САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

ФАКУЛЬТЕТ ИНФОКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ

Отчет по лабораторной работе №1

по курсу «Алгоритмы и структуры данных»

Тема: Сортировка вставками, выбором, пузырьковая.

Вариант 20

Выполнил:

Смирнов Георгий Валерьевич

К3139

Проверил:

Афанасьев А.В.

Санкт-Петербург

2024 г.

# Содержание отчета

[**Содержание отчета 2**](#_gjdgxs)

[**Задачи по варианту 3**](#_30j0zll)

[Задача №1. Сортировка вставкой 3](#_1y810tw)

[Задача №5. Сортировка выбором 6](#_4i7ojhp)

[Задача №8. Секретарь Своп](#_2xcytpi) [12](#_1ci93xb)

[**Вывод 23**](#_3whwml4)

# Задачи по варианту

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

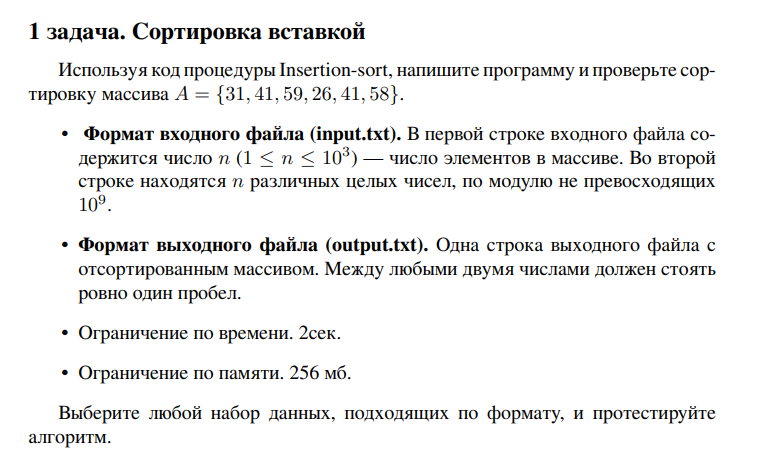
## 

## 

## 

## 

## Задача №1. Сортировка вставкой.



**Код программы**

import time

import tracemalloc

def insertion\_sort(arr):

for i in range(1, len(arr)):

key = arr[i]

j = i - 1

while j >= 0 and key < arr[j]:

arr[j + 1] = arr[j]

j -= 1

arr[j + 1] = key

def main():

# Чтение данных из файла input.txt

with open('input.txt', 'r') as file:

n = int(file.readline().strip())

array = list(map(int, file.readline().strip().split()))

# Начало отслеживания времени и памяти

start\_time = time.perf\_counter()

tracemalloc.start()

# Сортировка массива

insertion\_sort(array)

# Окончание отслеживания времени и памяти

current, peak = tracemalloc.get\_traced\_memory()

tracemalloc.stop()

end\_time = time.perf\_counter()

# Запись отсортированного массива в файл output.txt

with open('output.txt', 'w') as file:

file.write(' '.join(map(str, array)))

# Вывод времени и памяти

print(f"Время выполнения: {end\_time - start\_time:.6f} секунд")

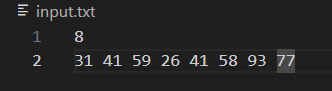
print(f"Использование памяти: {peak / 10\*\*6:.6f} МБ")

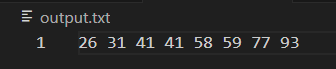
if \_\_name\_\_ == "\_\_main\_\_":

main()

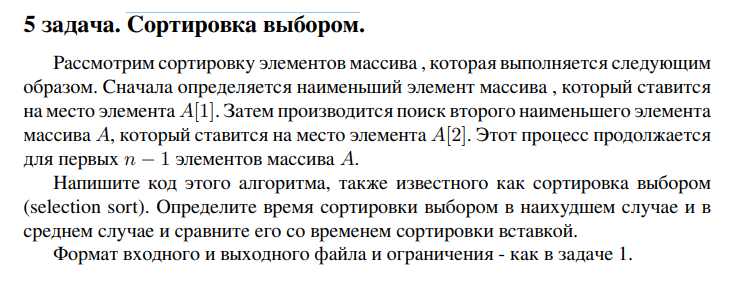
Результат работы кода на примерах из текста задачи:

1)





## Задача №5. Сортировка выбором.



**Код программы**

def insertion\_sort(arr):

for i in range(1, len(arr)):

key = arr[i]

j = i - 1

while j >= 0 and key < arr[j]:

arr[j + 1] = arr[j]

j -= 1

arr[j + 1] = key

def main():

# Чтение данных из файла input.txt

with open('input.txt', 'r') as file:

n = int(file.readline().strip())

array = list(map(int, file.readline().strip().split()))

