

Н.С. Толстова

Лабораторный практикум

Методы сортировки и поиска данных



Екатеринбург

2019

Оглавление

Введение	3
<i>Лабораторная работа 1. Простые алгоритмы внутренней сортировки</i>	5
<i>Лабораторная работа 2. Улучшенные алгоритмы внутренней сортировки данных</i>	8
<i>Лабораторная работа 3. Методы поиска данных</i>	10
<i>Лабораторная работа 4. Измерение времени работы алгоритмов сортировки</i>	12
Задачи для самостоятельного выполнения	17

Введение

Уважаемый преподаватель!

Лабораторные работы могут помочь в проведении занятий очной и заочной формы обучения.

Лабораторные работы играют большую роль в усвоении и закреплении изучаемого материала, а также в развитии у студентов практических навыков по использованию алгоритмов сортировки и поиска.

Цель лабораторно-практических работ: закрепление знаний и формирование умений по использованию алгоритмов сортировки и поиска.

Лабораторные работы предполагают решение следующих основных задач:

- приобретение навыков и приемов использования массивов, доступа к их элементам и преобразований массивов;
- практическое освоение принципов модульного программирования (использование процедур);
- развитие навыков самостоятельной работы.

В конце каждого блока лабораторных работ студенту может быть проведен тестовый контроль. После проведения всех лабораторных работ могут быть выполнены задания для самостоятельной работы, представленные в конце данного пособия.

Уважаемый студент!

Лабораторные работы данного пособия помогут самостоятельно изучить данный материал.

При выполнении лабораторных работ следует соблюдать следующие правила:

- текст, который никак не выделен, следует только читать;
- практические задания, отмеченные словом «Задание», следует обязательно и в полном объеме выполнять;

- дополнительные задания следует выполнять самостоятельно, лишь в случае полной неудачи следует обратиться к преподавателю;
- делать краткий конспект, что поможет ускорить усвоение материала и пройти тест, приведенный в конце каждой лабораторной работы.

После выполнения всех лабораторных работ необходимо выполнить задания для самостоятельной работы, представленные в конце данного пособия.

Лабораторная работа 1. Простые алгоритмы внутренней сортировки данных

1. Постановка учебно-практической задачи

Выполнение лабораторной работы направлено на освоение основных приемов использования массивов, методов доступа к элементам массивов, их реорганизации и модификации. В качестве практической проблемы, требующей решения, рассматривается известная задача сортировки (упорядочивания) массива в порядке возрастания (убывания) его элементов. При решении этой задачи требуется исходный массив, содержащий произвольные целые числа, преобразовать к виду, когда каждый элемент массива находится перед другим элементом этого массива, если его значение меньше (больше), чем значение сравниваемого элемента.

В данной лабораторной работе необходимо изучить ряд известных алгоритмов сортировки и создать комплекс программ, реализующий

- *метод сортировки выбором;*
- *метод сортировки простого обмена («пузырьковая сортировка»);*
- *метод сортировки прямого включения («простых вставок»).*

2. Учебно-методические цели работы

Данная задача представляет собой упрощенный учебно-ознакомительный вариант широко распространенной задачи сортировки, которая применяется в широком классе реальных задач, и умение и знание алгоритмов сортировки и их особенностей является необходимой частью образования специалиста по программному обеспечению. Наряду с ознакомлением с проблемой упорядочивания массивов и методами ее решения, выполнение работы направлено на достижение следующих учебно-методических целей:

- приобретение навыков и приемов использования массивов, доступа к их элементам и преобразований массивов;

- практическое освоение принципов модульного программирования (использование процедур).

Задание 1

Скопируйте в Ваш каталог файлы **Lab1_1.txt**, **Lab1_2.txt**, **Lab1_3.txt** из каталога **ISXOD**. Запустите среду Visual Studio.

Задание 2

Откройте файл **Сортировка выбором.avi** из папки **Видео** и посмотрите работу алгоритма сортировки выбором на примере.

