

Государственное образовательное учреждение
«Приднестровский государственный университет им. Т.Г. Шевченко»

Инженерно-технический институт

Кафедра программного обеспечения вычислительной техники
и автоматизированных систем

УТВЕРЖДАЮ

Заведующий кафедрой ПОВТ и АС



С.Г. Федорченко

«30» августа 2021 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
УЧЕБНОЙ ДИСЦИПЛИНЫ
Б1.В.08 «ТИПЫ И СТРУКТУРЫ ДАННЫХ»

Направление подготовки

09.03.04 Программная инженерия

Разработка программно-информационных систем

Квалификация
выпускника:

бакалавр

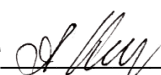
Форма обучения:

очная, заочная

Год набора:

2020 г.

Разработал:

доцент  / А.В. Кирсанова

«30» августа 2021 г.

Тирасполь, 2021

Паспорт фонда оценочных средств по учебной дисциплине

1. В результате изучения дисциплины: Б1.В.08 «Типы и структуры данных» у обучающегося должны быть сформированы следующие компетенции:

Категория (группа) компетенций	Код и наименование	Код и наименование индикатора достижения универсальной компетенции
Обязательные профессиональные компетенции выпускников и индикаторы их достижения		
	ПК-9. Владение навыками использования операционных систем, сетевых технологий, средств разработки программного интерфейса, применения языков и методов формальных спецификаций, систем управления базами данных	ИД-1 _{ПК-9} Знает методы формальных спецификаций и системы управления базами данных
		ИД-2 _{ПК-9} Умеет применять современные средства и языки программирования
		ИД-3 _{ПК-9} Имеет навыки использования операционных систем

2. Программа оценивания контролируемой компетенции:

Текущая аттестация	Контролируемые модули, разделы (темы) дисциплины их название	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
РУБЕЖНЫЙ КОНТРОЛЬ	<i>Раздел 1.</i> Реализации различных структур, данных и алгоритмов их обработки для решения задач.	ПК-9	КР1, ЛБ1-ЛБ8
	<i>Раздел 2.</i> Оценка эффективности использования различных алгоритмов обработки данных в зависимости от конкретной реализации данных.	ПК-9	КР1, ЛБ1-ЛБ8
РУБЕЖНАЯ АТТЕСТАЦИЯ	<i>Раздел 2.</i> Оценка эффективности использования различных алгоритмов обработки данных в зависимости от конкретной реализации данных.	ПК-9	КР1, КР2, ЛБ1-ЛБ8
	<i>Раздел 3.</i> Применение навыков тестирования, отладки и оформления программ, приобретение навыков работы в команде.	ПК-9	КР1, КР2, ЛБ1-ЛБ8
Промежуточная аттестация		ПК-9	Экзамен

3. Показатели и критерии оценивания компетенции по этапам формирования, описание шкал оценивания

Этапы оценивания компетенции	Показатели достижения заданного уровня освоения компетенции	Критерии оценивания результатов обучения			
		2	3	4	5
Первый этап	Знать ПК-1	Не знает методы формальных спецификаций и системы управления базами данных	Знает методы формальных спецификаций и системы управления базами данных, но допускает грубые ошибки	Знает методы формальных спецификаций и системы управления базами данных, но допускает неточности	Знает методы формальных спецификаций и системы управления базами данных
Второй этап	Уметь ПК-1	Не умеет применять современные средства и языки программирования	Умеет применять современные средства и языки программирования, но допускает грубые ошибки	Умеет применять современные средства и языки программирования, но допускает неточности	Умеет применять современные средства и языки программирования
Третий этап	Владеть ПК-1	Не имеет навыки использования операционных систем	Имеет навыки использования операционных систем, но допускает грубые ошибки	Имеет навыки использования операционных систем, но допускает неточности	Имеет навыки использования операционных систем

