|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования |
| **«МИРЭА – Российский технологический университет»** |
| **РТУ МИРЭА** |
|  |

|  |  |
| --- | --- |
| **Отчет по выполнению практического задания № 3** | |
| **Тема:** | |
| **«Определение эффективного алгоритма сортировки на основе эмпирического и асимптотического методов анализа»** | |
| Дисциплина: «Структуры и алгоритмы обработки данных» | |
|  | Выполнил студент: Чахуриди К.А. |
|  | Группа: ИКБО-74-23 |

Москва – 2024

СОДЕРЖАНИЕ

[1 ЦЕЛЬ 4](#_gjdgxs)

[2 ЗАДАНИЕ №1 5](#_30j0zll)

[2.1 Формулировка задачи 5](#_1fob9te)

[2.2 Описание выполнения и блок-схема алгоритма сортировки Шелла со смещениями Д.Кнута первым способом 6](#_2et92p0)

[2.3 Определение ситуаций лучшего, среднего и худшего случая и функции роста времени работы алгоритма сортировки Шелла со смещениями Д.Кнута первым способом 8](#_1t3h5sf)

2.4 Реализация алгоритма на языке C++ и проведение тестирования 8

[2.4.1 Реализация алгоритма сортировки Шелла со смещениями Д.Кнута первым способом сортировки на языке C++ 8](#_4d34og8)

[2.4.2 Тестирование алгоритма сортировки Шелла со смещениями Д.Кнута первым способом сортировки на языке C++ 10](#_2s8eyo1)

[2.5 Описание выполнения и блок-схема алгоритма быстрой сортировки (Хоара) 10](#_vdf352t4nrv)

[2.6 Определение ситуаций лучшего, среднего и худшего случая и функции роста времени работы алгоритма быстрой сортировки (Хоара) 12](#_lnxbz9)

[2.7 Реализация алгоритма на языке C++ и проведение тестирования 13](#_35nkun2)

[2.7.1 Реализация алгоритма быстрой сортировки (Хоара) на языке C++ 13](#_1ksv4uv)

[2.7.2 Тестирование 14](#_44sinio)

[2.8 Сортировка простыми вставками 15](#_3j2qqm3)

[2.9 Сравнение трёх алгоритмов на графике 15](#_1y810tw)

[2.10 Тестирование программы алгоритма сортировки Шелла со смещением Д.Кнута первым способом и быстрой сортировки(Хоара) в лучшем и худшем случае 17](#_4i7ojhp)

[2.10.1 Тестирование программы алгоритма сортировки Шелла со смещением Д.Кнута первым способом в худшем случае 17](#_2xcytpi)

2.10.2 Тестирование программы алгоритма сортировки Шелла со смещением Д.Кнута первым способом в лучшем случае 18

[2.10.3 Тестирование программы алгоритма быстрой сортировки(Хоара) в худшем случае 19](#_qsh70q)

[2.10.4 Тестирование программы алгоритма быстрой сортировки(Хоара) в лучшем случае 20](#_3as4poj)

[2.10.5 Заполнение таблиц для алгоритмов сортировок 21](#_1ci93xb)

[2.11 Вывод по заданию №1 22](#_147n2zr)

[3 ЗАДАНИЕ №2 23](#_23ckvvd)

[3.1 Формулировка задачи 23](#_ihv636)

[3.2 Формулы функции роста алгоритма сортировки простой вставкой в худшем и лучшем случае 23](#_32hioqz)

3.3 Асимптотическая оценка вычислительной сложности простого алгоритма сортировки вставкой 24

[3.4 Графическое представление функции роста и полученных асимптотических оценок сверху и снизу 24](#_1hmsyys)

[3.5 Справочная информация о вычислительной сложности алгоритмов сортировки Шелла со смещением Д.Кнута первым способом и быстрой сортировкой(Хоара) 25](#_41mghml)

[3.5.1 Справочная информация о вычислительной сложности алгоритма сортировки Шелла со смещением Д.Кнута первым способом 25](#_hvoudxzff51k)

[3.5.2 Справочная информация о вычислительной сложности алгоритма быстрой сортировки (Хоара) 26](#_67me1jqi1jww)

[3.6 Таблица асимптотической сложности трёх алгоритмов 26](#_2grqrue)

[3.7 Выводы по заданию №2 26](#_3fwokq0)

[4 ВЫВОДЫ 28](#_2u6wntf)

[5 ЛИТЕРАТУРА 29](#_19c6y18)

# 1 ЦЕЛЬ

Получить навыки по анализу вычислительной сложности алгоритмов сортировки и определению наиболее эффективного алгоритма.

# 2 ЗАДАНИЕ №1

## 2.1 Формулировка задачи

Вариант 12, в списке 28. Сортировка Шелла со смещением Д.Кнута первым способом, быстрая сортировка(Хоара).

Эмпирическая оценка эффективности алгоритмов.

1. Разработать алгоритм Шелла со смещениями Д. Кнута первым способом, реализовать код на языке С++. Сформировать таблицу 1.1 результатов эмпирической оценки сложности сортировки по формату табл. 1 для массива, заполненного случайными числами.

2. Определить ёмкостную сложность алгоритма Шелла со смещениями Д. Кнута первым способом.

3. Разработать алгоритм простого слияния, реализовать код на языке С++. Сформировать таблицу 1.2 результатов эмпирической оценки сортировки по формату табл. 1 для массива, заполненного случайными числами.

4. Определить ёмкостную сложность алгоритма быстрой сортировки (Хоара).

5. Добавьте в отчёт данные по работе любого из алгоритмов простой сортировки в среднем случае, полученные в предыдущей практической работе (в отчёте – таблица 1.3).

6. Представить на общем сравнительном графике зависимости Тп(n)=Cф+Mф для трёх анализируемых алгоритмов. График должен быть подписан, на нём – обозначены оси.

7. На основе сравнения полученных данных определите наиболее эффективный из алгоритмов в среднем случае (отдельно для небольших массивов при n до 1000 и для больших массивов с n>1000).

8. Провести дополнительные прогоны программ ускоренной и быстрой сортировок на массивах, отсортированных а) строго в убывающем и б) строго возрастающем порядке значений элементов. Заполнить по этим данным соответствующие таблицы 1.4, 1.5 для каждого алгоритма по формату табл. 1.

9. Сделайте вывод о зависимости (или независимости) алгоритмов сортировок от исходной упорядоченности массива на основе результатов, представленных в таблицах.

## 2.2 Описание выполнения и блок-схема алгоритма сортировки Шелла со смещениями Д.Кнута первым способом

Алгоритм сортировки Шелла со смещениями Д. Кнута (Shell's increments by Knuth) применяет последовательность округленных значений (3k-1) в порядке убывания, где (k) - индекс величины последовательности.

Шаги алгоритма:

1. Начинаем с выбора шага (h) равным наибольшему уменьшенному значению последовательности (3k -1), которое меньше длины списка.
2. Применяем сортировку вставкой в подсписках, каждый из которых состоит из элементов списка, отстоящих на расстоянии (h) друг от друга.
3. Уменьшаем шаг (h) до следующего наименьшего значения последовательности (3k - 1) и повторяем шаг 2.
4. Продолжаем уменьшать шаг (h) и применять сортировку вставкой до тех пор, пока не достигнем шага (h = 1). В этом случае выполняем обычную сортировку вставкой.

Используя этот подход, алгоритм сортировки Шелла со смещениями Д. Кнута позволяет улучшить эффективность сортировки на больших массивах данных по сравнению с обычной сортировкой вставкой.

Реализация данного описания выполнения алгоритма представлена в виде блок-схемы (рис.1,2).

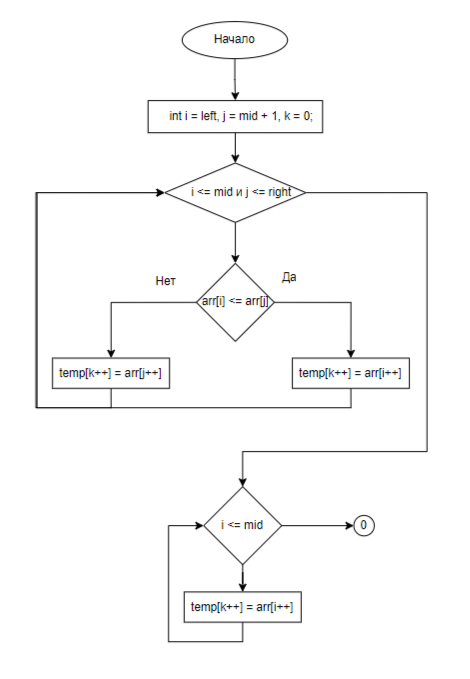


Рисунок 1 – Блок-схема алгоритма сортировки Шелла

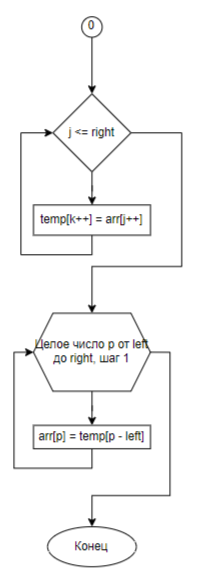


Рисунок 2 – Блок-схема алгоритма сортировки Шелла

## 2.3 Определение ситуаций лучшего, среднего и худшего случая и функции роста времени работы алгоритма сортировки Шелла со смещениями Д.Кнута первым способом

В лучшем случае массив уже отсортирован по возрастанию. Тогда количество операций сравнений и перемещений будет составлять O(n logn).

