

Министерство науки и высшего образования Российской Федерации

САНКТ-ПЕТЕРБУРГСКИЙ  
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО

---

*С. А. Фёдоров*

АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ  
КУРСОВАЯ РАБОТА

Учебное пособие

Санкт-Петербург  
2024

УДК 004.42

**Фёдоров С. А. Алгоритмы и структуры данных. Курсовая работа** : учеб. пособие. – СПб.: СПбПУ, 2023. – 22 с.

Учебное пособие посвящено курсовой работе дисциплины «Алгоритмы и структуры данных».

Студентам предлагается для выполнения курсовая работа для изучения и сопоставления различных структур данных: массивам строк, массивам символов, структурам массивов, массивам структур, однонаправленным и двунаправленным динамическим спискам и деревьям.

Учебное пособие предназначено для студентов, обучающихся по направлению подготовки бакалавров 09.03.04 «Программная инженерия», в частности, по онлайн-программе 09.03.04\_03 «Разработка программного обеспечения».

© Фёдоров С. А., 2024

© Санкт-Петербургский государственный политехнический университет Петра Великого, 2024

# Содержание

Введение.....	3
§ 1. Курсовая работа.....	4
1.1 Общая часть задания.....	4
1.2 Указание к проектам с массивами структур и со структурой массивов.....	6
§ 2. Индивидуальные задания.....	8
§ 3. Требования к оформлению и содержанию курсовой работы.....	20
3.1 Задание на курсовую работу.....	20
3.2 Требования к оформлению.....	20
3.3 Требования к содержанию.....	20
Библиографический список.....	23

## **Введение**

В данном пособии студентам, обучающимся на разработчиков программного обеспечения, предлагается для выполнения курсовая работа для изучения и сопоставления различных структур данных:

- массивам строк;
- массивам символов;
- структур массивов;
- массивам структур;
- однонаправленным динамическим спискам;
- двунаправленным динамическим спискам;
- деревьям.

При выполнении курсовой работы, создавая своё *произведение*, необходимо следовать дисциплине программирования. Каждая часть работы оформляется как отдельный программный проект. При сдаче принимается исходный код ПО и результаты его работы.

## **§ 1. Курсовая работа**

### **1.1 Общая часть задания**

1. Одна и та же информация из входного файла вводится по-разному в оперативную память с целью освоения работы с различными структурами данных — задание выполняется в виде отдельных программных проектов/

	Проекты курсовой работы							
Средства	1	2а	2б	3	4	5	6а	6б
регулярное программирование	+	+	+	+	+	+	+	+
модули	+	+	+	+	+	+	+	+
массивы строк	+	–	–	–	–	–	–	–
массивы символов	–	+	+	–	–	–	–	–
массив структур	–	–	–	+	–	?	–	–
структура массивов	–	–	–	–	+	?	–	–
файлы записей	–	–	–	+	+	+	–	–
хвостовая рекурсия	–	–	–	–	–	+	+	+
динамические однонаправленные списки <sup>1</sup>	–	–	–	–	–	–	+	–
рекурсивный размещаемый тип								+

2. Необходимо прочитать список из не менее чем 12 строк. Данные в одной строке имеют заданный формат и отделяются друг от друга дополнительным пробелом.
3. В проекте 5 структура выбирается и обосновывается студентом по критериям: сплошные данные, регулярный доступ к памяти, векторизация и выравнивание адресов.
4. В проектах 2а и 2б в двумерных массивах используется

---

1 Каждая строка исходного форматированного файла рассматривается как элемент списка, файл с неизвестным числом строк читается до конца. Длину списка можно запомнить при чтении, если только она используется при сортировке.

разное назначение индексов: `Array(i, j)` и `Array(j, i)`.

## 1.2 Указание к проектам с массивами структур и со структурой массивов

Для хранения и обработки данных необходимо выбрать *структуру массивов* или *массив структур*, руководствуясь требованиями к сплошным данным, регулярному доступу к памяти и векторизации.

В опорном варианте № 1.3 приведён пример использования *массива структур*.

При работе с *массивом структур* неформатированный файл записей формируется отдельной процедурой из форматированного файла *запись за записью*, а затем из него происходит чтение *одной записи* — всего массива структур.

При использовании компиляторов `ifort` или `ifx` при записи и чтении неформатированного (двоичного) файла остаётся использовать вместо прямого доступа потоковый (`stream`) без указания *размера записи и номера записи*: записывать последовательно структуры, а затем прочитать массив структур разом.