4. Шкала оценивания

Согласно Положению «О порядке организации аттестации в ИТИ ПГУ им. Т.Г. Шевченко, итоговая оценка представляет собой сумму баллов, полученных студентом по итогу освоения дисциплины (модуля):

Оценка в традиционной шкале	Оценка в 100-балльной шкале	Буквенные эквиваленты оценок в шкале ЗЕ (% успешно аттестованных)
5 (отлично)	88–100	А (отлично) – 88-100 баллов
4 (хорошо)	70–87	В (очень хорошо) – 80-87 баллов
		С (хорошо) – 70-79 баллов
3 (удовлетворительно)	50–69	D(удовлетворительно) – 60-69 баллов
		E(посредственно) – 50-59 баллов
2 (неудовлетворительно)	0–49	Fх– неудовлетворительно, с возможной пересдачей – 21-49 баллов
		F– неудовлетворительно, с повторным изучением дисциплины – 0-20 баллов

Расшифровка уровня знаний, соответствующего полученным баллам, дается в таблице, указанной ниже

A	“Отлично” - теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному.
B	“Очень хорошо” - теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом в основном сформированы, все предусмотренные программой обучения учебные задания выполнены, качество выполнения большинства из них оценено числом баллов, близким к максимальному.
C	“Хорошо” - теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками.
D	“Удовлетворительно” - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий, возможно, содержат ошибки.
E	“Посредственно” - теоретическое содержание курса освоено частично, некоторые практические навыки работы не сформированы, многие предусмотренные программой обучения учебные задания не выполнены, либо качество выполнения некоторых из них оценено числом баллов, близким к минимальному.
FX	“Условно неудовлетворительно” - теоретическое содержание курса освоено частично, необходимые практические навыки работы не сформированы, большинство предусмотренных программой обучения учебных заданий не выполнено, либо качество их выполнения оценено числом баллов, близким к минимальному; при дополнительной самостоятельной работе над материалом курса возможно повышение качества выполнения учебных заданий.
F	“Безусловно неудовлетворительно” - теоретическое содержание курса не освоено, необходимые практические навыки работы не сформированы, все выполненные учебные задания содержат грубые ошибки, дополнительная самостоятельная работа над материалом курса не приведет к какому-либо значимому повышению качества выполнения учебных заданий.

5. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций при изучении учебной дисциплины в процессе освоения образовательной программы

5.1 Типовой вариант задания на контрольную работу 1.

1. Приведите характеристики различных структур данных, методы их создания и представления в памяти.
2. Асимптотические обозначения скорости роста сложности алгоритма.
3. Напишите программу подсчета медианы трех целых чисел. Какой случай для алгоритма является наилучшим? Наихудшим? Средним? (Если наилучший и наихудший случаи совпадают, то перепишите Ваш алгоритм с простыми условиями, не пользуясь временными переменными, так, чтобы наилучший случай был лучше наихудшего.)

5.2 Типовой вариант задания на контрольную работу 2

1. Назовите параметры алгоритмов сортировки последовательностей.
2. Сгенерировать случайные данные не менее 50 и записать в массив. Отсортировать полученную последовательность методом вставки. Сколько сравнений выполнит алгоритм? Определите зависимость сложности от N .
3. Сгенерировать случайные данные не менее 50 и записать в файл. Отсортировать полученную последовательность естественного слияния. Сколько сравнений выполнит алгоритм? Определите зависимость сложности от N .

