МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное

учреждение высшего образования

«Южно-Уральский государственный университет

(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук

Кафедра системного программирования

**ОТЧЕТ**

о выполнении практического задания № 5

по дисциплине

«Структуры и алгоритмы обработки данных»

|  |  |
| --- | --- |
|  | Выполнил:  студент группы КЭ-203  Старостенок Д.В.  Проверил:  ст. преподаватель кафедры СП  Петрова Л.Н. |

Челябинск – 2022

ОГЛАВЛЕНИЕ

[1. Задача № 1 “Сортировки” 2](#_Toc100656473)

[1.1. Физические величины: 2](#_Toc100656474)

[1.2. Область допустимых значений физических величин: 2](#_Toc100656475)

[1.3. Единицы измерения физических величин: 2](#_Toc100656476)

[1.4. Ограничения: 2](#_Toc100656477)

[1.5. Анализ ожидаемых результатов: 2](#_Toc100656478)

[1.6. Листинг программы: 3](#_Toc100656479)

[1.7. Контрольный тест: 4](#_Toc100656480)

[Вывод 5](#_Toc100656481)

1. Задача № 1 “Сортировки”

Задача:

Отсортировать одномерный массив целых чисел по убыванию. Сортировки: простым включением и пузырьковая.

* 1. Физические величины:

int, var

* 1. Область допустимых значений физических величин:

int – от -2147483648 до 2147483647

* 1. Единицы измерения физических величин:

int – целочисленный тип, 4 байта

* 1. Ограничения:

Размер введенного массива

* 1. Анализ ожидаемых результатов:

Отсортированный в обратном порядке список

* 1. Листинг программы:

def time\_counter(sort, array): # Подсчет времени работы

time\_now = timer()

sort(array)

end\_time = timer()

working\_time = end\_time - time\_now

return format(working\_time, ".7f")

def bubble\_sort(array): # Пузырьковая сортировка

for i in range(0, len(array) - 1):

for j in range(0, len(array) - 1):

if array[j] < array[j + 1]: # смена условия, чтобы был обратный ход

array[j], array[j + 1] = array[j + 1], array[j]

return array

def insertion(array): # Сортировка простыми включениями

for i in range(len(array)):

j = i - 1

key = array[i]

while array[j] < key and j >= 0: # смена условия, чтобы был обратный ход

array[j + 1] = array[j]

j -= 1

array[j + 1] = key

return array

if \_\_name\_\_ == '\_\_main\_\_':

size\_list = int(input("Введите размер списка: "))

sort\_list = [] # Сортированный в обратном порядке

for i in range(size\_list):

sort\_list.append(i)

sort\_list\_half = [] # Сортированный на половину

for i in sort\_list[:(size\_list // 2) - 1:-1]:

sort\_list\_half.append(i)

for i in sort\_list[:size\_list // 2]:

sort\_list\_half.append(i)

sort\_list\_reversed = [] # Сортированный список

for i in sort\_list[::-1]:

sort\_list\_reversed.append(i)

print("Сортированный список ", sort\_list\_reversed)

print("Сортированный в обратном порядке ", sort\_list)

print("Сортированный на половину", sort\_list\_half)

sorts\_times = PrettyTable()

sorts\_times.field\_names = ["Метод", "Сортированный", "Сортированный в обратном порядке", "Сортированный на половину"]

sorts\_times.add\_row(["Пузырьковая", time\_counter(bubble\_sort, sort\_list\_reversed.copy()),

time\_counter(bubble\_sort, sort\_list.copy()),

time\_counter(bubble\_sort, sort\_list\_half.copy())])

sorts\_times.add\_row(["Простыми включениями", time\_counter(insertion, sort\_list\_reversed.copy()),

time\_counter(insertion, sort\_list.copy()),

time\_counter(insertion, sort\_list\_half.copy())])

print(sorts\_times)

* 1. Контрольный тест:

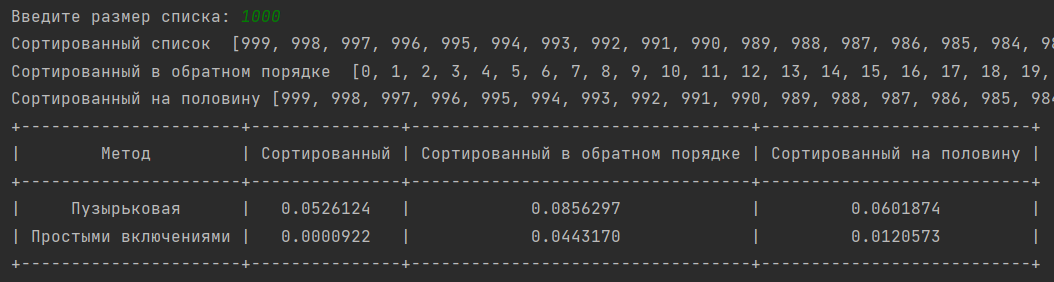


Рис. 1. Тест сортировок 1

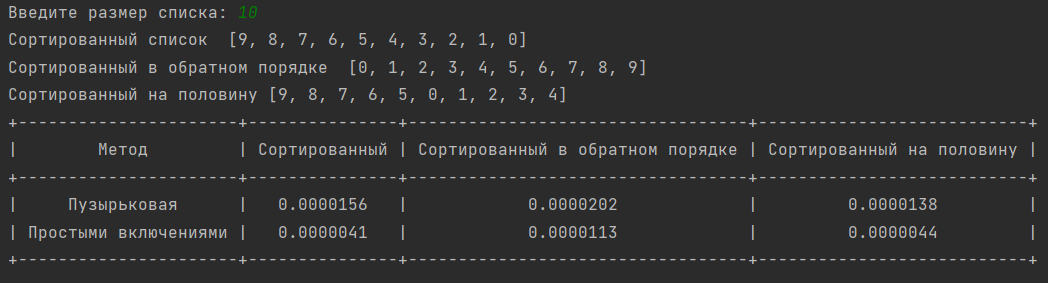


Рис. 2. Тест сортировок 2

Вывод

В результате проделанной работы были получены навыки реализации и работы с сортировками. По заданию создана 1 программа, которая помогла закрепить полученные навыки.