При работе со *структурой массивов* в № 1.4 работать с неформатированным файлом в следующем порядке:

1. Открыть форматированный файл. Прочитать из него все данные в отдельные *локальные массивы* — одним оператором чтения с неявным циклом (прямо как в проекте 1.1). Закрыть форматированный файл.
2. Открыть неформатированный файл в *потоковом режиме* (`stream`) без указания *размера записи*. Записать в него прочитанные локальные массивы — одним оператором

записи write, указав в списке этого оператора все локальные массивы, которые были прочитаны на предыдущем шаге (без указания номера записи, разумеется). Закрыть неформатированный файл.

3. Для чтения структуры массивов в отдельной процедуре открыть неформатированный файл *в потоковом режиме* (stream), прочитать из него *структуру массивов* – одним оператором чтения с указанием только экземпляра структуры массивов. Порядок полей-массивов в структуре должен быть тем же, что при их записи в неформатированный файл.



## § 2. Индивидуальные задания

1. Дан список группы в виде:

ФАМИЛИЯ	ПОЛ	РЕЗУЛЬТАТЫ СЕССИИ
15 симв.	1 симв.	4 симв.

Пример входного файла:

Иванов	М	4455
Петрова	Ж	3554

Определить лидеров среди мужчин и женщин по успеваемости и их средний балл (у мужчин и женщин отдельно). Пример выходного файла

Лидеры по успеваемости:

Иванов	4455	ср. балл	4.50
Петрова	3554	ср. балл	4.30

Средний балл среди мужчин: 4.2

Средний балл среди женщин: 4.1

**Указание.** Сформировать сперва массивы для мужчин и женщин, а потом с ними работать. Для обработки данных обоих полов повторно использовать одни и те же процедуры.

2. Дан список группы в виде:

ФАМИЛИЯ	И. О.	ПОЛ	ПРОПИСКА	СРЕДНИЙ_БАЛЛ
15 симв.	5 симв.	1 симв.	1 симв.	4 симв.

Пример входного файла (в графе прописки буква П стоит у петербурцев, С — у гостей Санкт-Петербурга):

Иванов	И. И.	М	П	4.35
--------	-------	---	---	------

Отсортировать по убыванию среднего балла по отдельности списки петербуржцев и гостей Санкт-Петербурга.

Пример выходного файла:

Петербуржцы:

Иванов И. И. М 4.35

Барабашкин И. И. М 4.32

Гости города:

Петров И. И. М 4.35

Петрыкин И. И. М 4.32

3. Дан список группы в виде:

ФАМИЛИЯ И. О. ПОЛ ГОД РОЖДЕНИЯ

15 сив. 5 симв. 1 симв. 4 симв.

Пример входного файла:

Иванов И. И. М 1995

Определить средний возраст юношей в группе. Примеры выходного файла:

Средний возраст юношей в группе: 21 год

Средний возраст юношей в группе: 22 года

Средний возраст юношей в группе: 25 лет

4. Дан список группы в виде:

ФАМИЛИЯ ГОД РОЖДЕНИЯ СЛУЖБА В АРМИИ ПРОПИСКА ПОЛ

15 симв. 4 симв. 3 симв. 1 симв. 1 симв.

Пример входного файла (в графе прописки буква П стоит у петербуржцев, с — у гостей Санкт-Петербурга):

Иванов 1992 нет П М

Петров 1993 да С М

Отсортировать в алфавитном порядке по отдельности

списки петербуржцев и гостей Санкт-Петербурга, служивших в армии. Пример выходного файла:

Служившие петербуржцы:

Барабашкин            1992 М

Служившие гости города:

Петров                1993 М

5. Дан список сотрудников научно-исследовательской лаборатории в виде:

ФАМИЛИЯ    ДОЛЖНОСТЬ

15 симв.    15 симв.

Пример входного файла:

Иванов                техник

Определить число одинаковых должностей. Пример выходного файла:

ведущий инженер - 2

старший инженер - 3

инженер            - 8

техник              - 2

**Указание.** Завести логический массив длиной N. При обработке должности очередного сотрудника сразу найти все такие должности и пометить в логическом массиве, где они встречаются. Должность следующего сотрудника обрабатывать, только если она прежде не встречалась (смотреть в логический массив). Использовать Count с маской, Апу. См. упражнение 5.16, 7.25.