В среднем случае массив заполнен случайными числами. Тогда количество операций сравнений и перемещений будет составлять O(n log n2).

В худшем случае массив отсортирован по убыванию. Тогда количество операций сравнений и перемещений будет составлять O(n log n2).

Функции роста времени: лучший случай: O(n logn), худший случай:O(n log n2).

## 2.4 Реализация алгоритма на языке C++ и проведение тестирования

### **2.4.1 Реализация алгоритма сортировки Шелла со смещениями Д.Кнута первым способом сортировки на языке C++**

Напишем программу, которая реализует алгоритм сортировки Шелла со смещением первым способом на языке C++(рис.3,4).

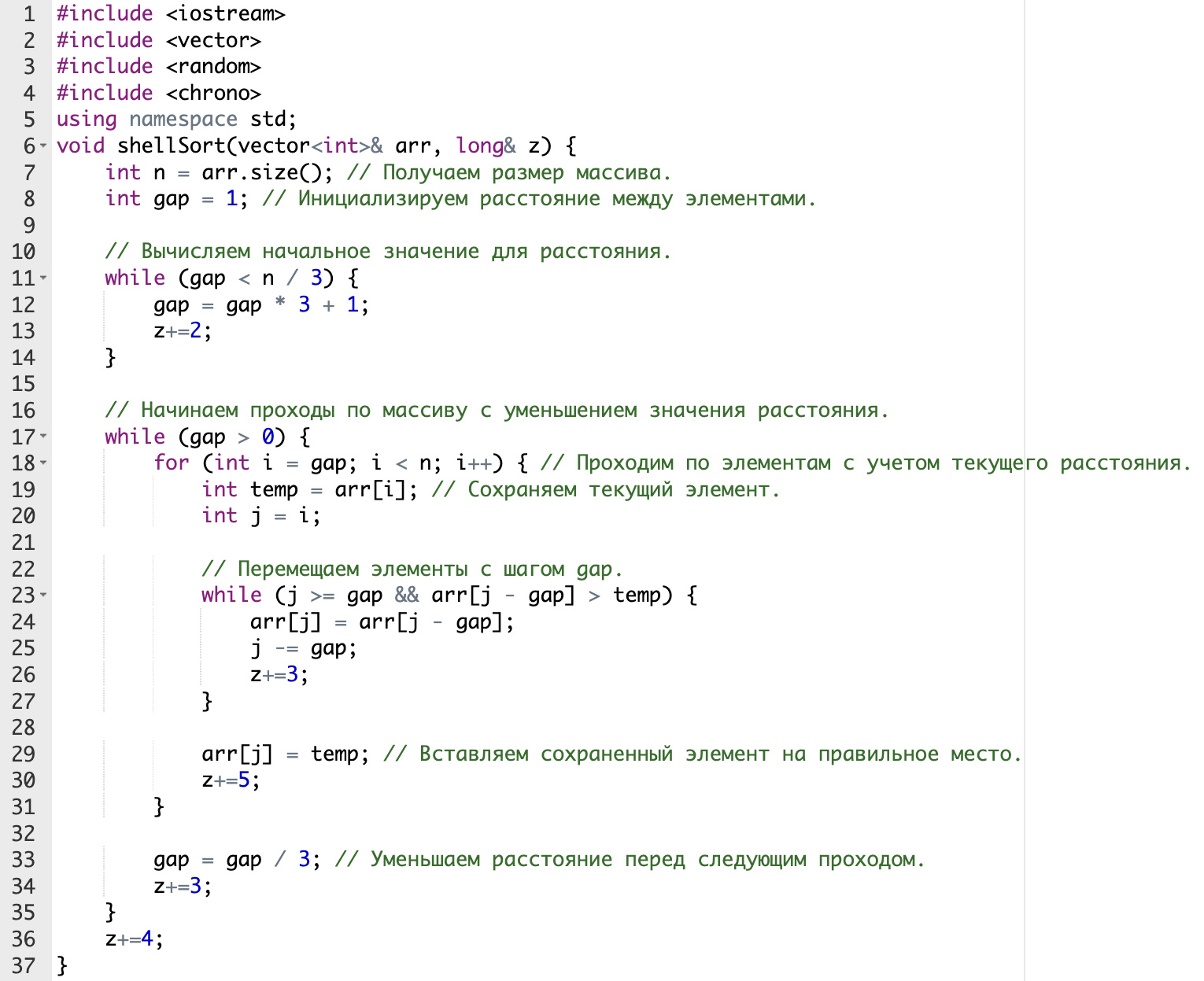


Рисунок 3 – Программа алгоритма

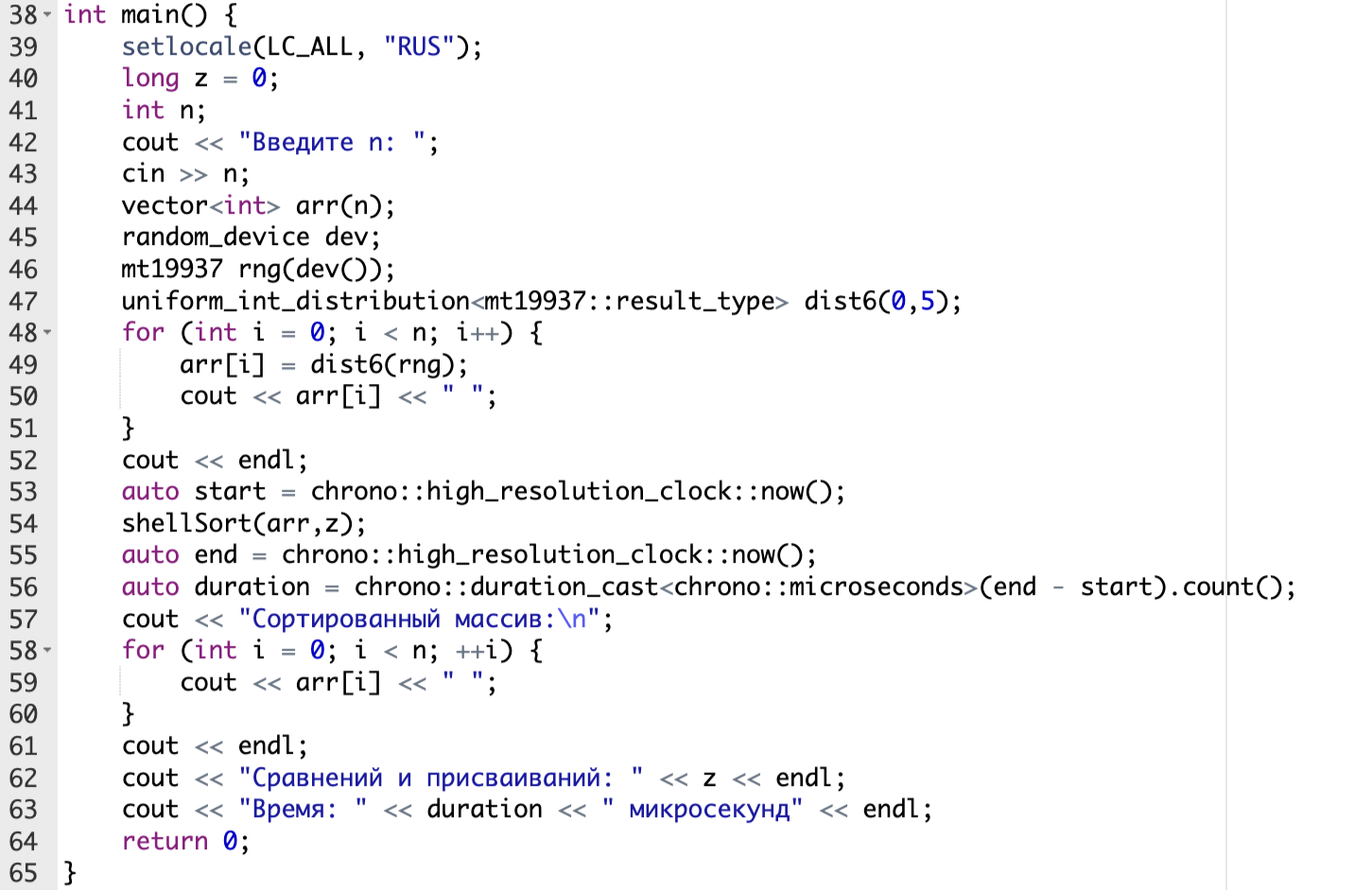


Рисунок 4 – Функция main

### **2.4.2 Тестирование алгоритма сортировки Шелла со смещениями Д.Кнута первым способом сортировки на языке C++**

Проведём тестирование алгоритма и предоставим результаты на рисунке 5 и таблице 1.1.

