

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМЕНИ М. В. ЛОМОНОСОВА  
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И КИБЕРНЕТИКИ

## ОТЧЕТ ПО ЗАДАНИЮ №1

### «Методы сортировки»

Вариант 2 / 1 / 1 / 3

Выполнил:  
студент 101 группы  
Ядерцов Д. А.

Преподаватель:  
Кузьменкова Е. А.

Москва  
2020

# Содержание

Постановка задачи	2
Результаты экспериментов	2
Комментарии к проекту	3
Структура программы и спецификация функций	3
Отладка программы, тестирование функций	5
Список цитируемой литературы	6

## Постановка задачи

Цель задания экспериментально сравнить две сортировки, которые сортируют массив 64-разрядных целых чисел по неубыванию, двумя методами: методом «Пузырька» и методом Шелла.

## Результаты экспериментов

Из таблицы 1 мы можем заметить, что число сравнений у сортировки методом «Пузырек» возрастает квадратично, это теоретически доказывается в книге [1]102-104 стр. А сортировка методом Шелла возрастает примерно как  $n^{1.4}$  (таблица 2), что примерно равно теоретической оценке [1]107-108 стр. Также можно заметить, что как в [1] число обменов не превышает число сравнений и тоже будет расти с такой же скоростью, как и сравнения.

n	Параметр	Номер сгенерированного массива				Среднее значение
		1	2	3	4	
10	Сравнения	45	45	45	45	45
	Перемещения	0	44	18	25	21.75
100	Сравнения	4950	4950	4950	4950	4950
	Перемещения	0	4839	2555	2371	2441.25
1000	Сравнения	499500	499500	499500	499500	499500
	Перемещения	0	486389	236899	242246	241383.5
10000	Сравнения	49995000	49995000	49995000	49995000	49995000
	Перемещения	0	48880360	24519783	24276089	24419058

Таблица 1: Результаты работы сортировки методом «Пузырек»

n	Параметр	Номер сгенерированного массива				Среднее значение
		1	2	3	4	
10	Сравнения	22	27	29	32	27.5
	Перемещения	0	12	13	16	10.25
100	Сравнения	503	649	855	916	730.75
	Перемещения	0	241	400	457	274.5
1000	Сравнения	8006	11124	13629	14289	11762
	Перемещения	0	4108	6129	6804	4260.25
10000	Сравнения	120005	163798	242209	234790	190200.5
	Перемещения	0	53780	127227	119726	75183.25

Таблица 2: Результаты работы сортировки методом Шелла

## Комментарии к проекту

В этом проекте реализованы почти все сортировки(кроме пирамидальной), закомментировав и раскомментировав несколько определенных строчек, то можно поменять с какими типами данных работать(за это отвечает `typ`) и в каком порядке сортировать, также сразу происходит сбор всей статистики по всем реализованным сортировкам и все сохраняется в соответствующие файлы в папке «Stats». Я реализовал проект в нескольких файлах, имеющих окончание «.h»(библиотека). Это было сделано для того, чтобы облегчить разработку, отладку и сделать код более понятным(название заголовочного файла говорит суть всех функций, написанных в этом файле). Заголовочные файлы:

- *headers.h* - файл в котором содержится тип чисел, с которыми мы работаем, и подключается библиотека для работы с `rand()`.
- *generate.h* - файл, в котором содержатся генераторы массивов.
- *checker.h* - в этом файле находится функция проверки
- *sort.h* - файл, в котором находятся все сортировки

## Структура программы и спецификация функций

Из файла *generate.h*:

- *int comp(const void\*, const void\*)* - компаратор для библиотечной быстрой сортировки
- *int min(int, int)* - возвращает минимум из двух элементов
- *typ ost(int n, long long P)* - возвращает случайное целое число(произведение n случайных чисел по модулю P)
- *typ fost(int a, int b)* - возвращает случайное вещественное число(произведение a случайных чисел делится на произведение b случайных чисел)
- *void generate(int size, typ\* a)* - создает массив случайных чисел
- *void reverse(int size, typ\* a)* - разворачивает массив
- *void certaingenerate(int size, typ\* a, int mode)* - оболочка *generate()*, чтобы можно было генерировать массивы с определенными условиями

Из файла *checker.h*:

- *int check(int size, typ\* a, typ\* b)* - проверяет на полное соответствие двух массивов

Из файла *sort.h*:

- *void swap(typ\*, typ\*)* - меняет в памяти местами два элемента

- *int cmp(typ, typ)* - компаратор для функций сортировки
- *void BubbleSort(int size, typ\* a)* - сортировка методом «Пузырек»
- *void SelectionSort(int size, typ\* a)* - сортировка методом выбора
- *void ShellSort(int size, typ\* a)* - сортировка методом Шелла
- *void QuickSort(typ\* a, long long leftpoint, long long rightpoint)* - быстрая сортировка(сортировка Хоара) рекурсивная реализация
- *void QuickCall(int size, typ\* a)* - оболочка над *QuickSort()*, чтобы было удобнее вызывать

Из файла *main.c*:

- *void view(int size, typ\* a)* - вывод массива в файл
- *void show(int size, FILE\* outputstream)* - заполнение таблицы собранными данными
- *void init(FILE\* outputstream)* - оформление шапки таблицы
- *void process(void (\*func)(int, typ\*), char\* sortname)* - функция, которая для определенной сортировки(передается входным параметром): генерирует массив, сортирует массив, собирает статистику по этой сортировке.
- *void Test(void (\*func)(int, typ\*), char\* sortname)* - функция, которая проверяет правильность работы сортировки с помощью библиотечной быстрой сортировки.
- *int main(void)* - запускает *process()* и/или *Test()*

## Отладка программы, тестирование функций

Для тестирования сортировки, достаточно вызвать функция ***Test(a, b)***, где a - функция сортировки(BubbleSort, ShellSort и тд), b - название сортировки в двойных кавычках("SelectionSort" и тд), параметр b для вывода в консоль. Тестирование происходит полным сопоставлением отсортированного массива с помощью встроенной функции и выбранной нами.

## Список литературы

- [1] Вирт Н. Алгоритмы и структуры данных. — М.: Мир, 1989.