# Правила выполнения лабораторных работ

Перед выполнением заданий лабораторной работы рекомендуется изучить теоретический материал по теме лабораторной работы и описание методов обработки данных на псевдокоде, используя конспекты лекционных занятий и литературу из списка.

Задания лабораторных работ выполняются только на языке программирования С/С++, среда программирования по выбору студента.
Изучаемые методы обработки данных рекомендуется программно реализовывать в виде отдельных функций (подпрограмм), массивы (последовательности) данных должны передаваться в подпрограммы в качестве параметров. Заполнение массивов данными, вывод их на экран, вычисление вспомогательных величин и пр. необходимо оформлять в виде отдельных подпрограмм.
При выполнении заданий следует обеспечить вывод на экран данных на всех шагах алгоритма. Программа должна иметь дружественный, интуитивно понятный интерфейс (меню пользователя, вывод подсказок, комментарии при вводе/выводе данных и т.д.).
В программе в ходе выполнения алгоритма необходимо предусмотреть подсчет количества сравнений С и количества пересылок М и вывести их экран.
Тестирование разработанной программы необходимо проводить для различных типов входных данных (случайный массив, упорядоченный массив в прямом и обратном порядке). После тестирования необходимо проанализировать полученные результаты, т.е. проверить соответствие полученных экспериментальным путем величин М и С теоретическим оценкам трудоемкости реализованных методов.

Для зачета по лабораторной работе студенту необходимо представить в отдельной папке

* Исходные тексты программ с подробными комментариями;
* Исполняемые файлы;
* Отчет по лабораторной работе.

Отчет должен включать в себя следующие разделы

* Формулировку задания
* Описание основных методов, используемых в лабораторной работе;
* Результаты работы программы (в виде файла или в виде скриншота);
* Анализ результатов.

## Лабораторная работа 1.

### Методы сортировки массивов с квадратичной трудоемкостью.

**Цель работы:** Освоить методы сортировки массивов с квадратичной трудоемкостью.

**Порядок выполнения работы:**

1. Разработать подпрограммы сортировки массива целых чисел методами прямого выбора, методом пузырьковой сортировки и методом шейкерной сортировки.
2. Правильность сортировки проверить путем подсчета контрольной суммы и числа серий в массиве (оформить в виде подпрограммы).

*Серией* называется неубывающая последовательность элементов массива максимальной длины.

**Пример:** в массиве 23145314 (23  145 3 14) содержится 4 серии

1. Составить таблицу следующего вида (данные получить экспериментально) для n=100, 200, 300, 400, 500. (n – количество элементов в массиве)

|  |  |  |
| --- | --- | --- |
| Размермассива | Мф+Сф м. Шелла | Мф+Сф пирам. (м. Хоара) |
| Убыв. | Случ. | Возр. | Убыв. | Случ. | Возр. |
| 100 |   |   |   |   |   |   |
| 200 |   |   |   |   |   |   |
| 300 |   |   |   |   |   |   |
| 400 |   |   |   |   |   |   |
| 500 |   |   |   |   |   |   |

1. Проанализировать полученные результаты, сравнить их с теоретическими оценками трудоемкости.

## Лабораторная работа 2.

### Быстрые методы сортировки массивов.

**Цель работы:** Освоить быстрые методы сортировки массивов

**Порядок выполнения работы:**

1. Разработать подпрограммы сортировки массива целых чисел методом Шелла и методом пирамидальной сортировки (или методом Хоара). Проверить правильность сортировки.
2. Исследовать трудоемкость метода Шелла для *n*=10, 100, …, 500, *n* – количество элементов в массиве. Определить последовательность шагов для предварительных сортировок по формуле Кнута. Построить таблицу и проанализировать полученные результаты:

|  |  |  |  |
| --- | --- | --- | --- |
| Длина массива | Количество шагов по формуле Кнута | Последовательность шагов по формуле Кнута | Мф+Сф Метод Шелла |
|   |   |   |   |

1. Составить таблицу следующего вида (данные получить экспериментально) для n= 100, 200, 300, 400, 500. (n – количество элементов в массиве)

|  |  |  |
| --- | --- | --- |
| Размермассива | Мф+Сф м. Шелла | Мф+Сф пирам. (м. Хоара) |
| Убыв. | Случ. | Возр. | Убыв. | Случ. | Возр. |
| 100 |   |   |   |   |   |   |
| 200 |   |   |   |   |   |   |
| 300 |   |   |   |   |   |   |
| 400 |   |   |   |   |   |   |
| 500 |   |   |   |   |   |   |

1. Проанализировать полученные результаты, сравнить их с теоретическими оценками трудоемкости.