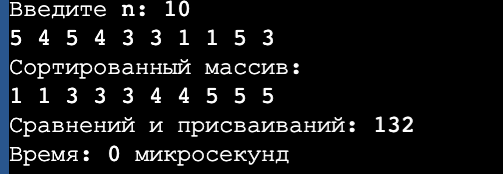


Рисунок 5 - Тестирование программы при n=10

Таблица 1.1. Сводная таблица результатов

|  |  |  |
| --- | --- | --- |
| **n** | **T(n), мс** | **Тп(n)=Cф+Mф** |
| 100 | 0.009 | 2374 |
| 1000 | 0.092 | 34142 |
| 10000 | 1.451 | 451640 |
| 100000 | 17.347 | 5655030 |
| 1000000 | 144.378 | 67454614 |

## 2.5 Описание выполнения и блок-схема алгоритма быстрой сортировки (Хоара)

Алгоритм быстрой сортировки, также известный как сортировка Хоара, является одним из самых эффективных алгоритмов сортировки. Он работает следующим образом:

Выбирается опорный элемент из массива. Это может быть любой элемент, например, средний элемент.

Все остальные элементы массива сравниваются с опорным элементом, и элементы меньше опорного перемещаются влево от него, а элементы больше опорного перемещаются вправо от него.

После этого опорный элемент будет расположен на своем финальном месте в отсортированном массиве.

Процедура разделения массива на две части выполняется рекурсивно для левой и правой частей массива, пока в каждой части будет содержаться только один элемент или пустой массив.

В итоге получается отсортированный массив.

Реализация данного описания выполнения алгоритма представлена в виде блок-схемы (рис.6).

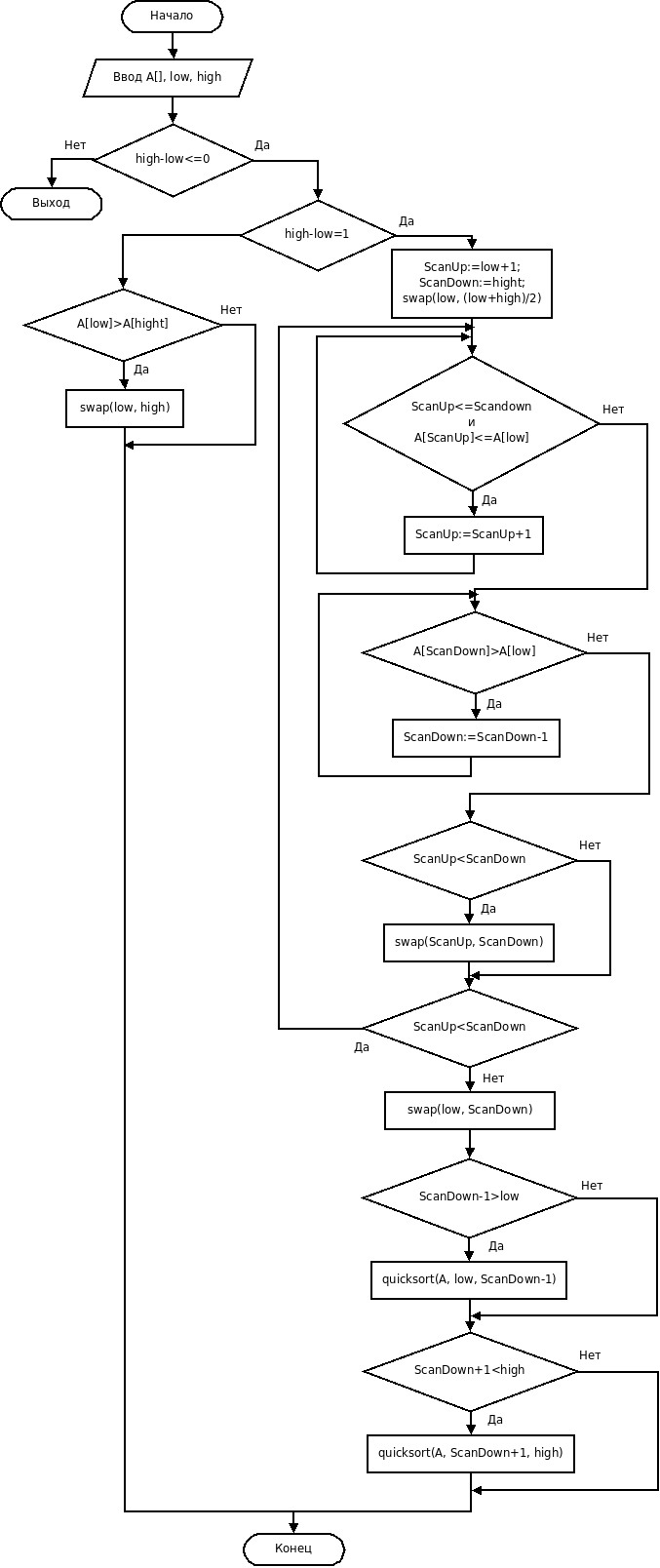


Рисунок 6 – Блок-схема алгоритма быстрой сортировки (Хоара)

## 2.6 Определение ситуаций лучшего, среднего и худшего случая и функции роста времени работы алгоритма быстрой сортировки (Хоара)

а. Идеальный сценарий - когда массив уже отсортирован, минимизируя количество операций до O(n log n). В среднем случае, когда числа в массиве случайные, сложность алгоритма также O(n log n). В худшем случае, если массив отсортирован в обратном порядке, количество операций возрастает до O(n2).

б. На основе функций роста времени: лучший случай - O(n log n), худший случай - O(n2).

## 2.7 Реализация алгоритма на языке C++ и проведение тестирования

### **2.7.1 Реализация алгоритма быстрой сортировки (Хоара) на языке C++**

Напишем программу, которая реализует алгоритм быстрой сортировка(Хоара) на языке C++(рис.7,8).