Задание 3

Скопируйте текст из **Lab1_1.txt** в свой проект. Содержащаяся в нем программа демонстрирует сортировку методом выбора (выделения) по убыванию. Разберитесь в том, как работает эта программа. Выполните задание, написанное в конце программы (после **END**). **Продемонстрируйте все задания преподавателю.**

Задание 4

Модифицируйте программу из файла **Lab1_1.txt** так, чтобы сортировались только положительные и целые элементы массива. **Продемонстрируйте готовую программу преподавателю.**

Задание 5

Еще одна модификация: замените числовой массив строковым и отсортируйте его в алфавитном порядке. **Продемонстрируйте готовую программу преподавателю.**

Задание 6

Имеется список спортсменов, участвующих в соревнованиях по прыжкам в длину, и их результаты. Составить программу, которая позволит вывести список участников на экран дисплея в порядке занятых ими мест. **Продемонстрируйте готовую программу преподавателю.**

Задание 7

Откройте файлы **Сортировка вставкой.avi**, **Сортировка обменом.avi** из папки **Видео** и посмотрите работу алгоритмов сортировки вставкой и

обмена на примере.

Задание 8

Считайте с диска файлы с именем **Lab1_2.txt, Lab1_3.txt**. В этих файлах содержится заголовок программ, комментарии по выполнению, описание переменных, оператор начала и конца программы, операторы ввода и вывода данных. Вам следует из этой "болванки" составить программы, позволяющие продемонстрировать сортировку разными методами. Составьте предложенные программы в различных окнах среды Visual Studio.

Продемонстрируйте все готовые программы преподавателю.

Задание 9

Выйдите из среды Visual Studio. Скопируйте из Вашего каталога файлы **lab1_1.txt, lab1_2.txt, lab1_3.txt** на дискету.

Лабораторная работа 2. Улучшенные алгоритмы внутренней сортировки данных

1. Постановка учебно-практической задачи

Выполнение лабораторной работы направлено на освоение основных приемов использования массивов, методов доступа к элементам массивов, их реорганизации и модификации. В качестве практической проблемы, требующей решения, рассматривается известная задача сортировки (упорядочивания) массива в порядке возрастания (убывания) его элементов. При решении этой задачи требуется исходный массив, содержащий произвольные целые числа, преобразовать к виду, когда каждый элемент массива находится перед другим элементом этого массива, если его значение меньше (больше), чем значение сравниваемого элемента.

В данной лабораторной работе необходимо изучить ряд известных алгоритмов сортировки и создать комплекс программ, реализующий

- *метод быстрой сортировки;*
- *метод сортировки Шелла.*

2. Учебно-методические цели работы

Данная задача представляет собой упрощенный учебно-ознакомительный вариант широко распространенной задачи сортировки, которая применяется в широком классе реальных задач, и умение и знание алгоритмов сортировки и их особенностей является необходимой частью образования специалиста по программному обеспечению. Наряду с ознакомлением с проблемой упорядочивания массивов и методами ее решения, выполнение работы направлено на достижение следующих учебно-методических целей:

- приобретение навыков и приемов использования массивов, доступа к их элементам и преобразований массивов;
- практическое освоение принципов модульного программирования

(использование процедур).

Задание 1

Скопируйте в Ваш каталог файлы **Lab2_1.txt**, **Lab2_2.txt** из каталога **ISXOD**. Запустите среду Visual Studio.

Задание 2

Откройте файл **Сортировка Шелла.avi**, **Быстрая сортировка.avi** из папки **Видео** и посмотрите работу алгоритмов быстрой сортировки, Шелла на примере.

Задание 3

Считайте с диска файлы с именем **Lab2_1.txt**, **Lab2_2.txt**. В этих файлах содержится заголовок программ, комментарии по выполнению, описание переменных, оператор начала и конца программы, операторы ввода и вывода данных. Скопируйте содержимое файлов в свои проекты. Вам следует из этой "болванки" составить программы, позволяющие продемонстрировать сортировку разными методами. Составьте предложенные программы в различных окнах среды Visual Studio. **Продемонстрируйте все готовые программы преподавателю.**

Задание 4

Выйдите из среды Visual Studio. Скопируйте из Вашего каталога файлы **lab2_1.txt**, **lab2_2.txt** на дискету.

Лабораторная работа 3. Методы поиска данных

1. Постановка учебно-практической задачи

Выполнение лабораторной работы направлено на освоение основных приемов использования массивов, методов доступа к элементам массивов, их реорганизации и модификации. В качестве практической проблемы, требующей решения, рассматривается известная задача поиска в массиве элемента с заданным значением. Основной вопрос задачи поиска является нахождение в заданной совокупности данных элемента, обладающего заданным свойством.

