

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИТМО
ФАКУЛЬТЕТ ИНФОКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ**

**Отчет по лабораторной работе №1
По курсу «Алгоритмы и структуры данных»**

Тема: Сортировка вставками, выбором, пузырьковая. Вариант 1

Выполнил:
Бен Шамех Абделазиз
Группа: К3239

Проверила:
Ромакина Оксана Михайловна

Санкт-Петербург
2025 г.

Содержание

1 Задача 1. Сортировка вставкой	2
1.1 Листинг кода	2
1.2 Объяснение решения	2
1.3 Результаты выполнения	2
1.4 Метрики	2
2 Задача 2. Сортировка вставкой +	3
2.1 Листинг кода	3
2.2 Результаты выполнения	3
3 Задача 3. Сортировка вставкой по убыванию	4
3.1 Листинг кода	4
3.2 Результаты выполнения	4
4 Задача 4. Линейный поиск	5
4.1 Листинг кода	5
4.2 Результаты выполнения	5
5 Задача 7. Знакомство с жителями Сортлэнда	6
5.1 Листинг кода	6
5.2 Результаты выполнения	6
6 Задача 10. Палиндром	7
6.1 Результаты выполнения	7
7 Вывод	8

1 Задача 1. Сортировка вставкой

Используя код процедуры Insertion-sort, напишите программу и проверьте сортировку массива $A = \{31, 41, 59, 26, 41, 58\}$.

1.1 Листинг кода

```
1 def task1():
2     f = open('input.txt')
3     f2 = open('output.txt', 'w')
4     n = int(f.readline())
5     arr = [int(i) for i in f.readline().split()]
6
7     def ins_sort(arr):
8         for j in range(1, len(arr)):
9             i = j - 1
10            k = arr[j]
11            while i >= 0 and arr[i] > k:
12                arr[i+1] = arr[i]
13                i = i - 1
14                arr[i+1] = k
15            return arr
16
17     print(*ins_sort(arr), file=f2)
18     f.close()
19     f2.close()
```

1.2 Объяснение решения

Перебирая каждый элемент массива, начиная с 1 индекса, ищем для него место, переставляя все элементы, которые больше него, "вправо пока не найдем корректную позицию.

1.3 Результаты выполнения

Пример из задачи:

- Ввод (input.txt): 6 \n 31 41 59 26 41 58
- Вывод (output.txt): 26 31 41 41 58 59

1.4 Метрики

Сценарий	Время	Память
Нижняя граница (n=1)	0.0065 sec	14.77 Mb
Пример из задачи	0.00407 sec	14.80 Mb
Верхняя граница (n=1000)	0.02734 sec	15.13 Mb

2 Задача 2. Сортировка вставкой +

Измените процедуру для вывода индексов, на которые перемещаются элементы в процессе сортировки.

2.1 Листинг кода

```
1 def task2():
2     f = open('input.txt')
3     f2 = open('output.txt', 'w')
4     n = int(f.readline())
5     arr = [int(i) for i in f.readline().split()]
6
7     def ins_sort(arr):
8         ind_arr = [-1] * n
9         ind_arr[0] = 1
10        for j in range(1, len(arr)):
11            i = j-1
12            k = arr[j]
13            while i >= 0 and arr[i] > k:
14                arr[i+1] = arr[i]
15                i = i-1
16            arr[i+1] = k
17            ind_arr[j] = i+2
18        return ind_arr, arr
19
20    arr1, arr2 = ins_sort(arr)
21    print(*arr1, file=f2)
22    print(*arr2, file=f2)
23    f.close()
24    f2.close()
```

2.2 Результаты выполнения

- Ввод (input.txt): 6 \n 31 41 59 26 41 58

- Вывод (output.txt):

```
1 2 3 1 4 5
26 31 41 41 58 59
```

3 Задача 3. Сортировка вставкой по убыванию

3.1 Листинг кода

```
1 def reverse_ins_sort(arr):
2     for i in range(len(arr)-1):
3         while i >= 0 and arr[i] < arr[i+1]:
4             arr[i+1], arr[i] = arr[i], arr[i+1]
5             i -= 1
6     return arr
```

3.2 Результаты выполнения

- Ввод: 6 \n 31 41 59 26 41 58
- Вывод: 59 58 41 41 31 26

4 Задача 4. Линейный поиск

4.1 Листинг кода

```
1 def task4():
2     f = open('input.txt')
3     arr = [int(i) for i in f.readline().split()]
4     v = int(f.readline())
5     finded = [i for i, x in enumerate(arr) if x == v]
6     # ... logic for printing ...
```

4.2 Результаты выполнения

- Ввод (Число 41): 31 41 59 26 41 58 \n 41
- Вывод: 2 \n 1 4 (Найдено 2 раза на индексах 1 и 4)

5 Задача 7. Знакомство с жителями Сортлэнда

5.1 Листинг кода

```
1 def task7():
2     f = open('input.txt')
3     n = int(f.readline())
4     arr = [(float(item), ind+1) for ind, item in enumerate(f.readline()
5 .split())]
6     # ... sorting by savings ...
7     print(arr[0][1], arr[n//2][1], arr[-1][1])
```

5.2 Результаты выполнения

- Ввод: 5 \n 10.1 55.0 40.0 20.0 70.5
- Вывод: 1 3 5

6 Задача 10. Палиндром

6.1 Результаты выполнения

- Ввод: 7 \n AABCDDBA
- Вывод: ABCDCBA

7 Вывод

В ходе лабораторной работы были освоены алгоритмы сортировки вставками и их модификации. Было выявлено, что:

1. Алгоритм сортировки вставками эффективен для массивов небольшого размера.
2. Временная сложность в худшем случае составляет $O(n^2)$, а затраты памяти — $O(1)$ (In-place), если не считать хранение индексов.
3. Использование системных метрик позволяет наглядно оценить эффективность кода при масштабировании данных.