Рисунок 7 – Программа алгоритма

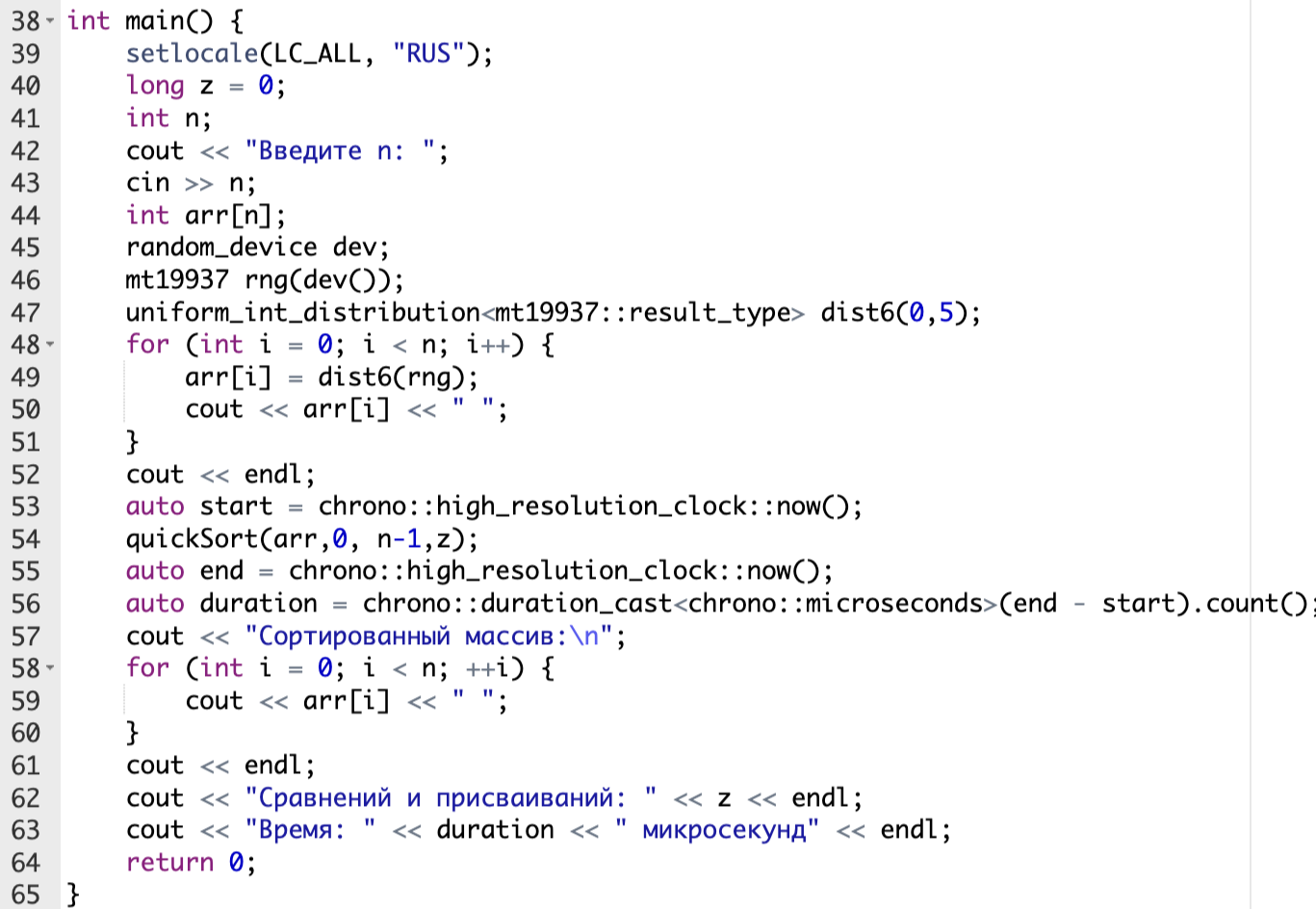


Рисунок 8 – Функция main

### **2.7.2 Тестирование**

Проведём тестирование алгоритма и предоставим результаты на рисунке 9 и таблице 1.1.

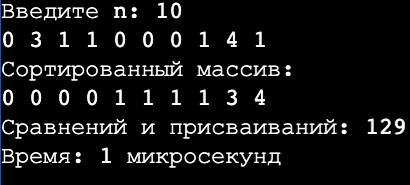


Рисунок 9 - Тестирование программы

Таблица 1.2. Сводная таблица результатов

|  |  |  |
| --- | --- | --- |
| **n** | **T(n), мс** | **Тп(n)=Cф+Mф** |
| 100 | 0.019 | 2068 |
| 1000 | 0.105 | 28853 |
| 10000 | 1.026 | 357234 |
| 100000 | 15.034 | 4243615 |
| 1000000 | 413.34 | 53542683 |

## 2.8 Сортировка простыми вставками

Добавим из предыдущей работы таблицу результатов тестирования простой сортировки обменом в среднем случае(табл.1.3).

Таблица 1.3. Сводная таблица результатов

|  |  |  |
| --- | --- | --- |
| **n** | **T(n), мс** | **Тп(n)=Cф+Mф** |
| 100 | 0.018 | 7144 |
| 1000 | 1.211 | 715681 |
| 10000 | 103.987 | 67338229 |
| 100000 | 10965.65 | 6749003338 |
| 1000000 | 1045738.24 | 67454963849 |

## 2.9 Сравнение трёх алгоритмов на графике

Графики функций роста для алгоритмов сортировки быстрой сортировки(Хоара), Шелла со смещением Д.Кнута и простыми вставками позволят провести сравнение. Построим график функции роста Тп от размера массива n. Построим два графика. Первый будет построен на значениях до 1000(рис.10), а второй от 10000 и до 1000000(рис.11).