В данной лабораторной работе необходимо изучить ряд известных методов поиска и создать комплекс программ, реализующий

- *метод линейного поиска элемента в массиве;*
- *метод двоичного поиска элемента в упорядоченном массиве.*

2. Учебно-методические цели работы

Данная задача представляет собой упрощенный учебно-ознакомительный вариант широко распространенной задачи поиска, которая применяется в широком классе реальных задач, и умение и знание алгоритмов поиска и их особенностей является необходимой частью образования специалиста по программному обеспечению. Наряду с ознакомлением с проблемой поиска элементов в массиве, выполнение работы направлено на достижение следующих учебно-методических целей:

- приобретение навыков и приемов использования массивов, доступа к их элементам и преобразований массивов;
- практическое освоение принципов модульного программирования (использование процедур).

Задание 1

Скопируйте в Ваш каталог файлы **Lab3_1.txt**, **Lab3_2.txt** из каталога **ISXOD**. Запустите среду Visual Studio.

Задание 2

Откройте файл **Линейный поиск.avi**, **Двоичный поиск.avi** из папки **Видео** и посмотрите работу алгоритмов линейного и двоичного поиска на примере.

Задание 3

Считайте с диска файлы с именем **Lab3_1.txt**, **Lab3_2.txt**. В этих файлах содержится заголовок программ, комментарии по выполнению, описание переменных, оператор начала и конца программы, операторы ввода и вывода данных. Вам следует из этой "болванки" составить программы, позволяющие продемонстрировать поиск элемента в массиве двумя методами. Составьте предложенные программы в различных окнах среды Visual Studio. Продемонстрируйте все готовые программы преподавателю.

Задание 4

Выйдите из среды Visual Studio. Скопируйте из Вашего каталога файлы **lab3_1.txt**, **lab3_2.txt** на дискету.

Лабораторная работа 4. Измерение времени работы алгоритмов сортировки

1. Постановка учебно-практической задачи

Выполнение лабораторной работы направлено на освоение основных приемов использования массивов, методов доступа к элементам массивов, их реорганизации и модификации. В качестве практической проблемы, требующей решения, рассматривается известная задача сортировки (упорядочивания) массива в порядке возрастания (убывания) его элементов. При решении этой задачи требуется исходный массив, содержащий произвольные целые числа, преобразовать к виду, когда каждый элемент массива находится перед другим элементом этого массива, если его значение меньше (больше), чем значение сравниваемого элемента.

В данной лабораторной работе необходимо создать программу, реализующую изученные алгоритмы в лабораторной работе 1 и 2:

- *метод сортировки выбором;*
- *метод сортировки простого обмена («пузырьковая сортировка»);*
- *метод сортировки прямого включения («простых вставок»);*
- *метод быстрой сортировки;*

Разрабатываемая программа должна обеспечивать:

- вывод на экран меню;
- ввод исходной информации;
- формирования массивов с большим числом элементов случайным образом;
- выбор метода сортировки или поиска;
- сортировку массива и поиска данных;
- печать результата;
- замеры времени выполнения сортировок массива.

Демонстрация работоспособности разработанных программных средств

должна обеспечивать два варианта контроля: контроль работоспособности каждого из методов и контроль временных характеристик всех реализованных методов.

2. Учебно-методические цели работы

Данная задача представляет собой упрощенный учебно-ознакомительный вариант широко распространенной задачи сортировки, которая применяется в широком классе реальных задач, умение и знание алгоритмов сортировки и их особенностей является необходимой частью образования специалиста по программному обеспечению. Наряду с ознакомлением с проблемой упорядочивания массивов и методами ее решения, выполнение работы направлено на достижение следующих учебно-методических целей:

- приобретение навыков и приемов использования массивов, доступа к их элементам и преобразований массивов;
- практическое освоение принципов модульного программирования (использование процедур);
- получения навыков сравнительного анализа эффективности разных алгоритмов.

3. Рекомендации по выполнению работы

Программа должна обеспечить сортировку массивов размером произвольной длины до 10000 элементов и выводить для контроля:

- при небольшом количестве элементов (например, менее 20) – неупорядоченный массив и массив после сортировки для каждого из предложенных алгоритмов;
- при значительном объеме данных (более 20) выводить время сортировки одного и того же массива для всех четырех предложенных алгоритмов.

Содержимое массива рекомендуется формировать с помощью

генератора псевдослучайных чисел, замеры времени производить средствами модулей System и Dos (Random, DateTime).

4. Рекомендации по организации программного комплекса

Разработку программы рекомендуется выполнить в виде программного комплекса, включающего:

- модули сортировки;
- модуль формирования массива;
- печать массива;
- модуль замера времени;
- программу, реализующую диалог с пользователем.