5.3. Типовой вариант задания на лабораторную работу 1. Анализ алгоритмов

- 1) Напишите программу, подсчитывающую количество прописных букв в текстовом файле. Сколько сравнений требуется этому алгоритму? Каково максимальное возможное значение числа операций увеличения счетчика? Минимальное такое число? (Выразите ответ через число N символов во входном файле.)
- 2) В файле записан некоторый набор чисел, однако мы не знаем, сколько их. Напишите программу для подсчета среднего значения чисел в файле. Какого типа операции делает Ваш алгоритм? Сколько операций каждого типа он делает?
- 3) Напишите программу, не использующую сложных условий, которая по трем введенным целым числам определяет, различны ли они все между собой. Сколько сравнений в среднем делает Ваш алгоритм? Не забудьте исследовать все классы входных данных.
- 4) Напишите программу, которая получает на входе три целых числа, и находит наибольшее из них. Каковы возможные классы входных данных? На каком из них Ваш алгоритм делает наибольшее число сравнений? На каком меньше всего? (Если разницы между наилучшим и наихудшим классами нет, перепишите свой алгоритм с простыми сравнениями так, чтобы он не использовал временных переменных, и чтобы в наилучшем случае он работал быстрее, чем в наихудшем).
- 5) Напишите программу для поиска второго по величине элемента в списке из N значений. Сколько сравнений делает Ваш алгоритм в наихудшем случае?
- 6) Напишите программу подсчета медианы трех целых чисел. Какой случай для алгоритма является наилучшим? Наихудшим? Средним? (Если наилучший и наихудший случаи совпадают, то перепишите Ваш алгоритм с простыми условиями, не пользуясь временными переменными, так, чтобы наилучший случай был лучше наихудшего.)
- 7) Напишите программу, проверяющую, верно ли, что данные четыре целых числа попарно различны. Какой из классов данных обеспечивает наилучший случай для Вашего алгоритма? Наихудший? Средний? (Если наилучший и наихудший случаи совпадают, то перепишите Ваш алгоритм с простыми условиями, не пользуясь временными переменными, так, чтобы наилучший случай был лучше наихудшего.)
- 8) Напишите программу, которая по данному списку чисел и среднему значению этих чисел определяет, превышает ли число элементов списка, больших среднего значения, число элементов, меньших этого значения, или наоборот. Опишите группы, на которые распадаются возможные наборы входных данных. Какой случай для алгоритма является наилучшим? Наихудшим? Средним? (Если наилучший и наихудший случаи совпадают, то перепишите Ваш алгоритм так, чтобы он останавливался, как только ответ на по-

ставленный вопрос становится известным, делая наилучший случай лучше наихудшего.)

9) Напишите программу, проверяющую, могут ли три положительных числа a , b , c быть сторонами треугольника. Какой случай для алгоритма является наилучшим? Наихудшим? Сколько сравнений требуется этому алгоритму?

10) Напишите программу, проверяющую, может ли кирпич, заданный тремя размерами X , Y , Z пройти через прямоугольное отверстие размером A , B . Какой случай для алгоритма является наилучшим? Наихудшим? Сколько сравнений требуется этому алгоритму?

5.4 Типовой вариант задания на лабораторную работу 2. Генерация псевдослучайных чисел.

1. Разберите предложенные алгоритмы генерации случайных чисел. Определите интервал повторения случайных чисел.

2. Напишите на основе алгоритма последовательного поиска функцию и протестируйте ее.

3. С помощью одного из методов сгенерируйте список случайных чисел в промежутке от 1 до N и проводите поиск в этом списке; здесь N может быть любым числом от 100 до 1000. В Вашей программе должен быть глобальный счетчик, который следует установить в нуль в начале программы и увеличивать на 1 непосредственно перед очередным сравнением.

4. Затем главная программа должна вызывать функцию последовательного поиска для каждого элемента от 1 до N . Подсчитанное после этого общее число сравнений следует разделить на N ; результатом будет среднее число сравнений при последовательном поиске.

5. Повторите шаг 3 для двоичного поиска в упорядоченном списке.

6. Повторите шаг 3 для выборки из списка заданного K -го минимального элемента.

7. Составьте отчет, в котором сравните результаты третьего, пятого и шестого шагов с предсказаниями анализа.

5.5. Типовой вариант задания на лабораторную работу 3. Фундаментальные структуры данных

№	Условие упражнения
1.	Реализовать базовые операции для массива.
2.	Реализовать базовые операции для одно- и двунаправленного списка, реализованного через массивы.