6. Дан список товаров с указанием цены вида:

НАИМЕНОВАНИЕ ВЕЩИ    ЦЕНА            ВАЛЮТА

20 симв.                    5 симв.    5 симв.

Пример входного файла:

Стол	2500 руб.
------	-----------

Определить три самых дорогих товара. Пример выходного файла:

Самые дорогие товары:

- |          |             |
|----------|-------------|
| 1. Диван | - 12000 руб |
| 2. Стул  | - 9000 руб. |
| 3. Тумба | - 6000 руб. |

**Указание.** Не следует сортировать весь список – нужно найти только первые три элемента. Например, можно три раза последовательно находить максимальную стоимость товара, но не учитывая уже найденные на предыдущем шаге (MaxLoc с mask).

В проекте 1.6 необходимо сформировать три ссылки на самые дорогие товары.

7. Дан список группы в виде:

ФАМИЛИЯ	И. О.
15 сив.	5 симв.

Пример входного файла:

Иванов	И. И.
--------	-------

Отсортировать список по алфавиту с учетом инициалов методом сравнения «один со всеми». Пример выходного файла:

Иванов	И. А.
Иванов	И. И.
Иванова	И. И.

**Указание.** При проверке равенства массивов символов

использовать встроенную функцию All (вторая вариация).

**8. Дан список владельцев телефонов в виде:**

ФАМИЛИЯ    ТЕЛЕФОН  
15 симв.    10 симв.

Пример входного файла:

Петров	9111634576
Фёдоров	9111635687

Отсортировать этот список в порядке убывания номеров телефонов, используя метод вставок. Пример выходного файла:

Фёдоров	9111635687
Петров	9111634576

**9. Дан список группы в виде:**

ФАМИЛИЯ                    И. О.    ГОД РОЖДЕНИЯ  
15 сив.                    5 симв.    4 симв.

Пример входного файла:

Иванов	И. И. 1994
Петров	Д. Т. 1992

Выделить первого по алфавиту и самого молодого. Пример выходного файла:

Первый по алфавиту:  
Иванов                    И. И. 1994  
Самый молодой:  
Петров                    Д. Т. 1992

**10. Дан список товаров в виде:**

НАИМЕНОВАНИЕ    ЦЕНА            ВАЛЮТА    ПРОЦЕНТ ИЗНОСА

10 симв.            5 симв. 5 симв. 3 симв.

Пример входного файла:

Стол	2500 руб. 45
------	--------------

Используя процент износа, сформировать продажную комиссионную цену. Пример выходного файла:

Комиссионная цена вещей:

Стол	1375 руб.
------	-----------

**11.** Дан список сотрудников научно-исследовательской лаборатории в виде:

ФАМИЛИЯ	ДОЛЖНОСТЬ
---------	-----------

15 симв. 15 симв.

Пример входного файла:

Иванов	техник
--------	--------

Отсортировать список в порядке повышения должности от «техника» до «вед. инженера». Пример выходного файла:

Иванов	техник
Петров	старший инженер

**Указание.** Для сравнения должностей завести логическую функцию, которая будет возвращать результат сравнения двух должностей. Должности: техник, инженер, старший инженер, ведущий инженер, главный инженер. Должности можно поместить в массив для организации сравнения с помощью FindLoc.

**12.** Дан список группы в виде:

ФАМИЛИЯ	И. О.	ГОД РОЖДЕНИЯ	ПРОПИСКА	ПОЛ
---------	-------	--------------	----------	-----

15 симв. 5 симв. 4 симв. 1 симв. 1 симв.

Пример входного файла (в графе прописки буква П стоит у петербуржцев, С — у гостей Санкт-Петербурга):

Иванов	И. И. 1995 П М
Петрова	Х. Л. 1994 С Ж

Выделить из них трёх наиболее молодых петербуржцев мужчин. Пример выходного файла:

Иванов	И. И. 1995
Галкин	В. И. 1997
Потапкин	Е. З. 1997

**Указание.** Не следует сортировать весь список – нужно найти только первые три элемента. Например, можно три раза последовательно находить максимальный год рождения, но не учитывая уже найденные на предыдущем шаге (MaxLoc с mask).

В проекте 1.6 необходимо сформировать три ссылки на самых молодых.

**13.** Дан список группы в виде:

ФАМИЛИЯ	ИМЯ	ОТЧЕСТВО
15 симв.	10 симв.	15 симв.

