

Федеральное государственное бюджетное образовательное учреждение высшего образования Московский государственный технологический университет "ФГБОУ ВО МГТУ «СТАНКИН»

Кафедра «Управление и информатика в технических системах

**Отчёт по лабораторным работам по дисциплине "Алгоритмы  
и структуры данных"**

Выполнил:  
Студент группы ИДБ-22-10  
Инкин Денис Васильевич  
Преподаватель:  
доцент Евдокимов С.А.

Москва, 2023

## Содержание

<b>1</b>	<b>Сортировка подсчетом</b>	<b>3</b>
1.1	Постановка задачи . . . . .	3
1.2	Описание алгоритма . . . . .	3
1.3	Блок-схема алгоритма . . . . .	5
1.4	Код алгоритма на языке Python . . . . .	6
1.5	График времени сортировки . . . . .	7
1.6	Выводы . . . . .	8
1.7	Библиография . . . . .	9

# 1 Сортировка подсчетом

## 1.1 Постановка задачи

Написать программу сортировки простыми вставками и сравнить ее время выполнения со стандартной функцией `qsort`. Для сравнения вычисляем время выполнения функции сортировки на массиве целых чисел следующих размеров: 16, 100, 500, 1000, 5000. Для работы программы сортировки генерируем случайные массивы целых чисел:

- для массивов размерами  $\leq 500$  элементов числа в интервале  $[100, 1000]$ ;
- для массивов размерами  $\leq 500$  элементов числа в интервале  $[1000, 10000]$ ;

Для проверки программы сравниваем результат работы с корректно отсортированным массивом.

Время работы функции сортировки рассчитываем с точностью до тысячных долей миллисекунд. Например, 0.002, 0.018, 0.377, 1.380, 35.806 мсек.

Для оценки поведения функции сортировки создаем графики в зависимости от числа элементов ( $N$ ) в массиве:  $N$  в массиве:

- график функции  $N^2$ ;
- график функции  $N \log_2 N$ ;
- график времени работы стандартной функции `qsort`;
- график времени работы функции сортировки подсчетом;

## 1.2 Описание алгоритма

Этот алгоритм сортирует записи  $R_1, \dots, R_n$  по ключам  $a_1, \dots, a_n$ . После завершения алгоритма величина  $\text{COUNT}[j]$  определяет положение в записи  $R_j$

Основные шаги алгоритма:

- 1 Установить в счетчиках  $\text{COUNT}[0] \dots \text{COUNT}[N-1]$  нули.
- 2 Выполнить шаг 3 при  $i = 0, 1, \dots, n-1$ ; затем завершить выполнение процедуры.
- 3 Выполнить шаг 4 при  $j = i+1, i+2, \dots, n-1$ .
- 4 Если  $a_i < a_j$ , то увеличить  $\text{COUNT}[j]$  на 1; в противном случае увеличить  $\text{COUNT}[i]$  на 1.

Формальная запись алгоритма посредством диаграмм Насси-Шнейдермана показана на рис.1.

Результат работы программы на массиве из 16 чисел показан ниже:

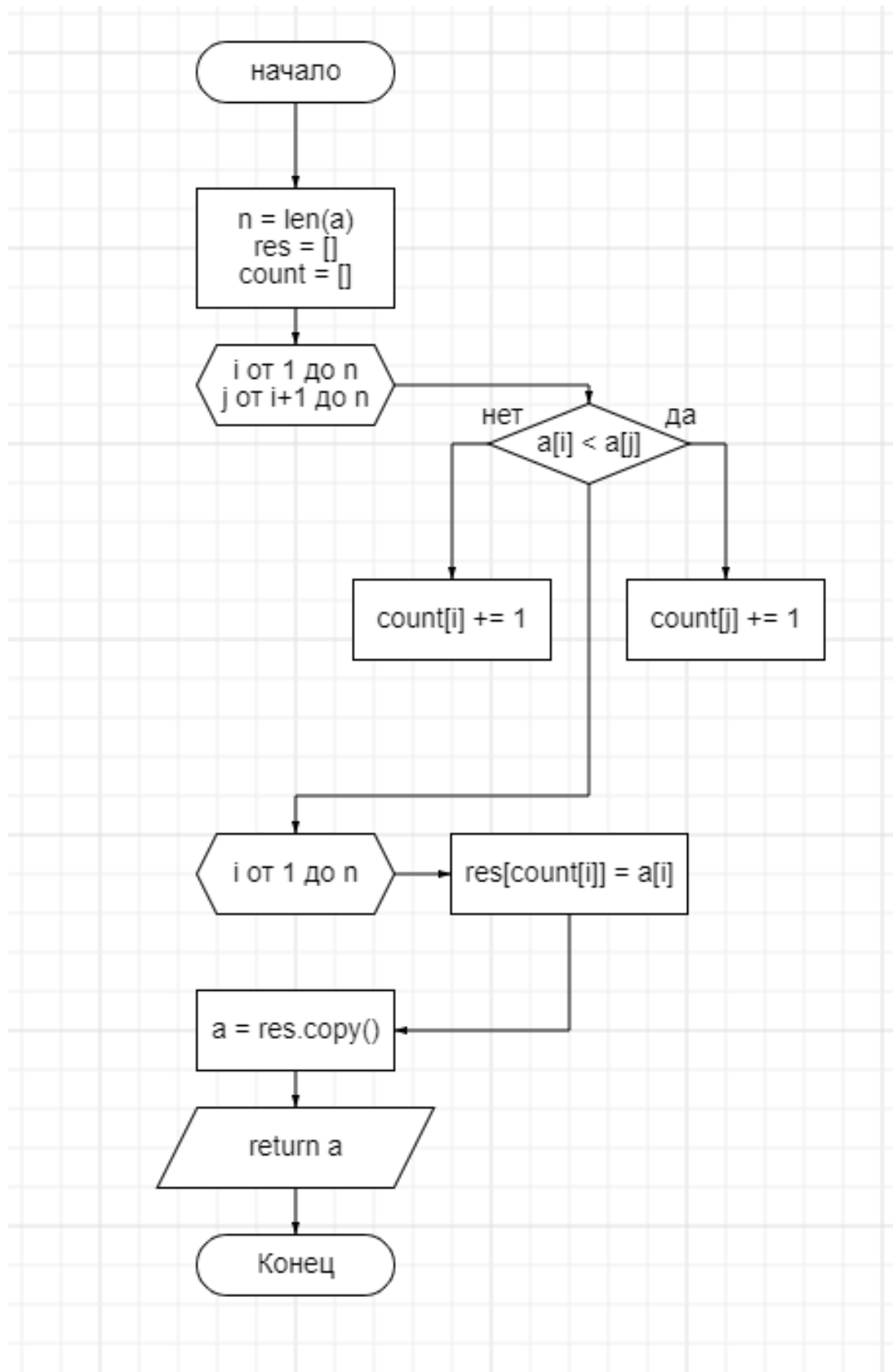
**Результат работы программы:** Сортируется массив из 16 чисел

Исходный массив: 643, 723, 209, 763, 827, 596, 338, 263, 734, 589, 773, 491, 207, 973, 168, 838

Отсортированный массив: 168, 207, 209, 263, 338, 491, 589, 596, 643, 723, 734, 763, 773, 827, 838, 973

Время сортировки: 0.0362 мсек.