3.	Реализовать базовые операции для одно- и двунаправленного списка, реализованного через последовательности связанных компонент.
4.	Реализовать операции конкатенации и расцепления одно- и двунаправленного списка, реализованного через массивы и через последовательности связанных компонент.
5.	Реализовать операции добавления и удаления элемента в стеке, реализованном через массивы и через последовательности связанных компонент.
6.	Реализовать операции добавления и удаления элемента в обычной и в циклической очереди, реализованном через массивы и через последовательности связанных компонент.
7.	Реализовать операции добавления и удаления элемента в дек (и в начало и в конец очереди), реализованном через массивы и через последовательности связанных компонент.

5.6. Типовой вариант задания на лабораторную работу 4. ВНУТРЕННИЕ СОРТИРОВКИ МАССИВОВ И ПОСЛЕДОВАТЕЛЬНОСТЕЙ И ИХ АНАЛИЗ

Сгенерировать случайные данные не менее 50 и записать в массив. Отсортировать полученную последовательность:

Вариант 1: методом вставки. Сколько сравнений выполнит алгоритм? Определите зависимость сложности от N.

Вариант 2: методом выбора. Сколько сравнений выполнит алгоритм? Определите зависимость сложности от N.

Вариант 3: методом классического пузырька. Сколько сравнений выполнит алгоритм? Определите зависимость сложности от N.

Вариант 4: методом шейкер сортировки. Сколько сравнений выполнит алгоритм? Определите зависимость сложности от N.

Вариант 5: методом Шелла. Сколько сравнений выполнит алгоритм? Определите зависимость сложности от N.

Вариант 6: методом быстрой сортировки. Сколько сравнений выполнит алгоритм? Определите зависимость сложности от N.

Вариант 7: методом пирамидальной сортировки. Сколько сравнений выполнит алгоритм? Определите зависимость сложности от N.

Вариант 8: методом вставки. Сколько сравнений выполнит алгоритм? Определите зависимость сложности от N.

Вариант 9: методом корневой сортировки. Сколько сравнений выполнит алгоритм? Определите зависимость сложности от N.

5.7. Типовой вариант задания на лабораторную работу 5. ВНЕШНИЕ СОРТИРОВКИ МАССИВОВ И ПОСЛЕДОВАТЕЛЬНОСТЕЙ И ИХ АНАЛИЗ

Сгенерировать случайные данные не менее 50 и записать в ФАЙЛ. Отсортировать полученную последовательность:

Вариант 1: методом прямого слияния. Сколько сравнений выполнит алгоритм? Определите зависимость сложности от N.

Вариант 2: методом естественного слияния. Сколько сравнений выполнит алгоритм? Определите зависимость сложности от N.

Вариант 3: методом многопутевого слияния. Сколько сравнений выполнит алгоритм? Определите зависимость сложности от N.

Вариант 4: методом многофазной сортировки. Сколько сравнений выполнит алгоритм? Определите зависимость сложности от N.

5.8. Типовой вариант задания на лабораторную работу 6. Сортировка списков

Разработать программу в соответствии с заданием к лабораторной работе.

Задание.

1. Разработать функции сортировки списков методами, выбранными по таблице в соответствии с вариантом.