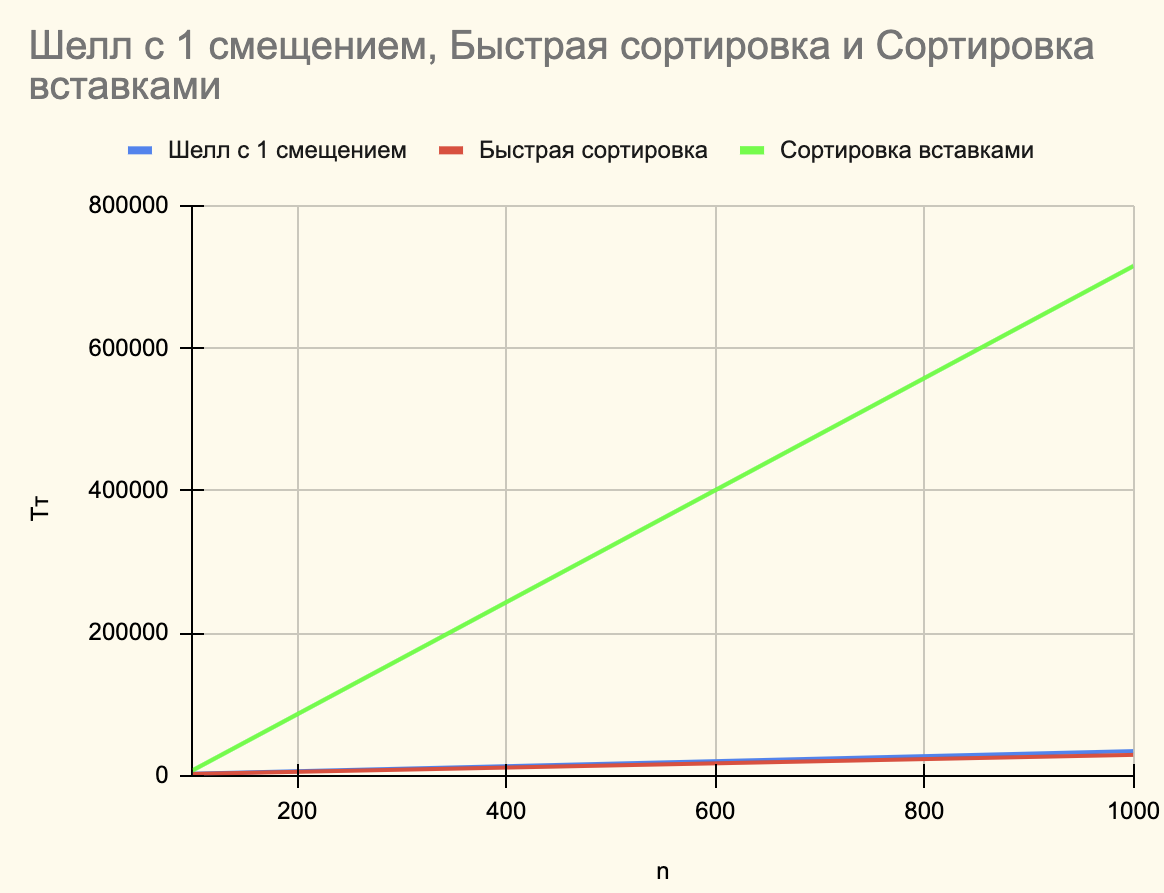


Рисунок 10 - График сравнения трёх сортировок в среднем случае при n до 1000

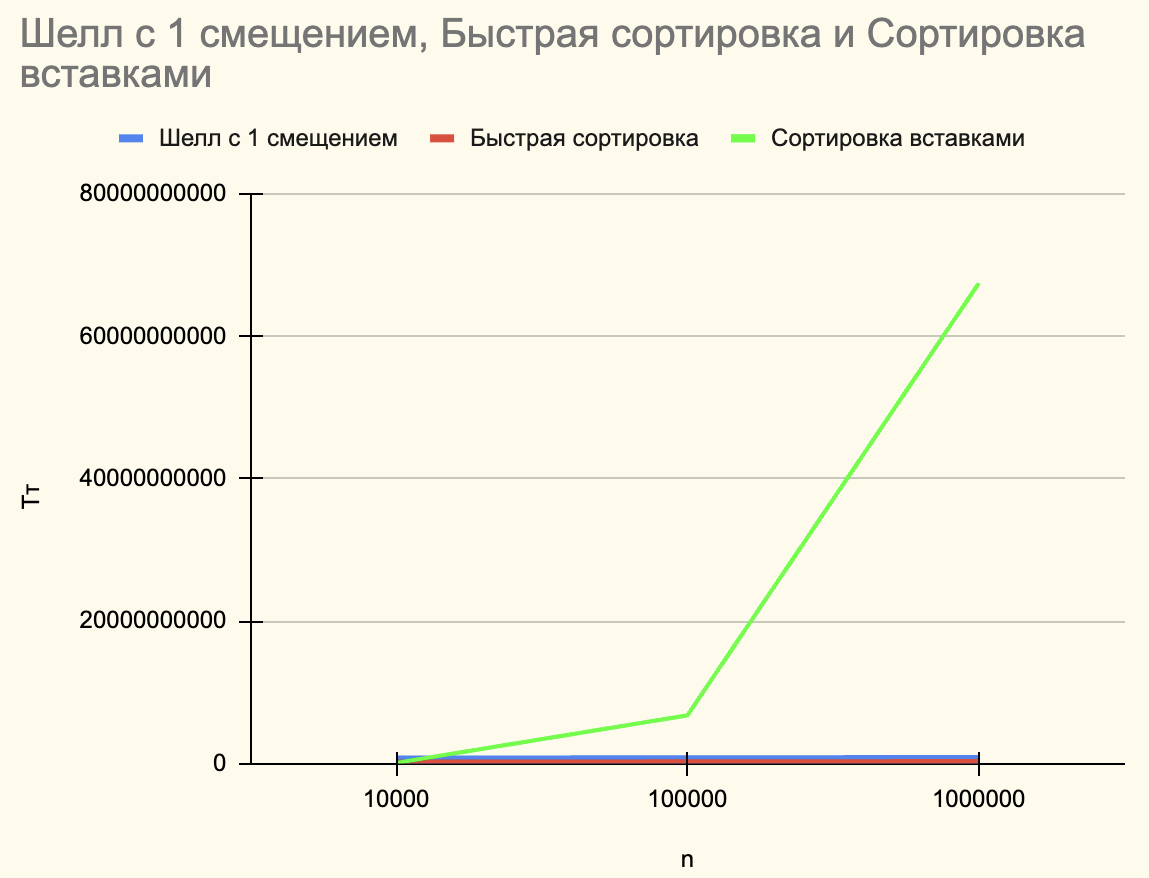


Рисунок 11 - График сравнения трёх сортировок в среднем случае при n от 10000 до 1000000

Анализ графиков показывает, что в среднем случае алгоритм сортировки простой вставкой является наименее эффективным, следом идет алгоритм быстрой сортировки(Хоара), а самым эффективным оказывается алгоритм сортировки Шелла со смещением Д.Кнута первым способом.

## 2.10 Тестирование программы алгоритма сортировки Шелла со смещением Д.Кнута первым способом и быстрой сортировки(Хоара) в лучшем и худшем случае

### **2.10.1 Тестирование программы алгоритма сортировки Шелла со смещением Д.Кнута первым способом в худшем случае**

Напишем программу, которая реализует алгоритм сортировки Шелла со смещением Д.Кнута первым способом с отсортированными значениями массива по убыванию на языке C++(рис.12,13).

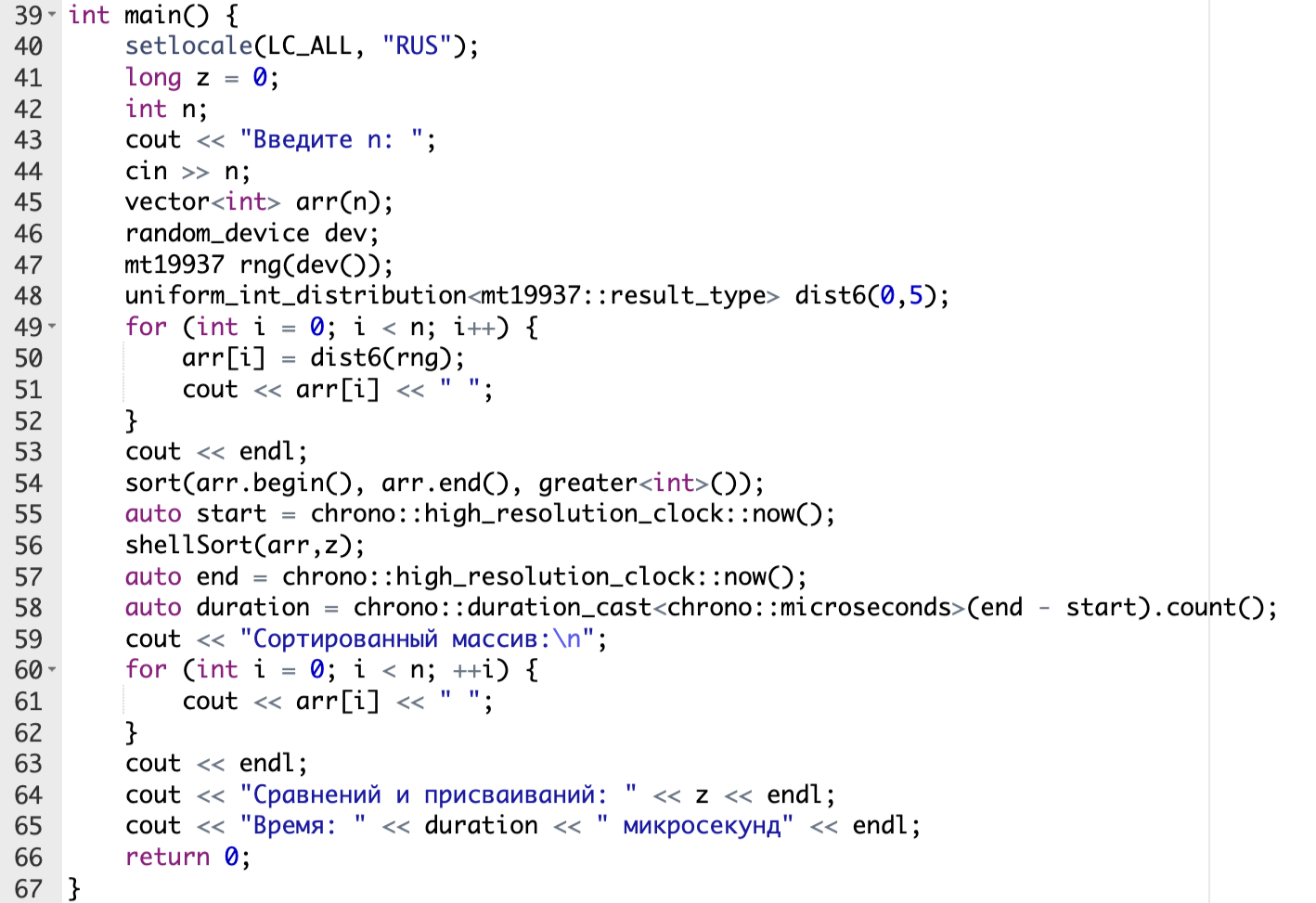


Рисунок 12 – Функция main



Рисунок 13 – Тестирования программы

### **2.10.2 Тестирование программы алгоритма сортировки Шелла со смещением Д.Кнута первым способом в лучшем случае**

Напишем программу, которая реализует алгоритм сортировки Шелла со смещением Д.Кнута первым способом с отсортированными значениями массива по возрастанию на языке C++(рис.14,15).