4.1. Состав разрабатываемых процедур и функций

Рекомендуется завести глобальные переменные и типы данных следующего вида:

Алгоритмы сортировок рекомендуется реализовать в виде процедур вида:

–procedure S_VKL (var a:massiv; n:integer);

{Процедура сортировки включением. Встроен счётчик времени работы. a-массив, n-число элементов в массиве}

–procedure S_VYB (var a:massiv; n:integer);

{Процедура сортировки выбором. Встроен счётчик времени работы. a-массив, n-число элементов в массиве}

–procedure S_PUZ (var a:massiv; n:integer);

{Процедура сортировки пузырьком. Встроен счётчик времени работы. a-массив, n-число элементов в массиве}

–procedure S_BIS (var a:massiv; low,hi:integer);

{Процедура быстрой сортировки массива. a-массив, low, hi – с какого по какой элементы сортировать}

–**procedure S_FAST;**

{Процедура сортирует массив быстрым способом и считает время его работы.}

–**procedure SHOW;**

{Процедура выводит отсортированный массив}

Основные действия программы:

- формирование нового массива;
- сортировка массива методом включения;
- сортировка массива методом выбора;
- сортировка массива методом пузырька;
- быстрая сортировка;
- сортировка одного и того же сформированного массива всеми видами сортировок, с печатью времени выполнения операций;
- выход из программы.

Задание 1

Скопируйте в Ваш каталог файл **Lab4_1.txt** из каталога **ISXOD**. Запустите среду Turbo Pascal.

Задание 2

Считайте с диска файл с именем **Lab4_1.txt**. В этом файле содержится заголовок программ, комментарии по выполнению, описание переменных, оператор начала и конца программы, операторы ввода и вывода данных, процедуры и меню программы. Вам следует вписать недостающие фрагменты. Продемонстрируйте готовую программу преподавателю.

Задание 3

Заполните таблицу, изобразите диаграмму и сделайте вывод о времени работы алгоритмов сортировки случайного массива в секундах для данного компьютера. Покажите преподавателю.

Кол-во элементов в массиве	Сортировка включением	Сортировка выбором	Сортировка пузырьком	Быстрая сортировка
N=500				
N=1000				
N=1500				
N=2000				
N=2500				
N=3000				
N=3500				
N=4000				
N=4500				
N=5000				
N=5500				
N=6000				
N=6500				
N=7000				
N=7500				
N=8000				
N=8500				
N=9000				
N=9500				
N=10000				

Задание 4

Выйдите из среды Visual Studio. Удалите из Вашего каталога файл **lab4_1.txt**.

Задачи для самостоятельного выполнения

1. Изменить решения в методах вставки, выбором и обменом так, чтобы осуществлялась сортировка:

- ✎ четных элементов массива;
- ✎ элементов, записанных на нечетных местах;
- ✎ отрицательных элементов массива и т.д.

2. Дан, например, массив 12 3 5 7 9 10. За один просмотр методом «пузырька» он становится отсортированным, остальные просмотры ничего не дают. Исключить лишние просмотры.

3. Объединить требования заданий 2 и 3 в единое целое. Этот метод называется «шейкер-сортировкой». Реализовать его. В сортировке простыми вставками убрать переменную x , т. е. внутренний цикл записать в виде: `While A[0] < A[j] Do`. Подсказка. Массив A необходимо сделать типа `Array[0..Nmax] Of Integer` и i -й элемент записывать на 0 место, так называемый прием «барьерного элемента».

4. Метод вставок. На момент вставки элемента с номером i элементы массива с номерами от 1 до $i-1$ отсортированы. Выберем из них средний элемент (или один из двух средних) и сравним его с элементом $A[i]$. Если $A[i]$ меньше этого элемента, то поиск места вставки следует продолжать в левой половине, иначе — в правой половине отсортированного массива. Эта схема получила название сортировки бинарными вставками. Она предложена Дж. Мочли в 1946 году и была одной из первых публикаций по методам компьютерной сортировки. Реализовать данную схему.