№	Алгоритмы сортировки	№	Алгоритмы сортировки
1	Линейный односвязный список, содержащий целые числа. 1) Отбором с процедурами вставки и удаления; 2) Вставками.	9	Кольцевой двусвязный список, содержащий символы. 1) Пузырьковая с заменой указателей; 2) Отбором с процедурами вставки и удаления.
2	Кольцевой односвязный список, содержащий символы. 1) Вставками; 2) Отбором с заменой указателей.	10	Кольцевой односвязный список, содержащий данные типа float. 1) Пузырьковая с заменой указателей; 2) Вставками.
3	Линейный односвязный список, содержащий данные типа float. 1) Пузырьковая с процедурами вставки и удаления; 2) Отбором с заменой указателей.	11	Кольцевой двусвязный список, содержащий целые числа. 1) Отбором с процедурами вставки и удаления; 2) Пузырьковая с переносом данных.
4	Линейный двусвязный список, содержащий целые числа. 1) Вставками; 2) Пузырьковая с переносом данных.	12	Линейный односвязный список, содержащий данные типа double. 1) Пузырьковая с процедурами вставки и удаления; 2) Отбором с процедурами вставки и удаления.
5	Линейный двусвязный список, содержащий символы. 1) Вставками; 2) Пузырьковая с заменой указателей.	13	Линейный двусвязный список, содержащий данные типа double. 1) Вставками; 2) Пузырьковая с процедурами вставки и удаления.
6	Линейный односвязный список, содержащий символы. 1) Отбором с заменой указателей; 2) Пузырьковая с переносом данных.	14	Кольцевой односвязный список, содержащий данные типа double. 1) Отбором с заменой указателей; 2) Вставками.
7	Кольцевой односвязный список, содержащий целые числа. 1) Пузырьковая с переносом данных; 2) Вставками.	15	Кольцевой двусвязный список, содержащий данные типа double. 1) Вставками; 2) Пузырьковая с заменой указателей.

8	Линейный двусвязный список, содержащий данные типа float. 1) Отбором с заменой указателей; 2) Пузырьковая с заменой указателей.	16	Кольцевой двусвязный список, содержащий данные типа float. 1) Вставками; 2) Отбором с заменой указателей.
---	---	----	---

5.9. Типовой вариант задания на лабораторную работу 7. Алгоритмы поиска и выборки целевого элемента

1. Разберите предложенные алгоритмы генерации случайных чисел. Определите интервал повторения случайных чисел.

2. Напишите на основе алгоритма последовательного поиска функцию и протестируйте ее.

3. С помощью одного из методов сгенерируйте список случайных чисел в промежутке от 1 до N и проводите поиск в этом списке; здесь N может быть любым числом от 100 до 1000. В Вашей программе должен быть глобальный счетчик, который следует установить в нуль в начале программы и увеличивать на 1 непосредственно перед очередным сравнением.

4. Затем главная программа должна вызывать функцию последовательного поиска для каждого элемента от 1 до N. Подсчитанное после этого общее число сравнений следует разделить на N; результатом будет среднее число сравнений при последовательном поиске.

5. Повторите шаг 3 для двоичного поиска в упорядоченном списке.

6. Повторите шаг 3 для выборки из списка заданного K-го минимального элемента.

7. Составьте отчет, в котором сравните результаты третьего, пятого и шестого шагов с предсказаниями анализа.

8. Введите строку текста. Выполните алгоритм Кнута-Морриса-Пратта. Подсчитайте количество сравнений, требуемых при поиске образца без повторений и с повторениями.

9. Введите строку текста. Выполните алгоритм Бойера-Мура. Подсчитайте количество сравнений, требуемых при поиске образца без повторений и с повторениями.

5.10. Типовой вариант задания на лабораторную работу 8. Решение задач методом динамического программирования.

Решите задачу. Оформите решение в соответствии с требованиями к оформлению задач.

Варианты заданий:

№	Условие
1.	Задача 99 «Шифр» Заданы три натуральных числа: два исходных числа и номер члена, который является ключом к шифру. Каждый последующий член последовательности получается путем прибавления к сумме двух предыдущих членов, суммы цифр суммы двух предыдущих членов. Написать программу, которая определяла бы ключ к шифру. Исходные данные - натуральные числа до 1 0001 000. Ввод корректен.