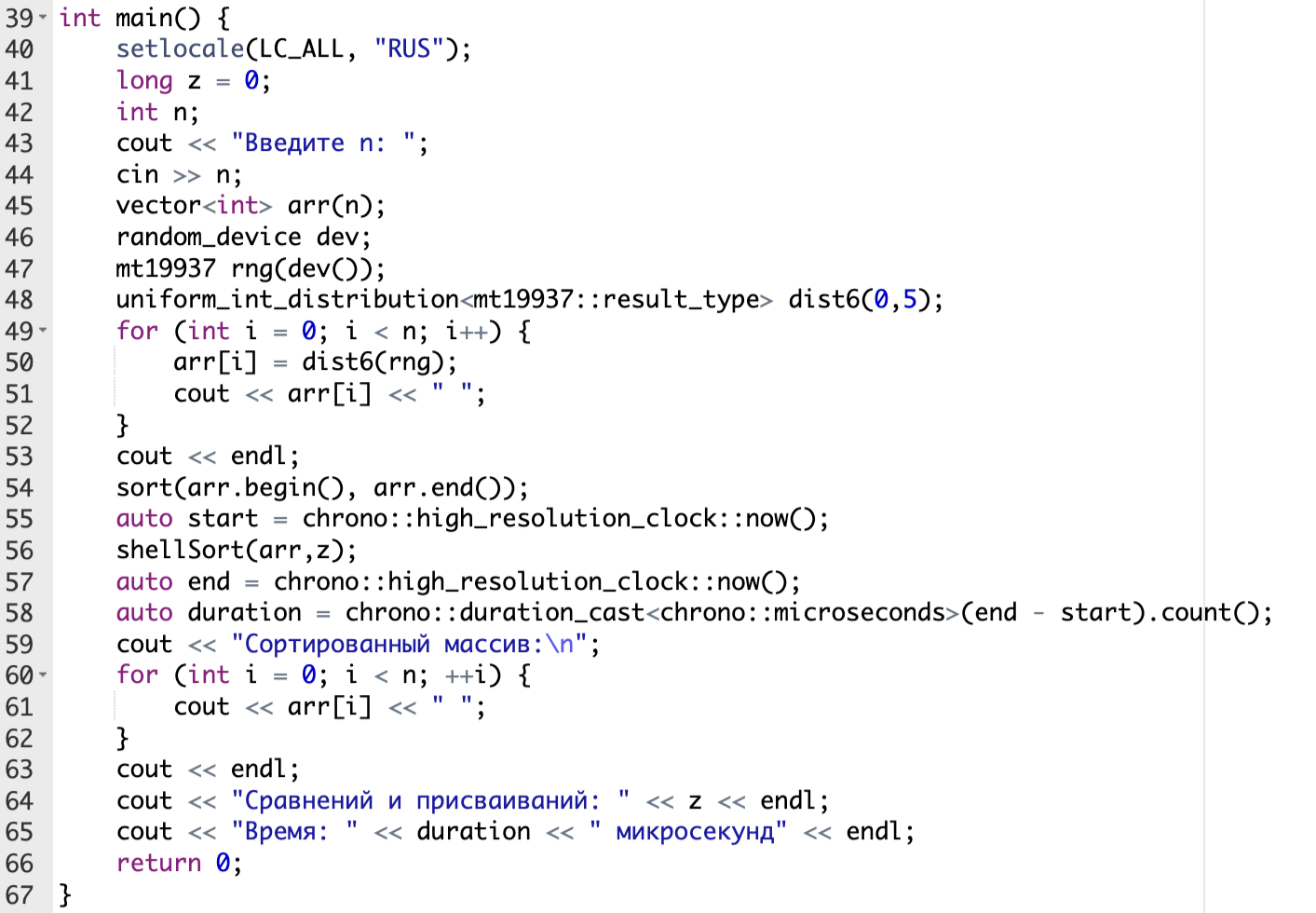


Рисунок 14 – Функция main



Рисунок 15 – Тестирования программы

### **2.10.3 Тестирование программы алгоритма быстрой сортировки(Хоара) в худшем случае**

Напишем программу, которая реализует алгоритм быстрой сортировка(Хоара) с отсортированными значениями массива по убыванию на языке C++(рис.16,17).

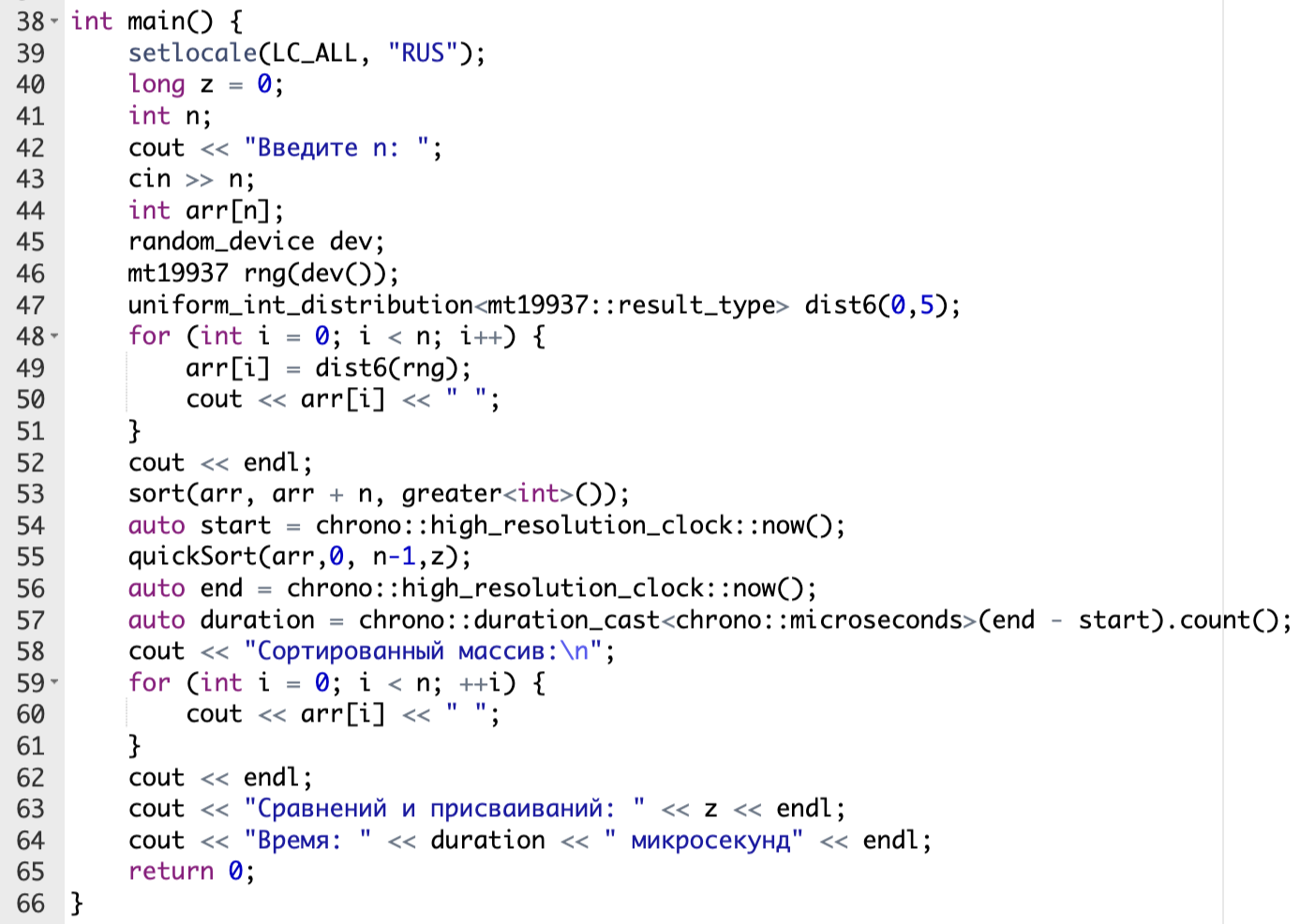


Рисунок 16 – Функция main

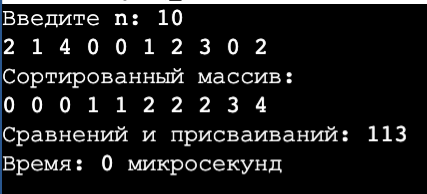


Рисунок 17 – Тестирования программы

### **2.10.4 Тестирование программы алгоритма быстрой сортировки(Хоара) в лучшем случае**

Напишем программу, которая реализует алгоритм быстрой сортировки(Хоара) с отсортированными значениями массива по возрастанию на языке C++(рис.18,19).