Пример входного файла:

Безруков	Сергей	Викторович
Лебедев	Пётр	Станиславович

Определить число встречающихся имен. Пример выходного файла:

Сергей	- 2
Пётр	- 1

**Указание.** Независимо по итерациям составить маску из уникальных имён (неявный цикл). Определить количество уникальных имён (Count). Выделить переменные для хранения уникальных имён и для хранения числа повторений каждого из них. Независимо по итерациям (do concurrent по числу уникальных имён): записывать в нужные позиции переменных уникальное имя и число его повторений (Count над сечением).

**14.** Дан список владельцев телефонов в виде:

ФАМИЛИЯ    ТЕЛЕФОН  
15 симв.    10 симв.

Пример входного файла:

Петров	9111634576
Фёдоров	9111635687
Петров	9111634573

Найти строку с первой по алфавиту фамилией и с наименьшим номером телефона. Пример выходного файла:

Петров	9111634573
--------	------------

**15.** Дан список группы в виде:

ФАМИЛИЯ    ИМЯ            ОТЧЕСТВО  
15 симв.    10 симв.    15 симв.

Пример входного файла:

Безруков	Сергей	Викторович
Лебедев	Пётр	Станиславович
Глухих	Сергей	Алексеевич

Удалить из списка всех, у которых имена совпадают, кроме первого. Пример выходного файла:



Безруков  
Лебедев

Сергей  
Пётр

Викторович  
Станиславович

**Указание.** Работать с исходным массивом. При каждом нахождении повторяющегося имени будет неэффективным тут же удалять его и сдвигать все остальные элементы в массиве, потому что дальше это имя опять может встретиться.

Будет неэффективным также нахождение сперва всех мест в массиве, где находится текущее имя, а потом удаление его оттуда сдвигом остальных элементов, потому что среди остальных имён могут оказаться другие повторяющиеся, которые всё равно в итоге придётся удалять.

Необходимо сперва сформировать массив-маску для всех уникальных имён. При этом не стоит проверять имя, если оно уже встречалось. Использовать подход `uniq(i) = uniq(i) .and. ...` Затем по маске упаковать исходные массивы (Pack).

При работе с динамическим списком обработку проводить двумя рекурсивными процедурами – в первой рекурсии берётся очередное имя, во второй оно удаляется дальше по списку.

**16.** Дан список группы в виде:

ФАМИЛИЯ	ГОД РОЖДЕНИЯ	ПОЛ
15 симв.	4 симв.	1 симв.

Пример входного файла:

Иванов	1985	М
Петрова	1983	Ж

Найти самого пожилого мужчину и самую молодую жен-

щину. Пример выходного файла:

Самый пожилой мужчина:

Иванов 1985

Самая молодая женщина:

Петрова 1983 ж

**Указание.** Предусмотреть, что самых может быть несколько.

**17.** Дан список группы в виде:

ФАМИЛИЯ ПОЛ РЕЗУЛЬТАТЫ СЕССИИ

15 симв. 1 симв. 4 симв.

Пример входного файла:

Иванов М 4455

Петрова Ж 3554

Отсортировать по убыванию среднего балла мужчин и женщин по отдельности. Пример выходного файла:

Мужчины:

Иванов М 4455 4.50

Женщины:

Петрова Ж 3554 4.25

**18.** Дан список группы в виде:

ФАМИЛИЯ И. О. ПОЛ ГОД РОЖДЕНИЯ ПОЛ

15 симв. 5 симв. 1 симв. 1 симв.

Пример входного файла:

Иванов И. Л. М 1985

Петрова Д. О. Ж 1983

Сформировать отсортированные по убыванию возраста списки мужчин и женщин. Использовать для сортировки

метод “выбором”. Пример выходного файла:

Мужчины:

Иванов                      И. Л. М 1985

Женщины:

Петрова                    Д. О. Ж 1983

**19.** Дан список сотрудников лаборатории в виде:

ФАМИЛИЯ    ПРОФЕССИЯ

15 симв.    10 симв.

Пример входного файла:

Иванов                    повар

Петрова                    фармацевт

Сформировать список профессий с указанием их количества. Пример выходного файла:

повар                    - 1

фармацевт              - 1

**Указание.** См. указание к 13.

**20.** В первом входном файле дан список товаров в виде:

НАИМЕНОВАНИЕ    ЦЕНА            ВАЛЮТА

10 симв.            5 симв.    5 симв.

Пример первого входного файла:

стол                      9500    руб.

бра                        1000    руб.

диван                    12000   руб.

люстра                    500     руб.