## 1.3 Блок-схема алгоритма



## 1.4 Код алгоритма на языке Python

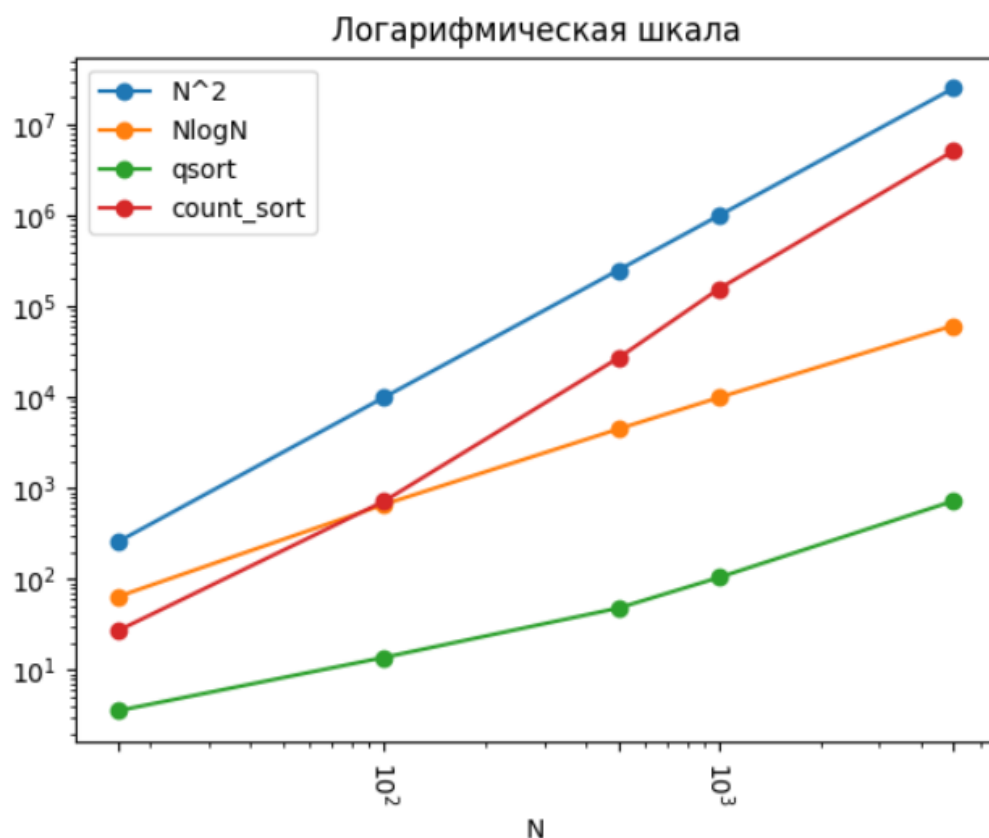
```
def count_sort(a):  
    n = len(a)  
    res = [0 for i in range(n)]  
    count = [0 for i in range(n)]  
    for i in range(n):  
        for j in range(i + 1, n):  
            if a[i] < a[j]:  
                count[j] += 1  
            else:  
                count[i] += 1  
  
    for i in range(n):  
        res[count[i]] = a[i]  
    a = res.copy()  
    return a
```

## 1.5 График времени сортировки

Для оценки поведения функции сортировки создаем графики в зависимости от числа элементов ( $N$ ) в массиве:

- график функции  $N^2$ ;
- график функции  $N \log_2 N$ ;
- график времени работы стандартной функции `qsort`;
- график времени работы функции сортировки подсчетом;

N	N^2	N*LOGN	мксек	
			qsort	count_sort
16	256	64	3.576	27
100	10000	664	13.828	721
500	250000	4483	48.160	24964
1000	1000000	9966	104.90	202056
5000	25000000	61439	726.46	5123699



## 1.6 Выводы

Из анализа графиков следует:

- Сортировка простыми вставками имеет временную сложность  $O(N^2)$
- На малом числе элементов массива ( $N$ ) в массиве время сортировки простыми вставками лучше, чем у функции  $N\log_2 N$ ;



## 1.7 Библиография

1. Д.Кнут. Искусство программирования для ЭВМ. Сортировка и поиск. Том 3. М.:Мир, Москва, 1978. — 844 с.