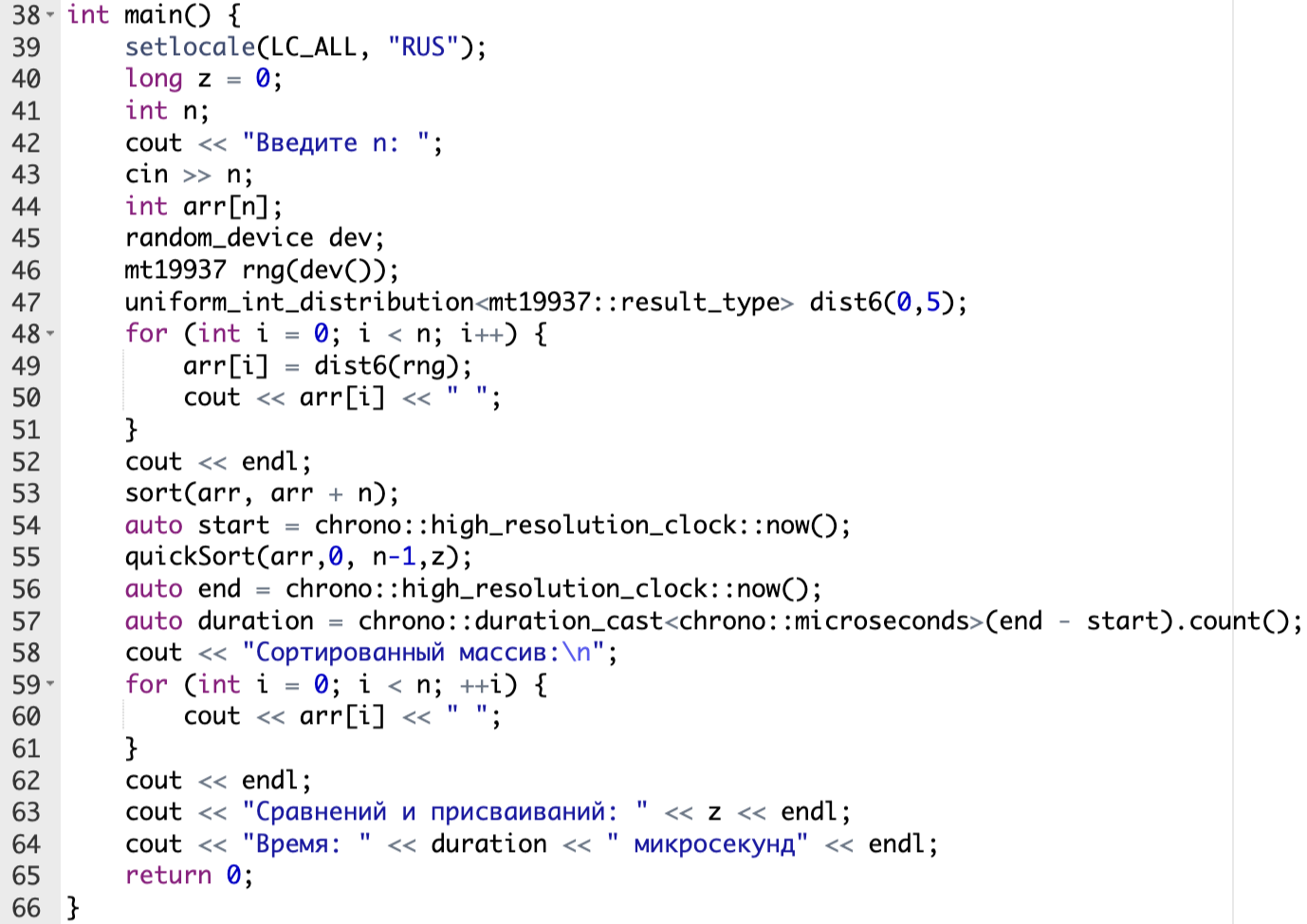


Рисунок 18 – Функция main

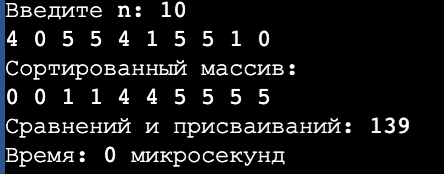


Рисунок 19 – Тестирование программы

### **2.10.5 Заполнение таблиц для алгоритмов сортировок**

Заполним таблицу 1.4 данными, полученными в результате тестирования программ сортировки Шелла со смещением Д.Кнута первым способом со значениями массива отсортированными значениями по убыванию и возрастанию.

Таблица 1.4. Сводная таблица результатов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **n** | **T(n), мс** | **Тп(n)=Cф+Mф в худшем случае** | **T(n), мс** | **Тп(n)=Cф+Mф в лучшем случае** |
| 100 | 0.006 | 2101 | 0.004 | 1732 |
| 1000 | 0.063 | 31688 | 0.049 | 27317 |
| 10000 | 0.941 | 430034 | 0.609 | 376262 |
| 100000 | 7.989 | 5194272 | 10.349 | 4835787 |
| 1000000 | 410.743 | 62940478 | 83.85 | 59021392 |

### 

Заполним таблицу 1.5 данными, полученными в результате тестирования программ быстрой сортировки(Хоара) со значениями массива отсортированными значениями по убыванию и возрастанию.

Таблица 1.5. Сводная таблица результатов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **n** | **T(n), мс** | **Тп(n)=Cф+Mф в худшем случае** | **T(n), мс** | **Тп(n)=Cф+Mф в лучшем случае** |
| 100 | 0.005 | 1931 | 0.005 | 1921 |
| 1000 | 0.041 | 26843 | 0.039 | 26611 |
| 10000 | 0.467 | 349281 | 0.464 | 350450 |
| 100000 | 5.956 | 4472806 | 5.824 | 4432964 |
| 1000000 | 93.509 | 67092090 | 97.843 | 66494460 |

## 2.11 Вывод по заданию №1

В зависимости от структуры исходного массива, алгоритм сортировки Шелла со смещением Д. Кнута и быстрая сортировка могут проявлять различную производительность. Если массив частично упорядочен или имеет какие-то специфические особенности, алгоритм сортировки Шелла со смещением Д. Кнута может быть предпочтительнее. В противном случае, быстрая сортировка Хоара может быть более эффективной.

# 

# 3 ЗАДАНИЕ №2

## 3.1 Формулировка задачи

Асимптотический анализ сложности алгоритмов

Требования по выполнению задания

1. Из материалов предыдущей практической работы приведите в отчёте формулы Тт(n) функций роста алгоритма простой сортировки вставками в лучшем и худшем случае.

2. На основе определений соответствующих нотаций получите асимптотическую оценку вычислительной сложности простого алгоритма сортировки вставками:

- в О-нотации (оценка сверху) для анализа худшего случая;

- в Ω-нотации (оценка снизу) для анализа лучшего случая.

3. Получите (если это возможно) асимптотически точную оценку вычислительной сложности алгоритма в нотации θ.

4. Реализуйте графическое представление функции роста и полученных асимптотических оценок сверху и снизу.

5. Привести справочную информацию о вычислительной сложности алгоритмов шейкерной сортировки и быстрой сортировки (Хоара).

6. Общие результаты свести в табл. 2.

7. Сделать вывод о наиболее эффективном алгоритме из трёх.

## 3.2 Формулы функции роста алгоритма сортировки простой вставкой в худшем и лучшем случае

Идеальный сценарий для сортировки - уже упорядоченный массив. В этом случае число сравнений и перемещений будет минимальным и составит Тт(n)=n. Наихудший вариант - массив, отсортированный в обратном порядке. Это приведет к увеличению количества операций до Тт(n)=(n2-n)/2. В среднем случае, когда массив частично отсортирован, количество операций будет Тт(n)=n2/4.

## 3.3 Асимптотическая оценка вычислительной сложности простого алгоритма сортировки вставкой

По асимптотической оценке сложности простого алгоритма сортировки вставкой для наихудшего случая - О(n2). Для наилучшего случая - Ω(n). А для среднего случая - θ(n2). Объем памяти, затрачиваемый на алгоритм простой сортировки вставкой, равен O(1).

## 3.4 Графическое представление функции роста и полученных асимптотических оценок сверху и снизу

Построим график на основе пунктов 3.2 и 3.3(рис.20).