	Ввод: C1 C2 NC1 C2 N - разделенные пробелом исходные числа и номер искомого члена. Вывод: число-результат. Пример: Ввод: 39 11 539 11 5 Вывод: 140140
2.	Задача 1010 «Ход конем» На клетчатой доске размером $N \cdot N$ клеток в клетке с координатами AA, BB (координаты отсчитывать от левого нижнего угла), стоит шахматный конь. Конь за один ход перемещается одновременно на две клетки вперед и одну в сторону в любом горизонтальном или вертикальном направлении в пределах заданной доски. Написать программу, которая вычисляла бы количество клеток, в которые конь не сможет попасть за MM и менее MM ходов. Ввод корректен, MM и NN: $1 < N < 101$, $1 < N < 101$, $0 \leq M < 100$, $10 \leq M < 100$. Ввод: N A B M N A B M - размер доски, координаты коня и количество ходов. Вывод: число-результат. Пример: Ввод: 5 1 1 35 1 1 3 Вывод: 33
3.	Задача "Разноделяющиеся числа" Будем называть два целых числа «разноделяющимися», если большее по модулю делится без остатка на меньшее по модулю и нет ни одной такой цифры, которая входила бы в десятичную запись обоих этих чисел. Меньшее по модулю из этих чисел будем называть «разноделяющимся» делителем большего по модулю числа. Составить программу находящую на заданном интервале число (максимальное) имеющее наибольшее количество различных «разноделяющихся» делителей. Вывести это число и количество его различных «разноделяющихся» делителей. В отдельно взятом одном из «разноделяющихся» чисел цифры могут повторяться; AA и BB - границы интервала числа, не превосходящие по модулю число 2 000 000 2 000 000; границы входят в интервал, задаются и вводятся в произвольном порядке; длина интервала не превосходит 202202; если на заданном интервале несколько чисел имеют равное наибольшее количество различных «разноделяющихся» делителей, то взять максимальное из них; число «1» считать делителем (обычным) любого числа; знак числа цифрой не считать. Ввод: два числа AA и BB - границы интервала. Вывод: два числа: первое - на заданном интервале число, имеющее наибольшее количество различных «разноделяющихся» делителей, второе - количество различных «разноделяющихся» делителей первого числа. Пример: Ввод: 9 169 16 Вывод: 16 616 6

5.11 Типовой тест промежуточной аттестации

1. Структура данных представляет собой

- а) набор правил и ограничений, определяющих связи между отдельными элементами и группами данных
- б) набор правил и ограничений, определяющих связи между отдельными элементами данных

- c) набор правил и ограничений, определяющих связи между отдельными группами данных
- d) некоторую иерархию данных

2. Линейный список, в котором доступен только последний элемент, называется

- a) стеком
- b) очередью
- c) деком
- d) массивом
- e) кольцом

3. Структура данных работа с элементами которой организована по принципу FIFO (первый пришел - первый ушел) это –

- a) Стек
- б) Дек
- в) Очередь
- г) Список

4. Линейный последовательный список, в котором включение исключение элементов возможно с обоих концов, называется

- a) стеком
- b) очередью
- c) деком
- d) кольцевой очередью

5. В чём особенности очереди ?

- a) открыта с обеих сторон ;
- b) открыта с одной стороны на вставку и удаление;
- c) доступен любой элемент.

6. В чём особенности стека ?

- a) открыт с обеих сторон на вставку и удаление;
- b) доступен любой элемент;
- c) открыт с одной стороны на вставку и удаление.

7. Какую дисциплину обслуживания принято называть FIFO ?

- a) стек;
- b) очередь;
- c) дек.

8. Какая операция читает верхний элемент стека без удаления ?

- a) pop;
- b) push;
- b) stackpop.

9. Каково правило выборки элемента из стека ?

- a) первый элемент;
- b) последний элемент;
- c) любой элемент.

10. Как освободить память от удаленного из списка элемента?

- a) p=getnode;

- b) ptr(p)=nil;
- c) freenode(p);
- d) p=lst.

11. Как создать новый элемент списка с информационным полем D?

- a) p=getnode;
- b) p=getnode; info(p)=D;
- c) p=getnode; ptr(D)=lst.

12. Как создать пустой элемент с указателем p?

- a) p=getnode;
- b) info(p);
- c) freenode(p);
- d) ptr(p)=lst.

13. Сколько указателей используется в односвязных списках?

- a) 1
- b) 2;
- c) сколько угодно.