## Лабораторная работа 3.

### Быстрые методы сортировки последовательностей.

**Цель работы:** Освоить быстрые методы сортировки последовательностей

**Порядок выполнения работы:**

1. Разработать подпрограммы сортировки последовательности целых чисел методом прямого слияния (или методом цифровой сортировки).
2. Разработать сервисные функции для работы со списками:
	* заполнение списка (стека) возрастающими числами;
	* заполнение списка (стека) убывающими числами;
	* заполнение списка (стека) случайными числами;
	* печать элементов списка;
	* подсчет контрольной суммы элементов списка;
	* подсчет количества серий в списке.
3. Составить таблицу следующего вида (данные получить экспериментально) для n= 100, 200, 300, 400, 500. (n – количество элементов в массиве)

|  |  |
| --- | --- |
| Длина списка | (Мф+Сф ) метод прямого слияния (цифровая сорт.) |
| Возрастающие числа | Убывающие числа | Случайные числа |
| 100 |   |   |   |
| 200 |   |   |   |
| 300 |   |   |   |
| 400 |   |   |   |
| 500 |   |   |   |

1. Проанализировать полученные результаты, сравнить их с теоретическими оценками трудоемкости. Сравнить полученные результаты с трудоемкостью метода прямого выбора и метода пирамидальной сортировки (использовать результаты предыдущих лабораторных работ).

## Лабораторная работа 4.

### Индексация и быстрый поиск.

**Цель работы:**Изучение методов построения индексных массивов и быстрого поиска с использованием индексации.

**Порядок выполнения работы:**

1. Написать программу «Телефонный справочник», которая обрабатывает данные об абонентах телефонной станции. Каждый абонент имеет имя, адрес, телефонный номер. В программе описать массив абонентов (назовем его справочник). В справочнике должно быть не менее 10 элементов, которые заполняются либо программно, либо считываются из файла.
2. Разработать подпрограмму создания в памяти компьютера индексного массива для упорядочивания справочника (воспользоваться любым методом сортировки, кроме пузырькового). Применить разработанную подпрограмму для создания индексных массивов упорядочивания (в прямом порядке) справочника по имени, адресу и номеру телефона абонента. Вывести на экран исходный массив абонентов и содержимое построенных индексных массивов.
3. Разработать подпрограмму вывода на экран упорядоченного справочника. Применить разработанную подпрограмму для вывода на экран справочника, упорядоченного по возрастанию имени абонента, адреса абонента и номера телефона абонента.
4. Разработать подпрограмму поиска в справочнике с использованием индексного массива. Применить разработанную подпрограмму для поиска абонента по имени, адресу и номеру телефона. Ключ для поиска вводить с клавиатуры.

## Лабораторная работа 5.

### Хэширование и поиск.

**Цель работы:**Изучение возможности хэширования данных для организации поиска.

**Порядок выполнения работы:**

1. Разработать подпрограмму хеширования массива целых чисел методом прямого связывания и подпрограмму поиска в хэш-таблице элемента по заданному ключу. Вывести на экран построенную хэш-таблицу.
2. Реализовать подпрограмму хеширования массива целых чисел методом открытой адресации. Для разрешения коллизий использовать линейные и квадратичные пробы. Вывести на экран заполненные хеш-таблицы для *m*=11 в виде

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер ячейки | 0 | 1 | 2 | 3 | … |   |   | … | *m*-1 |
| Число |   |   |   |   |   |   |   |   |   |

1. Подсчитать и сравнить количество коллизий при линейных и квадратичных пробах. Построить таблицу и проанализировать полученные результаты:

|  |  |  |
| --- | --- | --- |
| Размер хеш-таблицы | Количество исходных чисел | Количество коллизий |
| Линейные пробы | Квадратичные пробы |
| 13 | 15 |   |   |
| 29 | 30 |   |   |
| 43 | 45 |   |   |
| 67 | 70 |   |   |
| 83 | 85 |   |   |

1. Организовать поиск элемента с заданным ключом для метода открытой адресации (линейные и квадратичные пробы).