5. Пусть N является точным квадратом натурального числа, например, 3. Разделим массив на $\text{Sqrt}(N)$ групп по $\text{Sqrt}(N)$ элементов в каждой. Выберем максимальный элемент в каждой группе... Проще рассмотреть пример. Дан массив 7 10 3 5 15 9 6 12 8. При разбивке на группы — (7 10 3) (5 15 9) (6 12 8). Максимальные элементы 10 15 12. Максимальный из них 15, он во второй группе. Если оставшиеся элементы из второй группы, а это 5 и 9, меньше 10

и 12, то мы нашли сразу три элемента, записываемые на свои места. Если нет, то заменяем максимальные элементы элементами из группы. Этот метод предложен Э. Х. Фрэндом в 1956 году и получил название метода квадратичного выбора. Количество сравнений имеет порядок $O(N*\sqrt{N})$. Реализовать метод.

6. Алгоритм фон Неймана. Упорядочить массив $a_1 a_2 \dots, a_n$ по убыванию с помощью алгоритма сортировки слияниями:

✎ каждая пара соседних элементов сливается в одну группу из двух элементов (последняя группа может состоять из одного элемента);

✎ каждая пара соседних двухэлементных групп сливается в одну четырехэлементную группу и т.д.

✎ При каждом слиянии новая укрупненная группа упорядочивается.

7. Сортировка подсчетом. Выходной массив заполняется значениями – 1. Затем для каждого элемента определяется его место в выходном массиве путем подсчета количества элементов строго меньших данного. Естественно, что все одинаковые элементы попадают на одну позицию, за которой следует ряд значений – 1. После этого оставшиеся в выходном массиве позиции со значением – 1 заполняются копией предыдущего значения.

8. «Хитрая» сортировка. Из массива путем однократного просмотра выбирается последовательность элементов, расположенных в порядке возрастания, переносится в выходной массив и заменяется во входном на – 1. Затем оставшиеся элементы включаются в полученную упорядоченную последовательность методом погружения», когда очередной элемент путем ряда обменов «погружается» до требуемой позиции в уже упорядоченную часть массива.

9. Дан линейный массив целых чисел. Проверьте, является ли он упорядоченным по убыванию.

10. Дан целочисленный линейный массив из 10 элементов. Найдите наименьшее число K элементов, которое нужно исключить из

последовательности $A[1], A[2], \dots, A[10]$, чтобы осталась возрастающая последовательность.

11. Ввести массив a_1, a_2, \dots, a_{15} . Расположить ненулевые элементы по убыванию.

12. Ввести массив x_1, x_2, \dots, x_{20} . Элементы, на нечетных местах, расположить в порядке возрастания, а на четных в порядке убывания.

13. Ввести массив a_1, a_2, \dots, a_{15} . Требуется упорядочить его по возрастанию абсолютных значений элементов.

14. Ввести массив x_1, x_2, \dots, x_{20} . Требуется расположить отрицательные элементы в порядке убывания.

15. Упорядочить первые n элементов данного ряда в порядке возрастания. Напечатать эти элементы в порядке убывания.

16. Найти второй по величине (после наименьшего) элемент данного ряда.

17. Найти так называемую медиану ряда, т. е. такой его элемент, который больше любого из одной половины элементов и меньше любого из другой (если число элементов ряда четно, следует взять среднее значение из двух значений, обладающих указанным свойством).

18. В целочисленном массиве найти наибольшее число одинаковых элементов.

19. Дано n целых чисел. Сколько чисел лежит между данными a и b ?

20. Отсортировать положительные элементы массива с помощью простого выбора.

21. Даны список футбольных команд высшей лиги России и количество очков, набранных каждой командой в чемпионате России. Известно, что нет команд с равным числом очков. Распечатать список призеров.

22. В неупорядоченном массиве могут быть совпадающие элементы. Из каждой группы одинаковых элементов оставить только один, удалив остальные и «поджав» массив к его началу.

23. Дан массив из строк (например, фамилий). Отсортировать его по

алфавиту и написать процедуру вставки новой фамилии после заданной так, чтобы алфавитный порядок не нарушился. Предусмотреть ситуацию, когда массив заполнен «до отказа» и вставка нового элемента невозможна.

24. Имеется железнодорожное расписание, содержащее номер рейса поезда, времена отправления и прибытия и станцию прибытия. Организовать поиск номера поезда, время отправления и прибытия, если задана станция.

25. Дана последовательность, элементы которой есть целые двузначные числа. Упорядочить последовательность по возрастанию сумм цифр соответствующих элементов.

26. Дана последовательность, элементы которой есть целые двузначные числа. Упорядочить последовательность по убыванию произведений цифр соответствующих элементов.

27. Дана последовательность, расположить ее элементы, кратные 3, по убыванию.