# Сортировка массива

insertion\_sort(array)

# Запись отсортированного массива в файл output.txt

with open('output.txt', 'w') as file:

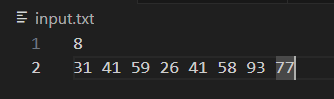
file.write(' '.join(map(str, array)))

if \_\_name\_\_ == "\_\_main\_\_":

main()

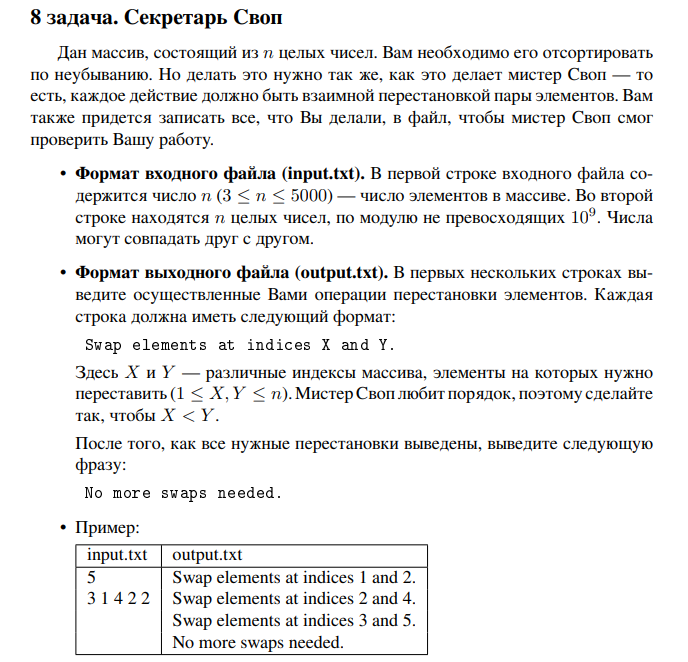
Результат работы кода на примерах из текста задачи:

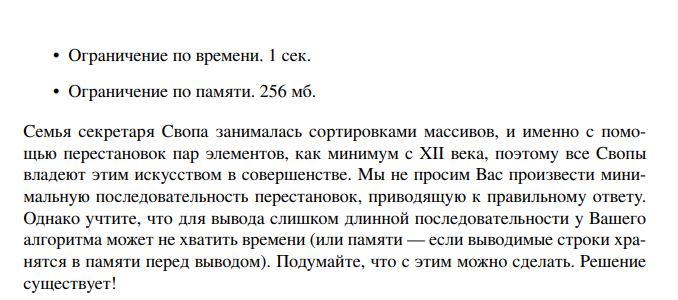
1)





## Задача №8. Секретарь Своп.





**Код программы**

import time

import tracemalloc

def insertion\_sort\_with\_swaps(arr):

swaps = []

for i in range(len(arr)):

min\_index = i

for j in range(i + 1, len(arr)):

if arr[j] < arr[min\_index]:

min\_index = j

if min\_index != i:

arr[i], arr[min\_index] = arr[min\_index], arr[i]

swaps.append(f"Swap elements at indices {i + 1} and {min\_index + 1}.")

return swaps

def main():

# Чтение данных из файла input.txt

with open('input.txt', 'r') as file:

n = int(file.readline().strip())

array = list(map(int, file.readline().strip().split()))

# Начало отслеживания времени и памяти

start\_time = time.perf\_counter()

tracemalloc.start()

# Получение списка операций перестановки

swaps = insertion\_sort\_with\_swaps(array)

# Окончание отслеживания времени и памяти

current, peak = tracemalloc.get\_traced\_memory()

tracemalloc.stop()

end\_time = time.perf\_counter()

# Запись операций в файл output.txt

with open('output.txt', 'w') as file:

for swap in swaps:

file.write(swap + '\n')

file.write("No more swaps needed.\n")

# Вывод времени и памяти

print(f"Время выполнения: {end\_time - start\_time:.6f} секунд")

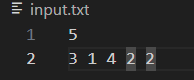
print(f"Использование памяти: {peak / 10\*\*6:.6f} МБ")

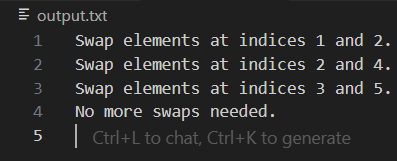
if \_\_name\_\_ == "\_\_main\_\_":

main()

Результат работы кода на примерах из текста задачи:

1)





|  | Время выполнения, с | Затраты памяти, МБ |
| --- | --- | --- |
| Пример из задачи | 0 0.000088 секунд | 0.000370 МБ |

Вывод по задаче: Программа корректно работает на всех приведенных тестах и укладывается в ограничения по времени и памяти