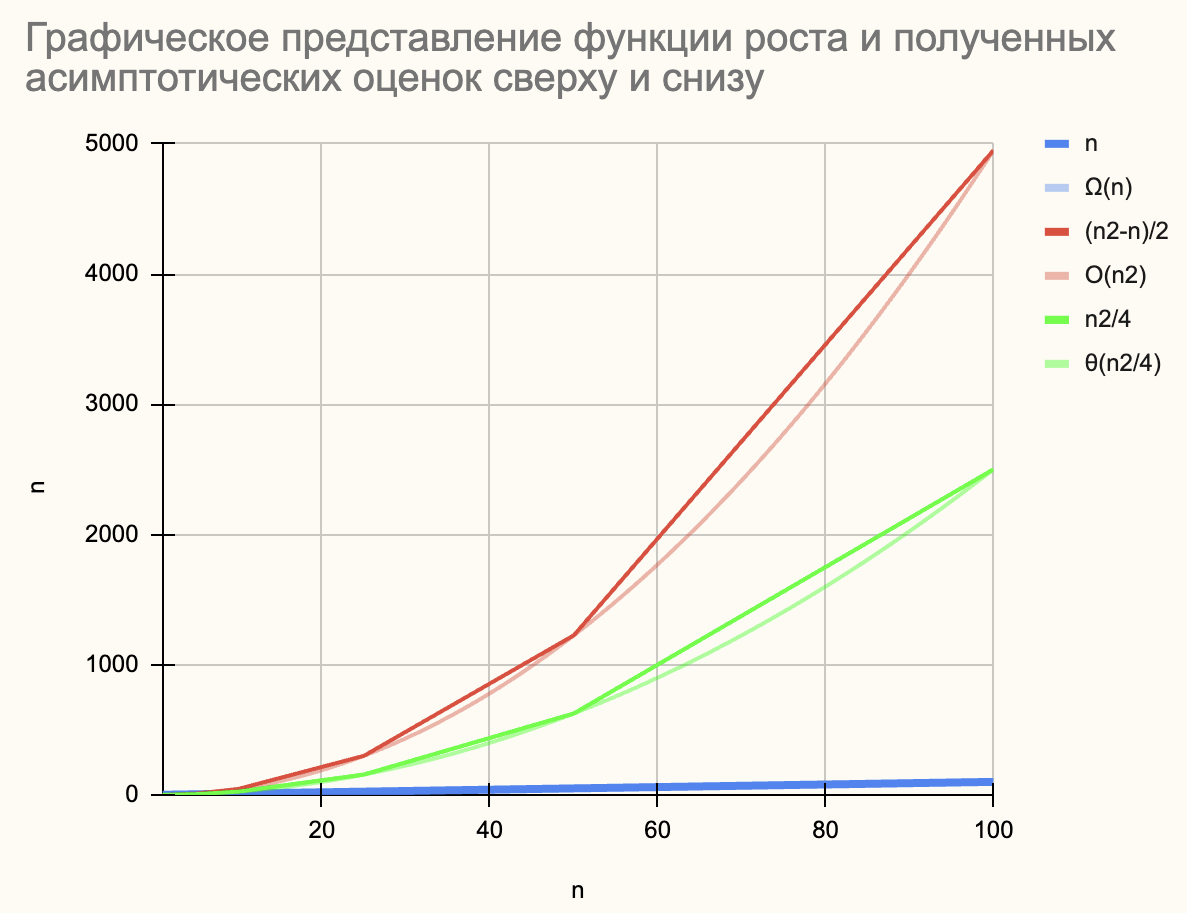


Рисунок 20 - Графическое представление функции роста и полученных асимптотических оценок сверху и снизу

## 3.5 Справочная информация о вычислительной сложности алгоритмов сортировки Шелла со смещением Д.Кнута первым способом и быстрой сортировкой(Хоара)

### **3.5.1 Справочная информация о вычислительной сложности алгоритма сортировки Шелла со смещением Д.Кнута первым способом**

Худший случай: Оценка вычислительной сложности - О(n log2 n).

Лучший случай: Оценка вычислительной сложности - Ω(n).

Средний случай: Оценка вычислительной сложности - θ(n1.25).

Ёмкостная сложность - O(1).

### **3.5.2 Справочная информация о вычислительной сложности алгоритма быстрой сортировки (Хоара)**

Худший случай: Оценка вычислительной сложности - О(n2).

Лучший случай: Оценка вычислительной сложности - Ω(n log n).

Средний случай: Оценка вычислительной сложности - θ(n log n).

Ёмкостная сложность - O(log n).

## 3.6 Таблица асимптотической сложности трёх алгоритмов

Заполним таблицу 2 асимптотической сложности алгоритма для алгоритмов сортировки простой вставки, Шелла со смещением Д.Кнута первым способом и быстрой сортировки(Хоара). А также укажем ёмкостную сложность данных алгоритмов сортировок.

Таблица 2. Сводная таблица результатов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Алгоритм | Асимптотическая сложность алгоритма | | | |
| Наихудший случай (сверху) | Наилучший случай (снизу) | Средний случай (точная оценка) | Ёмкостная сложность |
| Простая вставка | О(n2) | Ω(n) | θ(n2) | О(1) |
| Сортировка Шелла со смещением Д.Кнута вторым способом | О(nlog2n) | Ω(n) | θ(n1.25) | О(1) |
| Быстрая сортировка (Хоара) | О(n2) | Ω(n logn) | θ(n logn) | О(logn) |

## 

## 3.7 Выводы по заданию №2

Простая сортировка вставкой, быстрая сортировка (Хоара) и сортировка Шелла с улучшением Д.Кнута являются популярными алгоритмами сортировки.

Простая сортировка вставкой имеет сложность O(n2) в худшем случае, что делает ее не самым эффективным алгоритмом для больших наборов данных. Однако, для небольших наборов данных она может быть эффективной из-за простоты реализации.

Быстрая сортировка (Хоара) имеет среднюю сложность O(n log n) и является одним из самых быстрых алгоритмов сортировки. Однако, в худшем случае ее сложность может быть O(n2), если плохо выбран опорный элемент.

Сортировка Шелла с улучшением Дональда Кнута также имеет сложность в среднем O(n log n) и может быть эффективной для некоторых типов данных, особенно когда данные уже частично отсортированы.

Итак, если необходимо выбрать наиболее эффективный алгоритм из простой сортировки вставкой, быстрой сортировки и сортировки Шелла с улучшением Д.Кнута, то наилучшим выбором будет быстрая сортировка (Хоара) из-за ее более высокой скорости выполнения для общих случаев.

# 

# 4 ВЫВОДЫ

В ходе практической работы были выполнены следующие задачи:

- Получены навыки по анализу вычислительной сложности алгоритмов сортировки и определению наиболее эффективного алгоритма;

- Проведён анализ алгоритмов быстрой сортировки(Хоара) и Шелла со смещением Д.Кнута первым способом;

- Были реализованы программы алгоритмов быстрой сортировки(Хоара) и Шелла со смещением Д.Кнута первым способом;

- Проведено тестирование программ для алгоритмов быстрой сортировки(Хоара) и Шелла со смещением Д.Кнута первым способом;

- Построены графики функции роста Тп алгоритмов быстрой сортировки(Хоара) и Шелла со смещением Д.Кнута первым способом;

- Произведено сравнение алгоритмов простой сортировки вставками, быстрой сортировки(Хоара) и Шелла со смещением Д.Кнута первым способом;

- Проведен анализ асимптотической сложности алгоритмов сортировки простыми вставками, быстрой сортировки(Хоара) и Шелла со смещением Д.Кнута первым способом;

-Произведено сравнение асимптотической сложности алгоритмов сортировки простыми вставками, быстрой сортировки(Хоара) и Шелла со смещением Д.Кнута первым способом;

- Проведено определение наиболее эффективного алгоритма.

Таким образом, главную цель практической работы, а именно получение навыков по анализу вычислительной сложности алгоритмов сортировки и определению наиболее эффективного алгоритма, можно считать выполненной.

# 5 ЛИТЕРАТУРА

1. Бхаргава А. Грокаем алгоритмы. Иллюстрированное пособие для программистов и любопытствующих. – СПб: Питер, 2017. – 288 с.

2. Вирт Н. Алгоритмы + структуры данных = программы. – М.: Мир, 1985. – 406 с.

3. Кнут Д.Э. Искусство программирования, том 3. Сортировка и поиск, 2-е изд. – М.: ООО «И.Д. Вильямс», 2018. – 832 с.

4. Кораблин Ю.П. Структуры и алгоритмы обработки данных: учебно-методическое пособие / Ю.П. Кораблин, В.П. Сыромятников, Л.А. Скворцова. – М.: РТУ МИРЭА, 2020. — 219 с.

5. Кормен Т.Х. и др. Алгоритмы: построение и анализ, 3-е изд. – М.: ООО «И.Д.Вильямс», 2013. – 1328 с.

6. Макконнелл Дж. Основы современных алгоритмов. Активный обучающий метод. 3-е доп. изд., - М.: Техносфера, 2018. – 416 с.

7. Седжвик Р. Фундаментальные алгоритмы на C++. Анализ/Структуры данных/Сортировка/Поиск. – К.: Издательство «Диасофт», 2001. – 688 с.

8. Скиена С. Алгоритмы. Руководство по разработке, - 2-е изд. – СПб: БХВ-Петербург, 2011. – 720 с.

9. Хайнеман Д. и др. Алгоритмы. Справочник с примерами на C, C++, Java и Python, 2-е изд. – СПб: ООО «Альфа-книга», 2017. – 432 с.

10. AlgoList – алгоритмы, методы, исходники [Электронный ресурс]. URL: http://algolist.manual.ru/ (дата обращения 15.03.2022).

11. Алгоритмы – всё об алгоритмах / Хабр [Электронный ресурс]. URL: https://habr.com/ru/hub/algorithms/ (дата обращения 15.03.2022).

12. НОУ ИНТУИТ | Технопарк Mail.ru Group: Алгоритмы и структуры данных [Электронный ресурс]. URL: https://intuit.ru/studies/courses/3496/738/info (дата обращения 15.03.2022).