14. В чём отличительная особенность динамических объектов?

- a) порождаются непосредственно перед выполнением программы;
- b) возникают уже в процессе выполнения программы;
- c) задаются в процессе выполнения программы.

15. При удалении элемента из кольцевого списка...

- a) список разрывается;
- b) в списке образуется дыра;
- c) список становится короче на один элемент .

16. Для чего используется указатель в кольцевых списках

- a) для ссылки на следующий элемент;
- b) для запоминания номера сегмента расположения элемента;
- c) для ссылки на предыдущий элемент ;
- d) для расположения элемента в списке памяти.

17. Чем отличается кольцевой список от линейного?

- a) в кольцевом списке последний элемент является одновременно и первым;
- b) в кольцевом списке указатель последнего элемента пустой;
- c) в кольцевых списках последнего элемента нет ;
- d) в кольцевом списке указатель последнего элемента не пустой.

18. Сколько указателей используется в односвязном кольцевом списке ?

- a) 1;
- b) 2;
- c) сколько угодно.

19. В каких направлениях можно перемещаться в кольцевом двунаправленном списке ?

- a) в обоих;
- b) влево;
- c) вправо.

20. С помощью какой структуры данных наиболее рационально реализовать очередь ?

- а) стек;
- б) список;
- в) дек.

21. В памяти ЭВМ бинарное дерево удобно представлять в виде:

- а) связанных линейных списков;
- б) массивов;
- в) связанных нелинейных списков.

22. Элемент t , на который нет ссылок:

- а) корнем;
- б) промежуточным;
- в) терминальным (лист).

23. Дерево называется полным бинарным, если степень исходов вершин равна:

- а) 2 или 0;
- б) 2;
- в) M или 0;
- г) M .

24. Даны три условия окончания просеивания при сортировке прямым включением. Найдите среди них лишнее.

- а) найден элемент $a(i)$ с ключом, меньшим чем ключ u ;
- б) найден элемент $a(i)$ с ключом, большим чем ключ u ;
- в) достигнут левый конец готовой последовательности.

25. Какой из критериев эффективности сортировки определяется формулой $M = 0,01 \cdot n^3 + 10 \cdot n$?

- а) число сравнений;
- б) время, затраченное на написание программы;
- в) количество перемещений;
- г) время, затраченное на сортировку.

26. Как называется сортировка, происходящая в оперативной памяти?

- а) сортировка таблицы адресов;
- б) полная сортировка;
- в) сортировка прямым включением;
- г) внутренняя сортировка;
- д) внешняя сортировка.

27. Как можно сократить затраты машинного времени при сортировке большого объёма данных?

- а) производить сортировку в таблице адресов ключей;
- б) производить сортировку на более мощном компьютере;
- в) разбить данные на более мелкие порции и сортировать их.

28. Существуют следующие методы сортировки. Найдите ошибку.

- а) строгие;
- б) улучшенные;
- в) динамические

29. Метод сортировки называется устойчивым, если в процессе сортировки...

- а) относительное расположение элементов безразлично;

- b) относительное расположение элементов с равными ключами не меняется
- c) относительное расположение элементов с равными ключами изменяется;
- d) относительное расположение элементов не определено.

30. 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.

31. Что из перечисленных ниже понятий является одним из типов сортировки?

- a) внутренняя сортировка;
- b) сортировка по убыванию;
- c) сортировка данных;
- d) сортировка по возрастанию.

32. Сколько сравнений требует улучшенный алгоритм сортировки?

- a) $n \cdot \log(n)$;
- b) en ;
- c) $n \cdot n/4$.

33. Сколько сравнений и перестановок элементов требуется в пузырьковой сортировке?

- a) $n \cdot \log(n)$;
- b) $(n \cdot n)/4$;
- c) $(n \cdot n - n)/2$.

34. Сколько дополнительных переменных нужно в пузырьковой сортировке помимо массива, содержащего элементы ?