Во втором входном файле дан список обновлённых товаров в виде:

ЦЕНА      ВАЛЮТА  
5 симв. 5 симв.

В конце списка приводится массив из значений Т и F (истина и ложь), указывающий у каких товаров в исходном списке была изменена цена. Пример второго входного файла:

1200 руб.  
450 руб.

F T F T

Сформировать обновлённый список товаров. Пример выходного файла:

стол	9500 руб.
бра	1200 руб.
диван	12000 руб.
люстра	450 руб.

**Указание.** В 1.1 (и местами далее) использовать Merge, Unpack.

### **§ 3. Требования к оформлению и содержанию курсовой работы**

#### **3.1 Задание на курсовую работу**

Курсовая работа выполняется в течение семестра, а её код сдаётся как л/р №№ 1.1 – 1.6.

#### **3.2 Требования к оформлению**

Отчёт готовится в LaTeX. Титульный лист берётся с л/р по Введению в проф. деятельность. Отчёт по курсовой – не лабораторной – работе. Требования к *оформлению* отчёта такие же, как для ВКР [1].

#### **3.3 Требования к содержанию**

Титульный лист. Содержание. Задание на работу (берёте задание из л/р № 1). Введение. Глава 1. ... 2. Глава 2. ... Выводы. Приложение (по желанию).

1. Введение. Цель работы — выбор структуры данных для решения поставленной задачи на современных микроархитектурах. Задачи:

1. Реализовать задание с использованием массивов строк.
2. Реализовать задание с использованием массивов символов.
3. Реализовать задание с использованием массивов структур.
4. Реализовать задание с использованием структур массивов.

5. Реализовать задание с использованием массивов структур или структур массивов (на выбор) и с использованием хвостовой рекурсии при обработке данных.

6. Реализовать задание с использованием динамического списка.

6. Провести анализ на регулярный доступ к памяти.

7. Провести анализ на векторизацию кода.

8. Провести сравнительный анализ реализаций.

2. «Глава 1. Реализация и анализ применения различных структур данных». Описываете последовательно все проекты с разными структурами — по параграфу на каждый: массив строк, массив символов (а и б), структура массивов, массив структур, структура массивов/массив структур с хвостовой рекурсией (на выбор с обоснованием), динамический список, динамический список на основе рекурсивного размещаемого типа:

- \* приводите объявление структуры данных (для конкретной вариации)

- \* приводите основные операторы **обработки данных**

- \* указываете где регулярный доступ, а где нет

- \* указываете где код был векторизован, а где потенциально векторизуем (например, требуется выравнивание данных)

- \*\* приводите обоснование выбора назначения индексов в массиве символов (Names(LEN, AMOUNT) или Names(AMOUNT, LEN))

3. «Глава 2. Сравнение реализаций»

Приводите сравнительную таблицу реализаций по критериям: сплошные данные, регулярный доступ, векторизация, по-

тенциальная векторизация, а также по показателям: время работы *участка кода по обработке данных*, сложность *участка кода по обработке данных*, эффективность *участка кода по обработке данных*.

Сложность участка кода брать как число строк кода. Эффективность участка кода брать как отношение производительности участка кода к сложности этого участка кода. Производительность участка кода брать как обратное время работы этого участка кода. *Для удобства эффективность стоит умножить на  $10^6$  и взять целую часть.*

Проверяете на *относительно больших данных* – время работы должно быть порядка нескольких секунд (массивы должны быть динамическими).

Делаете вывод о том, какая структура данных предпочтительна для решения данной задачи на современных микроархитектурах.

5. Выводы. Приводите решённые задачи. Указываете предпочтительную структуру данных. Делаете вывод о том, что цель была достигнута.

## ***Библиографический список***

1. Правила оформления студенческих текстовых документов: дипломных (курсовых) проектов (работ), отчётов и рефератов [Электронный ресурс] : методические рекомендации / В.И. Маслов, Л.Н. Шуткевич ; Санкт-Петербургский государственный политехнический университет, Институт металлургии, машиностроения и транспорта, Кафедра конструкторско-технологических инноваций .— Электрон. текстовые дан. (1 файл : 519 Кб) .— Санкт-Петербург, 2013 .— Загл. с титул. экрана . — Свободный доступ из сети Интернет (чтение, печать) .— Текстовый документ. — Adobe Acrobat Reader 7.0 .— <URL:<http://elib.spbstu.ru/dl/2/2976.pdf>>.