожидаемый результат: Сравнительный анализ методов хеширования с обоснованием выбора наилучшей хеш-функции для каждого подхода.

Примерные вопросы на экзамен:

1. Сортировка пузырьком
2. Сортировка вставками
3. Сортировка выбором
4. Шейкерная сортировка
5. Сортировка Шелла
6. Алгоритм бинарного поиска
7. Быстрая сортировка
8. Внешняя сортировка слиянием (прямое, естественное, многопутевое) (может быть одно из перечисленных)
9. Сортировка с помощью двоичного дерева
10. Поразрядная сортировка (radix sort)
11. Хэш-таблицы с разрешением коллизий методом цепочек
12. Хэш-таблицы с разрешением коллизий методом открытой адресации
13. ABC-сортировка (для строк) (не будет)
14. Реализовать стек и базовые операции работы со стеклом, с использованием собственного двусвязного списка
15. Реализовать очередь и базовые операции работы с очередью, с использованием собственного двусвязного списка
16. Алгоритм Крускала (поиска минимального остовного дерева)
17. Алгоритм Прима (поиска минимального остовного дерева)
18. Обход графа в глубину
19. Обход графа в ширину
20. Алгоритм Дейкстры (поиск кратчайшего пути)



4. Порядок проведения и критерии оценивания промежуточной аттестации

4.1. Порядок приема практических работ

В течение семестра студентами должны быть выполнены 6 практических работ, и по каждой из них готовится отчет и проводится защита работы.

Практические работы оцениваются по следующим критериям:

Зачтено

1. Работа выполнена в достаточном объеме
2. В работе возможны ошибки, не приводящие к сильным искажениям результатов, либо отсутствуют полностью.
3. Студент, в случае необходимости, демонстрирует свободное владение материалом и может ответить на дополнительные вопросы.

Не зачтено

1. Выполнено менее 60% задания
2. В отчете по практической работе допущены грубые ошибки или неточности.
3. Студент не ориентируется в материале практического занятия и не владеет в достаточной мере знаниями, необходимыми для выполнения практического задания.

Штрафы:

-20 баллов за каждую неделю просрочки сдачи лабораторной работы

Штраф применяется к баллам за конкретную работу (минимальный итог — 0)

Просрочка считается от дедлайна, установленного преподавателем

Каждая защищенная практическая работа дает 10 баллов к оценке за дисциплину.
Максимум можно набрать 60 баллов за практические работы.

4.2. Порядок проведения промежуточной аттестации

На экзамене студент получает билет с тремя задачами различной сложности:

Каждая из 3 задач оценивается отдельно:

Группа сложности	Макс. за задачу	Требования
Группа 1 (базовая)	25 баллов	Теоретическое описание + работающая реализация базового алгоритма
Группа 2 (средняя)	35 баллов	Описание + реализация + анализ сложности + обработка граничных случаев
Группа 3 (продвинутая)	40 баллов	Полное решение: теория, код, оптимизация, сравнение с альтернативами, обоснование выбора

Задача считается выполненной, только если предоставлены **и теория, и код**. Отсутствие одного из компонентов — 0 баллов за задачу.

Итоговая оценка за экзамен умножается на коэффициент 0.4 и суммируется с баллами за практические работы.

В итоге студент получает общую оценку в стобалльной системе, которая в дальнейшем переводится в пятибалльную систему согласно критериям.



4.3. Критерии оценивания промежуточной аттестации по видам оценочных средств

Общая оценка формируется в системе электронного обучения MOODLE. Максимальный балл — 100 баллов.

Оценка	Отлично/ Зачтено	Хорошо/ зачтено	Удовлетворитель но/зачтено	Неудовлетворительно/ незачтено
Баллы	100-90 баллов	89-75 баллов	74-60 баллов	59-0 баллов
Уровень освоения проверяемых компетенций	высокий	средний	базовый	недостаточный

4.4. Результаты промежуточной аттестации и уровни сформированности компетенций

При подведении итогов учитываются результаты выполнения практических работ и промежуточной аттестации:

- 0-59 баллов – неудовлетворительно/незачтено;
- 60-74 баллов – удовлетворительно/зачтено;
- 75-89 баллов – хорошо/зачтено;
- 90-100 баллов – отлично/зачтено;

Особенности проведения процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья обозначены в рабочей программе дисциплины (модуля).

Уровни сформированности компетенций определяется следующим образом:

1. Высокий уровень сформированности компетенций соответствует оценке отлично:
 - предполагает формирование компетенций на высоком уровне;
 - знание теоретических разделов изучаемой дисциплины на уровне не ниже оценки отлично;
 - студент умеет применять на практике знания, полученные в рамках изучения дисциплины
 - формируются навыки использования теоретических и практических разделов дисциплины для решения задач профессиональной деятельности;
2. Средний уровень соответствует оценке хорошо:
 - предполагает формирование компетенций на среднем уровне;
 - знание теоретических разделов изучаемой дисциплины на уровне не ниже оценки хорошо;
 - студент умеет применять знания, полученные в рамках изучения дисциплины, для решения задач профессиональной деятельности;
3. Базовый уровень соответствует оценке удовлетворительно:
 - предполагает формирование компетенций на базовом уровне;
 - знание теоретических разделов изучаемой дисциплины на уровне не ниже оценки удовлетворительно;



4. Недостаточный уровень соответствует оценке неудовлетворительно.