- a) 0 (не нужно);
- b) всего 1 элемент;
- c) n переменных (ровно столько, сколько элементов в массиве).

35. Как рассортировать массив быстрее, пользуясь пузырьковым методом?

- a) одинаково;
- b) по возрастанию элементов;
- c) по убыванию элементов.

36. В чём заключается идея метода QuickSort?

- a) выбор 1, 2, ..., n – го элемента для сравнения с остальными;
- b) разделение ключей по отношению к выбранному ;
- c) обмен местами между соседними элементами.

37. Массив сортируется “пузырьковым” методом. За сколько проходов по массиву самый “лёгкий” элемент в массиве окажется вверху ?

- a) за 1 проход;
- b) за $n-1$ проходов;
- c) за n проходов, где n – число элементов массива.

38. При обходе дерева слева направо получаем последовательность...

- a) отсортированную по убыванию;
- b) неотсортированную ;
- c) отсортированную по возрастанию.

39. При обходе дерева слева направо его элемент заносится в массив...

- a) при втором заходе в элемент;
- b) при первом заходе в элемент;
- c) при третьем заходе в элемент.

40. Где эффективен линейный поиск?

- a) в списке;
- b) в массиве;
- c) в массиве и в списке.

41. Какой поиск эффективнее?

- a) линейный;
- b) бинарный;
- c) без разницы.

42. В чём суть бинарного поиска?

- a) нахождение элемента массива x путём деления массива пополам каждый раз, пока элемент не найден;
- b) нахождение элемента x путём обхода массива;
- c) нахождение элемента массива x путём деления массива.

43. Как расположены элементы в массиве бинарного поиска?

- a) по возрастанию;
- b) хаотично;
- c) по убыванию.

44. В чём суть линейного поиска?

- a) производится последовательный просмотр от начала до конца и обратно через 2 элемента;
- b) производится последовательный просмотр элементов от середины таблицы;
- c) производится последовательный просмотр каждого элемента.

45. Где наиболее эффективен метод транспозиции?

- a) в массивах и в списках;
- b) только в массивах;
- c) только в списках.

46. В чём суть метода транспозиции?

- a) перестановка местами соседних элементов;
- b) нахождение одинаковых элементов;
- c) перестановка найденного элемента на одну позицию в сторону начала списка.

47. Что такое уникальный ключ?

- a) если разность значений двух данных равна ключу;
- b) если сумма значений двух данных равна ключу;
- c) если в таблице есть только одно данное с таким ключом.

48. В чём состоит назначение поиска?

- a) среди массива данных найти те данные, которые соответствуют заданному аргументу;
- b) определить, что данных в массиве нет;
- c) с помощью данных найти аргумент.

5.12 Вопросы к экзамену

1. Концепция типа данных. Абстрактные типы данных.
2. Структуры с индексированием. Поиск и выборка в структурах с индексированием.
3. Анализ сложности алгоритмов. Классы сложности. Правила.
4. Метод «разделяй и властвуй». Задача об умножении двух n -битовых чисел. Оценка сложности.
5. Основная теорема. Правила сумм, произведений, о времени выполнения операторов.
6. Алгоритмы сортировки: основные понятия, параметры оценки. Примеры видов внутренней сортировки.
7. Сортировка вставками: суть, пример и код.
8. Сортировка выбором: суть, пример и код.
9. Сортировка пузырьком (классическим): суть и код.
10. Быстрая сортировка: суть, пример и код.
11. Пирамидальная сортировка: суть, пример и код.
12. Шейкер-сортировка: суть, пример и код.
13. Корневая сортировка: суть, пример и код.
14. Сортировка Шелла: суть, пример и код.
15. Сортировка слиянием: суть, пример и код.
16. Пирамидальная сортировка: суть, пример и код.
17. Внешние сортировки. Суть, пример прямого слияния.
18. Внешние сортировки. Суть, пример естественного слияния.
19. Внешние сортировки. Суть, пример многофазной сортировки.
20. Динамические структуры данных.